# Scanless Fast Handoff Technique Based on Global Path Cache for WLANs

Weetit Wanalertlak, *Student member, IEEE,* Ben Lee, *Member, IEEE,*
and Chansu Yu, *Member, IEEE*

***Abstract*— Wireless LANs (WLANs) have been widely adopted and are more convenient as they are inter-connected as wireless campus networks and wireless mesh networks. However, time-sensitive multimedia applications, which have become more popular, could suffer from long end-to-end latency in WLANs. This is due mainly to handoff delay, which in turn is caused by channel scanning. This paper proposes a technique called *Global Path-Cache* (GPC) that provides fast handoffs in WLANs. GPC properly captures the dynamic behavior of the network and MSs, and provides accurate next AP predictions to minimize the handoff latency. Moreover, the handoff frequencies are treated as time-series data, thus GPC calibrates the prediction models based on short term and periodic behaviors of mobile users. Our simulation study shows that GPC virtually eliminates the need to scan for APs during handoffs and results in much better overall handoff delay compared to existing methods.**

***Index Terms*— Wireless LANs, handoff, channel scanning, mobility prediction, time series analysis.**

## I. Introduction

Wireless communication technology together with the advancements in systems and network software allow users to be connected and be productive while on the road. Wireless LANs (WLANs) based on the IEEE 802.11 standard [2], better known as *Wi-Fi hot spots*, are already prevalent in residential as well as public areas, such as airports, university campuses, shopping malls, coffee shops, etc. Moreover, numerous efforts have already been underway to connect Wi-Fi hot spots to offer a better connectivity over a larger geographical area such as community networks that cover metropolitan areas of major US cities [3]–[6].

One of the greatest benefits of Wi-Fi hotspots or community networks is mobility support, which allows a user, for example, to continually talk on a *Voice over IP* (VoIP) application or watch a video stream while walking or riding a bus between city blocks. However, mobility incurs a large handoff delay when a *mobile station* (MS) switches connection from one *access point* (AP) to another. The key to reducing the handoff delay is to minimize the *scanning process*, which involves probing all the communication channels in order to find the best available AP. Recent studies found that passively scanning for APs during a handoff can be as much as a second [7] and actively scanning for APs requires 350∼500 ms [7]. This becomes a major concern for mobile multimedia applications such as VoIP where the end-to-end delay is recommended to be not greater than 50 ms [8].

Since the scanning process represents more than 90% of the overall handoff delay, a number of techniques have been proposed to specifically optimize the scanning process [9]–[12]. These methods employ extra hardware, either in the form of additional radios [9] or an overlay sensor network [12], to detect APs, selectively scan channels to probe based on the topological placement of APs [10], and predict the next point-of-attachment based on signal strength [11]. Unfortunately, these techniques either do not provide next AP predictions that can eliminate the need to scan for APs or consider the mobility patterns of MSs, which are dictated by the structure of a building or a city block and the past behaviors of MSs. There are also methods that consider mobility history of MSs to provide next-AP predictions [13]. However, these methods tend to be general and thus they do not consider the special characteristics of WLANs, such as highly overlapped cell coverage, MAC contention, and variations in link quality.

In order to illustrate these characteristics of WLANs, Figure 1 shows the coverage areas of the four-story, 153,000-ft$^2$ Kelley Engineering Center (KEC) at Oregon State University, and MetroFi, which is a public WLAN service that covers 2.5-mile$^2$ area of Portland, Oregon [5]. The APs in KEC are connected by Ethernet switches, while APs in the MetroFi network are interconnected by a wireless mesh network [14]. Besides the obvious difference in the sizes of the coverage areas, these two networks share many similarities and challenges based on the following observations. First, APs are installed in relative close proximity to users, *e.g.*, offices and classrooms in KEC and residential and business districts in Portland. Thus, the topological placement of APs does not follow an ideal hexagonal cell layout. Second, some cells are highly overlapped to provide high bandwidth for MSs in high traffic areas, *e.g.*, classrooms in KEC and to overcome RF signal fading due to "urban canyons" (especially in downtown Portland west of the river). Third, adjacent cells use only non-overlapped channels to reduce the electromagnetic interference among cells. Fourth, the signals transmitted from APs are not limited to just a single floor but extends omni-directionally beyond the ceilings, floors and walls. Therefore, an MS on the $1^{st}$ floor can detect signals from APs on the $2^{nd}$, $3^{rd}$, and $4^{th}$ floors. Finally, the operating environment of WLANs changes frequently and drastically due to multipath effects, user mobility, and electromagnetic interference. Therefore, the quality of signals from APs cannot be guaranteed over time. All these factors contribute to more frequent handoffs as well as higher handoff latency.

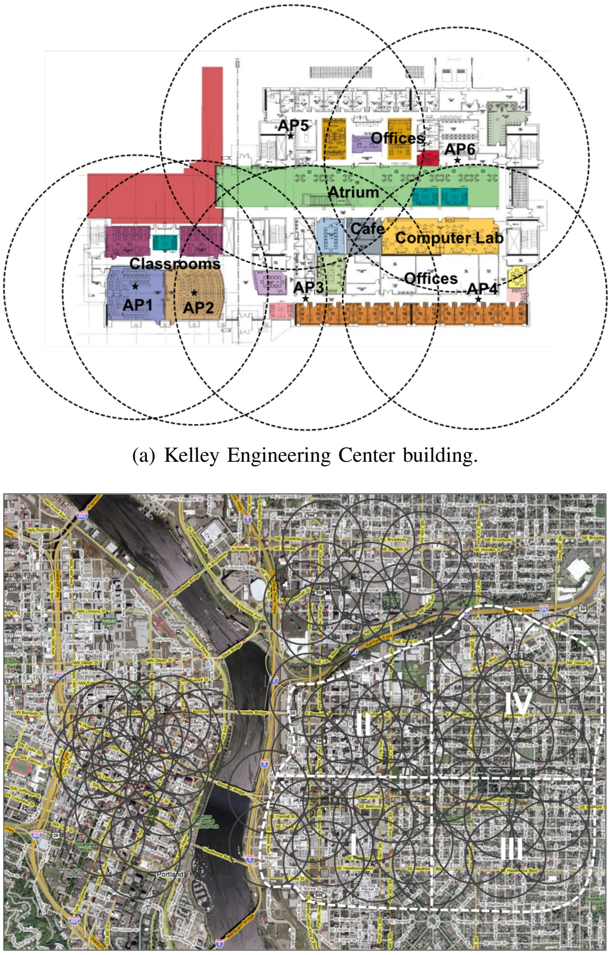
In [1], we presented a solution, called the *Global Path-Cache* (GPC) technique, which eliminates the need to perform scanning for available APs and thus results in faster handoffs. The key

This work was published in part at The International Conference on Computer Communications and Networks (ICCCN), 2007 [1].

Weetit Wanalertlak is a Ph.D. student in School of Electrical Engineering and Computer Science, Oregon State University. Email: wanalewe@eecs.orst.edu

Ben Lee is an Associate Professor in School of Electrical Engineering and Computer Science, Oregon State University. Email: benl@eecs.orst.edu

Chansu Yu is an Associate Professor in Dept. of Electrical and Computer Engineering, Cleveland State University. Email: c.yu91@csuohio.edu



(a) Kelley Engineering Center building.

(b) Public WLAN in Portland, Oregon (MetroFi®).

Fig. 1. Example WLAN coverage areas.

idea of GPC is to predict the next point-of-attachments based on the history of mobility patterns of MSs. This is achieved by maintaining the handoff history of all the MSs in the network, and then monitoring a MS's direction of movement relative to the topological placement of APs to predict its next point-of-attachment. In addition, next AP predictions are based on the frequencies of occurrences rather than signal strength. Therefore, it takes into consideration that mobility patterns are dictated by the structure of a building or a city block and the past behaviors of MSs. The GPC technique is an adaptive algorithm, which is independent of the topological placement of APs and the number of channel frequencies used.

This paper extends our earlier work on GPC by also considering short-term handoff behaviors and significantly expanding its evaluation. Therefore, in addition to providing a discussion of the basic GPC scheme, the specific contributions of this paper are as follows:

- First, the basic GPC scheme presented in [1] provides next AP predictions based on long-term frequency of handoffs and is unable to capture short-term and periodic handoff behaviors that are crucial for improving the prediction accuracy for all scenarios. This paper enhances the basic GPC scheme by treating the handoff frequencies as time-series data, thus GPC calibrates the prediction models based on specific characteristics of WLAN by applying *AutoRegressive Integrated Moving Average* (ARIMA) and *Exponential Weight Moving Average* (EWMA).
- Second, performance evaluation is significantly expanded to include a much larger network (*i.e.*, MetroFi Portland), and specifically analyze the performance effects of different types of users and the improvements provided by the time series-analysis.

Our simulation study shows that GPC results in superior handoff delay compared to *Selective Scan with Caching* (SSwC) [11] and *Neighbor Graph* (NG) [10]. The average handoff latency in GPC is 27∼28 ms and 37∼39 ms for KEC and Portland, respectively, based on the current off-the-shelf NIC delay parameters. In contrast, SSwC requires 149 and 297 ms for KEC and Portland, respectively. This is due to the fact that GPC provides a much higher Next-AP prediction accuracy, especially the $1^{st}$ Next-AP prediction, than SSwC. GPC achieves 100% accuracy and thus requires no probing while SSwC achieves prediction accuracy of only 31%∼54%. Note that NG does not provide any Next-AP predictions, and thus incurs latency of 328 and 422 ms for KEC and Portland, respectively. The time-series-based GPC scheme further improves the $1^{st}$ Next-AP prediction accuracy by as much as 17.1% and reduces the handoff latency as much as 8.5% compared to the basic GPC scheme.

The paper is organized as follows. Section 2 presents the background of the IEEE 802.11 handoff procedure. Section 3 discussed the related work. Section 4 discusses the proposed GPC technique. Section 5 evaluates the performance of the proposed method and compares it with the existing solutions. Finally, Section 6 concludes the paper and discusses future work.

## II. Background - Scanning Process in IEEE 802.11

In the IEEE 802.11 standard, when a MS moves from one cell to another, the network interface senses the degradation of signal quality in the current channel. The signal quality continues to degrade as the MS moves further away from the current AP, and the MS initiates a handoff to a new cell when the signal quality reaches a preset threshold [10]. This process starts with probing for a new cell using either passive or active scanning. In *passive scanning*, MS switches its transceiver to a new channel and waits for a beacon to be sent by the new AP, typically every 100 ms, or until the waiting time reaches a predefined maximum duration, which is longer than the beacon interval. Moreover, the time MS has to wait varies since beacons sent by APs are not synchronized. For these reasons, a recent study has shown that MSs can spend up to 1 second to search all possible channels [7], which results in unacceptable handoff delay.

In *active scanning* shown in Figure 2, a MS broadcasts a probe request and waits for a response. If the MS receives a response from an AP, it assumes there may be other APs in the channel and waits for *MaxChannelTime*. Otherwise, the MS only waits for *MinChannelTime*. MinChannelTime is shorter than MaxChannelTime to keep the overall handoff delay low, but it should be long enough for MS to receive a possible response. A typical duration for scanning each channel is around 25 ms and 350∼500 ms for all 11 channels [7].

After scanning, MS typically joins the network with the strongest signal strength, which is done by performing authentication and association/reassociation. Authentication is the process that a MS uses to announce its identity to the new AP. In the IEEE

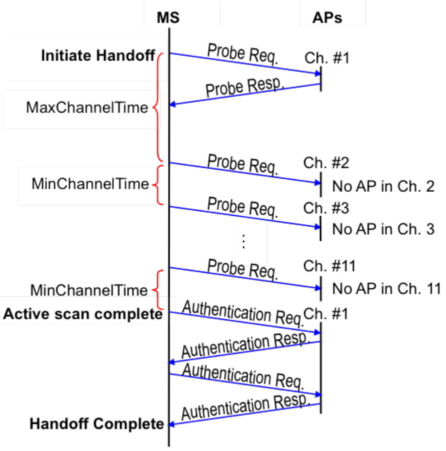


Fig. 2. Active Scanning in IEEE 802.11.

802.11 standard, authentication is performed using open system or shared key. Open system authentication is the default method for IEEE 802.11, and involves the MS sending authentication request frame, which contains source address in the frame header and information in the frame body to indicate the type of authentication, to AP. Then, the AP sends the authentication response frame back to MS. This frame has the authentication result and the information to indicate the type of authentication.

The next step is association/reassociation, which allows the distribution system to keep track of the location of each MS, so frames destined for the MS can be forwarded to the correct AP. How association/reassociation requests are processed is implementation specific, but typically involves allocation of frame buffers and, in the case of reassociation, communicating with the old AP so that any frames buffered at the old AP are transferred to the new AP and the old AP terminates its assocation with the MS. Finally, the last step involves the new AP resetting the Ethernet Address Table in the switch that connects both the old and the new APs so that the network traffic can be rerouted.

## III. Related Work

### A. Mobility Prediction

Mobility prediction is crucial for mitigating the effects of handoffs and therefore improving QoS. There has been a plethora of work on mobility prediction for variety of wireless networks, such as cellular [11], [12], [15], WLANs [9]–[11], [13], [16], ad hoc networks [17], and mesh networks [14], and applied to reduce handoff latency [7], [10], [11], [18], provide efficient resource reservation [13], [15], [19]–[23], improve routing protocols [17], and conserve power [24].

Although many different mobility prediction techniques have been proposed, these techniques can be broadly classified into the following three categories. First, data-mining techniques use a database to track and characterize the long-term mobility patterns of MSs, which are then used to predict locations of MSs. These techniques reduce the signaling overhead during handoff and provide resource reservation to MSs in cellular networks [15], [22], [23]. Second, topology-based techniques use the knowledge of geographical locations of APs and directional movement of MSs to provide resource reservation in cellular networks [21]. Third, stochastic techniques provide mobility predictions using probabilistic model. These techniques apply the knowledge of geographic coordinates of MSs from either GPS or triangulation of signal strengths to predicted future locations [19], [20].

Although all these techniques provide mobility prediction in cellular networks, they are not efficient solutions for WLANs. For example, data-mining techniques require large storage and fast processors to analyze long-term mobility behavior. In addition, the latter two techniques typically require a GPS device to obtain information about locations and directions of MSs. For systems that rely on signal triangulation, their effectiveness may be limited due to the fact that WLANs are mainly used for indoors and crowded outdoor areas where the signal strength is highly affected by noise rather than distance [25].

The technique closest to ours is Markov-based mobility predictions, which rely on the fact that the probability of the future outcome is based on the current and past outcomes [16]. Typically, a Markov mobility predictor performs the following two operations: The first operation is to maintain a collection of past locations of MSs. The second operation is to predict future locations of MSs based on the value of conditional probability that matches with the past locations of MSs. Since the mobility patterns in WLANs tend to be non-random and periodic, the Markov-based technique can be found in many mobility prediction algorithms, including ours, which aim to minimize the scanning process to provide fast handoffs in WLANs.

### B. Handoff Delay

There has been a lot of work done to reduce the handoff delay in WLANs. The related work discussed here focuses on optimizing the probing or scanning process, which is the most time consuming part of a handoff [26], [27]. *MultiScan* uses multiple WLAN network interfaces to opportunistically scan and pre-associate with alternative APs to avoid disconnections [9]. The basic idea is to have the first WLAN interface communicate with the current AP while the second WLAN interface scans for new APs. This scan information is then used to connect to the new AP before the connection is lost from the current AP. *Selective Active Scanning* uses an overlay sensor network to obtain information on the presence of APs and the quality of their transmission channels [12]. This way, a MS broadcasts an AP-list request to surrounding sensor nodes to obtain a precise information about neighboring APs, and initiates a scanning process solely based on this list. Although both techniques can provide fast handoffs, they require extra hardware, implemented either on the client side or as a separate control plane, which may be impractical and/or power inefficient.

Another technique to reduce the handoff delay is to either passively or actively scan for available APs in the background [7], [28]. *SyncSacn* is a passive method that requires APs to send staggered periodic beacons to allow a MS to scan for additional APs while it is still connected to the current AP. In contrast, a MS actively probes for APs in [28]. Both methods rely on the power saving mode to buffer packets at the AP during background probing. Although the handoff delay can be reduced by both methods, there is a hidden cost since a MS has to occasionally suspend its communication to either listen or probe for other APs. Nonetheless, the GPC method proposed in this paper is an orthogonal approach to the background scanning and thus they can be deployed together to reduce the cost of performing a full scan.

Other methods that are closest to ours in terms of reducing the scanning process are *Neighbor Graph* [10], *Pre-Authentication path* [18], and *Selective Scan with Caching* [11]. The Neighbor Graph and Pre-Authentication path techniques reduce the number of channels to scan by defining a directed graph that represents the topological placement of APs and the mobility patterns of MSs. Moreover, edges between APs that represent handoffs are added or deleted to reflect the changing conditions. In addition, the Pre-Authentication path technique reduces the signaling overhead between MS and AP by allowing MSs to pre-authenticate and pre-reassociate to APs within a directed graph before the actual handoff occurs. Although both Neighbor Graph and Pre-Authentication path techniques significantly reduce the average number of channels probed, they do not provide next point-of-attachment predictions and thus all edges (*i.e.*, adjacent channels) emanating from a node needs to be scanned.

Selective Scan with Caching minimizes the need to probe during a handoff by predicting next point-of-attachment based on signal strength. A MS joining the network for the first time performs a full scan. Then, the corresponding bits in the channel mask are set for all the probe responses received from APs, as well as bits for channels 1, 6, and 11 with the premise that these channels are more likely be used by APs. As MS connects to the AP with the strongest signal, the corresponding bit in the channel mask is reset based on the assumption that the likelihood of adjacent APs having the same channel is very small. In addition, two other APs' addresses representing the second and third strongest signals are stored in the *AP-cache* using the current AP's address as the key. These two APs represent the best and second best candidates for subsequent handoffs. During the next handoff, the MS will attempt to reassociate with these two APs in order. If it fails to reassociate with both APs or an entry is not found in the AP-cache, a selective scan is performed based on the channel mask to choose two additional APs with the strongest signals and stores them in the AP-cache. If no APs are discovered with the current channel mask, bits in the channel mask are inverted and another scan is performed. If the partial scan fails to discover APs, a full scan is performed. However, in order to use the information from the last scanning period for the current handoff, the direction of MS movement relative to the cell layout must be identical to the one in the last handoff. This is often not the case and thus the AP-cache will frequently fail to provide correct Next-AP predictions.

Recently, there has been a growing interest in expanding the coverage area of WLANs using wireless mesh networking. In *SMesh* [29], multiple APs are used to monitor the connectivity quality of MSs in their vicinity to coordinate which of them should serve the client. This is achieved by having each MS associate with a unique multicast group of mesh nodes that are in the vicinity of the MS and the mesh node with the best connectivity to the MS sends a gratuitous ARP message to force a handoff. In contrast, the proposed GPC technique is a MS-initiated handoff method, which does not require the overhead of maintaining multicast groups. Moreover, monitoring the signal quality of MSs requires all APs to be operating in the same channel and thus limiting the range of coverage area.

## IV. The Proposed GPC Technique

In order to reduce the handoff delay, GPC tracks past associated APs and then use this information to perform mobility predictions for future handoffs. This virtually eliminates the need to scan channels when MSs move through the coverage area of the same set of APs. Section IV.A starts off with the discussion of the basic GPC method that prioritizes multiple Next-AP predictions based simply on frequency of handoff sequence occurrences. Then, Section IV.B discusses the application of time-series analysis on handoff occurrences to formulate a better model based on user behavior to improve the Next-AP prediction accuracy.

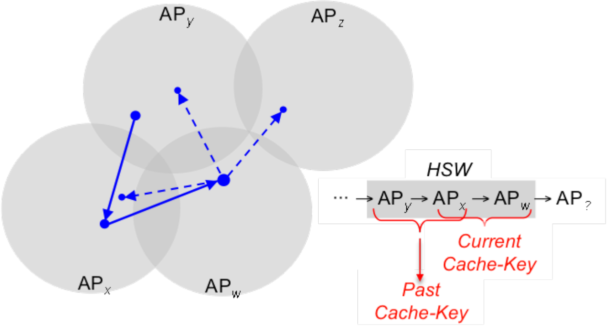


Fig. 3. Local history using HSW for $k$=3.

TABLE I
Global History in the Path-Cache for Figure 3.

| Cache-Key | | Next-AP | Counter |
|---|---|---|---|
| Past-AP | Current-AP | | |
| $AP_x$ | $AP_y$ | $AP_x$ | 6 |
| $AP_x$ | $AP_y$ | $AP_w$ | 2 |
| $AP_x$ | $AP_y$ | $AP_z$ | 10 |
| ··· | ··· | ··· | ··· |
| $AP_y$ | $AP_x$ | $AP_y$ | 6 |

### A. The Basic GPC Scheme

The basic idea behind GPC is to track past mobility patterns and then use this information to predict future handoffs. In order to illustrate the motivation behind GPC, Figure 3 shows an example of a coverage area that contains four APs. As the MS moves away from $AP_w$, it is unclear which AP it will attach to next since there are three possible candidates (*i.e.*, $AP_x$, $AP_y$ or $AP_z$). Therefore, the history of handoff sequences is maintained and used to predict behavior of future handoffs.

In order to keep track of a MS's handoff sequence, a *local history* is maintained using a $k$-entry *Handoff-Sequence Window* (HSW) containing information of the current AP as well as $k-1$ past APs (*i.e.*, the MAC address and the channel number). Figure 3 illustrates HSW for $k$=3. A MS joining the network for the first time has no local history and thus its HSW contains null entries. When the MS associates with a cell, the information of the current AP is queued in HSW. During each subsequent handoff, the MS sends to the server a *Path-Cache request* containing HSW as part of an authentication request.

When the server receives path-cache requests from MSs, a *global history* of all the MSs in the network is maintained in the *Path-Cache*, where each entry contains a *Cache Key* represented by *Current-AP* and $k-2$ *Past-APs*, *Next-AP*, and a *Counter* indicating the number of hits on this entry. Table I shows the partial content of the Path-Cache for Figure 3.

The following operations are performed when a MS sends a Path-Cache request to the server. Note that this process is initiated when the MS senses the signal strength of the current AP to be weaker than a certain threshold.

- *Path-Cache update* - The server uses the *past cache-key* represented by the handoff sequence $AP_0$, $AP_1$, $\cdots$, $AP_{k-2}$ in HSW to search in the Path-Cache for a matching Cache-Key. If a match is found, a check is made to see if $AP_{k-1}$ also matches the Next-AP entry. If it matches, the server increments the counter for that entry by one. If the server does not find a match, it means the HSW is new. Therefore, the server stores the new handoff sequence in the Path-Cache and initializes its counter to one.
- *Next-AP prediction* - The server uses the *current cache-key* represented by the handoff sequence $AP_1$, $AP_2$, $\cdots$, $AP_{k-1}$ in HSW to search in the Path-Cache for a matching Cache-Key. If a match or multiple matches are found, the server sends to MS a *Path-Cache response* with a list of Next-AP predictions sorted in descending order of their counter values as part of an authentication response. Otherwise, a null Next-AP prediction is sent back to notify of Path-Cache miss. If the HSW in the Path-Cache request is null, it indicates the MS is joining the network for the first time. Therefore, the servers uses a special handoff sequence $null_1$, $null_2$, $\cdots$, $AP_{tuned-in}$, where $AP_{tuned-in}$ represents the current AP the MS is tuned into, to search in the Path-Cache.

Note that the size of $k$ depends on the complexity of the network topology and the building structure. If the coverage area is small and yet there are many APs, a longer handoff history will be preferred. However, our study shows that in general $k$=3 is sufficient to provide a good next-AP prediction. In addition, all the Path-Cache entry counters are periodically decremented to prevent saturation.

The algorithm for the GPC technique is illustrated in Figure 4 based on the assumption that the next-AP prediction has been determined from the previous handoff and both the Path-Cache and Authentication servers are collocated:

Step 1: MS directly tunes into the AP provided by the Next-AP prediction. If Next-AP prediction is null, MS performs a full-scan and tunes into the AP with the strongest signal.

Step 2: MS sends authentication request, *Auth_Req*, containing Path-Cache request, *PC_Req(HSW)*, to the server to obtain Next-AP predictions for the next handoff (1).

Step 3: If authentication is successful, the server performs Path-Cache Update (2) and Next-AP Prediction (3) based on the received HSW, and sends authentication response, *Auth_Resp*, containing Path-Cache response, *PC_Resp(Predicted_Next-AP)* (4). Otherwise, choose the next element in the Next-AP prediction list and go to Step 1.

Step 4: MS sends reassociation request (5) to the AP and receives reassociation response (Step 6). If no reassociation response is received, move to the next element in the Next-AP prediction list and go to Step 1.

Step 5: Information of the new AP is queued in HSW (7).

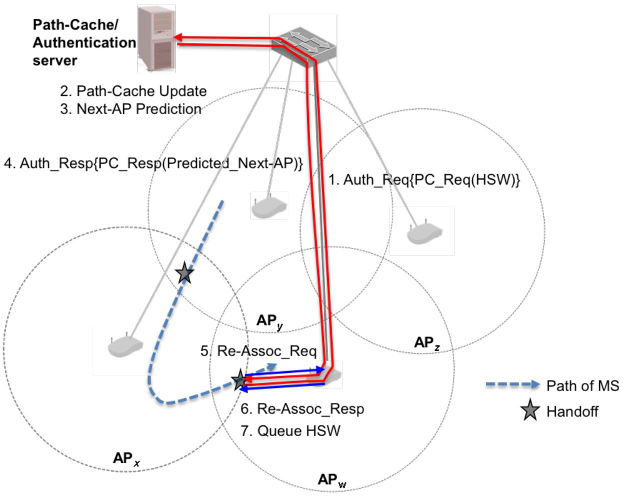


Fig. 4. The steps in the GPC Technique.

If a Path-Cache request hits on the Path-Cache and its 1st Next-AP prediction is successful, GPC will reduce overall handoff delay down to only the time required for MS to perform a channel switch plus authentication and reassociation. With each additional Next-AP misprediction, the overall handoff delay increases incrementally by the channel switching time plus authentication timeout period. For example, if the 1st Next-AP prediction fails but the 2nd Next-AP prediction is successful, MS first tunes into the first predicted Next-AP and waits until the authentication times out, then tunes into the second predicted Next-AP.

In case of a Next-AP misprediction, or authentication failure, MS will revert back to the conventional handoff, which requires a full scan. A Path-Cache miss will occur if a handoff sequence is encountered for the first time. Afterwards, the new sequence will be recorded in the Path-Cache and used to predict future handoffs. Therefore, as long as the Path-Cache is current, all MSs can benefit from this information to provide fast handoffs. Finally, note that Path-Cache requests/responses are piggy-backed on authentication requests/responses. Therefore, no extra messages are needed. Also note that the discussion of GPC thus far has been based on a centralized scheme. However, GPC can also be implemented using a distributed scheme where each AP maintains its own portion of the global Path-Cache.

### B. *Time-Series Based Prediction Model for GPC*

The previous subsection discussed how the basic GPC scheme uses the handoff history to effectively predict Next-APs. However, the returned Next-AP predictions are prioritized based on long-term frequency of handoff sequences using counters. However, these counters are unable to capture short-term and periodic handoff behaviors that are crucial for improving Next-AP predictions for all scenarios. This is addressed by treating the frequency of handoff sequences as a time-series data using *AutoRegressive Integrated Moving Average* (ARIMA) and *Exponential Weight Moving Average* (EWMA).

Our analysis of the time-series data shows that the predicted frequency of handoff sequence $\overline{z}_{t+1}$ for $AP_4 \rightarrow AP_5 \rightarrow AP_6$ can be represented by ARIMA(0, 2, 2) as shown below (see Appendix)

*1) ARIMA Based Prediction Model for GPC:* ARIMA is known to work well for non-stationary processes [30], [31], and has been used to model automotive traffic flow [32], [33] and mobility prediction [19], [20]. The frequency of handoff sequence is treated as a time-series data where the basic discrete time interval *t* is one minute. This archival data series can be aggregated to generate longer time intervals as needed. The period of the handoff data series *T* depends on the system under study.

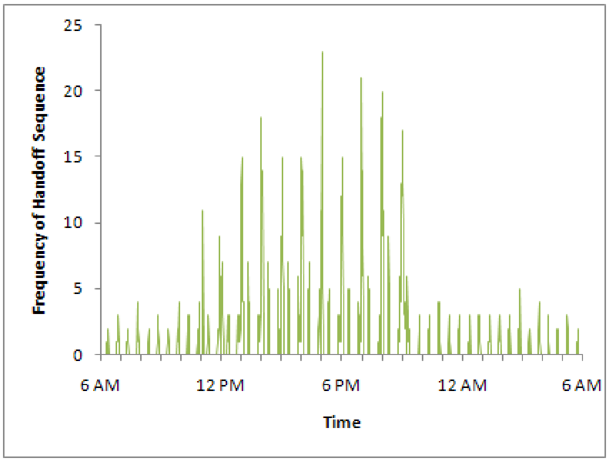


Fig. 5. Frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ for KEC.

For a typical WLAN environment, such as ours, the recommended period will be at least one day to capture all possible trends within a day. Figure 5 shows an example time-series data representing a simulated user mobility for the handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ in the KEC building (see Section V.A), which shows that there are more handoff activities between 11 AM to 9 PM than 10 PM to 10 AM. Our analysis of the time-series data shows that the predicted frequency of handoff sequence $\overline{z}_{t+1}$ for $AP_4 \rightarrow AP_5 \rightarrow AP_6$ can be represented by ARIMA(0, 2, 2) as shown below (see Appendix)

$$\overline{z}_{t+1} = 0.0217z_t - 0.0216z_{t-1} + 1.9783\overline{z}_t - 0.9784\overline{z}_{t-1}$$

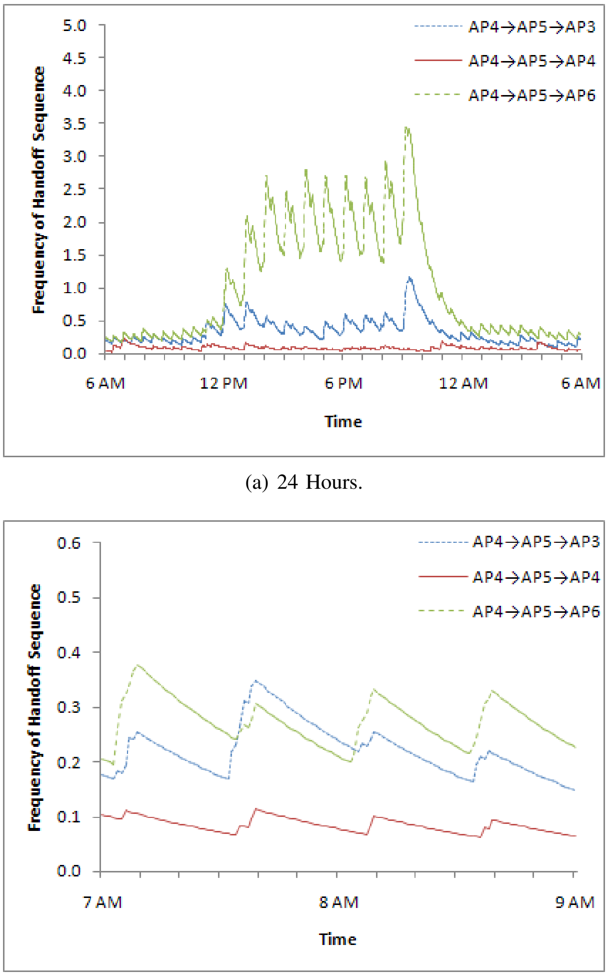
where $z_t$ and $z_{t-1}$ are the sampled time-series data, $\overline{z}_t$ and $\overline{z}_{t-1}$ are the predicted time-series data.

Figure 6 shows the plot of predicted frequency of handoff sequences $AP_4 \rightarrow AP_5 \rightarrow AP_3$, $AP_4 \rightarrow AP_5 \rightarrow AP_4$, and $AP_4 \rightarrow AP_5 \rightarrow AP_6$ using the ARIMA model, which represent the three possible paths through $AP_4 \rightarrow AP_5$. This figure shows that in general the handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ occurs the most often. One of the advantages of GPC based on ARIMA is that it can better track of short-term changes in the mobility pattern. They occur when the frequencies of handoff sequences are relatively close together as in Figure 6(a) between 12 AM to 11 AM. For example, Figure 6(b) shows a magnified view of the frequency of handoff sequences between 7 to 9 AM of Figure 6(a). The ARIMA model is able to determine that the frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_3$ overtakes the frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ and becomes the highest around 7:40 AM. Even a small increase in handoff activities can cause the mobility prediction to change. Therefore, GPC based on ARIMA correctly provide $AP_3$ as the $1^{st}$ Next-AP prediction. However, the basic GPC scheme based only on long-term history cannot capture this short term variations causing mispredictions.

2) *EWMA Based Prediction Model for GPC:* EWMA is equivalent to ARIMA(0, 1, 1) [29, 30] and is much simpler to formulate than the general ARIMA model. EWMA can be defined as

$$\overline{z}_{t+1} = (1 - \lambda)\overline{z}_t + \lambda z_t$$

where $z_t$ is the sampled time-series data, $\overline{z}_t$ is the predicted time-series data, and $\lambda$ is the smoothing factor $0 < \lambda < 1$. The parameter $\lambda$ determines characteristic of the EWMA model and is typically chosen experimentally. Based on our analysis, $\lambda$ for the time-series data representing frequency of handoff sequences in KEC is chosen to be 0.1. Figure 7(a) shows the plot of predicted frequency of handoff sequences for $AP_4 \rightarrow AP_5 \rightarrow AP_3$, $AP_4 \rightarrow AP_5 \rightarrow AP_4$, and $AP_4 \rightarrow AP_5 \rightarrow AP_6$ using the EWMA model. Figure 7(b) shows that EWMA, despite some noise, is also able to capture the fact that frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_3$ becomes the highest around 7:40 AM. Although EWMA does not rely on the full statistical analysis to estimate the order and the coefficients, our simulation result show that this simple model gives results that are relatively close to ones from ARIMA.



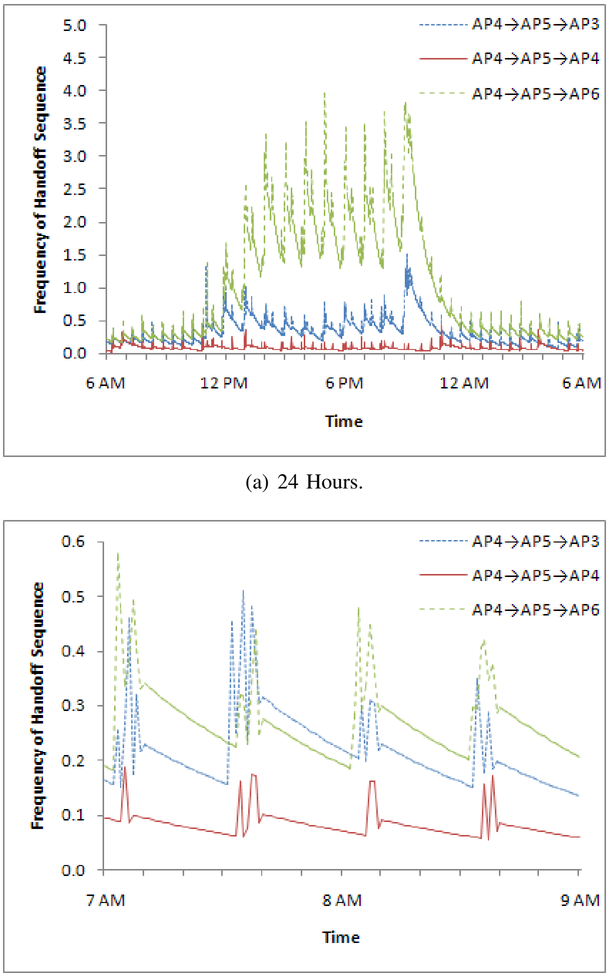
(a) 24 Hours.

(b) 7 AM to 9 AM.

Fig. 6. Predicted Frequency of Handoff Sequences based on ARIMA(0, 2, 2) for KEC.

## V. Performance Evaluation

This section presents the performance evaluation of the proposed GPC technique. Section V.A describes the simulation environment as well as the two key components of the simulator - *path generator* and *handoff detector*. Section V.B discusses the delay parameters used in the study. Section V.C compares the results of the basic GPC scheme against the *Selective Scan with Caching* (SSwC) [11] and *Neighbor Graph* (NG) [10], [34]



(a) 24 Hours.

(b) 7 AM to 9 AM.

Fig. 7. Predicted Frequency of Handoff Sequences based on EWMA for KEC.

techniques, as well as presents the performance improvement using the ARIMA and EWMA models.

### *A. Simulation Environment*

The two network topologies used in the simulation study are the coverage areas for the KEC building and part of Portland (indicated by a dotted line) as shown in Figures 1(a) and (b), respectively. The simulated coverage area for KEC contains 6 APs and 450 MSs, while the coverage area for Portland contains 40 APs and 4,500 MSs. The paths taken by MSs are limited to hallways and the atrium in KEC and sidewalks in Portland. There are three groups of users for KEC, *i.e.*, 200 students, 200 graduate students, and 50 staff members, with each having different types of mobility behaviors. For example, *students* mostly move between the atrium, the cafe, and the computer lab. In addition, students move in and out of the classrooms during the last ten minutes of each class hour between 8 AM and 6 PM. In contrast, *graduate students* mainly move between their offices, the atrium and the computer lab. Finally, *staff* moves mainly between their offices and the atrium.

The results for Portland were generated based on nine different groups with each group consisting of 500 users. *Nomadic* represents a group of MSs that can move anywhere within the simulated area. The next four groups represent *commuters* (C) who work in each of the four quadrants or regions, *i.e.*, C-I C-II, C-III, and C-IV in Figure 1(b), which are likely to travel long distances (*i.e.*, 15-20 blocks) to work. Moreover, these groups of MSs only move between 6 AM to 10 AM and 6 PM to 10 PM. The last four groups represent *residents* (R) who live in each of the four regions, *i.e.*, R-I, R-II, R-III, and R-IV in Figure 1(b). These groups of MSs can move anytime but are likely to only move within few blocks (5-10 blocks) from their homes.

In order to accurately simulate mobility patterns and handoffs, we developed our own simulator that implements a WLAN radio model, generates mobility patterns based on building and city layouts, and supports management frames (which is currently not supported in exiting network simulators, such as ns-2) needed to implement scanning, authentication, and reassociation. The two main modules of the simulator are the *path generator* and the *handoff detector*. For each MS, the path generator randomly selects a location within the preassigned region on the network topology at a predefined time, then uses the path-finder algorithm [35] to generate a path for MS. The handoff detector monitors a MS's movement and performs a handoff when the distance between the MS and the associated AP reaches the maximum radius of the coverage area , which is based on log-distance path loss model [25]. This process is performed at a resolution of one meter. The handoff detector records the number of channel switches, the number of times MS has to wait for $t_{max}$, $t_{min}$, $t_{auth}$, and $t_{assoc}$ (see Section V.B). The simulation steps are described below:

Step 0: Initially, each MS is assigned to a random location within a predefined region. Then, a full scan is performed to choose an AP to associate with.

Step 1: For each MS, a destination location is randomly selected within a predefined region at a predefine time.

Step 2: For each MS, a moving path is generated between its current location and the next location in one-meter increments.

Step 3: For each one-meter step of a MS's movement, the distance is determined between the MS and the current AP. If the distance reaches the maximum radius of the coverage cell, handoff is performed. If the number of handoffs is equal to the maximum number of handoffs, stop simulation. Otherwise, go to Step 1.

### *B. Simulation Delay Parameters*

The delay parameters used in the simulation are shown in Table II: *Channel Switching Time* ($t_{switch}$) is the time required to switch from one channel to another; *MinChannelTime* ($t_{min}$) is the minimum amount of time a MS has to wait on an empty channel; *MaxChannelTime* ($t_{max}$) is the maximum amount of time a MS has to wait to collect all the probe responses, which is used when a response is received within MinChannelTime; *Authentication delay/timeout* ($t_{auth}$) is the time required to perform authentication based on MAC addresses; and *Reassociation delay* ($t_{assoc}$) is the time requires to perform reassocation.

The Parameter Set 1 represents the current off-the-shelf NICs, and was obtained using an experimental setup that consisted of two laptops with PCMCIA 802.11a/b/g NICs based on Atheros AR 5002X chipsets [36] (running Linux 2.6 on Laptop #1 as a traffic generator and FreeBSD 6.1 on Laptop #2 as a traffic

TABLE II
DELAY PARAMETERS USED IN THE SIMULATION.

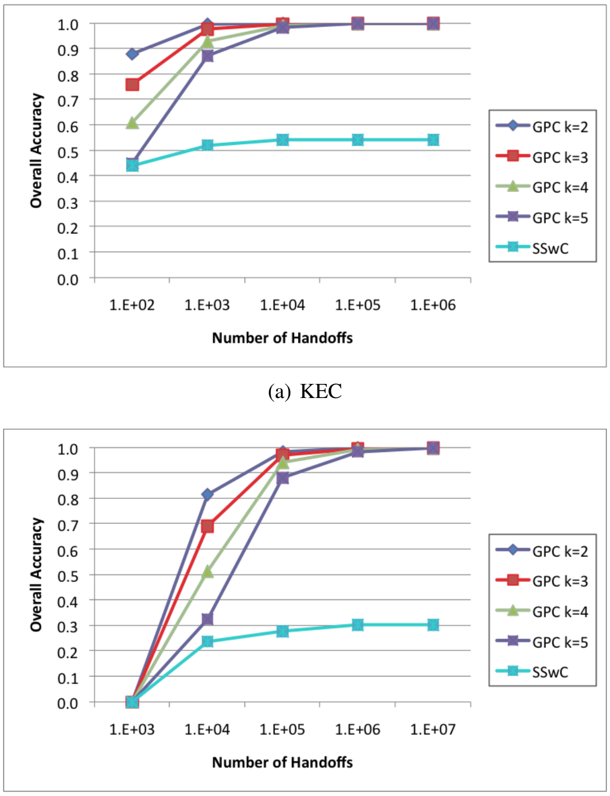
| Parameters | Set 1<br>(Measured) | Set 2<br>(Optimized) |
|---|---|---|
| Channel Switching Time ($t_{switch}$) | 11.4 ms | 11.4 ms |
| MinChannelTime ($t_{min}$) | 20 ms | 1 ms |
| MaxChannelTime ($t_{max}$) | 200 ms | 10 ms |
| Authentication delay ($t_{auth}$) | 6 ms | 6 ms |
| Reassociation delay ($t_{reassoc}$) | 4 ms | 4 ms |

observer), a Sun SPARC Server with Ethernet LAN NIC (running SunOS 5.1), and an HP ProCurve Wireless Access Point 420. The NICs on the AP and on both laptops are operating on Ch. 1. Measurements were obtained by having the first laptop transmit a stream of 16-byte UDP packets to the server, while tcpdump running on the second laptop sniffs the traffic. $t_{switch}$ was determined by forcing the NIC on the first laptop to switch to Ch. 2, which has no APs, and then immediately switch back to Ch. 1. The observed time between the last UDP packet and the probe request from the first laptop was 22.8 ms, which represents $2 \cdot t_{switch}$, and thus $t_{switch}$ is assumed to be 11.4 ms. $t_{auth}$ was determined by measuring the longest possible time between an authentication request and response. Our experiment shows that the MS receives an authentication response within approximately 1$\sim$5 ms. Therefore, $t_{auth}$=6 ms ensures that it is longer than the time between the authentication request and response. Similarly, $t_{assoc}$ is estimated from the average round-trip time of reassociation request and response, which is $t_{assoc}$=4 ms. $t_{max}$ was estimated by observing the time between a probe request and an authentication request, which is 199.4 ms. This is consistent with the $t_{max}$ value provided in the source code of the open source wireless network device driver [35]; therefore, $t_{max}$ is assumed to be 200 ms. On the other hand, there is no direct method to measure $t_{min}$. Thus, the reference value of $t_{min}$ = 20 ms is assumed as in [37]. The delay values were obtained from average of 2400 measurements over a period of a day to reduce variations due to network traffic.

The Parameter Set 2 represents possible future NICs with reduced handoff delays based on optimized $t_{min}$ and $t_{max}$ values from [27]. This study determined that the value of $t_{min}$ that leads to minimized handoff delay are given by $t_{min} \geq DIFS + (aCWmin \times aSlotTime)$ [27], where $DIFS$ is Distributed Inter-Frame Space, $aCWmin$ is the number of slots in the minimum contention window, and $aSlotTime$ is the length of a slot. In the IEEE 802.11g standard [2], the values for $DIFS$, $aCWmin$, and $aSlotTime$ are 28 $\mu$s, 15 $\mu$s, and 9 $\mu$s, respectively, which results in $t_{min} \geq 163$ $\mu$s. However, $t_{min}$ is defined in terms of Time Units (TU), where 1 TU = 1024 $\mu$s. Therefore, the smallest possible value of $t_{min}$ is 1024 $\mu$s. Moreover, $t_{max}$ is estimated as the transmission delay required when 10 MSs try to access the same AP. In their simulation [27], the bit rate of the channel is set to 2 Mbps, which is the maximum possible rate for management frames. The same bit rate for control frame also applies to IEEE 802.11g [2], [37]. Therefore, the estimated $t_{max}$ is 10 ms.

### *C. Simulation Results*

This subsection compares the performance of GPC against Selective Scan with Caching (SSwC) [11] and Neighbor Graphs (NG) [10] described in Section III.B. We first investigates the amount of handoff history needed to provide accurate Next-AP predictions. Then, the basic GPC scheme is compared against SSwC and NG in terms of Next-AP accuracy and handoff delay. Finally, we show the performance gains by adopting time-series based prediction models for GPC.



(a) KEC

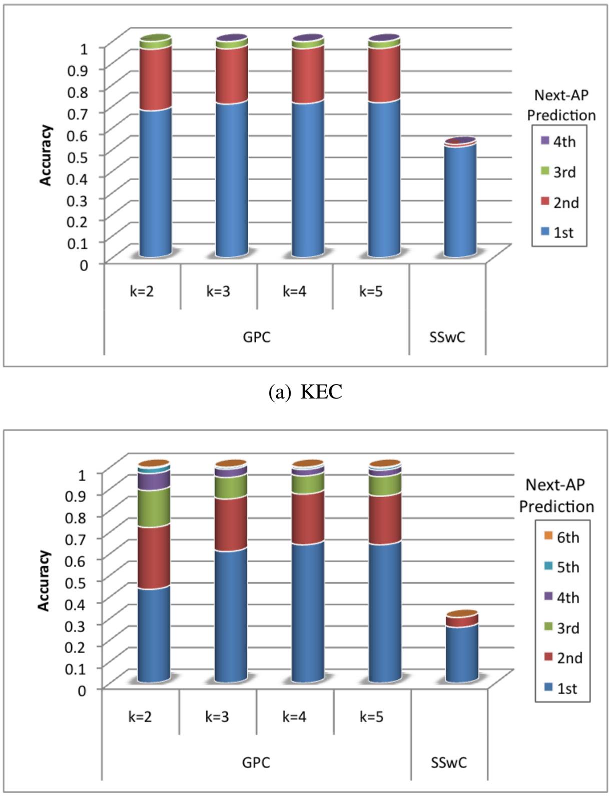
(b) Portland

Fig. 8. Overall Next-AP accuracy as function of history or number of handoffs. (NG is not included because it does not provide Next-AP predictons. At least $10^4$ and $10^6$ handoffs are needed in KEC and Portland, respectively, for system initialization.)

In order to provide a fair comparison, SSwC was extended to have an unlimited number of AP-Cache entries and Next-AP predictions per entry. Note that the original SSwC algorithm assumes only 10 AP-Cache entries and two Next-AP predictions per entry (*i.e.*, Best AP and $2^{nd}$ Best AP) [11].

*1) Number of handoffs for system initialization:* Figure 8 compares the overall accuracy of GPC and SSwC as function of history, which is represented as the number of handoffs. The overall accuracy is defined as the number of correct predictions divided by the total number of handoffs. The NG technique is not included in this comparison since it does not provide a Next-AP prediction mechanism. As can be seen, when the number of handoffs is low (below $10^4$ in KEC and $10^6$ in Portland), GPC lacks sufficient history and thus, the overall accuracy is below 100% and decreases as $k$ increases. This is because a larger $k$ leads to a larger number of possible handoff sequences, and thus a longer history is required to record all possible handoff sequences.

For the KEC building, the overall accuracy for GPC becomes 100% beyond $10^4$ handoffs because all the possible handoff sequences have been recorded in the Path-Cache. Thus, all path-cache requests will be provided with correct Next-AP predictions. In contrast, the larger Portland area requires at least $10^6$ handoffs



(a) KEC

(b) Portland

Fig. 9. Accuracy of Next-AP Predictions. (GPC reaches a total of 100% prediction hits while SSwC reaches 52% and 30% in KEC and Portland, respectively.)

before the overall accuracy becomes 100%. Although the number of handoffs required is much greater than KEC, Portland has many more MSs. Therefore, 45,000 users in Portland for example can produce $10^6$ handoffs within only ∼3.5 hours.

The overall accuracy of SSwC also increases as function of number of handoffs, but saturates at ∼54% and 31% for KEC and Portland, respectively, as shown in Figures 8(a) and 8(b).

Based on the aforementioned discussion, all the subsequent results in this section were obtained based on the assumption that (1) GPC maintains a complete history of handoff patterns, (2) AP-cache of SSwC contains entries for all the APs in the network, and (3) NG was preconfigured. This is done by first running the simulations for $10^4$ handoffs for KEC and $10^6$ handoffs for Portland to fill up the respective caches and performing NG construction, and then gathering statistics for up to $10^7$ handoffs.

*2) Basic GPC versus SSwC and NG - Prediction accuracy:* Figure 9 compares the accuracies of Next-AP predictions. Again, NG is not included in this discussion. The set of returned predictions is prioritized based on their hit counter values for GPC and signal strengths for SSwC. The significance of these priorities is that each misprediction adds to the overall handoff delay. For GPC, the accuracy for the $1^{st}$ Next-AP prediction for KEC is 68% and increases slightly as function of $k$ as shown in Figure 9(a). The $1^{st}$ Next-AP predictions that fail are satisfied by the $2^{nd}$ Next-AP predictions with accuracy of 89%, which make up 28.5% of all predictions. Similarly, the $3^{rd}$ Next-AP predictions that succeed make up 3.5% of all predictions as in Figure 9(a).

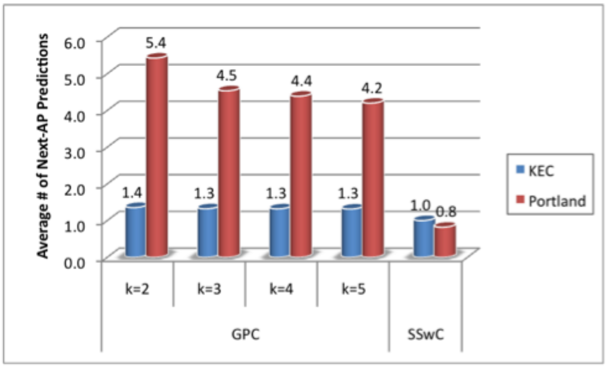


Fig. 10. Average Number of Next-AP Predictions. (Represents the average number adjacent cells in GPC and overlapped cells in SSwC.)

In contrast, SSwC provides significantly lower $1^{st}$ and $2^{nd}$ prediction accuracies of 51% and 2.6%, respectively. For Portland, the $1^{st}$ Next-AP prediction accuracy starts at 43% with GPC and increases slightly as function of $k$ as shown in Figure 9(b), which is similar to the case for the KEC building. In comparison, SSwC provides lower $1^{st}$, $2^{nd}$, and $3^{rd}$ prediction accuracies of 25%, 6% and 0.02%, respectively.

The GPC's superior prediction accuracy is attributed to a larger Next-AP prediction pool (a larger number of cache entries) and its counter-based prediction prioritization. First, the average number of Next-AP predictions returned per handoff is shown in Figure 10. As can be seen, GPC provides a higher average number of Next-AP predictions per handoff than SSwC. In short, SSwC provides at most only two predictions while GPC offers up to four predictions for the KEC building and six predictions for Portland. The reason for this can be explained from the characteristic of overlapped cells. Our simulations show that 40% of the overlapped regions in the KEC building are covered by two cells, and only 5% have three cells. Thus, SSwC will have at most two Next-AP predictions because a MS can connect at most two other APs (besides the current one). In contrast, the maximum number of Next-AP predictions with GPC is four because it depends on the number of adjacent cells. Similarly, 36.1% of the overlapped regions in Portland are covered by two cells, 24.9%, 3.34%, and 0.04% have three, four, and five cells, respectively. Since the area covered by five cells is relatively small, SSwC will have at most three Next-AP predictions (excluding the current AP). In contrast, the maximum number of Next-AP predictions with GPC is six.

This can also be explained by the maximum number cache entries needed, which is shown in Figure 11. The AP-cache used in SSwC requires only 6 and 40 entries, which are the number available APs in the $1^{st}$ floor of the KEC building and Portland, respectively. In contrast, GPC keeps track of MSs' more complex moving paths as $k$ increases and thus requires more entries. Note that the number of entries cannot be compared directly because each entry in the Path-Cache for GPC provides one Next-AP prediction where as each entry in AP-cache for SSwC provides multiple Next-AP predictions.

In addition, the set of returned predictions in GPC is prioritized based on how often these paths are encountered. In contrast, SSwC relies only on signal strength, which is often different from actual paths taken by MSs. Moreover, the AP-cache used in SSwC only caches all the unique APs in the network. Therefore, when

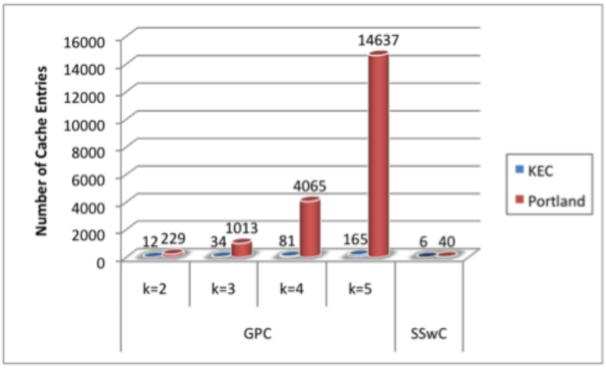
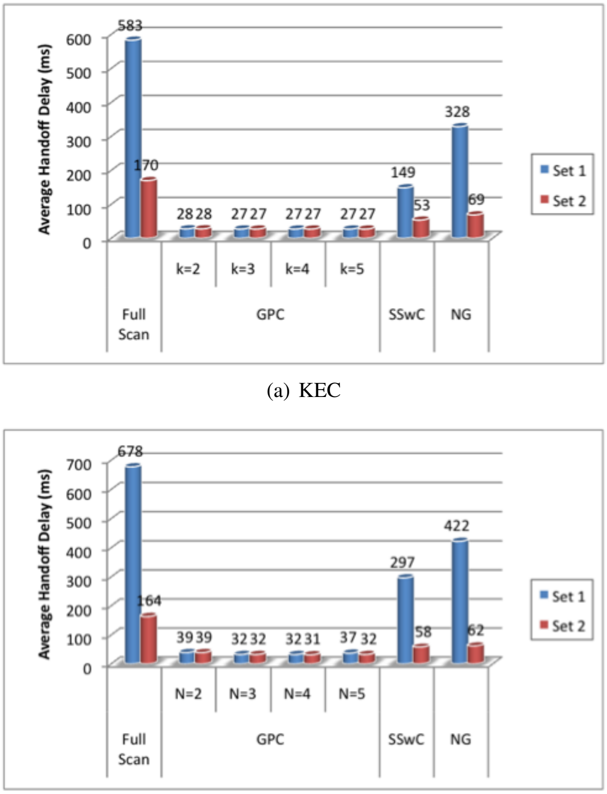


Fig. 11. Number of cache entries. (AP-cache in SSwC and Path-Cache in GPC)



(a) KEC

(b) Portland

Fig. 12. Average Handoff Delay.

an AP with different set of Next-AP predictions is discovered, it overwrites the existing entry, which leads to higher mispredictions as well as larger overall handoff delay.

*3) Basic GPC versus SSwC and NG - Handoff delay:* The mispedictions mentioned above are reflected in the average number of channels probed per handoff. The SSwC scheme probes on average 1.6 and 2.1 channels for KEC and Portland, respectively. This is because Next-AP prediction provided by SSwC has very low accuracy (see Figure 9) that cause 47.7% and 70% of the handoffs in KEC and Portland, respectively, to mispredict and have to rely on selective scanning, which involves selecting the best AP from channels 1, 6, 11, and channels heard from either a previous full scan or selective scan. The average number of channels probed for NG is higher at 2.9 for both topologies, and depends on the number of neighbor nodes encountered at each point-of-attachment. For GPC, the number channels probed per handoff is zero because once the GPC has a complete history it is guaranteed to provide accurate Next-AP predictions.

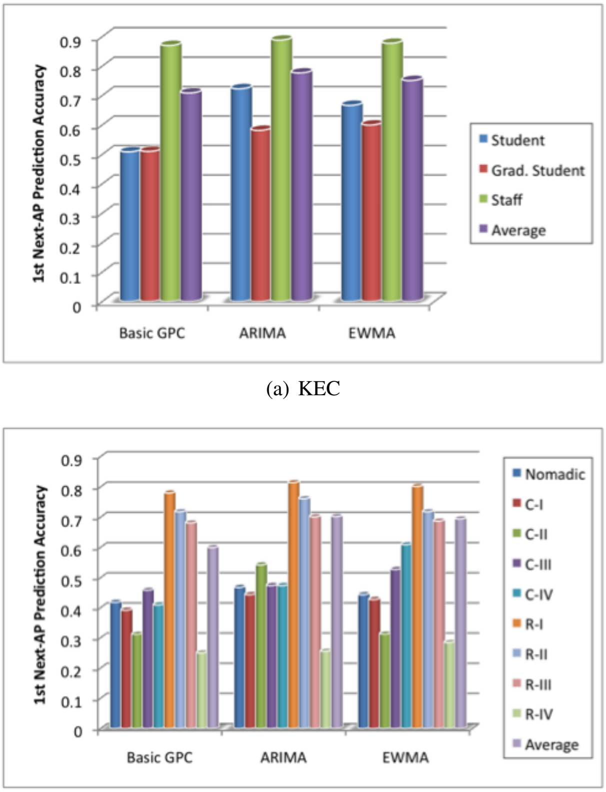
Figure 12 shows the average handoff delays for all three techniques based on the two parameter sets defined in Table II, and includes the result for full scan as a reference. These results show that GPC results in the lowest average handoff delay due to better Next-AP prediction accuracy. Overall, GPC incurs average handoff delay of 27∼28 ms for both parameters sets and is significantly lower than SSwC and NG. Finally, the suggested size for $k$ is 3 because the average handoff delay is relatively constant as $k$ increases beyond 3 and yet it requires only a minimal number of entries in GPC.

*4) Time-Series Based GPC versus SSwC and NG:* Although the basic GPC scheme based on long-term history can significantly reduce the handoff delay, Figure 9 shows that ∼30% and ∼40% of handoffs in KEC and Portland, respectively, require more than one Next-AP prediction. This adds to the handoff delay and illustrates the importance of having highly accurate $1^{st}$ Next-AP prediction. Therefore, Figure 13 compares the $1^{st}$ Next-AP prediction accuracy with $k$=3 using ARIMA(0, 2, 2) and EWMA against the basic GPC scheme. The average improvements using ARIMA for KEC and Portland are 9.6% and 17.1%, respectively. This is because the time-series based GPC properly captures the handoffs caused by short term and periodic behaviors of mobile users. The improvements vary for different users groups. For example, ARIMA improves the $1^{st}$ Next-AP prediction for all three groups in KEC. However, the largest improvement of 42% comes from students because their behaviors are dictated by the class schedules, which causes their handoffs to be periodic and their predictions to become more accurate during those periods. Similarly, all of the user groups in Portland resulted in ∼10% improvement. However, Nomadic and commuter groups (C-I, C-II, C-III, and C-IV) exhibit larger improvements due to short-term surges in handoffs caused by groups of users commuting during rush hour. Next, EWMA resulted in average improvements of 6% and 15.8% for KEC and Portland, respectively, but provided less improvements than the more complex ARIMA since EWMA does not rely on the full statistical analysis to generate the time-series model.

Finally, Figure 14 compares the handoff delays based on the parameter set defined in Table II. Note that both sets of delay parameters yield the same delay results since GPC does not require channel probing after a sufficient amount of history. These results show that GPC with ARIMA provides 4.4% and 8.5% improvement, while EWMA provides 5.6% and 8.5% improvement for KEC and Portland, respectively. This may appear to be only a small improvement compared to the basic GPC scheme, but when individual handoff delays are considered, they resulted in significant improvements for some user groups. For example, the Student group in KEC resulted in 15.2% for ARIMA and 9.1% for EWMA. This was also the case for Portland, where group C-IV, which refers to commuters who work in region IV, resulted in 27% improvement over the basic GPC scheme.

## VI. Conclusion and Future Work

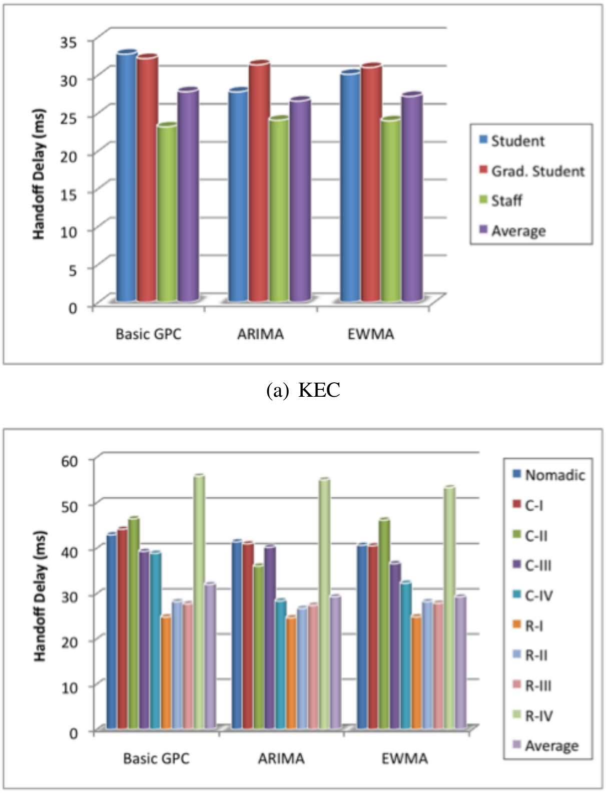
This paper described the GPC technique to minimize the time required to scan for APs in WLANs. GPC is different from the



(a) KEC

(b) Portland

Fig. 13. $1^{st}$ Next-AP prediction Accuracy based on Time-Series Analysis.



(a) KEC

(b) Portland

Fig. 14. Handoff Delay based on Time-Series Analysis. (The results are the same for the two parameter sets because GPC does not probe channels and thus $t_{min}$ and $t_{max}$ are not used.)

other existing methods because it uses global history of handoffs to determine directions of moving MSs. Therefore, it captures the mobility patterns of MSs much like NG and at the same time provides a much more accurate Next-AP predictions than SSwC. Our simulation study shows that the basic GPC scheme eliminates the need to perform scanning and thus results in much lower overall handoff delay compared to the existing techniques. In addition, the time-series based models further reduce the overall handoff delay by increasing the accuracy of $1^{st}$ Next-AP predictions.

For future work, we plan to investigate couple of issues. First, we plan to investigate the effectiveness of GPC in high traffic areas where a large number of packets are lost due to MAC contention. This can cause MSs to be disconnected and require scanning for an alternative AP, which makes it difficult to predict the next-point-of-attachment. Moreover, authentication/reassociation requests may be lost during contention causing multiple requests to be sent and further aggravating the contention problem [37]. Therefore, understanding how GPC will perform under this type of network condition is crucial for properly adjusting some of the parameters, e.g., the timeout period for authentication and reassociation, to reduce the effects of MAC layer contention. Second, we would like to investigate how GPC can be utilized to speed up vertical handoffs.

## Appendix

### A. Derivation of the ARIMA Based Prediction Model for GPC

The order of an ARIMA model is typically denoted by the notation ARIMA($p$, $d$, $q$), where $p$, $d$, and $q$ refer to the order of the *autoregressive*, the *differencing*, and the *moving average* parts of the model, respectively. ARIMA($p$, $d$, $q$) in general can be defined as

$$(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)\nabla^d z_t=(1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q)\varepsilon_t,$$

where $z_t$ is the time-series data, $\phi$ is the autoregressive parameter, $\theta$ is the moving average parameter, $B$ is the *backshift* operator defined by $Bz_t = z_{t-1}$ or $B^m z_t = z_{t-m}$, $\nabla$ is the backward difference operator of the form of $\nabla^d = (1-B)^d$, and $\varepsilon_t$ is white noise. There are two steps involved in formulating the ARIMA model. The first step is the *model identification* based on *autocorrelation function* (ACF) and *partial autocorrelation function* (PACF). The second step is the *model estimation* that determines the parameters $\phi$ and $\theta$ using an estimator algorithm.

The model identification determines the parameters $p$, $d$, and $q$ for the ARIMA model. This process begins with determining whether the time-series data is non-stationary. If so, the differencing transforms the time-series data to become stationary. Some time-series data may require additional differencing, but a typical value for $d$ ranges from 0 to 2. Once $d$ is set, $\nabla^d z_t$ is replaced by a stationary time-series data $x_t$, and ARIMA($p$, $d$, $q$) can be rewritten as

$$(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)x_t = (1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q)\varepsilon_t.$$

The above equation represents a general *AutoRegressive Moving Average* (ARMA) model.

TABLE III
BEHAVIOR OF ACF AND PACF FOR THE ARIMA MODEL.

| | ARIMA<br>($p$, $d$, 0) | ARIMA<br>(0, $d$, $q$) | ARIMA<br>($p$, $d$, $q$) |
|---|---|---|---|
| ACF | Tail off | Cut off after lag $q$ | Tail off |
| PACF | Cutoff after lag $p$ | Tail off | Tail off |

The next step in the model identification is to calculate ACF and PACF of $x_t$. In general, ACF and PACF at lag $h$ are defined as:

$$\begin{aligned} ACF(h) &= corr(x_t, x_{t+h}) \\ PACF(h) &= \begin{cases} corr(x_1, x_0), & h = 1 \\ corr(x_h - x_h^{h-1}, x_0 - x_0^{h-1}), & h \geq 2 \end{cases} \end{aligned}$$

where $corr()$, is the correlation function given by

$$corr(x_t, x_{t+h}) = \frac{cov(x_t, x_{t+h})}{\sigma_x^2} = \frac{E[(x_t - \mu)(x_{t+h} - \mu)]}{\sqrt{E[(x_t - \mu)^2 (x_{t+h} - \mu)^2]}}$$

and $x_h^{h-1}$ and $x_0^{h-1}$ are a $h-1$-term linear regression model defined by $x_h^{h-1} = \beta_1 x_{h-1} + \beta_2 x_{h-2} + \cdots + \beta_{h-1} x_1$ and $x_0^{h-1} = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{h-1} x_{h-1}$, where $\beta_1, \cdots, \beta_{h-1}$ are regression coefficients.

The parameters $p$ and $q$ of ARIMA($p$, $d$, $q$) can be determined by examining the plots for ACF and PACF and applying the criteria defined in Table III. For example, ARIMA(0, $d$, $q$) is chosen when the ACF values are non-zero up to lag $q$ and the PACF values decay exponentially after the first lag. On the other hand, ARIMA($p$, $d$, 0) is chosen when the ACF values decay exponentially after the first lag and the PACF values are non-zero up to lag $p$. Finally, ARIMA($p$, $d$, $q$) is chosen when both ACF and PACF values decay exponentially after the first lag.

After the order of ARIMA is defined, the model estimation determines the parameters $\phi$ and $\theta$. This step typically involves curve fitting, which can be done in many different ways. The method used in our simulation is *Maximum Likelihood Estimator* (MLE). In general, MLE is given by

$$L(\beta) = \prod_{t=2}^{n} f(x_t | x_{t-1} \cdots x_1)$$

where $x$ is Gaussian, $\beta$ is a vector of parameters $\phi$ and $\theta$, and $f(x_t | x_{t-1} \cdots x_1)$ is a conditional density function. The MLE method estimates $\beta$ by finding the value of $\beta$ that maximizes $L(\beta)$.

The following steps show how the time-series data that represents the frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ in Figure 5 can be represented by ARIMA(0, 2, 2). The model identification starts by transforming the time-series data to become stationary. Since the time-series data becomes stationary after the second differencing, parameter $d$ is defined as 2. Then, the transformed time-series data $x_t$ is analyzed using ACF and PACF as shown in Figure 15. Based on the criteria given in Table III, the parameters $p$ and $q$ are defined as 0 and 2, respectively. Therefore, ARIMA(0, 2 ,2) can be rewritten as

$$\nabla^2 z_t = (1 - \theta_1 B - \theta_2 B^2) \varepsilon_t. \tag{1}$$

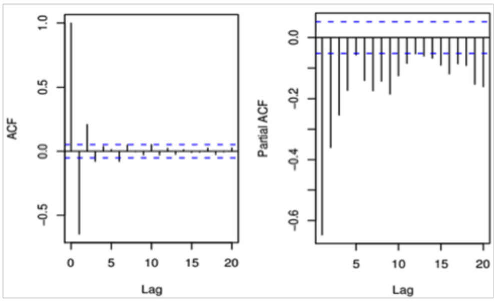


Fig. 15. ACF and PACF from the transformed time-series data in Figure 5.

Finally, the parameters $\theta_1$ and $\theta_2$ are estimated as 1.9783 and -0.9784, respectively, using a graphical method that searches for the maximum $L(\beta)$. Since our goal is to provide a prediction based on past information, the model can be represented as

$$z_t = \sum_{j=1}^{\infty} \pi_j z_{t-j} + \varepsilon_t, \tag{2}$$

where $\pi_j$ is a weighted average coefficient and $\sum_{j=1}^{\infty} \pi_j = 1$

Based on (2), the prediction model $\overline{z}_{t+1}$ can be written as

$$\overline{z}_{t+1} = \sum_{j=1}^{\infty} \pi_j z_{t+1-j}. \tag{3}$$

Next, (2) can be rewritten as

$$\varepsilon_t = (1 - \pi_1 B - \pi_2 B^2 - ...) z_t \tag{4}$$

Using $\varepsilon_t$ from (4), (1) can be rewritten as

$$(1 - 2B + B^2) z_t = (1 - \theta_1 B - \theta_2 B^2)(1 - \pi_1 B - \pi_2 B^2 - ...) z_t \tag{5}$$

From (5), the weighted average coefficients can be defined as $\pi_1 = 2 - \theta_1$, $\pi_2 = \theta_1 \pi_1 - (1 + \theta_2)$ and $\pi_j = \theta_1 \pi_{j-1} + \theta_2 \pi_{j-2}, j \geq 3$. Substituting the coefficient $\pi_j$ into (3) gives

$$\begin{aligned} \overline{z}_{t+1} &= \pi_1 z_t + \pi_2 z_{t-1} + \sum_{j=3}^{\infty} (\theta_1 \pi_{j-1} + \theta_2 \pi_{j-2}) z_{t+1-j} \\ &= \pi_1 z_t + \pi_2 z_{t-1} + [\theta_1 \sum_{j=1}^{\infty} \pi_j z_{t-j} - \theta_1 \pi_1 z_{t-1})] \\ &\quad + \theta_2 \sum_{j=1}^{\infty} \pi_j z_{t-1-j} \end{aligned} \tag{6}$$

Finally, substituting $\sum_{j=1}^{\infty} \pi_j z_{t-j}$ and $\sum_{j=1}^{\infty} \pi_j z_{t-1-j}$ with $\overline{z}_t$ and $\overline{z}_{t-1}$, respectively, (6) can be rewritten as

$$\begin{aligned} \overline{z}_{t+1} &= \pi_1 z_t + (\pi_2 - \theta_1 \pi_1) z_{t-1} + \theta_1 \overline{z}_t + \theta_2 \overline{z}_{t-1} \\ &= (2 - \theta_1) z_t - (1 + \theta_2) z_{t-1} + \theta_1 \overline{z}_t + \theta_2 \overline{z}_{t-1} \end{aligned} \tag{7}$$

## REFERENCES

[1] W. Wanalertlak and B. Lee, "Global path-cache technique for fast handoffs in WLANs," in *International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2007, pp. 45–50.

[2] *Local and Metropolitan Area Network, Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications*, IEEE Std. 802.11, 2007.

[3] Rooftop@Media. [Online]. Available: http://rooftops.media.mit.edu

[4] NYCwireless. [Online]. Available: http://NYCwireless.net

[5] MetroFi Portland Free Wi-Fi. [Online]. Available: http://www.metrofiportland.com

[6] SeattleWireless. [Online]. Available: http://SeattleWireless.net

[7] I. Ramani and S. Savage, "Syncscan: practical fast handoff for 802.11 infrastructure networks," in *IEEE INFOCOM*, Mar. 2005, pp. 675–684.

[8] "ITU-T recommendation G.114," International Telecommunication Union, Tech. Rep., 1993.

[9] V. Brik, A. Mishra, and S. Banerjee, "Eliminating handoff latencies in 802.11 WLANs using multiple radios: applications, experience, and evaluation," in *Internet Measurement Conference (IMC)*, Oct. 2005, pp. 27–27.

[10] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graphs," in *The International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, Jun. 2004, pp. 70–83.

[11] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing mac layer handoff latency in IEEE 802.11 wireless LANs," in *ACM International Workshop on Mobility Management and Wireless Access (MOBIWAC)*, Sep. 2004, pp. 19–26.

[12] S. Waharte, K. Ritzenthaler, and R. Boutaba, "Selective active scanning for fast handoff in WLAN using sensor networks," in *Mobile and Wireless Communications Networks (MWCN)*, Oct. 2004, pp. 59–70.

[13] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain, "Predictability of WLAN mobility and its effects on bandwidth provisioning." in *IEEE INFOCOM*, Apr. 2006, pp. 1–13.

[14] *Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirement - Part 11:Wireless LAN Medium Access Control and Physical Layer Specifications: Amendment: ESS Mesh Networking*, IEEE Unapproved draft Std. P802.11s/D1.02, Mar. 2007.

[15] D. Katsaros, A. Nanopoulos, M. Karakaya, G. Yavas, zgr Ulusoy, and Y. Manolopoulos, *Clustering Mobile Trajectories for Resource Allocation in Mobile Environments*, ser. Lecture Notes in Computer Science. Springer, Sep. 2003, vol. 2779/2003.

[16] J.-M. Franois, "Performing and making use of mobility prediction," Ph.D. dissertation, University of Lige, 2007.

[17] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, Jan. 2001.

[18] S. Pack and Y. Choi, "Fast handoff scheme based on mobility prediction in public wireless LAN systems," in *IEE Proceedings Communications*, vol. 151, no. 5, Oct. 2004, pp. 489–495.

[19] A. Aljadhai and T. F. Znat, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1915–1930, Oct. 2001.

[20] T.-H. Kim, Q. Yang, J.-H. Lee, S.-G. Park, and Y.-S. Shin, "A mobility management technique with simple handover prediction for 3G LTE systems," in *Vehicular Technology Conference (VTC)*, Jun. 2007, pp. 259–263.

[21] W.-S. Soh and H. S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions," in *IEEE INFOCOM*, vol. 4, Mar. 2004, pp. 2766–2777.

[22] H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke, "Personal paging area design based on mobile's moving behaviors," in *IEEE INFOCOM*, vol. 1, Apr. 2001, pp. 21–23.

[23] G. Yavas, D. Katsaros, O. Ulusoy, and Y. Manolopoulos, "A data mining approach for location prediction in mobile environments," *Data and Knowledge Engineering*, vol. 54, no. 2, pp. 121–146, Aug. 2005.

[24] C.-W. You, Y.-C. Chen, J.-R. Chiang, P. Huang, H.-H. Chu, and S.-Y. Lau, "Sensor-enhanced mobility prediction for energy-efficient localization," in *Sensor and Ad Hoc Communications and Networks (SECON)*, vol. 2, Sep. 2006, pp. 565–574.

[25] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2002.

[26] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Computert Communications Review*, vol. 33, no. 2, pp. 93–102, Apr. 2003.

[27] H. Velayos and G. Karlsson, "Techniques to reduce IEEE 802.11b MAC layer handover time," in *IEEE International Conference on Communications (ICC)*, Jun. 2004, pp. 3844–3848.

[28] S. Pal, S. Kundu, and K. Basu. Handoff : Ensuring seamless mobility in IEEE 802.11 wireless networks. [Online]. Available: http://crewman.uta.edu/corenetworking/projects/handoff/newhandoff.html

[29] Y. Amir, C. Danilov, M. Hilsdale, R. Musăloiu-Elefteri, and N. Rivera, "Fast handoff for seamless wireless mesh networks," in *The International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, Jun. 2006, pp. 83–95.

[30] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*, 3rd ed. Prentice Hall, 1994.

[31] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, 2nd ed. Springer, May 2006.

[32] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, Dec. 2003.

[33] G. Yu and C. Zhang, "Switching arima model based forecasting for traffic flow," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004, pp. 429–432.

[34] A. Mishra, M. ho Shin, and W. A. Arhaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *IEEE INFOCOM*, Mar. 2004, pp. 351–361.

[35] Amit's thoughts on path-finding and a*. [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming

[36] Atheros ar5002x 802.11a/b/g universal WLAN solution. [Online]. Available: http://www.atheros.com/pt/AR5002XBulletin.htm

[37] MadWIFI_0.9.2. [Online]. Available: http://www.madwifi.org