Scanless fast handoff technique based on global Path-Cache for WLANs

Scanless Fast Handoff Technique Based on Global Path Cache for WLANs

**Weetit Wanalertlak · Ben Lee · Chansu
Yu · Myungchul Kim · Seung-Min
Park · Won-Tae Kim**

Abstract Wireless LANs (WLANs) have been widely adopted and are more
convenient as they are inter-connected as wireless campus networks and wire-
less 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 scan-
ning. This paper proposes a technique called *Global Path-Cache* (GPC) that
provides fast handoffs in WLANs. GPC properly captures the dynamic be-
havior of the network and MSs, and provides accurate next-AP predictions to
minimize the handoff latency. Moreover, the handoff frequencies are treated

Weetit Wanalertlak and Ben Lee
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR, 97331, USA
Tel.: 1-541-737-3148
E-mail: {wanalewe,benl}@eecs.orst.edu

Chansu Yu
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, OH 44115, USA
and
WCU Division of IT Convergence Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja, Pohang, Gyeongbuk, Korea 790-784
E-mail: c.yu91@csuohio.edu

Myungchul Kim
Department of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
119 Munji-ro, Yuseong-gu, Daejeon, 305-702, Korea
E-mail: mckim@cs.kaist.ac.kr

Seung-Min Park and Won-Tae Kim
Electronics Telecommunications Research Institute (ETRI)
Daejeon, Korea
E-mail: {minpark,wtkim}@etri.re.kr

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.

Keywords Wireless LANs · handoff · channel scanning · mobility prediction · time series analysis

1 Introduction

Wireless communication technology together with the advancements in applications 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 [12], 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 [30, 23, 19, 31].

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 to find the best available AP. Recent studies found that passively scanning for APs during a handoff can be as much as a second [28] and actively scanning for APs requires 350 ms~500 ms [28]. 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 [9].

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 [6, 32, 33, 41]. These methods employ extra hardware, either in the form of additional radios [6] or an overlay sensor network [41] to detect APs, selectively scan channels based on the topological placement of APs [32], and predict the next point-of-attachment based on signal strength [33]. 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 [36]. 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 [42], 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 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 an 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 channels 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 [42] 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 Weighted Moving Average* (EWMA).
- Second, performance evaluation is significantly expanded to include a much larger network (*i.e.*, MetroFi Portland [19]), and analyze the performance effects of different types of users and the improvements provided by the time series-analysis.

Our simulation study shows that the basic GPC scheme results in superior handoff delay compared to *Selective Scan with Caching* (SSwC) [33] and *Neighbor Graph* (NG) [32]. The time-series-based GPC scheme further improves the average 1st 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. Moreover, the handoff latency improvements for some groups of MSs are high as $15.1\% \sim 27.1\%$.

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 basic GPC technique. Section 5 presents the time-series based prediction model for GPC. Section 6 evaluates the performance of the proposed method. Finally, Section 7 concludes the paper and discusses future work.

2 Background - Scanning Process in IEEE 802.11

In the IEEE 802.11 standard, when an MS moves from one cell to another, the network interface senses the degradation of signal quality in the current
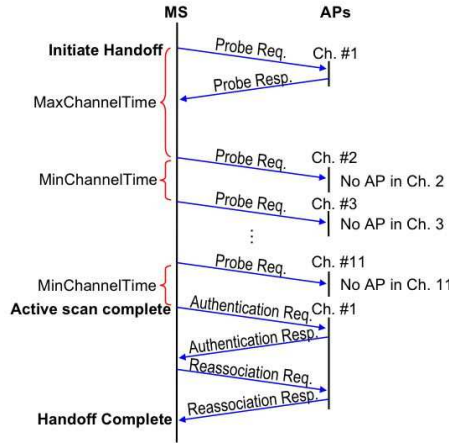
Fig. 1 Active Scanning in IEEE 802.11.

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 [32]. This process starts scanning for new cells 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 a 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 scan all possible channels [28], which results in unacceptable handoff delay.

In *active scanning* shown in Figure 1, an 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 [28].

After scanning, MS typically joins the network with the strongest signal strength, which is done by performing authentication and association/re-association. Authentication is the process that an MS uses to announce its identity to the new AP. In the IEEE 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 the AP. Then, the AP sends the authentication response frame back to the 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 association 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.

3 Related Work

3.1 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 a variety of wireless networks, such as cellular [14, 1, 16, 35, 44, 45], WLANs [10, 26, 33, 36, 8, 15, 24], sensor networks [47] ad hoc networks [38], and wireless mesh networks [13, 3], and applied to reduce handoff latency [33, 24], minimize handoff frequency [15], provide efficient resource reservation [14, 1, 16, 35, 36, 44, 45], improve routing protocols [38], and conserve power [47].

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 to reduce the signaling overhead during handoff [14, 44, 45]. Second, *topology-based* techniques use the knowledge of geographical locations of APs and directional movement of MSs [10, 35]. 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 [1, 16].

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, including methods for WLANs [26], 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 [29].

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 [37, 8]. Typically, a Markov mobility predictor

maintains a collection of past locations of MSs and predicts future locations of MSs based on the value of conditional probability that matches with the past locations of MSs. The Markov-based technique can be found in many mobility prediction algorithms, including ours. However, the proposed GPC scheme is designed to work seamlessly with the 802.11 MAC layer protocol and can be used to enhanced these methods by considering short-term mobility patterns using time-series analysis.

3.2 Reducing Handoff Delay in WLANs

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 [20, 40]. *MultiScan* uses an extra WLAN network interface to opportunistically scan and pre-associate with alternative APs to avoid disconnections [6]. 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. A similar technique called *Make-Before-Break* also uses two WLAN cards, but allows the card that scanned the channels to also perform authentication and association to eliminate MAC layer handoff delay [27]. In contrast, *LeapFrog* uses an extra WLAN interface on the AP side to broadcast beacon messages across all the channels. *Selective Active Scanning* uses an overlay sensor network to obtain information on the presence of APs and the quality of their transmission channels [41]. This way, an MS can broadcast 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 these techniques can provide fast handoffs, they require extra hardware, implemented either on the client side, AP side, or as a separate control plane, which may be impractical and/or power inefficient. The proposed GPC method requires only one WLAN card.

Another technique to reduce the handoff delay is to either passively or actively scan for available APs in the background. *SyncScan* is a passive method that requires APs to send staggered periodic beacons to allow an MS to scan for additional APs while it is still connected to the current AP [28]. In *Smooth Handoff*, an MS actively scans for APs in multiple sub-phases with data transmission in between sub-phases [17]. Although the handoff delay or packet loss can be reduced, there is a hidden cost since an MS has to occasionally suspend its communication to either listen or partially scan 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 delay are *Neighbor Graph* [32], *Pre-Authentication path* [24], *Selective Scan with Caching* [33], *Enhanced FastScan* [26], and *Direction Handoff* [10]. 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 represent handoffs that 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 of these techniques significantly reduce the average number of channels probed, they do not provide next point-of-attachment predictions and thus all active edges (*i.e.*, adjacent channels) emanating from a node need to be scanned.

Selective Scan with Caching minimizes the need to scan 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.

Enhanced FastScan (EFS) reduces scanning delay by restricting the number of candidate APs to scan based on the predicted location of an MS. EFS divides the coverage area of the current AP into four areas: NE, NW, SE, and SW. To determine the area where MS is located, Wi-Fi Positioning System (WPS) is used, and the coordinates of APs are added to beacon frames. However, WPS requires a large amount of scanned data to be processed, which is time consuming, and as stated before WPS may not provide accurate location information due to interference. The Directional Handoff scheme predicts the direction of an MS using a geomagnetic sensor and thus limits the number of APs to scan. However, automatic construction of an AP table is still an issue, and the number of APs to scan will increase when a predicted direction has many APs.

Recently, there has been a growing interest in expanding the coverage area of WLANs using wireless mesh networking. In *SMesh* [2], multiple APs are used to monitor the connectivity quality of MSs in their vicinity to coordinate which one 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 an 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.

4 The Basic GPC Technique

GPC tracks past associated APs and then use this information to perform mobility predictions to reduce the handoff delay. This virtually eliminates the need to scan channels when MSs move through the coverage area of the same set of APs. This section starts off with the discussion of the basic GPC method that prioritizes multiple next-AP predictions based simply on frequency of handoff sequences. Then, Section 5 discusses the application of time-series analysis on handoff occurrences to formulate a better model to improve the next-AP prediction accuracy.

In order to illustrate the motivation behind GPC, Figure 2 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 associate with 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 an 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 2 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 an AP, its information 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 handoffs 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 1 shows the partial content of the Path-Cache for Figure 2.

The following operations are performed when the server receives a Path-Cache request.

- *Path-Cache update* - The server uses the *past cache-key* represented by the handoff sequence $\langle AP_0, AP_1, \cdots, AP_{k-2} \rangle$ in HSW to search in the Path-Cache for a matching Cache-Key. If a match is found, a check is made
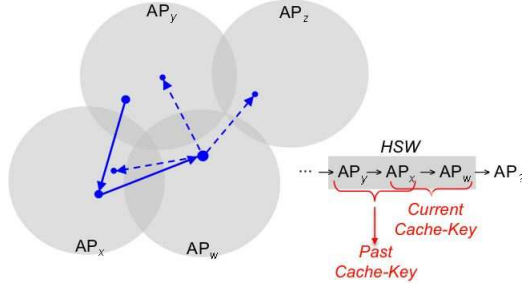
Fig. 2 Local history using HSW for k=3.

Table 1 Global History in the Path-Cache for Figure 2.

Cache-Key		Next-AP	Counter
Past-AP	Current-AP		
AP_x	AP_w	AP_x	6
AP_x	AP_w	AP_y	2
AP_x	AP_w	AP_z	10
$\cdots$	$\cdots$	$\cdots$	$\cdots$
AP_y	AP_x	AP_w	$6 \rightarrow 7$

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 (see Table 1). 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 $\langle AP_1, AP_2, \cdots, AP_{k-1} \rangle$ 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 *prediction list* containing a set 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 a 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 $\langle null_1, null_2, \cdots, AP_{tuned-in} \rangle$, 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 (see Section 6.3). In addition, all the Path-Cache entry counters are periodically decremented to prevent saturation.

The algorithm for the GPC technique is illustrated in Figure 3, where both the Path-Cache and Authentication servers are assumed to be collocated. Each

MS maintains *current* and *future* prediction lists. A MS performs a handoff based on the current prediction list received from the server during the previous handoff and receives the future prediction list for the future handoff. For example, the MS in Figure 3 performs a handoff from AP_x to AP_w based on the prediction list received during handoff from AP_y to AP_x. Also, note that Path-Cache requests/responses are piggy-backed on authentication requests/responses. Therefore, no extra messages are needed.

Step 0: MS selects the first element from the current prediction list as the next-AP prediction.

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. If the end of the current prediction list is reached, MS performs a partial scan of channels not in the current prediction list 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 the future prediction list for the future handoff.

Step 3: If authentication is successful, the server performs Path-Cache Update based on the received HSW. Otherwise, authentication will time out and MS chooses the next element in the current prediction list as the next AP-prediction and go to Step 1.

Step 4: The server performs Next-AP Prediction based on the received HSW and generates the future prediction list for the future handoff (*i.e.*, from AP_w to $AP_?$).

Step 5: The server sends authentication response, *Auth_Resp*, containing Path-Cache response with the future prediction list, *PC_Resp(Predicted_Next-APs)* to the MS.

Step 6: MS sends reassociation request to the AP and receives reassociation response. If no reassociation response is received, MS selects the next element in the current prediction list as the next-AP prediction and go to Step 1.

Step 7: Information of the new AP is queued in HSW, and the future prediction list becomes the current prediction list.

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, MS first tunes into and sends an authentication request the first predicted next-AP and waits for an authentication response. If the authentication times out, *i.e.*, the 1st next-AP prediction fails, then it tunes into the second predicted next-AP and sends an authentication request. This process repeats until one of the predictions in the next-AP prediction list provides the correct prediction.
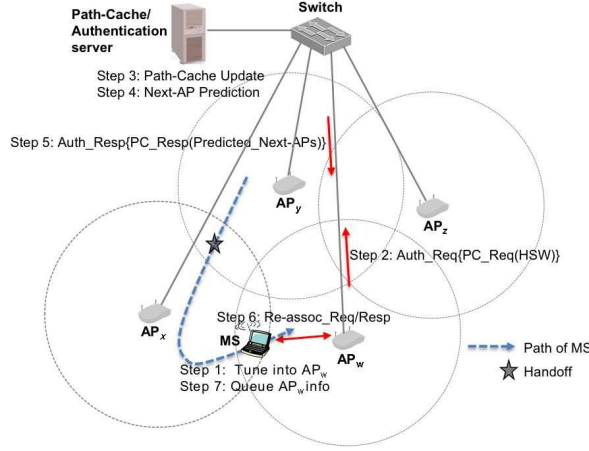
Fig. 3 The steps in the GPC Technique.

If either a Path-Cache miss occurs or all of the next-AP predictions fail, MS will revert back to the conventional handoff requiring a full scan. This happens when a handoff sequence is encountered for the first time. Afterwards, the new handoff sequence will be recorded in the Path-Cache and all MSs can benefit from this information to provide fast handoffs. Therefore, as long as the Path-Cache is up to date, no scanning is necessary since one of the predictions in the next-AP prediction list will provide the correct prediction.

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. This can be achieved by relaying Path-Cache requests using reassociation requests. For example, consider again the handoff between AP_x and AP_w shown in Figure 3. After authentication, MS sends reassociation request containing the Path-Cache request to AP_w. Then, AP_w performs Next-AP Prediction and returns the Path-Cache response as a part of the reassociation response. Finally, AP_w sends HSW as a part of either *Inter-Access Point Protocol* (IAPP) [11] or a vendor specific protocol to AP_x. After AP_x receives HSW, Path-Cache Update is performed. This allows MSs handing off from AP_x to obtain the proper next-AP prediction list.

5 Time-Series Based Prediction Model for GPC

The previous section 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. These counters are unable to capture short-term handoff behaviors that are crucial for improving next-AP predictions for all scenarios. This is

addressed by treating the frequency of handoff sequences as time-series data using *AutoRegressive Integrated Moving Average* (ARIMA) and its simpler form *Exponential Weight Moving Average* (EWMA).

5.1 ARIMA Based Prediction Model for GPC

ARIMA is known to work well for non-stationary processes [5, 34], and has been used to model automotive traffic flow [43, 48] and mobility prediction [1, 16]. The frequency of a handoff sequence z is treated as time-series data where the discrete time interval t is one minute. This archival data series can be aggregated to generate longer time intervals as needed. The observation time period for handoff sequences T depends on the system under study. For a typical WLAN environment, such as ones shown in Figure 10, the recommended period will be at least one day to capture all possible trends within a day.

The order of an ARIMA model is 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. In general, ARIMA(p, d, q) can be defined as

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

where z_t is the observed data (*i.e.*, frequency of handoff sequence) during the current time period t, ϕ is the autoregressive parameter, θ is the moving average parameter, B is the *backshift* operator defined by $B^m z_t = z_{t-m}$, ∇ is the backward difference operator of the form of $\nabla^d = (1 - B)^d$, and ε_t is the error terms.

There are two steps involved in formulating the ARIMA model. The first step is the *model identification* that determines the parameters p, d, and q. The second step is the *model estimation* that determines the parameters ϕ and θ using an estimator algorithm. These two steps will be illustrated using the time-series plot of frequency of handoff sequence shown in Figure 4, which was collected from simulation of user mobility in Figure 10(a) for the handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ during a 24 hour period (see Section 6.1). As can be seen, the figure shows an academic environment where there are more handoff activities between 11 AM to 9 PM than 10 PM to 10 AM. Moreover, these plots are characterized by extremely high fluctuations and sharp peaks during each hour exemplifying bursts of handoff activities based on class schedule.

The model identification begins with determining whether or not the time-series data is stationary. A stationary time-series data has no trend or seasonality. If not, 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. As will be explained shortly, our analysis
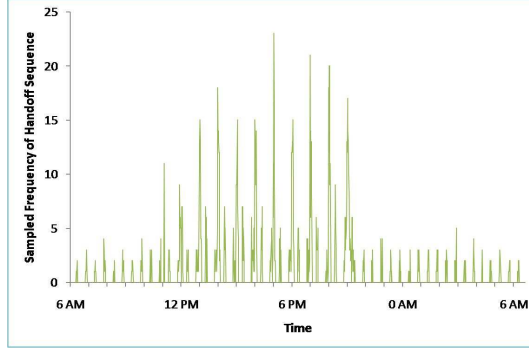
Fig. 4 Sampled frequency of handoff sequence z for $AP_4 \rightarrow AP_5 \rightarrow AP_6$ in KEC.

of the time-series data in Figure 4 becomes stationary after the second differencing, therefore the parameter d is defined as 2. Once d is set, $\nabla^d z_t$ in Eq. 1 is replaced by a stationary time-series data x_t, and ARIMA(p, d, q) can be rewritten as a general *AutoRegressive Moving Average* (ARMA) model shown below.

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

The next step in the model identification is to calculate *autocorrelation function* (ACF) and *partial autocorrelation function* (PACF) of x_t to determine the autoregressive (p) and moving average (q) parts of ARMA(p, q). In general, ACF at lag h is given as

$$ACF(h) = corr(x_t, x_{t+h}), \quad (3)$$

where $corr()$ is the *correlation* function defined as the ratio of *covariance*, $cov()$, and *variance*, σ^2:

$$corr(x_t, x_{t+h}) = \frac{cov(x_t, x_{t+h})}{\sigma^2_{x_t}}$$
$$= \frac{E[(x_t - \mu)(x_{t+h} - \mu)]}{E[(x_t - \mu)^2]}, \quad (4)$$

where μ is the mean. On the other hand, PACF at lag h is defined as:

$$PACF(h) = \begin{cases} corr(x_0, x_1), & h = 1 \\ corr(x_0 - x_0^{h-1}, x_h - x_h^{h-1}), & h \geq 2 \end{cases} \quad (5)$$

where x_0^{h-1} and x_h^{h-1} are estimated from the $(h-1)^{th}$-term linear regression model [34].

Table 2 Behavior of ACF and PACF for the ARMA model.

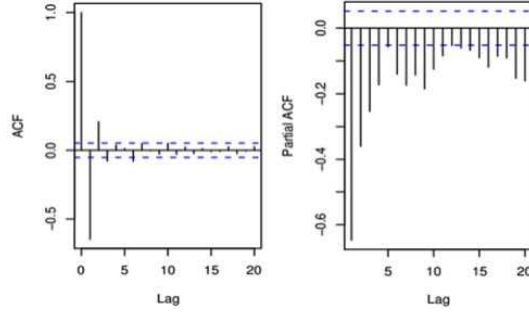|	ARMA $(p, 0)$	ARMA $(0, q)$	ARMA (p, q)
ACF	Exp/Sinu decay	Cut off after lag q	Exp/Sinu decay
PACF	Cutoff after lag p	Exp/Sinu decay	Exp/Sinu decay

Fig. 5 ACF and PACF of the transformed time-series data x_t in Figure 4.

Figure 5 shows the ACF and PACF plots for the transformed time-series data x_t of Figure 4. First, the ACF pattern indicates x_t is stationary since it cuts off fairly quickly. In contrast, ACF values for non-stationary data have very slow decay patterns. The parameters p and q of ARMA(p, q) can be determined by examining the plots for ACF and PACF and applying the criteria defined in Table 2. For example, ARMA$(0, q)$ is chosen when the ACF values cutoff after lag q and the PACF values have exponential or sinusoidal decay. On the other hand, ARMA$(p, 0)$ is chosen when the ACF values have exponential or sinusoidal decay and the PACF values cutoff after lag p. Finally, ARMA(p, q) is chosen when both ACF and PACF values have exponential or sinusoidal decay. Based on Figure 5 and the criteria defined in Table 2, the parameters p and q are identified as 0 and 2, respectively. Therefore, the time-series data in Figure 4 can be modeled as ARIMA$(0, 2, 2)$, which can be rewritten as

$$\nabla^2 z_t = (1 - \theta_1 B - \theta_2 B^2)\varepsilon_t. \tag{6}$$

We then substitute the *one-step-ahead forecast errors* $\varepsilon_t = z_t - \hat{z}_t$ [5], where $\hat{z}_t$ represents the predicted frequency of handoff sequence during the time period t, and $\nabla = (1 - B)$ to obtain

$$\hat{z}_t = (2 - \theta_1)z_{t-1} - (1 + \theta_2)z_{t-2} + \theta_1 \hat{z}_{t-1} + \theta_2 \hat{z}_{t-2}. \tag{7}$$

Therefore, the predicted frequency of handoff sequence during the time period $t+1$, $\hat{z}_{t+1}$, can be written as

$$\hat{z}_{t+1} = (2 - \theta_1)z_t - (1 + \theta_2)z_{t-1} + \theta_1\hat{z}_t + \theta_2\hat{z}_{t-1} \qquad (8)$$

where z_t and z_{t-1} are the current and previous sampled frequencies of handoff sequences, $\hat{z}_t$ and $\hat{z}_{t-1}$ are the current and previous predicted frequencies of handoff sequences, and θ_1 and θ_2 are parameters.

   After the prediction model is defined, the model estimation determines the parameters θ_1 and θ_2. 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

$$L(\beta) = \prod_{t=2}^{n} f(x_t|x_{t-1}\cdots x_1) \qquad (9)$$

where x is Gaussian, β is a vector of parameters ϕ and θ, and $f(x_t|x_{t-1}\cdots x_1)$ is a conditional density function. The MLE method estimates β by finding the value of β that maximizes $L(\beta)$. Using a graphical method that searches for the maximum $L(\beta)$, the parameters θ_1 and θ_2 are estimated as 1.9783 and -0.9784, respectively. Thus, the prediction model of the handoff sequence $AP_4{\rightarrow}AP_5{\rightarrow}AP_6$ is given by

$$\hat{z}_{t+1} = 0.0217z_t - 0.0216z_{t-1} + 1.9783\hat{z}_t - 0.9784\hat{z}_{t-1} \qquad (10)$$

   Figure 6 shows the plot of predicted frequencies for the three handoff sequences $AP_4{\rightarrow}AP_5{\rightarrow}AP_6$, $AP_4{\rightarrow}AP_5{\rightarrow}AP_3$, and $AP_4{\rightarrow}AP_5{\rightarrow}AP_4$ using the ARIMA model. This figure shows that in general the handoff sequence $AP_4{\rightarrow}AP_5{\rightarrow}AP_6$ occurs the most often. The major advantage 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 0 AM to 11 AM. For example, Figure 6(b) shows a magnified view of the frequency of handoff sequences between 7 AM 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 8:20 AM. Even a small increase in handoff activities can cause the mobility prediction to change. Therefore, ARIMA-based GPC correctly provide AP_3 as the 1st next-AP prediction. However, the basic GPC scheme based only on long-term history cannot capture this short term variations causing mispredictions.
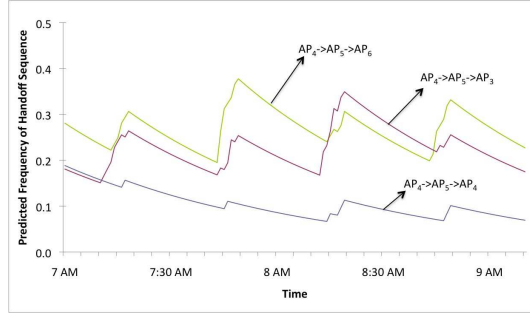
5.2 EWMA Based Prediction Model for GPC

EWMA is equivalent to ARIMA(0, 1, 1) [34, 43] and is much simpler to formulate than the general ARIMA model. EWMA can be defined as

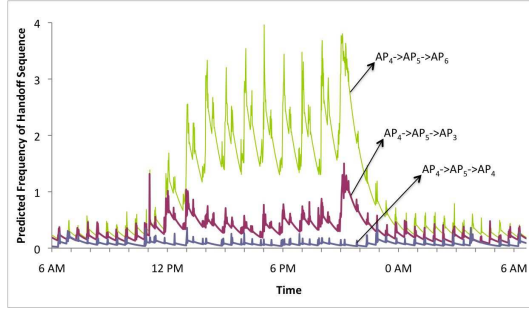$$\hat{z}_{t+1} = (1 - \lambda)\hat{z}_t + \lambda z_t \qquad (11)$$
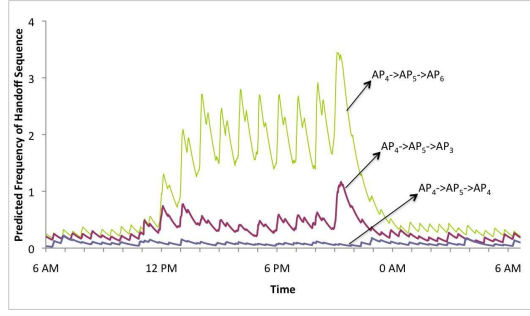
(a) 24 Hours.

(b) 7 AM to 9 AM.

Fig. 6 Predicted frequency of handoff sequences $\hat{z}$ for $AP_4 \rightarrow AP_5 \rightarrow AP_6$, $AP_4 \rightarrow AP_5 \rightarrow AP_3$, and $AP_4 \rightarrow AP_5 \rightarrow AP_4$ based on ARIMA(0, 2, 2) for KEC.

where λ is the smoothing factor $0 < \lambda < 1$. The parameter λ determines characteristic of the EWMA model and is typically chosen experimentally. Based on our analysis, λ 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 Figure 6(a) based on 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 8:15 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.

6 Performance Evaluation

This section presents the performance evaluation of the proposed GPC technique. Section 6.1 describes the simulation environment as well as the various components of the simulator. Section 6.2 discusses the delay parameters used in the study. Section 6.3 compares the results of the basic GPC scheme against the *Selective Scan with Caching* (SSwC) [33] and *Neighbor Graph* (NG) [32, 21]

(a) 24 Hours.
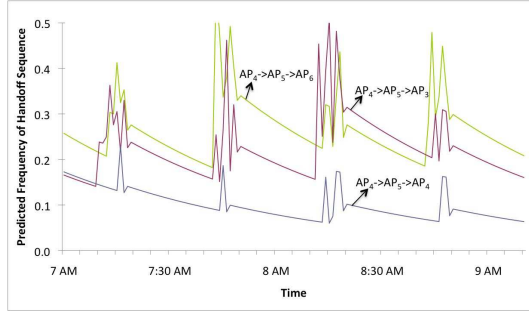
(b) 7 AM to 9 AM.

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.

6.1 Simulation Environment

In order to accurately simulate mobility patterns and handoffs, we developed a simulator that implements a WLAN radio model, generates realistic mobility patterns based on building and city layouts, and supports management frames needed to implement scanning, authentication, and reassociation. The structure of the simulator is shown in Figure 8, which consists of Path Generator, Path Finder, Mobility & Network Simulator, and Data Analysis modules.

The *Path Generator* generates a new destination waypoint based on the Path Graph definition, User Mobility definition, and the current Simulation State for each MS that has completed its trip between the original and destination waypoints. The *Path Graph* definition is a graphical representation of all the possible paths MSs can traverse within a simulated area, which is similar to the ones proposed in [46, 39]. Figure 9 shows the path graphs for the two network topologies used in the simulation study - KEC and Portland in Figure 10. They consist of vertices representing waypoints and segments
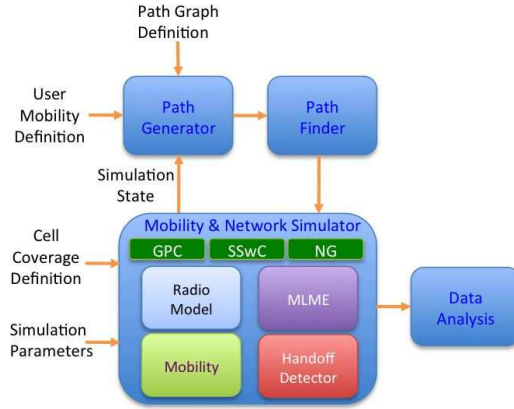
Fig. 8 Simulation model.

representing paths between adjacent waypoints. The *User Mobility* definition defines the number of MSs and APs as well how MSs move, including when and where an MS moves to and how long it stays at a waypoint. The *Simulation State* defines the current time and the locations of all the MSs in the simulated network area. The Path Generator uses these two definitions together with the current state of the simulator to randomly select destination waypoints based on a *modified* Random Waypoint model [7, 22]. Our modified Random Waypoint model allows a probability distribution to be assigned sub-areas or regions within a path graph based on different groups of MSs at different times. The *Path Finder* module then uses the path-finder algorithm [25] to generate the shortest path between the source and destination waypoints. The resulting path consists of multiple segments, which are then fed to the *Mobility & Network Simulator*.

The Mobility & Network Simulator consists of Mobility module, Radio Model, Handoff Detector, and MAC subLayer Management Entity (MLME). The *Mobility* module simulates the movements of MSs on the segments at one meter resolution. The *Handoff Detector* monitors each MS's movement, and based on the *Cell Coverage* definition and the *Radio Model*, which is based on log-distance path loss model [29], performs a handoff when the distance between the MS and the associated AP reaches the maximum radius of the coverage area. The handoff is performed using *MLME* module, which supports beacons, probing, authentication, and re-association. Finally, the *Data Analysis* module 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 6.2).

The two network topologies simulated are shown in Figure 10, which consist of the first floor 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 [19]. The APs in KEC are connected
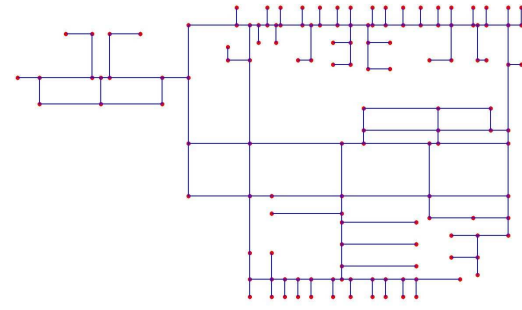
(a) KEC
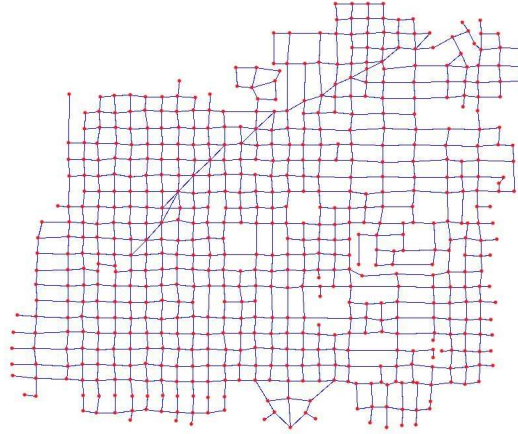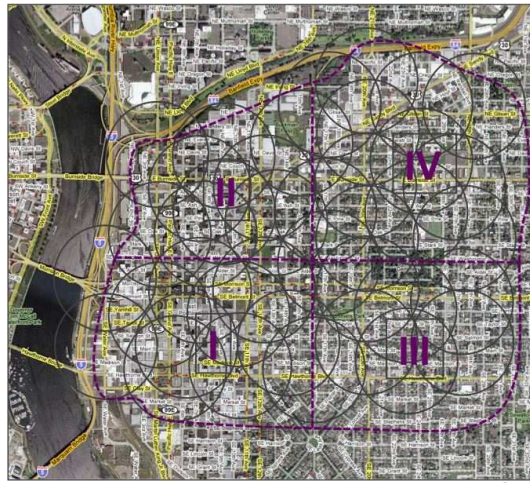

(b) Portland

Fig. 9 Paths graphs for Figure 10

by Ethernet switches, while APs in the MetroFi network are interconnected by a wireless mesh network [13].

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. There are three groups of users in KEC; 200 students, 200 graduate students, and 50 staff members, with each having different types of mobility behaviors. For example, *students* in Figure 10(a) 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
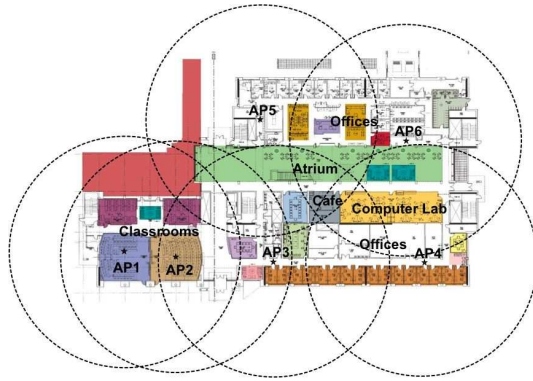
(a) Kelley Engineering Center building.

(b) Public WLAN in Portland, Oregon (MetroFi$^{®}$).

Fig. 10 Example WLAN coverage areas.

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 10(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 10(b). These groups of MSs can move anytime but are likely to only
move within few blocks (5-10 blocks) from their homes.

6.2 Simulation Delay Parameters

The delay parameters used in the simulation are shown in Table 3: *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 an MS has to wait on an empty channel; $MaxChannelTime$ (t_{max}) is the maximum amount of time an 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 PCM-CIA 802.11a/b/g NICs based on Atheros AR 5002X chipsets [4] (running Linux 2.6 on Laptop #1 as a traffic generator and FreeBSD 6.1 on Laptop #2 as a traffic 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∼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 [18]; therefore, t_{max} is assumed to be 200 ms. There are a number of ways to measure t_{min}. One way is to modify the network interface driver of the MS to send a probe request on an empty channel, wait t_{min}, and then send out another probe request on the same channel. This resulted in time between probe requests to be 20.04 ms. Therefore, t_{min} is assumed to be 20 ms, which is again consistent with the default value in the open source wireless network device driver [18]. 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 [40]. This study determined that the value of t_{min} that leads to minimized handoff delay are given by $t_{min} \geq DIFS + (aCWmin \times aSlotTime)$ [40], 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 [12], the values for $DIFS$, $aCWmin$, and $aSlotTime$ are 28 μs, 15 μs, and 9 μs, respectively, which results in $t_{min} \geq 163$ μs. However, t_{min} is defined in terms of Time Units (TU), where 1 TU = 1024 μs. Therefore, the

Table 3 Delay parameters used in the simulation.

Parameters	Set 1 (Measured)	Set 2 (Optimized)
t_{switch}	11.4 ms	11.4 ms
t_{min}	20 ms	1 ms
t_{max}	200 ms	10 ms
t_{auth}	6 ms	6 ms
$t_{reassoc}$	4 ms	4 ms

smallest possible value of t_{min} is 1024 μs. Moreover, t_{max} is estimated as the transmission delay required when 10 MSs try to access the same AP. In their simulation [40], 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 [18, 12]. Therefore, the estimated t_{max} is 10 ms.

6.3 Simulation Results

This subsection compares the performance of GPC against Selective Scan with Caching (SSwC) [33] and Neighbor Graphs (NG) [32] described in Section 3.2. 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 prediction accuracy and handoff delay. Finally, we show the performance gains by adopting time-series based prediction models for GPC.

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 2nd best AP) [33].

6.3.1 Number of handoffs for system initialization

Figure 11 compares the *overall* accuracy of GPC and SSwC as function of history, which is represented as the number of handoffs (shown in legend from left to right). The overall accuracy is defined as the percentage of correct prediction per handoff. In other words, if a Path cache request returns a next-AP prediction list and one of the next-AP prediction is correct then the prediction for the handoff is considered successful. On the other hand, if all the predictions in the next-AP prediction list fail or a null next-AP prediction is received, then the prediction for the handoff is considered unsuccessful. 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
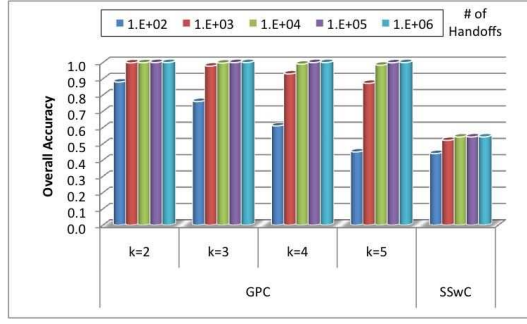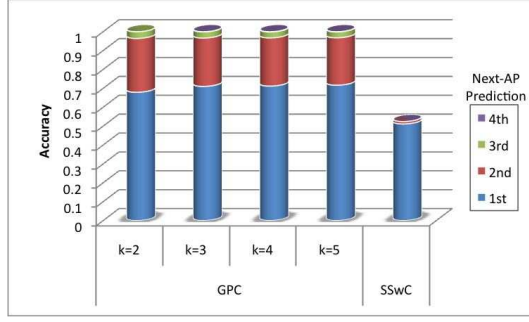
(a) KEC
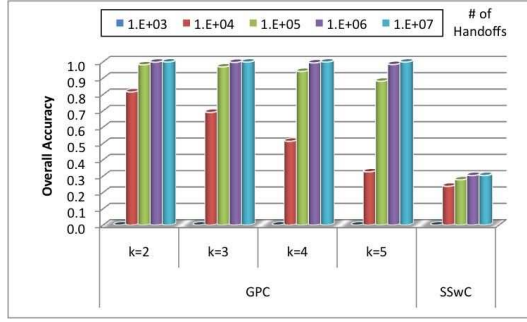
(b) Portland

Fig. 11 Overall next-AP accuracy as function of history or number of handoffs.

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 KEC, 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, each path-cache request will be provided with a next-AP prediction list and one of these predictions will be correct. In contrast, the larger Portland area requires at least 10^6 handoffs before the overall accuracy becomes 100%. Although the number of handoffs required is much greater than KEC, Portland has many more MSs. Therefore, 4,500 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 11(a) and 11(b).

Based on the aforementioned discussion, all the subsequent results in this section were obtained based on the assumption that (1) GPC contains a complete history of handoff patterns, (2) AP-cache of SSwC contains entries for all the APs in the network, and (3) NG is preconfigured. This is done by first running the simulations for 10^4 handoffs for KEC and 10^6 handoffs for Port-

(a) KEC
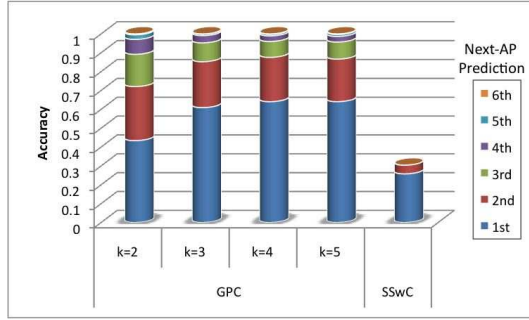
(b) Portland

Fig. 12 Accuracy of individual next-AP predictions.

land to fill up the respective caches and performing NG construction, and then gathering statistics for up to 10^7 handoffs.

6.3.2 Basic GPC versus SSwC and NG - Prediction accuracy

Figure 12 compares the accuracies of *individual* next-AP predictions based on the prediction order (shown in legend from bottom to top starting with the 1st prediction result on the bottom). 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. As mentioned before, the significance of these priorities is that each misprediction adds to the overall handoff delay. For GPC, the accuracy for the 1st next-AP prediction for KEC is 68% and increases slightly as function of k as shown in Figure 12(a). The 1st next-AP predictions that fail are satisfied by the 2nd next-AP predictions with accuracy of 89%, which make up 28.5% of all predictions. Similarly, the 3rd next-AP predictions that succeed make up 3.5% of all predictions as in Figure 12(a).

In contrast, SSwC provides significantly lower 1st and 2nd prediction accuracies of 51% and 2.6%, respectively. For Portland, the 1st next-AP prediction accuracy starts at 43% with GPC and increases slightly as function of k as
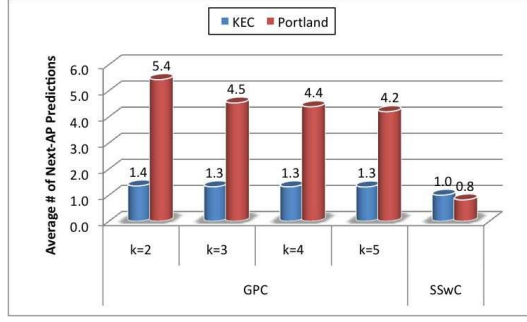
Fig. 13 Average number of next-AP predictions. (Represents the average number adjacent cells in GPC and overlapped cells in SSwC.)

shown in Figure 12(b), which is similar to the case for KEC. In comparison, SSwC provides lower 1st, 2nd, and 3rd next-AP 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. This can be seen by the average number of next-AP predictions returned per handoff shown in Figure 13 (shown in legend with KEC on the left and Portland on the right), which shows that GPC provides a higher average number of next-AP predictions per handoff than SSwC. In short, SSwC provides at most only two and three predictions for KEC and Portland, respectively, while GPC offers up to four and six predictions for KEC and Portland, respectively. The reason for this can be explained from the cell coverage characteristics.

Our simulations show that 40% of the *overlapped* regions in KEC are covered by two cells, and only 5% have three cells. Thus, SSwC will have at most two next-AP predictions because an MS can detect at most two other APs (besides the current AP). In contrast, the maximum number of *adjacent* cells that an MS can handoff to is four, thus maximum number next-AP predictions with GPC is four. Similarly, 36.1% of the overlapped regions in Portland are covered by two cells, and 24.9%, 3.34%, and 0.04% have three, four, and five cells, respectively. Since the area covered by five cells is very small, SSwC will have at most three next-AP predictions. In contrast, the maximum number of adjacent cells, and thus the maximum number of next-AP predictions with GPC, is six.

This can also be explained by the maximum number cache entries needed as shown in Figure 14. In order to provide a fair comparison, each entry for GPC contain multiple next-AP predictions, rather than one prediction per entry as shown in Table 1. The AP-cache used in SSwC requires only 6 and 40 entries, which are the number available APs in the 1st 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.
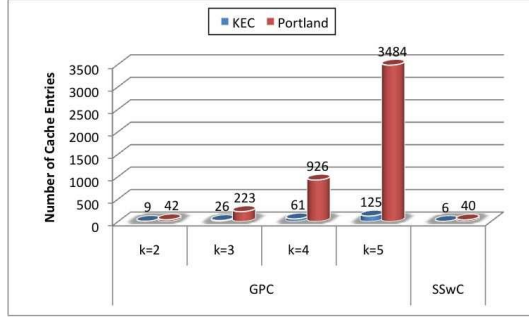
Fig. 14 Number of cache entries. (AP-cache in SSwC and Path-Cache in GPC)

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 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.
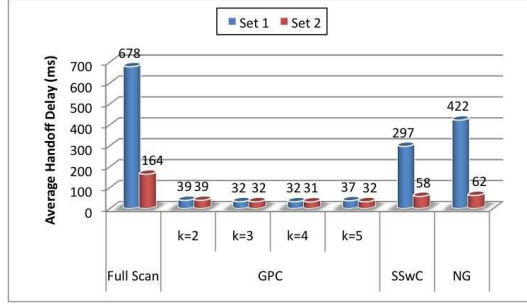
6.3.3 Basic GPC versus SSwC and NG - Handoff delay

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 12) 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 network topologies, and depends on the number of active edges 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 overall next-AP predictions.

Figure 15 shows the average handoff delays for all three techniques based on the two parameter sets defined in Table 3, and includes the result for full scan as a reference. These results show that GPC results in the lowest average handoff delay due to superior 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 (see Figure 14).
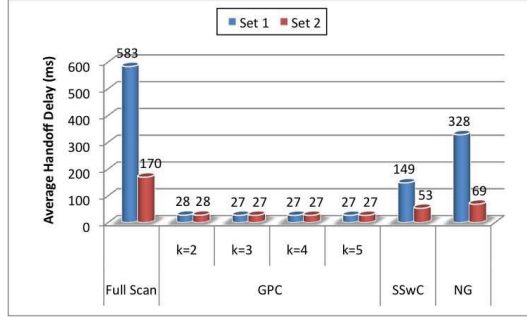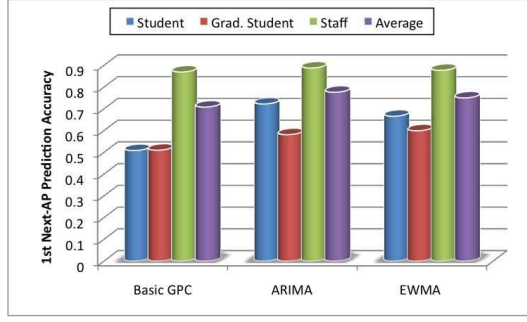
(a) KEC

(b) Portland

Fig. 15 Average handoff helay.

6.3.4 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 12 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 1st next-AP prediction. Therefore, Figure 16 compares the 1st next-AP prediction accuracy with $k=3$ using ARIMA and EWMA against the basic GPC scheme (shown in legend from left to right). The average improvements using ARIMA for KEC and Portland are 9.6% and 17.1%, respectively. This is because the time-series based GPC more accurately captures the handoffs caused by short-term behavior of mobile users as seen in Figures 6(b) and 7(b). As can be seen from Figure 16, the improvements vary for different users groups. For example, ARIMA improves the 1st 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 results in 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 im-
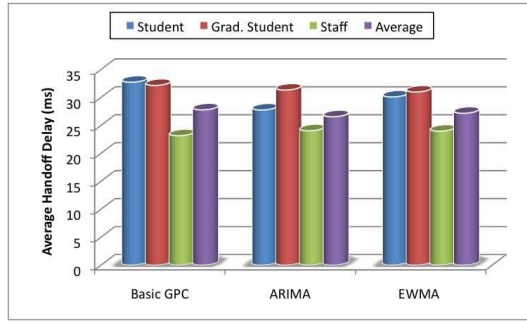
(a) KEC
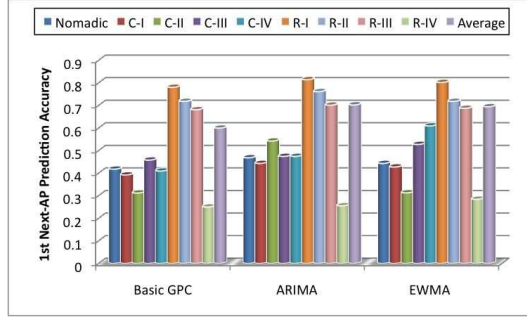
(b) Portland

Fig. 16 1st next-AP prediction accuracy based on time-series analysis.

provements due to short-term surges in handoffs caused by groups of users commuting during rush hour. 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 17 compares the handoff delays based on the parameter set defined in Table 3. Note that both sets of delay parameters yield the same results since GPC does not require channel scanning once a sufficient amount of handoff history has been collected. These results show that GPC with ARIMA provides 4.4% and 8.4% improvement, while EWMA provides 2.2% 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% and 9.1% improvement for ARIMA and EWMA, respectively. This was also the case for Portland, where group C-IV, which refers to commuters who work in region IV, resulted in 27.1% and 16.9% improvement for ARIMA and EWMA, respectively.

(a) KEC
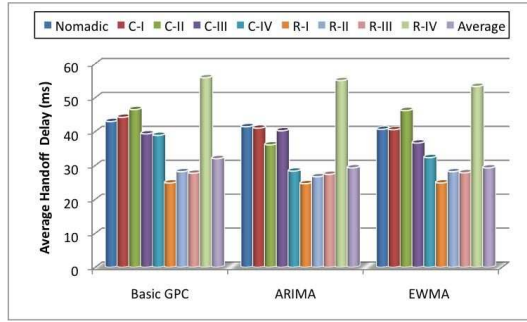
(b) Portland

Fig. 17 Handoff delay based on time-series analysis.

7 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 other existing methods because it uses global history of handoffs to determine directions of moving MSs. Therefore, it captures the mobility patterns of MSs similar to 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 1st 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 [18]. Therefore, understand-

ing 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.

Acknowledgment

References

1. A. Aljadhai and T. F. Znat. Predictive mobility support for QoS provisioning in mobile wireless environments. *IEEE Journal on Selected Areas in Communications*, 19(10):1915–1930, Oct. 2001.
2. 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)*, pages 83–95, June 2006.
3. G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas. Dynamic cross-layer association in 802.11-based mesh networks. In *IEEE INFOCOM*, pages 2090–2098, May 2007.
4. Atheros. Atheros AR5002X 802.11a/b/g Universal WLAN Solution. URL http://www.atheros.com/pt/AR5002XBulletin.htm.
5. G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Prentice Hall, 3rd edition, 1994.
6. 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)*, pages 27–27, Oct. 2005.
7. J. Broch, D. A. Maltz, D. B. Johnson, Y. chun Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 85–97, Oct. 1998.
8. J.-M. François. *Performing and Making Use of Mobility Prediction*. PhD thesis, University of Liege, 2007.
9. G.114. ITU-T recommendation G.114. Technical report, International Telecommunication Union, 1993.
10. S. Han, M. Kim, B. Lee, and S. Kang. Directional handoff using geomagnetic sensor in indoor WLANs. In *IEEE international conference on Pervasive Computing and Communications Conference (PERCOM)*, pages 128–134, Mar. 2012.

11. IAPP. IEEE 802.11f Standard: Recommended Practices for Multi-Vendor Access Point Interpretability via an Inter-Access Point Protocol. URL `http://grouper.ieee.org/groups/802/11/private/Draft_Standards/11f/802.11f-D3.1.pdf`.

12. IEEE802.11. Local and Metropolitan Area Network, Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications, 2007.

13. IEEEMesh. IEEEMesh. Draft Standard for Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11:Wireless LAN Medium Access Control and Physical Layer Specifications: Amendment: ESS Mesh Networking, Mar. 2007.

14. D. Katsaros, A. Nanopoulos, M. Karakaya, G. Yavas, O. Ulusoy, and Y. Manolopoulos. *Clustering Mobile Trajectories for Resource Allocation in Mobile Environments*, volume 2779/2003 of *Lecture Notes in Computer Science*. Springer, Sept. 2003.

15. M. Kim, Z. Liu, S. Parthasarathy, D. Pendarakis, and H. Yang. Association control in mobile wireless networks. In *IEEE INFOCOM*, pages 1256–1264, Apr. 2008.

16. 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)*, pages 259–263, June 2007.

17. Y. Liao and L. Gao. Practical schemes for smooth MAC layer handoff in 802.11 wireless networks. In *Proceedings of the International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 181–190, June 2006.

18. MadWiFi. MadWIFI 0.9.2. URL `http://www.madwifi.org`.

19. MetroFi. MetroFi Portland Free Wi-Fi. URL `http://www.metrofiportland.com`.

20. A. Mishra, M. Shin, and W. A. Arbaugh. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *ACM SIGCOMM Computer Communications Review*, 33(2):93–102, Apr. 2003.

21. A. Mishra, M. Shin, and W. A. Arbaugh. Context caching using neighbor graphs for fast handoffs in a wireless network. In *IEEE INFOCOM*, pages 351–361, Mar. 2004.

22. NS2. Network Simulator (NS2). URL `http://www.isi.edu/nsnam/ns`.

23. NYCWireless. NYCWireless. URL `http://NYCwireless.net`.

24. S. Pack and Y. Choi. Fast handoff scheme based on mobility prediction in public wireless LAN systems. In *IEE Proceedings Communications*, volume 151, pages 489–495, Oct. 2004.

25. Pathfinding. Amit's Thoughts on Pathfinding and A*. URL `http://theory.stanford.edu/~amitp/GameProgramming`.

26. I. Purushothaman and S. Roy. Fastscan: a handoff scheme for voice over ieee 802.11 WLANs. *Wirel. Netw.*, 16(7), Oct. 2010.

27. K. Ramachandran, S. Rangarajan, and J. C. Lin. Make-before-break MAC layer handoff in 802.11 wireless networks. In *IEEE International Confer-

ence on Communications (ICC), pages 4818–4823, June 2006.

28. I. Ramani and S. Savage. SyncScan: Practical fast handoff for 802.11 infrastructure networks. In *IEEE INFOCOM*, pages 675–684, Mar. 2005.

29. T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2nd edition, 2002.

30. RoofNet. Rooftop@Media. URL `http://rooftops.media.mit.edu`.

31. SeattleWireless. SeattleWireless. URL `http://SeattleWireless.net`.

32. 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)*, pages 70–83, June 2004.

33. 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)*, pages 19–26, Sept. 2004.

34. R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples.* Springer, 2nd edition, May 2006.

35. W.-S. Soh and H. S. Kim. Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions. In *IEEE INFOCOM*, volume 4, pages 2766–2777, Mar. 2004.

36. 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*, pages 1–13, Apr. 2006.

37. L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive Wi-Fi mobility data. *IEEE Transactions on Mobile Computing*, 5(12):1633–1649, Dec. 2006.

38. W. Su, S.-J. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1): 3–30, Jan. 2001.

39. T. Umedu, H. Urabe, J. Tsukamoto, K. Sato, and T. H. T. Higashino. A MANET protocol for information gathering from disaster victims. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 442–447, Mar. 2006.

40. H. Velayos and G. Karlsson. Techniques to reduce IEEE 802.11b MAC layer handover time. In *IEEE International Conference on Communications (ICC)*, pages 3844–3848, June 2004.

41. 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)*, pages 59–70, Oct. 2004.

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

43. 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*, 129(6):664–672, Dec. 2003.

44. 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*, volume 1, pages 21–23, Apr. 2001.
45. G. Yavas, D. Katsaros, O. Ulusoy, and Y. Manolopoulos. A data mining approach for location prediction in mobile environments. *Data and Knowledge Engineering*, 54(2):121–146, Aug. 2005.
46. J. Yoon, B. D. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *The International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, pages 177–190, June 2006.
47. 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)*, volume 2, pages 565–574, Sept. 2006.
48. G. Yu and C. Zhang. Switching ARIMA model based forecasting for traffic flow. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 429–432, May 2004.