Directional Handoff using Geomagnetic Sensor in Indoor WLANs

Sangyup Han*, Myungchul Kim*, Ben Lee† and Sungwon Kang*
* *Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea*
E-mail: {ilu8318, mck, sungwon.kang}@kaist.ac.kr
† *School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97330, USA*
E-mail: benl@eecs.orst.edu

***Abstract*—More and more mobile devices, such as smartphones and pad/tab devices, are being used in IEEE 802.11 Wireless LANs (or Wi-Fi). However, mobile users are currently unsatisfied with using Wi-Fi on the move due to large handoff delay. In order to perform fast handoff, this paper proposes a new scheme using a geomagnetic sensor (or a digital compass) embedded in mobile devices. The proposed scheme predicts the direction of movement of a Mobile Station (MS) from the currently associated Access Point (AP) and performs active scanning with a reduced number of channels. The proposed scheme was implemented in Android smartphones and their performance was evaluated in a real indoor WLAN environment. Our test results show that the proposed scheme reduces handoff delay compared to conventional handoff and selective scanning scheme. In addition, the proposed scheme does not require modification to existing APs, which makes it very practical for using real-time multimedia services on current mobile devices.**

***Keywords*-IEEE 802.11; Fast handoff; Directional handoff; Geomagnetic sensor; Digital compass; AP Table**

I. Introduction

With the popularity of smartphones and pad/tab devices, more and more mobile users are relying on IEEE 802.11 WLANs (or Wi-Fi) for multimedia services due to their high data rate and low cost. However, since WLANs do not support fast handoff from one AP to another, mobile users may experience significant delay or even lose connectivity on the move during real-time multimedia services, such as Voice over IP (VoIP) and video conferencing.

Fast handoff in WLANs is achieved by minimizing the time required to scan for available APs. This requires knowing the location and/or direction of movement of Mobile Stations (MSs) to improve the accuracy of the next AP prediction for association. The easiest way to determine the location is to utilize Global Positioning System (GPS) [1]. However, its power requirement and delay incurred in retrieving the location information are very high, and it cannot be used in an indoor environment. Another way is to use a Wi-Fi Positioning System (WPS) [2]. However, WPS may not be practical since its implementation and deployment require extra time and manpower. In the absence of GPS or WPS, most of the prior work on reducing the scanning delay involves predicting a limited number of channels to scan based on the history of past scans [3] or mobility patterns [4] and periodically scanning neighboring channels [5]. However, these schemes either require modification to the existing APs [4], [5] or do not reduce the scanning delay for the case of IEEE 802.11a/n [3].

Therefore, this paper proposes a new method called *Directional Handoff (DH) scheme* that predicts the direction of an MS by using a geomagnetic sensor embedded in the MS. The proposed scheme reduces handoff delay by limiting the number of candidate APs to scan by one or two AP(s) without any modification to the existing APs. The effectiveness of the proposed method is demonstrated through implementation on real smartphones in an indoor WLAN environment.

This paper is organized as follows: Section II discusses the background and previous studies on fast handoff in WLANs. Section III presents the proposed Directional Handoff (DH) scheme. Section IV describes the implementation of the DH scheme on real smartphones, and discusses the experimental setup and results. Finally, Section V concludes the paper and discusses future work.

II. Background and Related Work

Handoff in IEEE 802.11 WLANs consists of three steps: scanning, authentication, and re-association [6]. *Scanning* is the process of finding new APs to connect to. The next AP to handoff to is chosen from scanning results by considering various factors, such as Received Signal Strength Indicator (RSSI). After deciding on the AP, the *authentication* process sends information of the MS to the new AP. After authentication, the *re-association* process requests reconnection to the new AP. The handoff process is complete when the MS is allowed to reconnect. Therefore, handoff delay is defined as the sum of the time spent in each step and is given as

$$T_{Handoff} = T_{Scanning} + T_{Authen.} + T_{Reassoc.} \quad (1)$$

The time required for an MS to switch to another channel is negligible and thus is not included in Eq. 1. The overall handoff delay is typically in the range of 200∼400 ms, and the scanning step represents approximately 90% of this delay [6]. The authentication and re-association steps each take up to 10 ms and have relatively minimal impact on the overall delay. The reasons why the scanning process dominates handoff delay are related to the scanning method

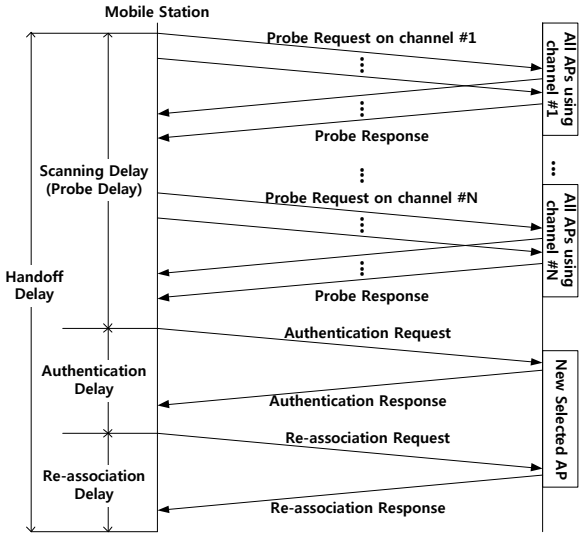


Figure 1: Frame exchange during handoff using active scanning.

used and the number of channels scanned. There are two scanning methods: passive scanning and active scanning. *Passive scanning* finds APs by listening to beacon frames periodically transmitted by the APs, which is typically every 100 ms. As a result, an MS has to wait as long as 100 ms on a single channel to receive beacon frames. Therefore, the time taken for passive scanning, T_{PS}, with N channels is defined as follows:

$$T_{PS} = N \times BeaconInterval \quad (2)$$

There are at least 11 channels in IEEE 802.11b/g [7], and thus passive scanning requires 1,100 ms (Eq. 2), and handoff delay takes 1,120 ms (Eq. 1). Since International Telecommunication Union (ITU) recommends handoff delay for VoIP application not to exceed 150 ms [8], passive scanning causes severe disruption for VoIP services.

In *active scanning*, an MS sends probe request frames to APs by either broadcasting or unicasting. Fig. 1 illustrates frames exchanged during handoff using active scanning. An MS sends probe request frames and then waits for corresponding response frames for each channel. Therefore, the time taken for active scanning, T_{AS}, can be defined as follows:

$$N \times T_{MinChannel} \leq T_{AS} \leq N \times T_{MaxChannel}, \quad (3)$$

where $T_{MinChannel}$ and $T_{MaxChannel}$ represent *MinChannelTime* and *MaxChannelTime*, respectively. An MS first sends probe request frames and waits for *MinChannelTime*. If it does not receive any probe response frames, it switches to the next channel. If more than one probe response frame is received within *MinChannelTime*, it waits until *MaxChannelTime* and then switches to the next channel.

A prior study showed that 6.5 ms and 11 ms are a good choice for *MinChannelTime* and *MaxChannelTime*, respectively. Therefore, active scanning delay in IEEE 802.11b/g can be up to 121 ms, and handoff delay can be up to 141 ms. However, since IEEE 802.11a/n using the 5 GHz band has at least 16 channels, the time required for scanning increases even more. Therefore, the conventional handoff scheme that actively scans all the channels is not suitable for real-time multimedia services.

There have been many studies on fast handoff for WLANs. Shin *et al.* proposed *selective scanning and caching* [3]. Selective scanning reduces the number of channels to scan using a data structure called *channel mask*. However, the number of channels to be scanned will increase for the case of IEEE 802.11a/n. *Caching* enhances selective scanning by skipping the scanning step and directly performing authentication and re-association based on the previously cached scanning results. However, the cached predictions will be incorrect if MSs move in the opposite directions of the previous movements.

Ramani and Savage introduced *SyncScan*, which finds new APs by periodically performing passive scanning on the surrounding channels [5]. However, existing APs need to be modified to synchronize the transmission timing of beacon frames, which may not always be guaranteed.

Purushothaman and Roy proposed the *Enhanced FastScan* (EFS) [9]. EFS reduces scanning delay by restricting the number of candidate APs to scan based on the predicted location of an MS. EFS divides the coverage area of the current AP into four areas: NE, NW, SE, and SW. To determine the area where MS is located, Wi-Fi Positioning System (WPS) is used, and the coordinates of APs are added to beacon frames. However, WPS requires a large amount of scanned data to be processed, which is time consuming. Furthermore, EFS requires modification to the existing APs.

Wanalertlak *et al.* proposed the *Behavior-based Mobility Prediction* (BMP) mechanism [4], which performs fast handoff without scanning by predicting the next AP to connect based on the mobility patterns of MSs. BMP enhances the accuracy of the next AP prediction by considering the behavior of users that includes past handoff sequences, users' affiliation or status, time-of-day, etc. However, BMP has the disadvantage that it requires modification to the existing APs to process modified authentication frame.

III. Directional Handoff Scheme

The proposed DH scheme predicts the direction of an MS using a geomagnetic sensor embedded in the mobile device, which reduces the handoff delay by scanning only the candidate APs on the predicted direction. This is done by using an *Access Point Table* (AP Table), which stores at most two target APs to scan with respect to locations of APs and direction of movement of MS.

A. Overall Procedure

Fig. 2 shows the flowchart of the proposed DH scheme. First, an MS checks RSSI of the currently associated AP.

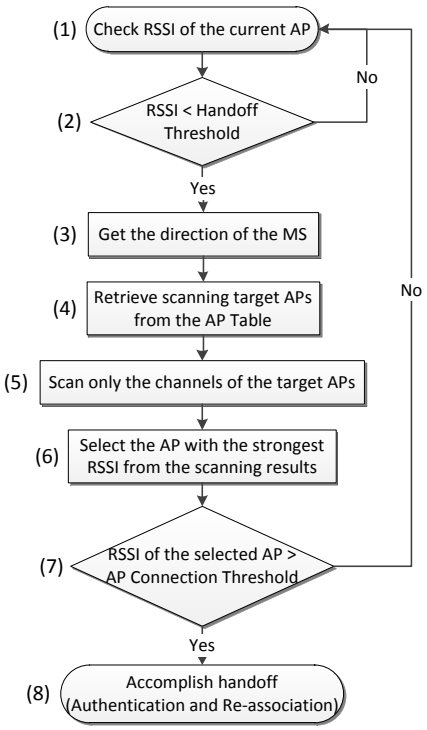


Figure 2: Flowchart for the Directional Handoff scheme.

In this study, whenever an MS receives a packet, the current RSSI, $RSSI_{Current}$, is calculated using the following formula:

$$RSSI_{Current} = RSSI_{Previous} \times (1 - \lambda) + RSSI_{Latest} \times \lambda, \quad (4)$$

where $RSSI_{Previous}$ is the previous RSSI value calculated using Eq. 4, $RSSI_{Latest}$ is the latest RSSI value measured from the received packet, and λ ($0 \leq \lambda \leq 1$) is a RSSI filter weight determining how much $RSSI_{Latest}$ reflects the current RSSI. In this study, λ value of 0.5 is used.

An MS determines whether or not to perform a handoff by comparing $RSSI_{Current}$ with the handoff threshold (Step 2). If $RSSI_{Current}$ is less than the threshold, then the MS initiates the scanning process. Before scanning, target APs and their channels must be chosen. In the proposed DH scheme, an MS selects the target APs depending on its direction of movement. An MS uses a geomagnetic sensor to obtain its direction of movement (Step 3, the detailed procedure is described in Section III-B.). Thereafter, the MS retrieves indices of the candidate next APs from the AP Table (Step 4). These indices are used to retrieve MAC addresses and channel frequencies from a *Mapping Table*, which stores MAC addresses and channel frequencies for all the APs.

If there are no target APs, then the MS performs active scanning on all the channels (Step 5). Our implementation of active scanning transmits probe request frames by either broadcasting or unicasting. If target APs exist, the MS uses unicasting because it can retrieve their MAC addresses and channel frequencies from the Mapping Table. This way, if the MS receives probe response frames before *MaxChannelTime*, it does not have to wait for additional probe response frames and can immediately switch to the next channel. On the other hand, if there are no target APs, the MS uses broadcasting to find all APs.

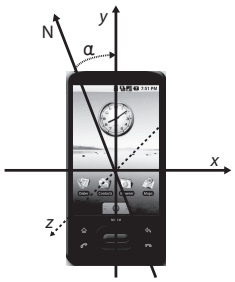


Figure 3: 3-axis of a geomagnetic sensor and the azimuth α.

After scanning, if the number of target APs is more than one, the MS selects the AP with the strongest RSSI (Step 6). Afterwards, this value is compared to the AP connection threshold (Step 7). For example, if the threshold is -75 dBm, an MS only connects to an AP whose RSSI is greater than -75 dBm. This minimizes the ping-pong effect (i.e., oscillating handoffs) [5]. Finally, the MS performs authentication and re-association steps to complete the handoff.

B. Getting Direction

This section describes the procedure to obtain the direction of movement of an MS (Step 3 of Fig. 2). Current smartphones and pad/tab devices have a variety of sensors, such as a gyro sensor (gyroscope), an acceleration sensor (accelerometer), a geomagnetic sensor (digital compass or orientation sensor), etc. The proposed DH scheme uses a geomagnetic sensor to obtain the azimuth of an MS, which is defined as the angle measured clockwise from the magnetic north of the earth to the y-axis of an MS. The symbol α in Fig. 3 denotes the azimuth of an MS. α varies from 0° to 360°, where 90° is east, 180° is south, 270° is west, and 0° and 360° represent north.

Fig. 4 shows the procedure for obtaining the direction of movement of an MS. The azimuth can be obtained by receiving an event from the *sensor manager* (Steps 1 and 2), which is one of the components in a mobile operating system, such as Android OS. The sensor manager tracks the changes in angle from the geomagnetic sensor and provides this information as events to other components and the application software. In order to receive events, the procedure shown in Fig. 4 specifies the type of sensor to keep track of and the time interval to the sensor manager. In our implementation, MS uses a time interval of 50$\sim$200 ms, which is sufficient to obtain the direction of an MS.

There are two considerations to be made in using the azimuth. First is the screen orientation of an MS: portrait mode vs. landscape mode. Fig. 3 shows a portrait mode. If

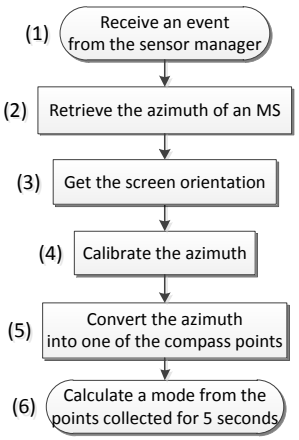


Figure 4: Flowchart for obtaining the direction of an MS.

the MS is put on its sides, then it becomes a landscape mode. If the screen mode is changed from portrait to landscape, the azimuth should be calibrated as follows (Steps 3 and 4):

- If an MS is put on its left side,
  (α_{Left}: before the calibration, α: after the calibration)

$$\alpha = \begin{cases} \alpha_{Left} - 90^\circ, & \text{if } \alpha_{Left} \geq 90^\circ \\ \alpha_{Left} - 90^\circ + 360^\circ, & \text{otherwise} \end{cases} \quad (5)$$

- If an MS is put on its right side,
  (α_{Right}: before the calibration, α: after the calibration)

$$\alpha = \begin{cases} \alpha_{Right} + 90^\circ, & \text{if } \alpha_{Right} \leq 270^\circ \\ \alpha_{Right} + 90^\circ - 360^\circ, & \text{otherwise} \end{cases} \quad (6)$$

The screen orientation (or screen mode) can be obtained by receiving events from mobile OS or by invoking Application Program Interface (API) commands. Since mobile OS must display its screen differently according to the screen orientation, it periodically traces the changes in the values of the geomagnetic sensor. In our implementation, the procedure receives the azimuth and the screen orientation information only through events.

Note that the geomagnetic sensor products developed by Asahi [11] and Yamaha [12] consume very little power (typically 1∼30 mW), which is lower than other sensors discussed in [13]. Therefore, continuous use of the geomagnetic sensor has minimal impact on the total power consumption.

The second consideration is related to phone calls. When an MS changes to the phone call mode, the azimuth must be calibrated based on Eq. 5 or 6 depending on which ear is used. However, this paper only considers the case where users are holding mobile devices in hand and, for example, using VoIP through speaker or earphones. The consideration for the phone call mode is left as future work.

After calibration, the procedure converts the azimuth into one of the eight compass points shown in Table I (Step 5). However, these compass points may not correspond to actual directions of MSs due to sensitivity and variations in the geomagnetic sensor. Therefore, the procedure calculates a mode from compass points collected over a period of 5 seconds (Step 6).

Table I: Conversion between azimuths and compass points.

Azimuth	Direction	Azimuth	Direction
0∼22, 338∼360	N	158∼202	S
23∼67	NE	203∼247	SW
68∼112	E	248∼292	W
113∼157	SE	293∼337	NW

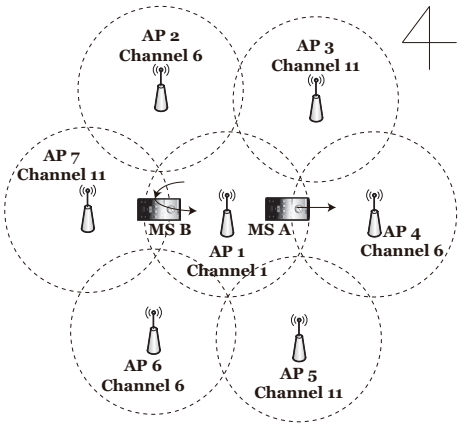


Figure 5: An example of WLAN deployment in an environment with no obstacles (MSs *A* and *B* are associated with AP1).

C. AP Table Structure

The AP Table is a data structure that stores target APs to scan with respect to locations of APs and direction of movement of MS. In this study, we assume that network administrators can construct the AP Table since the locations of APs are known during deployment. This is achieved by considering the available APs for each possible direction from an associated AP. If there are multiple APs in that direction, up to two APs with the strongest RSSI are stored in the AP Table. This process is repeated for all the APs in the network. The constructed AP Table is then stored in a server (co-located with the authentication server), which is downloaded to MSs when they first join the network.

Table II shows an example AP Table for the network topology shown in Fig. 5, where both MS *A* and MS *B* are associated with AP1 and moving towards the east. Each row stores current AP index, direction, and indices of at most two candidate APs to scan on that direction. The AP Table contains up to two target APs on each direction since at most two APs with the same Service Set IDentifier (SSID) cover each direction in Fig. 5.

Table II shows only the entries whose current AP index is AP1, but the information of other APs are recorded in the same manner. In addition, the table only uses indices of APs to eliminate redundancy. As mentioned before, MAC addresses and channel frequencies of APs are stored in the Mapping Table (not shown), which simplifies the updating

Table II: AP Table for the environment shown in Fig. 5.

Current AP	Direction	Next AP1	Next AP2
AP1	N	AP2	AP3
AP1	NE	AP3	AP4
AP1	E	AP4	-
AP1	SE	AP4	AP5
AP1	S	AP5	AP6
AP1	SW	AP6	AP7
AP1	W	AP7	-
AP1	NW	AP7	AP2

process when APs and/or channel frequencies change.

From the AP Table, an MS can easily find out in advance the target APs to scan. For example, MS *A* of Fig. 5 is moving to the east and discovers that the target AP is AP4 by searching the AP Table. The MS will then perform handoff to AP4 by scanning only the channel 6 of AP4.

The AP Table may look similar to Enhanced FastScan (EFS) database proposed in [9]. However, the difference is that the AP Table is constructed by considering *the direction* of an MS, while the EFS database is constructed by considering *the location* of an MS. As discussed before, it is easier and more cost effective to find out the direction of an MS than to find out its location.

Although the proposed DH scheme does not distinguish locations of the two MSs of Fig. 5, only MS *A* can handoff to AP4 because when MS *B* scans channel 6 of AP4, the scanning results will not return AP4 as a viable AP. Therefore, the proposed DH scheme prevents MSs from handing off to incorrect APs, because the direction of movement and the scanning result together indicate the approximate location of the MS. This approximate location is only useful when the signal radius of APs is less than 100 m (as in APs deployed in indoor areas).

Fig. 5 is an example of WLAN deployment with no obstacles. However, most indoor environments have walls and other obstacles, which limit the movement of MSs, e.g., from the east to the west or from the south to the north, etc. In these situations, predicting the movement of an MS and target APs is simpler, and the AP Table can easily be constructed beforehand.

IV. Implementation and Analysis

A. Cross-layer Implementation

Mobile devices used in this study are listed below:

- HTC Dream (G1) [14]: 528 MHz CPU clock, 192 MB RAM, Texas Instruments WL1251B Wi-Fi chipset
- Motorola Defy [15]: 800 MHz CPU clock, 512 MB RAM, Texas Instruments WL1273 Wi-Fi chipset

Both of these devices are Android platform-based smartphones. Android is an open source software [16] and Wi-Fi driver source code for the two devices is also open source [17]. Note that the proposed DH scheme does not require modification to the existing APs.

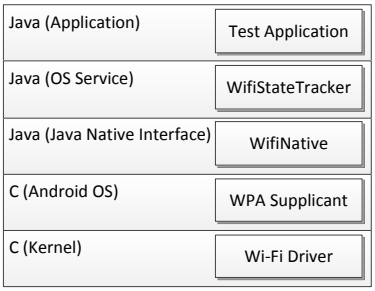


Figure 6: Cross-layer implementation.

The DH scheme uses a cross-layer implementation as shown in Fig. 6. As a result, the AP Table can be passed from the upper layers, such as the application and the OS service, to the lower layers. The following briefly explains the implementation. The test application generates Constant Bit Rate (CBR) data for VoIP application and uses User Datagram Protocol (UDP). WifiStateTracker receives a value from the geomagnetic sensor as an event from the sensor manager, and uses this information together with the AP Table and the Mapping Table to determine candidate APs to scan. The Wi-Fi driver performs active scanning and then selects the appropriate AP to reconnect and carries out the handoff. In addition, it logs the time when a UDP packet delivered from the test application is transmitted or received through the wireless medium.

A prior study in [3] showed that an MS may transmit a packet at the UDP/IP layer during a handoff, which contributes to increased handoff delay. To reduce handoff delay, the UDP/IP layer was modified to prevent it from delivering its packet to the MAC layer during a handoff. However, this does not mean packet will be lost. The application still generates a packet every 20 ms, but it is stored temporarily in a queue of the UDP/IP layer. After the handoff is completed, packets in the queue are delivered to the MAC layer and sent over the wireless medium.

B. Test Environment

The proposed DH scheme was implemented on smartphones and tested on the public WLAN environment deployed on the 1^{st} floor of Computer Science building at KAIST as shown in Fig. 7 (private APs are not shown). All the APs have the same SSID; therefore, only the link layer handoff is considered. In addition, Open System Authentication method is used, thus the time taken for the authentication step was minimized. At the beginning of each experiment, MS is associated with AP1 as shown in Fig. 7 and then it moves along the direction of the arrows and back to the position of AP1 at pedestrian speed. Accordingly, a total of 7$\sim$8 handoffs can occur in a single experiment, and several experiments were performed and the results are based on the two experiments with the best performance.

The performance evaluation is based on data transmission between a smartphone and a PC. However, during data

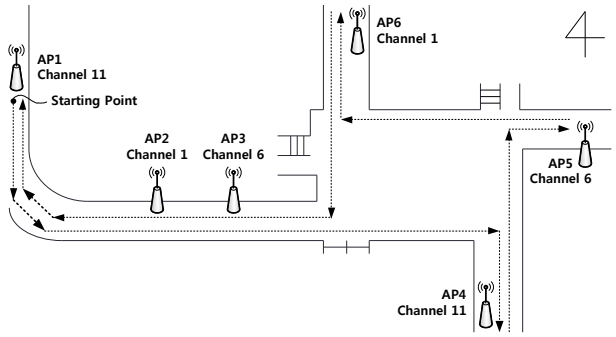


Figure 7: Public WLAN environment of the 1^{st} floor of Computer Science building at KAIST.

Table III: Handoff-related parameters and their default values.

	CH	SS	DH
Scanning Method	Active	Selective	
Handoff Threshold	-70 dBm	-80 dBm	-70 dBm
MinChannelTime	6.5 ms		
MaxChannelTime	11 ms		
Early Termination	Off		On
Probe Request Trans.	Broadcasting		Unicasting
AP Connection Threshold	-75 dBm	-80 dBm	-75 dBm
Background Scanning	Off		
RSSI Filter Weight	0.5		

transmission from a PC to a smartphone, there is *bridging delay* caused by update of the MAC address of an MS in the Ethernet switch [3]. Due to this delay, application-level handoff delay cannot be properly analyzed. Therefore, only the data transmission from a smartphone to a PC was considered.

The size of the data packet used in the test application is 200 bytes including IP and UDP headers (Transfer rate: 50 packets/sec = 80 Kbps). The PC is located in a building next to Computer Science building and uses the Windows 7 operating system. Finally, the proposed DH scheme was compared against the Conventional Handoff (CH) scheme that performs active scanning of all channels and the Selective Scanning (SS) with the channel mask used in [3]. The *caching* mechanism in [3] was not used because it causes MSs to connect to incorrect APs in their opposite direction leading to frequent disruptions.

The handoff-related parameters and their default values are summarized in Table III. Since the manufacturer specifies the default values by considering various situations, these values are not necessarily optimized for handoff. Therefore, the default values in the driver source code were modified according to the values in Table III. In addition to the parameters mentioned in the previous sections, Table III includes several other parameters. 'Early Termination' is the function that stops scanning when MS receives probe response frames or beacon frames before *MaxChannelTime*. 'Background Scanning' performs passive scanning every 5 or 10 seconds based on the current RSSI value. This feature is turned off for this study to reduce power consumption.

Table IV: The average delays for the three schemes.

	CH in Dream	SS in Dream	DH in Dream	DH in Defy
Application-Level Handoff Delay (ms)	196.1	159.1	103.5	87.2
Link-Level Handoff Delay (ms)	153.1	110.7	58.5	42.3
Scanning Delay (ms)	108.7	55.4	11.3	10.5
Authentication Delay (ms)	4.8	7.6	5.3	2.8
Re-association Delay (ms)	5.8	7.7	5.8	3

C. Results

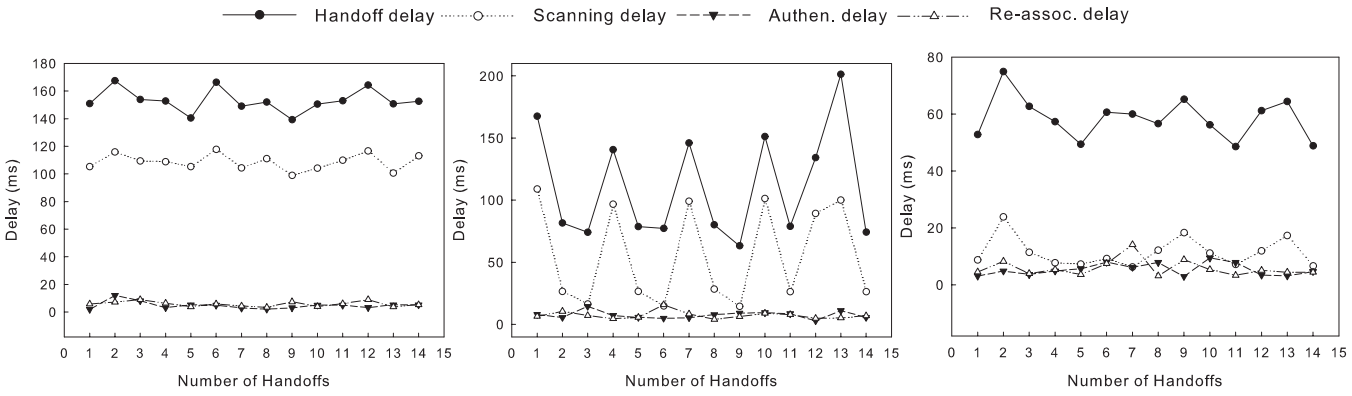
Figs. 8 (a)-(c) compare the link-level delays in HTC Dream. In the CH scheme, handoff delay is in the range of 140∼180 ms, and the scanning delay is in the range of 100∼120 ms, which represents a large percentage of the handoff delay. In the SS scheme, handoff and scanning delays fluctuate greatly because the channel mask is frequently inverted whenever Selective Scanning fails to find new APs. In addition, both the CH and SS schemes often connect to APs in the upper floor, which causes another handoff. In the DH scheme, handoff delay is in the range of 40∼80 ms, and the scanning delay is less than 20 ms.

In order to investigate the impact of handoff on applications such as VoIP, Fig. 9 shows application-level handoff delay, which is defined as the difference between the time when a UDP packet from the test application is sent before a handoff and the time when a UDP packet is sent after the handoff. In the DH scheme, application-level handoff delay is in the range of 80∼120 ms, which is about 100 ms lower than the delay in the CH scheme.

Table IV summarizes all the delays for the three schemes for HTC Dream. The DH scheme achieves 62% lower link-level handoff delay and 47% lower application-level handoff delay than the CH scheme. In addition, the DH scheme achieves 47% lower link-level handoff delay and 35% lower application-level handoff delay than the SS scheme. Table IV also shows the performance of the proposed DH scheme for Motorola Defy, which achieves even lower delay than HTC Dream due to a faster processor and a larger memory.

V. Conclusion and Future Work

The Directional Handoff scheme proposed in this paper performs fast handoff by scanning only a limited number of target APs. In order to reduce the number of APs to scan, the proposed scheme predicts the direction of an MS using a geomagnetic sensor embedded in the mobile device. The proposed scheme retrieves the target APs to scan from the predefined AP Table and performs active scanning on the target channels. Our experiments with real smartphones



(a) Conventional handoff. (b) Selective scanning with channel mask. (c) Directional handoff.

Figure 8: Analysis of handoff delay in HTC Dream.

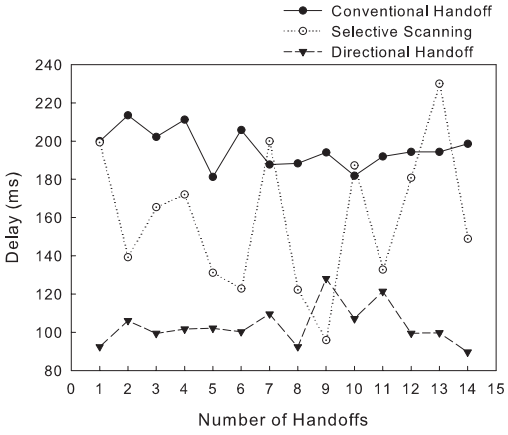


Figure 9: Application-level handoff delay in HTC Dream.

show that link-level and application-level handoff delays are reduced. Furthermore, the proposed scheme reduces application-level handoff delay to below 150 ms [8], which allows real-time multimedia services, such as VoIP, to be supported. The proposed scheme can be applied to IEEE 802.11a/b/g/n standards and is compatible with the existing WLANs as it does not require modification to the existing APs.

For future work, we plan to investigate automatic AP Table construction. Furthermore, we also plan to study fast handoff in an outdoor WLAN environment.

REFERENCES

[1] E. Kaplan, and C. Hegarty "Understanding GPS: Principles and Applications," Artech House, Boston, 2nd edition, 2006.

[2] P. Bahl, and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM, 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2000, pp. 775-784.

[3] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac, October 2004, pp. 19-26.

[4] W. Wanalertlak, B. Lee, C. Yu, M. Kim, S. Park, and W. Kim, "Behavior-based mobility prediction for seamless handoffs in mobile wireless networks," *Wireless Networks*, 3(17), April 2011, pp. 645-658.

[5] I. Ramani, and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," in *Proceedings of IEEE INFOCOM, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2005, pp. 675-684.

[6] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Computer Communication, Review*, 33(2) April 2003, pp. 93-102.

[7] IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std., June 2007.

[8] International Telecommunication Union. End-user multimedia QoS categories. ITU-T G.1010, 2001.

[9] I. Purushothaman, and S. Roy, "FastScan: a handoff scheme for voice over IEEE 802.11 WLANs," *Wireless Networks*, 7(16), October 2010, pp. 2049-2063.

[10] W. Kim, M. Kim, K. Lee, C. Yu, and B. Lee, "Link layer assisted mobility support using SIP for real-time multimedia communications," in *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac, October 2004, pp. 127-129.

[11] "Asahi KASEI Electronic Compass – AK8973N/B/S, AK8975/B" http://www.asahi-kasei.co.jp/akm/en/product/compass.html.

[12] "Yamaha Geomagnetic Sensors," http://www.yamaha.co.jp/english/product/lsi/magnetic_sensor/.

[13] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys, 2009, pp. 179-192.

[14] "HTC Dream – Wikipedia," http://en.wikipedia.org/wiki/HTC_Dream/.

[15] "Motorola Defy – Wikipedia," http://en.wikipedia.org/wiki/Motorola_Defy/.

[16] "Android Open Source," http://source.android.com/.

[17] "Android Source Code," http://android.git.kernel.org/.