Wireless Networks have been widely adopted into a major part of today’s network infrastructure. They have become a popular technology to not only expand the coverage of wired networks but also to interconnect a large wireless network, i.e., wireless mesh networks. As they allow more flexible communication than traditional wired-networks some challenges are raised, such as maintaining a seamless connectivity when Mobile Stations (MSs) move across the cells and dynamically adjusting resources for the transit MSs. Many solutions have proposed using mobility prediction to allow network devices and applications to prepare and adjust before the actual movement. However, none of the existing work considers mobility related to human factors. Therefore, this thesis proposes a technique called Behavior-based Mobility Prediction (BMP) that captures the dynamic behavior of MSs and the network by considering location, group, time-of-day, and duration factors. The proposed BMP is targeted to provide
accurate next Access Point (next-AP) predictions for WLANs to minimize the handoff latency. Moreover, the prediction can also apply to resource allocation in any type of Wireless Networks. Our simulation study shows that BMP virtually eliminates the need to scan for APs during handoffs and results in much better overall handoff delay compared to existing methods.
Behavior-based Mobility Prediction for Fast Handoffs in Wireless LANs

by

Weetit Wanalertlak

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented June 2, 2008
Commencement June 2009

APPROVED:

_________________________________________
Major Professor, representing Electrical and Computer Engineer

_________________________________________
Director of the School of Electric Engineering and Computer Science

_________________________________________
Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

_________________________________________
Weetit Wanalertlak, Author
I would like to express my sincere appreciation to my major advisor Prof. Ben Lee. I feel greatly honored to have worked under his supervision. He has been a great source of motivation and support through the entire process. This thesis would have not been possible without his tirelessly guidance and expertise. I would like to thank him for giving me the opportunity to continue pursue my graduate degree at Oregon State University.

I would also like to thank Prof. Wen-Tsong Shiue, my previous advisor for guidance and for giving me the opportunity to pursue my graduate degree at Oregon State University.

I would like to thank Prof. Bella Bose, Prof. Len. Forbes, Mr. Roger Traylor and Prof. John Nairn for serving on my graduate committee and for their comments.

Lastly, I would like to thank my family and friends for their support in everything. My parents, my wife, my sister and my brother, thanks for your caring and encouragement. David Zier, thank you for your support, encouragement and proof reading.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
</tr>
<tr>
<td>2</td>
<td>Global Path Cache</td>
</tr>
<tr>
<td>2.1</td>
<td>Related Work</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Mobility Prediction</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Handoff Delay</td>
</tr>
<tr>
<td>2.2</td>
<td>The Proposed GPC Technique</td>
</tr>
<tr>
<td>2.2.1</td>
<td>The Basic GPC Scheme</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Time-Series Based Prediction Model for GPC</td>
</tr>
<tr>
<td>2.2.3</td>
<td>ARIMA Based Prediction Model for GPC</td>
</tr>
<tr>
<td>2.3</td>
<td>Performance Evaluation</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Simulation Environment</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Simulation Delay Parameters</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Simulation Results</td>
</tr>
<tr>
<td>2.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>3</td>
<td>Behavior-based Mobility Prediction</td>
</tr>
<tr>
<td>3.1</td>
<td>Related Work</td>
</tr>
<tr>
<td>3.2</td>
<td>The Proposed Behavior-based Mobility Prediction Technique</td>
</tr>
<tr>
<td>3.2.1</td>
<td>User Behavior</td>
</tr>
<tr>
<td>3.2.2</td>
<td>The Proposed Method</td>
</tr>
<tr>
<td>3.3</td>
<td>Performance Evaluation</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Simulation Environment</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Simulation Delay Parameters</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Simulation Results</td>
</tr>
<tr>
<td>3.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>4</td>
<td>Future Work</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
</tr>
<tr>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>The Markov-based Mobility Prediction</td>
<td>88</td>
</tr>
<tr>
<td>Derivation of the ARIMA Based Prediction Model for GPC</td>
<td>90</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Active Scanning in IEEE 802.11.</td>
</tr>
<tr>
<td>2.1</td>
<td>Example WLAN coverage areas.</td>
</tr>
<tr>
<td>2.2</td>
<td>Local history using HSW for $k=3$.</td>
</tr>
<tr>
<td>2.3</td>
<td>The steps in the GPC Technique.</td>
</tr>
<tr>
<td>2.4</td>
<td>Frequency of handoff sequence $AP_4 \rightarrow AP_5 \rightarrow AP_6$ for KEC.</td>
</tr>
<tr>
<td>2.5</td>
<td>Predicted Frequency of Handoff Sequences based on ARIMA(0, 2, 2) for KEC.</td>
</tr>
<tr>
<td>2.6</td>
<td>Predicted Frequency of Handoff Sequences based on EWMA for KEC.</td>
</tr>
<tr>
<td>2.7</td>
<td>Overall accuracy as function of history.</td>
</tr>
<tr>
<td>2.8</td>
<td>Number of cache entries.</td>
</tr>
<tr>
<td>2.9</td>
<td>Average Number of Next-AP Predictions.</td>
</tr>
<tr>
<td>2.10</td>
<td>Next-AP Prediction Accuracy.</td>
</tr>
<tr>
<td>2.11</td>
<td>Average Number of Channels Probed.</td>
</tr>
<tr>
<td>2.12</td>
<td>Average Handoff Delay.</td>
</tr>
<tr>
<td>2.13</td>
<td>$1^{st}$ Next-AP prediction Accuracy based on Time-Series Analysis.</td>
</tr>
<tr>
<td>2.14</td>
<td>Handoff Delay based on Time-Series Analysis.</td>
</tr>
<tr>
<td>3.1</td>
<td>BMP Prediction request and response.</td>
</tr>
<tr>
<td>3.2</td>
<td>The Behavior-based Mobility Prediction Scheme.</td>
</tr>
<tr>
<td>3.3</td>
<td>The IMA structure.</td>
</tr>
<tr>
<td>3.4</td>
<td>$1^{st}$ prediction Accuracy.</td>
</tr>
<tr>
<td>3.5</td>
<td>Handoff Delay.</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Global History in the Path-Cache for Figure 2.2.</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Delay parameters used in the simulation.</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>Handoff Sequence Table and IMA.</td>
<td>66</td>
</tr>
<tr>
<td>3.2</td>
<td>Delay parameters used in the simulation.</td>
<td>76</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>ACF and PACF from the transformed time-series data in Figure 5.</td>
<td>93</td>
</tr>
</tbody>
</table>
## LIST OF APPENDIX TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Behavior of ACF and PACF for the ARIMA model.</td>
</tr>
</tbody>
</table>
Chapter 1 – Introduction

Wireless communication technology, along with advancements in systems and networking software, allow users to be connected and productive while on the road. Wireless LANs (WLANs)\(^1\), based on the IEEE 802.11 standard [1], are already prevalent in both residential and public areas, such as airports, university campuses, shopping malls, and coffee shops. Moreover, numerous efforts have already been underway to connect Wi-Fi hot spots in order to offer better connectivity over larger geographical areas, such as community networks that cover large metropolitan areas of several major US cities [2–5].

One of the greatest benefits of community networks is mobility support, which allows a user to continually talk on a Voice over IP (VoIP) application or watch a streaming video while walking or riding a bus between city blocks. However, mobility incurs a large handoff delay when a Mobile Station (MS) switches the connection from one wireless 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 take almost a second [6], while actively scanning for APs requires only 350~500 ms [6]. This becomes a major concern for mobile multimedia applications such as VoIP where

\(^1\)Also known as Wi-Fi hot spots
the end-to-end delay is should not be greater than 50 ms [7].

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 [8–11]. These methods employ extra hardware, either in the form of additional radios [8], or an overlay sensor network [11] that detect APs. The additional hardware is then used to selectively scan channels to probe based on the topological placement of APs [9], and predict the next point-of-attachment based on signal strength [10]. Unfortunately, these techniques do not predict the next AP. AP predictions can eliminate the need to scan for APs, consider mobility patterns for MSs, which are dictated by the structure of a building or a city block, and/or the past behaviors of MSs. There are also methods that consider mobility history of MSs that provide next-AP predictions [12]. However, these methods tend to be very generalized and do not consider the special characteristics of WLANs, such as highly overlapped cell coverage, MAC contention, and variations in link quality.

In this thesis, we present a solution, called the *Global Path-Cache* (GPC) and *Behavior-based Mobility Prediction* (BMP) techniques that eliminate the need to scan for available APs, resulting in faster handoffs. First, the key idea of GPC is to predict the next point-of-attachments based on the history of the mobility patterns for MSs. This is achieved by maintaining the handoff history for all the MSs in the network, and then monitoring a MS’s direction of movement relative to the topological placement of the APs in order to predict its next point-of-attachment. In addition, next AP predictions are based on frequencies of occurrences rather than signal strength. Therefore, it takes into consideration that mobility patterns
dictated by the structure of a building or a city block in addition to the past behaviors of MSs. Second, BMP further improves prediction for the next point-of-attachments by capturing the dynamic behavior of MSs and the network. BMP takes mobility patterns due factors of location, group, time-of-day, and duration, into consideration when determining the next point-of-attachment. The proposed technique uses an adaptive algorithm. When BMP fails to properly predict the next AP, the prediction models are recalibrated based on the different groups, where each group of MSs has similar mobility patterns, time-of-day characteristics as well as duration, which is reflected the activity of MS, while it associates to the AP.
1.1 Background

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 channel. As the MS moves further away from the current AP, the signal quality continues to degrade. When the signal quality reaches a preset threshold, it causes the MS to initiate a handoff to a new cell. This process starts with probing for 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 or until the waiting time reaches a predefined maximum duration, which is longer than the beacon interval. However, even though the beacon interval of each AP is factory preset, typically 100 ms, 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 a second to search all possible channels, which results in unacceptable handoff delay. Consequently, a large number of packets will be lost during handoff delay may trigger network application to drop a connection. Techniques such as store-and-forward can reduce the number of lost packets, but those packets will already incur a long delay. In a bi-directional audio application, such as VoIP, a long round trip delay will create echo during a conversation. Echo becomes a serious problem when the roundtrip delay exceeds 50 ms.

In active scanning illustrated in Figure 1.1, MS broadcasts a probe request and waits for a response. While MS waits for a response, it is also listening for possible network traffic transmitted on the channel. If MS receives a response
from AP or there are other AP in the channel or, MS waits for maximum channel
time (MaxChannelTime). The longer MaxChannelTime gives MS has a better
chance to discover more APs. Otherwise, MS only waits for minimum channel
time (MinChannelTime). MinChannelTime is a shorter than MaxChannelTime
to keep the overall handoff delay low, but it should be long enough for MS to
receive possible a response. This process is repeated for every channel. In order to
accelerate the scanning process, some channels may be skipped based on heuristics

Figure 1.1: Active Scanning in IEEE 802.11.
that require the knowledge of the topographical placement of the APs. A typical duration for scanning each channel is around 15 ms and 30-400 ms for all 11 channels.

After scanning, the last two steps of the handoff process involve authentication and reassociation. 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 following two steps:

- Step 1 - the MS sends authentication request management frame to AP. The frame contains source address (SA) in the frame header and information in the frame body to indicate the type of authentication. The source address helps AP to identify a new MS.

- Step 2 - AP sends the authentication management frame back to MS. This frame has similar information to the first frame with the authentication result and the information to indicate the type of authentication.

The last step of the handoff process is reassociation. MS performs this step before it can continue transmission with the new AP. Reassociation steps consists of the following:

- Step 1 - the MS sends reassociation request management frame to AP.

- Step 2 - AP sends the reassociation response management frame back to MS.
After reassociation, the last step in the handoff process is for the new AP to reset the Ethernet address table in the switch connecting both the old and new APs so that the network traffic can be re-routed.
IEEE 802.11 WLANs have become pervasive in our society. WLANs offer high
data transfer rates that allows portable devices such as laptops, PDAs, and smart-
phones to not only connect to the Internet, but also transfer real-time multimedia
data, such as streaming audio and video. Moreover, smart-phones, in the near
future, will be able to automatically turn off its connection to a GSM or CDMA
network and register to a lower cost Voice over IP (VoIP) service on a WLAN.
In fact, more and more WLAN APs are installed in public and commercial areas.
WLANs will soon cover majority downtown areas in many cities. One of the
greatest benefits of WLAN is mobility, which allows a user to continually talk on
a VoIP application or watch a video stream while walking between city blocks or
riding a bus. However, mobility incurs large handoff delays when the MS switches
connection from one AP to another. In a crowded network, such as office or
university campus environments, APs are installed close together. This causes
frequent handoffs that make the problem more severe. Long handoff delay is
undesirable and yet, a recent study found that the handoff delay in WLANs can
take as long as one second [6]. This becomes a major concern for mobile multimedia
applications, such as VoIP, where the end-to-end delay is recommended to be less
than 50 ms [7].

In order to illustrate these characteristics of WLANs, Figure 2 shows the cover-
age areas of the four-story, 153,000-ft$^2$ Kelley Engineering Center (KEC) at Oregon State University and the MetroFi, which is a public WLAN service that covers 2.5-mile$^2$ area of Portland, Oregon [4]. The APs in KEC are connected by Ethernet switches, while the APs in the MetroFi network are connected through a wireless mesh network [13]. Besides the obvious differences in coverage areas, these two networks share many similarities and challenges based on the following observations. First, the APs are installed in relative close proximity to users, such as offices and classrooms in KEC and the 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 (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 the cells. Fourth, the signals transmitted from the 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 change 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 [14, 15], we presented a solution, called Global Path-Cache (GPC), which eliminates the need to scan 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 the mobility patterns for the 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 the 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. GPC is an adaptive algorithm, which is independent of the topological placement of APs and the number of channel frequencies used.

Therefore, in addition to providing a discussion of the basic GPC scheme, [14,15] contributed the following:

- First, the basic GPC scheme presented in [14] 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. [14,15] 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, the performance evaluation is significantly expanded to include a much larger network (i.e., MetroFi Portland), and specifically analyzes 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) [10] and Neighbor Graph (NG) [9]. The average handoff latency in GPC is 27\textasciitilde{}28 ms and 32\textasciitilde{}39 ms for KEC and Portland, respectively. In contrast, SSwC requires as much as 147\textasciitilde{}149 ms with SSwC and 328\textasciitilde{}422 ms with NG. GPC provides a much higher accuracy than SSwC in its first Next-AP prediction. GPC achieves 100\% accuracy and thus requires no probing with at most six predictions while SSwC achieves a prediction accuracy of, at most, 24\%. The time series-based GPC scheme further improves the overall prediction accuracy and reduces the handoff latency as much as 8.5\% compared to the basic GPC scheme.
(a) Kelley Engineering Center building.

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

Figure 2.1: Example WLAN coverage areas.
2.1 Related Work

2.1.1 Mobility Prediction

Mobility prediction is crucial for mitigating the effects of handoffs and improving QoS. There has been a plethora of work on mobility prediction for variety of wireless networks, such as cellular [10, 11, 16], WLANs [8–10, 12, 17], ad hoc networks [18], and mesh networks [13], and has been applied to reduce handoff latency [6,9,10,19], provide efficient resource reservation [12,16,20–24], improve routing protocols [18], and conserve power [25].

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 the locations of MSs. These techniques reduce the signaling overhead during handoff and provide resource reservation to MSs in cellular networks [16,23,24]. Second, topology-based techniques use the knowledge of geographical locations of APs and directional movement of MSs to provide resource reservation in cellular networks [22]. Third, stochastic techniques provide mobility predictions using probabilistic models. These techniques apply the knowledge from the geographic coordinates of MSs, either from GPS or signal strength triangulation, to predicted future locations [20,21].

Although these techniques provide mobility prediction in cellular networks, they are not efficient solutions for WLANs. For example, data-mining techniques require large storage capacities 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 [26].

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 [17]. Typically, a Markov mobility predictor performs the following two operations; the first operation is to maintain a collection of past locations of MSs, while the second operation is to predict future locations of MSs based on the value of conditional probability that matches 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.

2.1.2 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 [27, 28]. MultiScan uses multiple WLAN network interfaces to opportunistically scan and pre-associate with alternative APs in order to avoid disconnections [8]. 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 [11]. This way, a MS broadcasts an AP-list request to the surrounding sensor nodes to obtain 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 [6, 29]. SyncSacn is a passive method that requires APs to send staggered periodic beacons that 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 [29]. Both methods rely on the power saving modes 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 to or probe for other APs. Nonetheless, the GPC method proposed in this paper is an orthogonal approach to 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 [6,9], Pre-Authentication path [19] and Selective Scan with
Caching [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 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-associate 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 need to be scanned.

Selective Scan with Caching minimizes the need to probe during handoffs 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 the 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 re-associate with these two APs in order. If
it fails to re-associate with both APs or an entry is not found in the AP-cache, a selective scan is performed based on the channel mask that chooses the 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 additional 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 [20], 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 on the same channel and thus limiting the range of coverage area.
2.2 The Proposed GPC Technique

In order to reduce the handoff delay, GPC tracks previously 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 2.2.1 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 2.2.2 discusses the application of time-series analysis on handoff occurrences to formulate a better model based on user behavior in order to improve the Next-AP prediction accuracy.

2.2.1 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 2.2 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 attach to next, as there are three possible candidates, $AP_x$, $AP_y$ and $AP_z$. 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) that contains the information of the current AP as well as $k-1$ past APs (i.e., the MAC address and the channel number). Figure 2.2 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
Figure 2.2: Local history using HSW for $k=3$.

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 the HSW as part of its 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 the *Current-AP*, $k-2$ *Past-APs*, the *Next-AP*, and a *Counter* indicating the number of hits on this entry. Table 2.1 shows the partial content of the Path-Cache for Figure 2.2.

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.
Table 2.1: Global History in the Path-Cache for Figure 2.2.

<table>
<thead>
<tr>
<th>Cache-Key</th>
<th>Next-AP</th>
<th>Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past-AP</td>
<td>Current-AP</td>
<td></td>
</tr>
<tr>
<td>$AP_x$</td>
<td>$AP_y$</td>
<td>$AP_x$</td>
</tr>
<tr>
<td>$AP_x$</td>
<td>$AP_y$</td>
<td>$AP_w$</td>
</tr>
<tr>
<td>$AP_x$</td>
<td>$AP_y$</td>
<td>$AP_z$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$AP_y$</td>
<td>$AP_x$</td>
<td>$AP_y$</td>
</tr>
</tbody>
</table>

Path-Cache update - The server uses the past cache-key represented by the handoff sequence $AP_0$, $AP_1$, $\ldots$, $AP_{k-2}$ 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 and 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 $AP_1$, $AP_2$, $\ldots$, $AP_{k-1}$ in HSW to search the Path-Cache for a matching Cache-Key. If a match or multiple matches are found, the server sends a Path-Cache response to the MS 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 \textit{null}_1, \textit{null}_2, \cdots, \textit{AP}_tuned−in, where \textit{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 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 2.3 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:

\textbf{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.

\textbf{Step 2:} MS sends authentication request, \textit{Auth\_Req}, containing Path-Cache request, \textit{PC\_Req}(HSW), to the server and obtains Next-AP predictions for the next handoff (1).

\textbf{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, \textit{Auth\_Resp}, containing Path-Cache response, \textit{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 the reassociation request (5) to the AP and receives a reassociation response (Step 6). If no reassociation response is received, the MS moves to the next element in the Next-AP prediction list and goes to Step 1.

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

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 1\textsuperscript{st} Next-AP prediction fails but the 2\textsuperscript{nd} 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, requiring 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. 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.
2.2.2 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 the 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 the handoff sequence, $z_{t+1}$, for $AP_4 \rightarrow AP_5 \rightarrow AP_6$ can be represented by ARIMA(0, 2, 2) as shown below (see Appendix).

2.2.3 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 [20, 21]. The frequency of handoff sequence frequency of handoff sequences 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. For a typical WLAN environment, such as ours, the recommended period will be at least one day (1,440 min.) to capture all possible trends within a day. Figure 2.4
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 2.3.1), 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 $z_{t+1}$ for $AP_4 \rightarrow AP_5 \rightarrow AP_6$ can be represented by ARIMA$(0, 2, 2)$ as shown below (see Appendix).

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

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

Figure 2.5 shows the plot of predicted frequency for the 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 keep 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 2.5(a) between 12 AM to 11 AM. For example, Figure 2.5(b) shows a magnified view of the frequency of handoff sequences between 7 to 9 AM of Figure 2.5(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 provides $AP_3$ as the $1^{st}$ Next-AP prediction. However, the basic GPC scheme based only on long-term history cannot capture this short term variations and can cause mispredictions.

2.2.3.1 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
\[ z_{t+1} = (1 - \lambda)z_t + \lambda z_t \]

where \( z_t \) is the sampled time series data, \( z_t \) is the predicted time series data, and \( \lambda \) is the smoothing factor \( 0 < \lambda < 1 \). The parameter \( \lambda \) determines characteristics of the EWMA model and is typically chosen experimentally. Based on our analysis, \( \lambda \) for the time-series data representing the frequency of handoff sequences in KEC is chosen to be 0.1. Figure 2.6(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 2.6(b) shows that EWMA, despite some noise, is also able to capture the fact that the frequency of the 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.
Figure 2.5: Predicted Frequency of Handoff Sequences based on ARIMA(0, 2, 2) for KEC.

(a) 24 Hours.

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

(b) 7 AM to 9 AM.

Figure 2.6: Predicted Frequency of Handoff Sequences based on EWMA for KEC.
2.3 Performance Evaluation

This section presents the performance evaluation of our proposed GPC technique. Section 2.3.1 describes the simulation environment as well as the two key components of the simulator; the \textit{path generator} and the \textit{handoff detector}. Section 2.3.2 discusses the delay parameters used in the study. Section 2.3.3 compares the results of the basic GPC scheme against the \textit{Selective Scan with Caching} (SSwC) \cite{10} and \textit{Neighbor Graph} (NG) \cite{9, 34} techniques, as well as presents the performance improvement using the ARIMA and EWMA models.

2.3.1 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 2.1(a) and 2.1(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 within KEC, students, graduate students, and staff, with each having different types of mobility behaviors. For example, \textit{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, \textit{graduate students} mainly move between their offices, the atrium and the computer lab. Finally, \textit{staff} moves mostly between their offices and the atrium.
The results for Portland were generated based on nine different groups of 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, C-I C-II, C-III, and C-IV in Figure 2.1(b), which are likely to travel long distances (about 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 the four regions, R-I, R-II, R-III, and R-IV in Figure 2.1(b). These groups of MSs can move anytime but are likely to only move within few 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 a *log-distance path loss model* [26]. This process is performed at a resolution of one meter. The handoff detector records the number of channel switches and the number of times MS
has to wait for $t_{max}$, $t_{min}$, $t_{auth}$, and $t_{assoc}$ \footnote{see Section 2.3.2}. 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 predefined 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, then the simulation is stopped. Otherwise, the simulation goes back to Step 1.

### 2.3.2 Simulation Delay Parameters

The delay parameters used in the simulation are shown in Table 2.2: *Channel Switching Time* ($t_{\text{switch}}$) is the time required to switch from one channel to another; *MinChannelTime* ($t_{\text{min}}$) is the minimum amount of time a MS has to wait on an empty channel; *MaxChannelTime* ($t_{\text{max}}$) is the maximum amount of time a MS has to wait to collect all the probe responses, which are 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 required to perform reassociation.

The Parameter Set 1 represents 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 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 were operating on Channel 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 Channel 2, which has no APs, and then immediately switch back to Channel 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, where \(t_{assoc}=4\) ms. \(t_{max}\) was estimated by observing the time between a probe request and an authentication request and is 199.4 ms. This is consistent with the \(t_{max}\) value provided in the
Table 2.2: Delay parameters used in the simulation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Set 1 (Actual)</th>
<th>Set 2 (Optimized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Switching Time ($t_{\text{switch}}$)</td>
<td>11.4 ms</td>
<td>11.4 ms</td>
</tr>
<tr>
<td>MinChannelTime ($t_{\text{min}}$)</td>
<td>20 ms</td>
<td>1 ms</td>
</tr>
<tr>
<td>MaxChannelTime ($t_{\text{max}}$)</td>
<td>200 ms</td>
<td>10 ms</td>
</tr>
<tr>
<td>Authentication delay ($t_{\text{auth}}$)</td>
<td>6 ms</td>
<td>6 ms</td>
</tr>
<tr>
<td>Reassociation delay ($t_{\text{reassoc}}$)</td>
<td>4 ms</td>
<td>4 ms</td>
</tr>
</tbody>
</table>

source code of the open source wireless network device driver [35]; therefore, $t_{\text{max}}$ is assumed to be 200 ms. On the other hand, there is no direct method to measure $t_{\text{min}}$. Thus, the reference value of $t_{\text{min}} = 20$ 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 $t_{\text{min}}$ and $t_{\text{max}}$ values from [28]. This study determined that the value of $t_{\text{min}}$ that leads to minimized handoff delay are given by $t_{\text{min}} \geq DIFS + (aCW_{\text{min}} \times aSlotTime)$ [28], where $DIFS$ is the Distributed Inter-Frame Space, $aCW_{\text{min}}$ is the number of slots in the minimum contention window, and $aSlotTime$ is the length of a slot. In the IEEE 802.11g standard [1], the values for $DIFS$, $aCW_{\text{min}}$, and $aSlotTime$ are 28 µs, 15 µs, and 9 µs, respectively, which results in $t_{\text{min}} \geq 163$ µs. However, $t_{\text{min}}$ is defined in terms of Time Units (TU), where 1 TU = 1024 µs. Therefore, the smallest possible value for $t_{\text{min}}$ is 1024 µs. Moreover, $t_{\text{max}}$ is estimated as the transmission delay required when 10 MSs try to access the same AP. In their simulation [28], 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 [1,37]. Therefore, the estimated $t_{max}$ is 10 ms.

2.3.3 Simulation Results

This subsection first compares the performance of the basic GPC against Selective Scan with Caching (SSwC) and Neighbor Graphs (NG). Then, the additional performance gains from applying the time-series models are analyzed. 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 rather than only 10 AP cache entries and two Next-AP predictions per entry (i.e., Best AP and 2$^{nd}$ Best AP) used in the original SSwC algorithm [10].

Figure 2.7 compares the overall accuracy of GPC and SSwC as function of history and is represented in 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 a longer history is necessary to record all possible handoff sequences in GPC. 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 GPC. Thus, all path-cache requests will be provided with the correct Next-AP predictions. 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, 45,000 users in Portland, for example, can produce $10^6$ handoffs within only $\sim 3.5$ hours. The overall accuracy of SSwC also increases as function of number of handoffs, but saturates at $\sim 54\%$ and 31\% for KEC and Portland, respectively.

In order to properly compare the performance, all subsequent results 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.

Figure 2.8 shows the maximum number cache entries needed for GPC and SSwC. Again, NG is not included in this comparison. 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 but requires more entries. Note that the number of entries cannot be compared directly because multiple GPC entries provide multiple Next-AP predictions, whereas each entry in AP-cache provides multiple Next-AP predictions. Therefore, a more accurate metric is the
average number of Next-AP predictions returned per handoff shown in Figure 2.9. As can be seen, GPC provides a higher average number Next-AP predictions per handoff than SSwC.

Figure 2.10 compares the accuracies of Next-AP predictions. 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 mis-prediction adds to the overall handoff delay. For GPC, the accuracy for the KEC building for the 1st Next-AP prediction starts at 68% and increases slightly as function of $k$. 1st Next-AP predictions that fail are satisfied by 2nd Next-AP predictions with accuracy of 89%. 3rd and 4th predictions only become effective with a longer handoff history and provide accuracies of 97%–100% and 100%, respectively. In contrast, SSwC provides significantly lower 1st and 2nd prediction accuracies of 51% and 2.6%, respectively. Since the complexity of moving paths is higher for Portland, GPC always provides up to 6 predictions for any number of $k$. However, the 1st Next-AP prediction starts at 43% and increases slightly as function of $k$, which is similar to the case for the KEC building. In comparison, SSwC provides lower 1st, 2nd, and 3rd prediction accuracies of 25%, 6% and 0.02%, respectively.

Note that 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 traveled by the MSs are cover by two cells, and only 5% have three cells. Thus, SSwC will have at most two Next-AP predictions. In contrast, the maximum number
of Next-AP predictions depends on the number of adjacent cells, which is four. Similarly, 36.1% of the overlapped regions in Portland traveled by the MSs are covered by two cells, 24.9%, 3.34%, and 0.04% have three cells, four cells, and five cells, respectively. Since the area covered by five cells is relatively small, SSwC will have at most three Next-AP predictions. In contrast, the maximum number of Next-AP predictions is six.

The GPC’s superior prediction accuracy comes from not only the history hand-off sequences but also the set of returned predictions and are prioritized based on how often these paths are encountered. In contrast, SScW relies only on the signal strength, which is often different from actual paths taken by the 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, leading to higher mispredictions that add to the overall handoff delay.

In order to illustrate this problem, consider the three cells adjacent to AP6 in Figure 2. The AP-cache entry for AP6 will have one of the following Next-AP predictions depending on where and when a full or selective scan was performed: AP3, AP4, AP5, AP3, AP3, or AP3, AP4. Suppose a MS travels along the path AP5 → AP6 → AP4, and AP2 is the current Next-AP prediction for AP5 and AP4 is the current Next-AP prediction for AP6. As the MS moves from AP5 to AP6 through the overlapped cells of AP3, AP5, and AP6, a misprediction occurs and selective scan will be performed based on the channel mask. If the selective scan is successful (i.e., the corresponding bits are set for AP3 and AP6) and AP6 is
determined to have the strongest signal, a new entry for $AP_6$ will be generated with $AP_3$ as the Next-AP prediction and the existing entry will be over-written. Later, when the MS moves from $AP_6$ to $AP_4$, the prediction will be wrong and again and a selective scan will be performed. In comparison, GPC records all the possible paths emanating from $AP_6$ and are prioritized based on their frequencies of occurrence. Thus, it is able to predict that a MS moving from $AP_5$ to $AP_6$ will likely to move to $AP_4$.

These mispredictions are reflected in the average number of channels probed per handoff shown in Figure 2.11, which also includes the result for NG. 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 a very low accuracy (see Figure ) that causes 47.7% and 70% of the handoffs in KEC and Portland, respectively, to mispredict and have to rely on selective scanning. This process 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 the probed channels 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 of channels probed per handoff is zero because once the GPC has a complete history it is guaranteed to provide accurate Next-AP predictions.

Figure 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 the GPC resulting 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.

Although the baseline GPC based on long-term history can significantly reduce the handoff delay, Figure 2.10 shows that \( \sim 30\% \) of handoffs in KEC and and \( \sim 40\% \) in Portland 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 compares the 1st Next-AP prediction accuracy with \( k=3 \) using ARIMA(0,2,2) and EWMA against the baseline 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 uses a time-series model to the predict frequency of handoff sequences, which can properly capture the handoff caused by short term and periodic behaviors of mobile users. The improvements vary in different users groups. For example, time-series based GPC shows 42% improvement for students in KEC since their behaviors are dictated by class schedules, which causes their handoffs to be periodic and very predictable by time-series models. Other groups of users have behaviors based on different schedules, they do have short-term mobility patterns caused by groups of users commuting in similar directions. Next, the EWMA resulted in an average improvement of 6% for KEC and 15.8% Portland, but provided less improvement than the more complex ARIMA since EWMA does not rely on the full statistical analysis to generate the
time-series models.

Finally, Figure 2.14 compares the handoff delays based on the parameter set defined in Table II. Note that both sets of delay parameters from Table II yield the same delay results because 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 only 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 example, Student and Graduate Student groups in KEC resulted in 15.2% and 2.6% improvement, respectively, for ARIMA and 8.07% and 3.68%, for EWMA. This was also the case for Portland, where group R-IV, which refers to users who live in region I, resulted in 56% improvement over the basic GPC scheme.
Figure 2.7: Overall accuracy as function of history.
Figure 2.8: Number of cache entries.

Figure 2.9: Average Number of Next-AP Predictions.
Figure 2.10: Next-AP Prediction Accuracy.
Figure 2.11: Average Number of Channels Probed.
Figure 2.12: Average Handoff Delay.

(a) KEC

(b) Portland
Figure 2.13: 1st Next-AP prediction Accuracy based on Time-Series Analysis.
Figure 2.14: Handoff Delay based on Time-Series Analysis.
2.4 Conclusion

This chapter described a GPC technique that minimizes the time required to scan for APs in WLANs. GPC is different from the other existing methods because it uses the 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 provide 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 other existing techniques. Additionally, the time-series based models further reduce the overall handoff delay by increasing the accuracy of 1st Next-AP predictions.
Although, the basic GPC scheme discussed in Chapter 2 is effective in predicting the next point-of-attachment for majority of MSs, some MSs behaving differently from the norm and suffer mispredictions. For example, a majority of MSs in the 1st floor of KEC are undergraduate students who often move between classrooms, computer lab, and café. Other groups of MSs are represented by graduate students, faculty, and staff who stay in their offices and occasionally move to café for a snack. Since most of the handoffs are caused by undergraduate students, the basic GPC scheme will provide predictions based on this dominant group and thus fail to properly capture the behavior of other groups leading to mispredictions. Similar situation occurs with user behavior relative to time. For example, there will be bursts of repetitive mobility patterns when students move between classes. However, during lunchtime, students congregate at the café to have lunch. These mobility patterns are different from long-term behavior of MSs causing the basic GPC technique to fail. Some of mispredictions can be reduced by assigning separate time series models or time-of-day to groups of MSs to isolate distinct mobility behaviors. Unfortunately, group and time-of-day characteristics may not be clearly defined as in the above example because MSs can possibly change their groups and time-of-day based on various other factors.

This chapter further improves the basic GPC scheme using a solution called
Behavior-based Mobility Prediction (BMP). The key idea of BMP is to model the regularity of mobility patterns based on the following four behavior factors: location, group, time-of-day, and duration. The location factor is already considered in the basic GPC scheme based on handoff history of all the MSs in the network.
3.1 Related Work

Although many different mobility prediction techniques have been proposed, these techniques share many similarities. This section categorizes and describes the various existing techniques.

*Location-based* schemes provide predictions based on the current and past locations (*i.e.*, cells). These are all applications of an order-k Markov predictor (see Appendix) [38], but are implemented differently. Although most techniques are based on mobility history of users, which include *Global Path Cache* [15], *Selective Scan with Caching* [10], *Movement Model* [39], and *Two-Tier prediction* [40, 41], some use directional vectors [20] to predict the next point-of-attachment. The disadvantage of location-based schemes is a lack of consideration of mobile user behavior, such as group, time-of-day, duration characteristics. Therefore, these techniques may not be able to properly capture mobility patterns that deviated from the norm, such as behaviors exhibited by a small group of users or repeated in certain periods of time. There is a technique that applies different predictors to different segments [42]. There are also techniques that reduce the number of location update operations in cellular networks by associating locations of individual users to different periods of time [43–45]. However, time is not the only factor that affects mobility patterns. Therefore, these techniques are unable to properly capture the behaviors exhibited by different groups.

*Topology-based* schemes define directed graphs that represent the topological placements of access points (APs) and the mobility patterns of MSs. These tech-
niques are typically applied to WLANs and include Neighbor Graph [6, 9] and Pre-Authentication path [19]. Although these techniques reduce the number of channels to scan, they do not provide next-cell predictions.

Activity-based schemes provide next-cell predictions by relating locations to the user’s interests, such as schedules and activities. These include Activity-based Mobility Prediction [46], ComMotion [47] and the techniques described in [48, 49], which apply geographical coordinates gathered from wearable devices such as GPS with user’s actual schedule to provide mobility prediction. The locations are associated with activities users perform, e.g., a restaurant is associated to the activity of eating and a theatre is associated with the activity of watching a movie. The prediction algorithm matches an activity from a user’s schedule or user’s special request with a location. If the multiple matches are found, these techniques typically prioritize the prediction outcomes based on the frequencies of occurrences. Then, the mobility prediction is made based on an assigned path between the current and the predicted locations. However, these techniques are not applicable for wireless networks because after a prediction is made, a path-finding algorithm needs to find the shortest path that the user will use from the current cell to the predicted cell. However, the shortest path may not be the preferred choice, since users may choose a path based on their point-of-interests, such as ATM to withdrawal money or simple idiosyncrasy of passing by a park. Data Mining based schemes consider a contextual information of a user, such as its geographical coordinates, current time, and schedule. The techniques presented in [16, 23, 24] reduce the signaling overhead during handoff and provide resource reservation to MSs in cellular net-
works by logging users’ visited cells and time in a database. In addition, some techniques also record geographic coordinates and directional movements of MSs from either GPS or triangulation of signal strengths [21, 22]. However, the basic idea to provide predictions by searching the database using user contexts stored in MSs. If a match or multiple matches are found, the matches are prioritized based on the number of occurrences acquired from the database. The data-mining based technique is an exhaustive approach for a mobility based prediction. Therefore, the disadvantages of data-mining based mobility prediction are first these techniques require large storage and fast processors to properly analyze long-term mobility behavior of users. Second, the best of our knowledge, none of the data-mining technique is able to capture mobility pattern caused by group characteristic. Third, most 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 mobile devices are mainly used for indoors and crowded outdoor areas where the signal strength is highly affected by noise rather than distance [26]. The previous techniques have implemented mobility model based on user behaviors regularity, such as location and time (i.e., schedule). These techniques still cannot provide accurate predictions for the the groups or the period of times which MSs behave deviate from the norm. Although, the data-mining based mobility prediction, which is an exhaustive approach may provide higher prediction accuracy. The accuracy is proportion of the mobility data required to record. In contrast, the proposed BMP is an adaptive scheme. Thus, the proposed technique recalibrates its models by monitoring the miss pre-
dictions. In addition, it can dynamically form new prediction models to capture
the mobility patterns based on the behavior factors of location, group, time-of-day,
and duration.
3.2 The Proposed Behavior-based Mobility Prediction Technique

The basic idea of Behavior-based Mobility Prediction (BMP) is to identify the temporal clusters and segments of time-series representation of handoff history to characterize the behavior of users. This is done by analyzing the history of handoff occurrences as groups during certain periods within an hour, a day, or even a week, and performing handoff predictions that pertain to those periods.

3.2.1 User Behavior

Behavior of mobile users can be characterized in many different ways. In the proposed BMP method, the four characteristics that define user behavior are location, group, time-of-day, and duration. The following discusses the motivation for using these characteristics.

The location factor, as discussed before, represents the history of mobility patterns that can be either static or dynamic. Static mobility patterns are dictated by fixed structures, such as roads, building structures, and city blocks. On the other hand, dynamic mobility patterns are caused by frequent and drastic changes in the operating environment of WLANs due to multipath effects, user mobility, and electromagnetic interference.

The group factor reflects the fact that MSs often behave as groups. For example, MSs in an academic setting can be categorized as students, graduate students, faculty, and staff, and the mobility behaviors of these four groups are very different. Moreover, MSs can be associated with specific events that are derived from
user habits, e.g., department faculty/staff spends most of the day near the administrative offices while students congregate in the atrium, classrooms, and computer labs. The group factor can be statically applied during the network registration phase. For example, users in a typical campus network are registered with Unix accounts that are grouped based on students, graduate students, faculty, and staff. Users in large community networks are also registered based on different types of memberships, such as residential, business, free subscriber, etc. In addition to these pre-assigned groups, other groups can be dynamically formed from a set of MSs that suffer from high misprediction rate.

The time-of-day factor indicates the fact that user behaviors change as function of time. For example, mobility patterns observed in an academic setting will change during the course of a day depending on the schedule of classes. There will be bursts of repetitive mobility patterns when students move between classes, and mobility behaviors during the evening will be different from the daytime. Most of MSs in an academic network in the evening are graduate students and, for most part, they tend move only within limited areas (i.e., graduate student offices, laboratories, and hallways). Similarly, most users in a community network in the evening tend to stay within residential areas. In addition, both environments typically exhibit periodic behaviors such as students attending classes and workers commuting.

The duration factor directly represents how long a MS is connected to a cell, and indirectly represents the speed at which it moves through a cell. The duration can be categorized as short, medium, and long. The motivation behinds the duration factor is based on the fact that MS moves into new cell to either transit or stay to
perform activities. In addition, the mobility behavior after a MS transit or stay within a cell may be different. For example, the MS representing a student in KEC as shown in Figure 2.1(a), which moves from an atrium area covered with AP6 to a classroom covered with AP2 to listen to a lecture will have perform a similar handoff pattern to the MS representing a student in other department which enters the building at the east entrance and moves from an atrium area covered with AP6 before exits the KEC at the west entrance. After the first MS finishes listening to the lecture, it may move back to the atrium area. However, the second MS already leaves the building, and may not come back until the nextday. Similarly, the MS representing a resident in Portland as shown in Figure 2.1(b), which commutes from residential area to work in office areas will perform a similar handoff pattern to the MS representing a nomadic which moves from the same residential area to shopping area but also passes the same office area. The first MS is likely to come back to the residential area after it finishes the work, but the second MS may not. Since, these activities are directly related with a duration. Therefore the duration factor can be used to distinguish a unique characteristic of MS.

A medium duration represents a typical handoff that occurs when a MS transits through a cell. Therefore, most handoffs are categorized as medium duration and is used together with group and time-of-day factors to accurately model user behavior.

A long duration represents a MS performing some activity at a destination cell. The purpose of using a long duration is that the MS may perform a different mobility behavior from the norm after it finished performing an activity such as a
student moves back to atrium after finishes listening to the lecture or resident in Portland goes back home after finish work. Moreover, a long duration handoffs are considered as a special case of medium duration and are associated with separate IMAs.

A short duration often represents an unnecessary or a false handoff. For instance, consider a MS moving across three adjacent cells $c_x$, $c_y$, and $c_z$. As the MS moves from $c_x$ to $c_y$, and then to $c_z$, if the connection duration for $c_y$ is very short then it indicates the three adjacent cells are highly overlapped. This is important because if the degree of overlapping is sufficiently high then the MS can move from $c_x$ directly to $c_z$, eliminating one extra handoff. This is also the case when a MS moves into a coverage area that is shared by two cells representing 1st and 2nd next-cell predictions. If the MS reassociates with 1st predicted cell, but actually moves to the 2nd predicted cell, there will be another handoff to the 2nd predicted cell. Therefore, short duration can be used to identify and eliminate these unnecessary handoffs.

3.2.2 The Proposed Method

In order to perform mobility prediction, MS sends a prediction req$(ID, HS, D)$, where $ID$ is the ID of the MS, $HS$ is the Handoff Sequence $(c_{n-k-1}, \ldots, c_n)_{HS}$ for the MS, where $c_n$ represents the current cell, and $D$ is the duration of time spent by the MS in the last cell, i.e., $c_{n-1}$, to the server to obtain predictions for the next handoff. This is shown in Figure 3.1, where the current cell is $w$, i.e.
Figure 3.1: BMP Prediction request and response.

c_n = w, HS is \langle \ldots, y, x, w \rangle, and D is t_n - t_{n-1}. The BMP server will respond with a set of ordered next-cell Prediction List for both medium (oPL_{medium}) and long (oPL_{long}) durations. During the next handoff to cell c_{n+1} the MS chooses between oPL_{medium} and oPL_{long} depending on the duration D of the MS in the current cell, which is defined as the time elapsed between association/reassociation with the current cell c_n and handoff to the next cell c_{n+1}. If the MS performs the next handoff within t_{duration} (e.g., t_{long} = 1 hour), which is the threshold time for long duration, oPL_{medium} will be used to perform the next-cell predictions. Otherwise, oPL_{long} will be used.
Figure 3.2: The Behavior-based Mobility Prediction Scheme.

The BMP scheme is illustrated in Figure 3.2. The server then performs the following sequence of operations. First, HS is used to search the HS Table for matching entries, which represent an unordered next-cell Prediction List (uPL). At the same time, ID is used to index the Group Table to obtain the group ID $G_i$ in $\{G_0, G_1, \ldots, G_{p-1}\}$, where $p$ represents the number of groups. Second, $G_i$ is used to select a particular group’s Time-of-Day characteristic ($ToD_i$). As the name suggests, ToDs model time-of-day characteristics of different groups of mobile users. The default time period for $ToD_i$ is $T_i$ (e.g., $T = 1$ day), which means MSs
that belong to this group exhibit steady handoff behaviors throughout the entire period $T$. On the other hand, a group of MSs that exhibit short-term and periodic handoff behaviors can be separately modeled as $ToD_i = \{\tau_0, \tau_1, \ldots, \tau_{q-1}\}$, where $\tau_j$ represents a time segment and $\sum_{j=0}^{q-1} \tau_j = T$. Thus, the current time (i.e., clock) determines $\tau_j$ for a particular group $G_i$, (i.e., ($ToD_i, \tau_j$)). The importance of ($ToD_i, \tau_j$) is that it uniquely defines a particular integrated moving average (IMA) to represent the mobility behavior (i.e., HS) of a particular group at a particular time-of-day with different durations (i.e., medium, and long durations). Therefore, indices $i$ and $j$ from ($ToD_i, \tau_j$) together with $HS_{match}$ determine the proper set of IMAs to be used in applying the priority for both $oPL_{medium}$ and $oPL_{long}$.

Finally, $HS$ Table, $ToDs$, $Group$ Table, and IMAs are appropriately updated to track variations in mobility behavior of users. In the following, we discuss in detail the operations of the major components in the proposed scheme.

### 3.2.2.1 Handoff Sequence Table

$HS$ represents the mobility history of a MS and is denoted as $\langle c_{n-k-1}, \ldots, c_n \rangle_{HS}$, where $c_i$ indicates the cell ID of $i^{th}$ visited cell and $k$ represents the length of handoff history. $HS$ Table is a collection of unique HSs representing the global history of mobility patterns in the network. In order to perform mobility prediction, $\langle c_{n-k}, \ldots, c_n \rangle_{HS}$ is used to search $\langle c_{n-k-1}, \ldots, c_{n-1} \rangle_{HS}$ in the HS Table. The $c_n$ of all the matched HSs represent the candidate cells that the MS may visit in the future.
Table 3.1: Handoff Sequence Table and IMA.

<table>
<thead>
<tr>
<th>HS</th>
<th>IMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_1, c_4, c_5, c_6, c_2)</td>
<td>1.67</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
<tr>
<td>(c_1, c_4, c_5, c_6, c_3)</td>
<td>2.12</td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
</tr>
<tr>
<td>(c_1, c_4, c_5, c_6, c_5)</td>
<td>0.75</td>
</tr>
</tbody>
</table>

In general, the mobility prediction based on the HS Table can be represented by a Markov process. For example, the estimate of probability of MS with HS of \(c_{n-k-1}, \ldots, c_n\) associating with the candidate cell ID \(c_{n+1}\) can be written as

\[
\hat{P}(X_{n+1} = c_{n+1}| X(n - k - 1, n) = \langle c_{n-k-1}, \ldots, c_n \rangle) = \frac{N(\langle c_{n-k-1}, \ldots, c_n, c_{n+1} \rangle, L)}{N(\langle c_{n-k-1}, \ldots, c_n \rangle, L)}
\]

(3.1)

where \(L\) is the overall history of mobility patterns. \(N\) is the average frequency of mobility patterns stored in IMA. Each HS Table entry, and thus each next-cell prediction, has an associated frequency of occurrence represented as an IMA. By default, these IMAs are for a period \(T\). For example, Table 3.1 shows three matches with \(c_2\), \(c_3\), and \(c_5\) representing the list of next-cell predictions based on HS of \(\langle c_1, c_4, c_5, c_6 \rangle\). Based on the IMA values, the estimate of probability of MS associating with \(c_2\), \(c_3\), or \(c_5\) based on HS of \(\langle c_1, c_4, c_5, c_6 \rangle\) is 0.466, 0.368, or 0.166. Thus, the ordered next cell prediction list is \(\{c_3, c_2, c_5\}\).
3.2.2.2 Group Formation/Dissolution

Separate time-series models (i.e., IMAs) are maintained for different user groups. As discussed before, group formation can be performed statically during the network registration phase. In addition, groups can be dynamically formed by applying heuristics to a set of MSs that suffer from high misprediction rate. In order to gather the necessary information for the heuristics, MSs would keep track of and submit their overall prediction accuracies to the mobility prediction server at the end of each period $T$ (e.g., $T = 1$ day). A new group is formed if $1^{st}$ prediction accuracies during the last period of $T$ of a group MSs are lower than the preset threshold $\rho_{form} = acc_{avg} - 0.5\sigma$, where $acc_{avg}$ and $\sigma$ represent the overall $1^{st}$ prediction accuracy and its and standard deviation from the last period $T$. If MSs in this newly formed group still suffer from high misprediction rate, which can be cause by differences or changes in mobility behaviors within the group, group dissolution is applied. In other words, if the server detects MSs in this group with the $1^{st}$ prediction accuracy lower than the preset threshold $\rho_{diss} = acc_{avg}$, the MSs are deleted from the group. In order to reduce the processing load, the server will not allow any group to be formed if the number of MSs in a group is below some threshold $N$. A reasonable value of $N$ is based on the traffic characteristics of the network and capability of the server. Note that a new group can be formed every period $T$ and the network can have arbitrary number of groups as long as each group has more than $N$ MSs.
3.2.2.3 Time-of-Day Characteristic

The time-of-day factor is applied when a group’s prediction accuracy is lower than some threshold for a period of time. This is done by associating a separate time-series models to different time segments of the day. The motivation for generating time-segments is to isolate periods where mobility patterns can be better identified. Therefore, each time-segment employs a separate IMA model to better predict time-dependent mobility behavior. The server does this by keeping track of the prediction accuracy of each group for a period of $\tau_{\text{min}}$, where $\tau_{\text{min}}$ is a minimal time-segment length. If the prediction accuracies of a particular group in last $\tau$ is lower than $\rho_{\text{diss}}$, the server assigns a new time-series model to the time-segment. Again, the value of $\tau_{\text{min}}$ is based on network traffic and server capability. If the IMA values of two consecutive time-segments are close together, time-segments can be combined.

3.2.2.4 IMA

IMAs are typically used in forecasting time series data, and can be derived from the more general AutoRegressive Integrated Moving Average (ARIMA) model. Since, the purpose of ARIMA is to perform a moving average on the time-series data, the autoregressive part can be ignored. Thus, IMA is simply ARIMA($0$, $d$, $q$), where $0$, $d$, and $q$ refer to the order of the autoregressive, the differencing, and the moving average parts of the model, respectively. Exponential Weighted Moving Average (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

\[ \tilde{z}_{t+1} = (1 - \lambda)\tilde{z}_t + \lambda z_t \]

where \( z_t \) is the sampled time-series data, \( \tilde{z}_t \) 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 0.01. Although EWMA does not rely on the full statistical analysis to estimate the order and the coefficients, our prior studies [15] show that this simple model gives results that are relatively close to ones from ARIMA.
The IMA structure is shown in Figure 3.3, which is essentially a three-dimensional array. The $HS_{match}$ and $i$ from $ToD_i$ are used to index to the first two dimensions of the IMA structure. This results in a set of EWMAs that correspond to a particular group’s ($G_i$) ToD characteristics ($ToD_i$). Each element in the last dimension is implemented as a linked-list indexed by $j$ from $\tau_j$ containing a pair of EWMAs corresponding to a particular time segment with a medium and long durations represented as EWMA$_M$ and EWMA$_L$, respectively. The total number of elements in the list is $q$, and $q$ will vary depending on the number of available time segments.

3.2.2.5 Update operation

After each handoff, the following set of update operations are performed.

- **HS Table and IMA Structure Update:** If $\langle c_{n-k-1}, \ldots, c_n \rangle_{HS}$ of MS matches with $\langle c_{n-k-1}, \ldots, c_n \rangle_{HS}$ in the HS Table, the server checks whether the duration is either medium (i.e., $t_{short} \leq D < t_{long}$) or long ($t_{long} < D$), where $t_{short} = 1$ sec and $t_{long} = 1$ hour, then the IMA value for the corresponding entry is updated. In addition, if MS is joining the cell for the first time, then a new HS entry is allocated in the HS Table and an IMA is created. IMAs are updated every $t = 1$ min., which is the time interval used to sample frequency of handoffs. Note that the handoff associated to short duration (i.e., $D < t_{short}$) will not be used to update IMA, since it represents a false handoff.
• **Group Table Update**: The group can be statically and dynamically formed. If a MS belongs to both statically and dynamically formed groups, then the priority is given to the dynamically formed group. Once a group is formed, a new $ToD_i$ and an IMA are generated for the group. Initially, $ToD_i$ starts with only a single time-segment, i.e., $\tau_j=T$, which will points to a newly generated IMA. After a group formation is performed, $ID_{MS}$s are registered to a new group. MSs are never registered with two dynamic groups at the same time. The MS which still exhibits high miss prediction rate (i.e., prediction accuracy is lower than $\rho_{diss}$) is unregistered from the group. If the total number of MSs in dynamic group is less than $N$ (i.e., $N$ is based on the traffic characteristics of the network and capability of the server.) the group is dissolved and the $ToD_i$ and an IMA are deleted.

• **$ToD$ Update**: The server monitors the prediction accuracy of each group for a period of $\tau_{min}$. If the prediction accuracies of a particular group in last $\tau_{min}$ is lower than $\rho_{diss}$, the server assigns a new time-series model to the time-segment. When the IMA values of two consecutive time segments are close together (e.g., closer than 10%) the two time segments are combined.
3.3 Performance Evaluation

This section presents the performance evaluation of the proposed Behavior-based Mobility Prediction Scheme. Subsection 3.3.1 describes the simulation environment as well as the two key components of the simulator - *path generator* and *handoff detector*. Subsection 3.3.2 discusses the delay parameters used in the study. Subsection 3.3.3 compares the accuracy and delay results of the proposed Behavior-based Mobility Prediction Scheme against long-term counter based [14] and time-series based GPC [15].

3.3.1 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.*, students, graduate students, and faculty/staff, 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, *faculty/staff* moves mainly between their offices and the atrium.
The results for Portland were generated based on nine different groups of 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), and 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 move every 45 to 75 minutes 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 [26]. 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 $t_{\text{max}}$, $t_{\text{min}}$, $t_{\text{auth}}$, and $t_{\text{assoc}}$ (see Section 3.3.2). 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 predefined 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.

### 3.3.2 Simulation Delay Parameters

The delay parameters used in the simulation are shown in Table 3.3.2: *Channel Switching Time* \( t_{\text{switch}} \) is the time required to switch from one channel to another; *MinChannelTime* \( t_{\text{min}} \) is the minimum amount of time a MS has to wait on an empty channel; *MaxChannelTime* \( t_{\text{max}} \) 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* \( t_{\text{auth}} \) is the time required to perform authentication based on MAC addresses; and *Reassociation delay* \( t_{\text{assoc}} \) is the time requires to perform reassociation.
The delay values 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 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. $t_{\text{switch}}$ 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 $2 \cdot t_{\text{switch}}$, and thus $t_{\text{switch}}$ is assumed to be 11.4 ms. $t_{\text{auth}}$ 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, $t_{\text{auth}}=6$ ms ensures that it is longer than the time between the authentication request and response. Similarly, $t_{\text{assoc}}$ is estimated from the average round-trip time of reassociation request and response, which is $t_{\text{assoc}}=4$ ms. $t_{\text{max}}$ was estimated by observing the time between a probe request and an authentication request, which is 199.4 ms. This is consistent with the $t_{\text{max}}$ value provided in the source code of the open source wireless network device driver [35]; therefore, $t_{\text{max}}$ is assumed to be 200 ms. On the other hand, there is no direct method to measure $t_{\text{min}}$. Thus, the reference value of $t_{\text{min}} = 20$ ms is assumed as in [37]. The
Table 3.2: Delay parameters used in the simulation.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Switching Time ($t_{\text{switch}}$)</td>
<td>11.4 ms</td>
</tr>
<tr>
<td>MinChannelTime ($t_{\text{min}}$)</td>
<td>20 ms</td>
</tr>
<tr>
<td>MaxChannelTime ($t_{\text{max}}$)</td>
<td>200 ms</td>
</tr>
<tr>
<td>Authentication delay ($t_{\text{auth}}$)</td>
<td>6 ms</td>
</tr>
<tr>
<td>Reassociation delay ($t_{\text{reassoc}}$)</td>
<td>4 ms</td>
</tr>
</tbody>
</table>

delay values were obtained from average of 2400 measurements over a period of a day to reduce variations due to network traffic.

3.3.3 Simulation Results

Figure 3.4(a) shows the 1st prediction accuracy of BMP versus the basic GPC scheme for KEC. The overall average improvements using BMP against the basic GPC scheme is 23.0% . Moreover, BMP improves the 1st prediction for all groups in KEC. The amount of improvements vary for different user groups. The stack plot shows that statically identifying groups resulted in the largest improvement of 7.7% 56% compared to the basic GPC scheme. The dynamic group formation provided another -0.14% 11.9% improvement. The time-of-day factor provides an additional improvement of 1.1% 7.5%. Among the three groups the largest improvement occurred for 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. The duration factor provided an
improvement of 0.4%  7.9% with students benefiting the most.

Similarly, most of the user groups in Portland resulted in significant improvements (see Figure 3.4(b)) with an overall average improvement of 42.9%. Among them, Nomadic and commuter groups (C-I, C-II, C-III, and C-IV) exhibited large improvements from formation of static groups. This is because the mobility behavior of the Nomadic group is much more far reaching and random, which is different from the rest of the user groups. On the other hand, commuter groups only move within limited areas and during rush hour. Therefore, commuter groups show additional improvement when the Time-of-Day factor is applied. In contrast, resident groups (except R-IV) gain the most from applying the duration factor. This is because the mobility behaviors of resident groups occur every 45 - 75 min. and within small areas.

Finally, Figure 3.3.3 compares the handoff delays based on the parameter set defined in Table 3.3.2. These results show that BMP provides 12.7% and 23.1% improvement for KEC and Portland, respectively. The handoff delay for KEC may appear to be a moderate improvement compared to the basic GPC scheme, but the resulting delay for BMP is very close to the lower bound delay, which is 21.4 ms = \( t_{\text{switch}} + t_{\text{auth}} + t_{\text{assoc}} \). More importantly, when individual handoff delays are considered, they resulted in significant improvements for some user groups. For example, the Student group in KEC resulted in 27.4% improvement, while Grad. Students had 21% improvement. This was also the case for Portland, where the Nomadic group resulted in 35.2% improvement over the basic GPC scheme. In addition, all groups resulted in similar average delay.
Figure 3.4: 1st prediction Accuracy.
Figure 3.5: Handoff Delay.
3.4 Conclusion

This thisis described the BMP technique to provide better next-AP prediction for MSs. BMP models the regularity in mobility patterns based the following four behavior factors: location, group, time-of-day, and duration. Therefore, it captures the dynamic mobility patterns of MSs. Although, the BMP only shows small improvement in overall handoff delay, when individual groups of MSs are considered the improvements are significant. This is because the proposed BMP increases prediction accuracy in every group of MSs.
Chapter 4 – Future Work

For future work, we plan to investigate couple of issues. First, we plan to investigate the effectiveness of BMP 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 BMP can be utilized to speed up vertical handoffs.
Bibliography


APPENDICES
The Markov-based Mobility Prediction

Consider a MS’s mobility history \( L = a_1a_2a_n \), where each symbol \( a_i \) represents the \( i^{th} \) visited AP. Let substring \( L(i, j) = a_ia_{i+1} \cdots a_j \) for any \( 1 \leq i \leq j \leq n \). Suppose the current state of the predictor is defined as \( c = L(n - k + 1, n) \). Let \( a \) be the set of all possible APs. Assuming MSs location as a random variable \( X \), let \( X(i, j) \) be a string \( X_iX_{i+1} \cdots X_j \) representing the sequence of random variables \( X_i, X_{i+1}, \ldots X_j \) for any \( 1 \leq i \leq j \leq n \). The order-\( k \) mobility prediction algorithm can be described by the following Markov process [50]:

\[
P(X_{n+1} = a|X(1, n) = L) = P(X_{n+1} = a|X(n - k + 1, n) = c) \\
= P(X_{i+k+1} = a|X(i + 1, i + k) = c) \tag{1}
\]

where the notation \( = P(X_i = a_i|\ldots) \) denotes the probability that \( X_i \) takes the value \( a_i \). The above equations indicate that stochastic variables that describes the probability depends only on the last \( k \) symbols and assumes a stationary distribution. These probabilities can be represented by a transitional matrix \( M \), where both rows and columns of \( M \) are indexed by length-\( k \) string so that \( P(X_{n+1} = a|X(1, n) = c) = L(1, n) = M(s, s') \) where the current context is represented by \( s = L(n - k + 1, n) \), and the next context is represented by
\( s = L(n - k + 2, n)a \). The elements of \( M \) can be generated from an estimate \( \hat{P} \) from the current history \( L \), the current context \( c \), and the equation

\[
\hat{P}(X_{n+1} = a|L) = \frac{N(c a, L)}{N(c, L)}
\]  

(2)

where \( N(s', s) \) denotes the number of times the substring \( s' \) occurs in string \( s \).

Therefore, prediction is made by scanning the row of \( M \) that corresponds to the current context \( c \) and choosing the entry with the highest predictions.
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

$$(1 - \phi_1B - \phi_2B^2 - \cdots - \phi_pB^p)\nabla^d z_t = (1 - \theta_1B - \theta_2B^2 - \cdots - \theta_qB^q)\varepsilon_t,$$

where $z_t$ is the time-series data, $\phi$ is the autoregressive parameter, $\theta$ is the moving average parameter, $B$ is the backshift operator, which is 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 $x_t$, 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.

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

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

$$PACF(h) = \begin{cases} 
  corr(x_1, x_0), & h = 1 \\
  corr(x_h - x_{h-1}, x_0 - x_{0}^{h-1}), & h \geq 2
\end{cases}$$

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

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

and $x_{h-1}^{h-1}$ 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.
Table 1: Behavior of ACF and PACF for the ARIMA model.

<table>
<thead>
<tr>
<th></th>
<th>ARIMA $(p, d, 0)$</th>
<th>ARIMA $(0, d, q)$</th>
<th>ARIMA $(p, d, q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>Tail off</td>
<td>Cut off after lag $q$</td>
<td>Tail off</td>
</tr>
<tr>
<td>PACF</td>
<td>Cutoff after lag $p$</td>
<td>Tail off</td>
<td>Tail off</td>
</tr>
</tbody>
</table>

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 fre-
Figure 1: ACF and PACF from the transformed time-series data in Figure 5.

The frequency of handoff sequence \( AP_4 \to AP_5 \to AP_6 \) 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 16. 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

\[
\nabla^2 z_t = (1 - \theta_1 B - \theta_2 B^2) \varepsilon_t, \tag{3}
\]

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 known information, the model can be rewritten as

\[
z_t = \sum_{j=1}^{\infty} \pi_j z_{t-j} + \varepsilon_t, \tag{4}
\]
where \( \pi_j \) is a weighted average coefficient and \( \sum_{j=1}^{\infty} \pi_j = 1 \).

Based on (4), the prediction model in general can be written as

\[
\tilde{z}_{t+1} = \sum_{j=1}^{\infty} \pi_j \tilde{z}_{t+1-j},
\]

where \( \tilde{z}_{t+1} \) is a predicted time series data.

Next, (4) can be rewritten as

\[
\varepsilon_t = (1 - \pi_1 B - \pi_2 B^2 - ...) z_t
\]

Using \( \varepsilon_t \) from (6), (3) can be rewritten as

\[
(1 - 2B + B^2) z_t = (1 - \theta_1 B - \theta_2 B^2)(1 - \pi_1 B - \pi_2 B^2 - ...) z_t
\]

From (7), the weighted average coefficient can be defined as \( \pi_1 = 2 - \theta_1 \), \( \pi_2 = \pi_1 - (1 - \theta_2) \) and \( \pi_n = \theta_1 \pi_{n-1} + \theta_2 \pi_{n-2}, n \geq 3 \). Equation (5) can be written as

\[
\tilde{z}_{t+1} = \sum_{j=1}^{\infty} \pi_j \tilde{z}_{t+1-j}
\]

\[
= \pi_1 z_t + \pi_2 z_{t-1} + \sum_{j=3}^{\infty} \pi_j z_{t+1-j}
\]

From (9), substitute \( \pi_j \) with \( \theta_1 \pi_{n-1} + \theta_2 \pi_{n-2}, n \geq 3 \). Equation (9) can be written as
\[ z_{t+1} = \pi_1 z_t + \pi_2 z_{t-1} + \sum_{j=3}^{\infty} (\theta_1 \pi_{j-1} + \theta_2 \pi_{j-2}) z_{t+1-j} \]

\[ = \pi_1 z_t + \pi_2 z_{t-1} + [\theta_1 \sum_{j=1}^{\infty} \pi_{j-1} z_{t+1-j} + \theta_1 \pi_1 z_{t-1})] \]

\[ + \theta_2 \sum_{j=1}^{\infty} \pi_j z_{t-2-j} \]

From (10), substitute \( \sum_{j=1}^{\infty} \pi_{j-1} z_{t+1-j} \) and \( \sum_{j=1}^{\infty} \pi_j z_{t-2-j} \) with \( z_t \) and \( z_{t-1} \) respectively. Equation (10) can be written as

\[ z_{t+1} = \pi_1 z_t + (\pi_2 - \theta_1 \pi_1) z_{t-1} + \theta_1 z_t + \theta_2 z_{t-1} \]

\[ = (2 - \theta_1) z_t - (1 + \theta_2) z_{t-1} + \theta_1 z_t + \theta_2 z_{t-1} \]