

Research Article

Adaptive Decision of Wireless Access Network for Higher User Satisfaction

Hyunsoon Kim , Mungyu Bae , Woonghee Lee , and Hwangnam Kim 

School of Electrical Engineering, Korea University, Seoul, Republic of Korea

Correspondence should be addressed to Hwangnam Kim; hncim@korea.ac.kr

Received 31 January 2018; Revised 30 May 2018; Accepted 21 June 2018; Published 10 July 2018

Academic Editor: Michael McGuire

Copyright © 2018 Hyunsoon Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Today's mobile devices mostly contain more than one wireless interface for Internet connection. Smartphones mostly have both LTE and Wi-Fi for data communication through Internet. Although the availability of Wi-Fi is incomparable to that of cellular network, its strength lies in the low cost and high data rate due to continuous PHY and MAC advancement of 802.11 protocol extensions. In this paper, we propose a device-centric system that performs cost-effective network connectivity to the mobile device by selectively activating Wi-Fi communication according to the device mobility and corresponding history of Wi-Fi usage. By analyzing the device mobility using attached sensors, the system selects appropriate Wi-Fi that has been often used in that mobility state. The system was implemented in actual mobile devices that were used to several experiments we designed to prove high performance of the system.

1. Introduction

Wide spread deployment of free access points, usually Wi-Fi, provides both easier access to Internet and high-speed communication to mobile device users. Traffic offloading from overloaded cellular network to Wi-Fi gave significant advantage to mobile carriers that had hard time to control unmanageable traffic. According to Ericsson Mobility Report [1], heavy video traffic users dominantly use Wi-Fi, about 10 times more than cellular in UK and 3 times more in US in terms of data consumption. Although the data rate of LTE outperforms that of Wi-Fi for most of the cases, this tendency shows that most of users prefer low cost wireless communications when their data consumption exceeds their monthly data plan. Therefore, we can expect that the demand of Wi-Fi communications will not diminish soon unless the cost of LTE or other cellular network declines significantly. In order to maintain this spontaneous Wi-Fi offloading lead by service users, mobile carriers installed thousands of public APs in urban areas with high population. Mobile service users affiliated to those carriers can easily access to Internet through public APs.

However, the management of different wireless band network service is quite challenging in perspective of mobile

carriers, especially when ISM (Industrial, Scientific, and Medical purpose) band [2] is included in the list of available bandwidths to handle. Unlike cellular network that is owned by a single mobile carrier in one region, there are many candidates of using an ISM channel in the same area since the bandwidth is available to anyone. Moreover, 2.4GHz band can be used in various link layer protocols such as Wi-Fi, Bluetooth, and 802.15.4 based sensor protocols. The extensive use of ISM band causes severe interference among wireless devices using the same channel, especially in urban areas with hundreds of access points within one hop range. From recent experiments we have done at tens of indoor environments near to Korea University, Seoul, Korea, we found that there were 100 to 500 Wi-Fi access points available within one hop range with 2.4 GHz bandwidth only. Although they are not all accessible due to WPA based authentication, there are still numbers of APs available by mobile carriers or local proprietors, such as cafés and restaurants that provide free Wi-Fi access for visitors.

Although hardware support for multiple wireless interfaces is given, current Wi-Fi usage and connection behavior are not smart enough to fully utilize the hardware appropriately. Most of mobile devices scan surrounding access

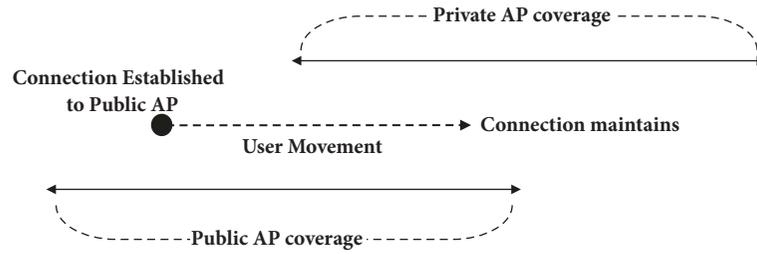


FIGURE 1: Example: a wrong behavior of Wi-Fi connection.

points and connect to the largest signal radiating AP that has been connected before. Once the AP is authenticated, the connection to Wi-Fi is automatic without user noticing the behavior, which can give both convenience and risk all together. There are usually more than one access point available, and the device user usually prefers one that provides stable performance. For instance, even if there is a public Wi-Fi AP provided by a mobile carrier, the user prefers the private AP installed in the room since the private AP mostly shows better performance. Therefore, the mobile device mostly connects to the private AP if the user is located in the room. However, that is not guaranteed to happen always. Figure 1 shows the worst case of using Wi-Fi; when the user comes back to his residence, the mobile device connects to a public AP since the AP came into the device's communication range, whereas the private AP is still far away to be used. However, the device maintains its connection even if the user arrives at the room because the device is still in the range of the public AP which is now transmitting very low quality signal.

Aforementioned exceptional case can be handled by link quality monitoring of currently connected AP, but monitoring link quality is quite difficult in reality. Link state is highly dynamic, which requires continuous measurement of status, and such monitoring requires both high computation and energy cost to the mobile device. Currently, most of users manually disconnect the bad AP and perform reconnection to the private AP. This manual reconfiguration may not be a significant problem to some people for now, but this is not true for the world of IoT [3] where billions of wireless equipment are installed and moving around for collecting data from the city. These communicating objects, ranging from small sensors to laptops, will be carried by a single user and will require always-on connectivity with low cost. IoT will provide people with network services through multiple wireless interfaces layered by cost and service quality. Along with mobile devices that will have multiple wireless interfaces each, the most important role of wireless devices would be the sufficient Internet connectivity with the lowest budget.

In order to achieve the goal, this paper proposes the system SIMLESS, Seamless Internet connectivity Management with Low Expense using Sensor based mobility State estimation. SIMLESS gathers up the Wi-Fi history for each user mobility state in order to prioritize Wi-Fi AP that is expected to provide the best Internet connectivity out of other Wi-Fi APs. We assume that the history of Wi-Fi usage best suits the user satisfaction because the list of APs within the

device history are the ones that are manually connected by the user at least once for one's benefit. Moreover, the most frequently used APs out of them are the ones favored by the user. Instead of evaluating the performance of different APs, we decided to use the history because we believe each person has different Wi-Fi availability and preference. Different mobile carriers have different Wi-Fi coverage map, and some people might avoid public APs due to security and instead use LTE. On the other hand, some people might have a private portable Wi-Fi that performs much better than public APs shared with lots of other users.

Focusing on better wireless connectivity to mobile devices targeting low cost, we propose SIMLESS that uses an accelerometer and a barometer to estimate the mobility state of the device and the user. There are various reasons of using aforementioned sensors instead of GPS or other localization systems. Firstly, these sensors are available anywhere unlike other localization systems that require external resources such as satellites and radio signals. Also, the sensors are available for most of devices whereas components like an infrared sensor are not. Secondly, monitoring IMU sensor or barometer data to estimate mobility state does not cost a lot in terms of energy consumption compared to GPS. Lastly, even if the user location is the same, the preference of Wi-Fi AP can differ by what the user is doing. For instance, the user might be located near to a bus station for waiting someone, or the user already took the bus and the bus is waiting at the station to bring more passengers to board. On the former occasion, the user might prefer a public Wi-Fi that is located at the bus station, whereas the user in the latter case will want a movable Wi-Fi installed in the waiting bus for the later case. Focusing on the fact that the user preference of using Wi-Fi depends on his/her mobility state, not just speed, SIMLESS uses inertial sensors and a machine learning method to estimate user mobility without external resources for aid. Therefore, it does not require any external aid but tracks the device regardless of environment such as indoor and outdoor. Contributions of the system SIMLESS are listed as follows.

- (i) SIMLESS is a low power system that does not require extra power consumption for link state estimation and GPS based localization.
- (ii) SIMLESS uses a mobility state estimation method which is based on low power sensors and machine learning algorithm using Bayesian probabilistic model and decision tree classification.

- (iii) From a set of real-life experiments, it is discovered that Wi-Fi usage of a user can greatly differ by the mobility state of the user. The variance on Wi-Fi preference is shown to become greater as portable access points are carried by people.
- (iv) Through experimental results from various realistic evaluations using SIMLESS, automatic connection to qualified Wi-Fi station is proved to behave well, which increased the user satisfaction in great degree.

This paper is organized as follows. Section 2 lists a number of researches focused on vertical handover control, sensor based mobility tracking systems, and Wi-Fi selection mechanisms. Section 3 describes an overview of SIMLESS, which is composed of three major modules: Mobility Trace Module (MTM), History Trace Module (HTM), and Connectivity Management Module (CMM). Sections 4, 5, and 6 describe details of each module and the main algorithm designed for proper Wi-Fi connection according to user mobility state which works in continuous manner. Section 7 describes overall evaluation process of proposed solution in terms of throughput, cost-effectiveness, and power consumption of the mobile device. Moreover, the section analyzes results of applying the solution to variable models of smartphone in order to prove both applicability and scalability of SIMLESS. Finally, Section 8 concludes the paper. The overall system was implemented in various Android smartphones, such as Galaxy S5, Galaxy S3, Galaxy Note 2, Galaxy Nexus, and Nexus 5.

2. Related Work

Vertical handover between cellular and Wi-Fi network is already researched and commercialized by numerous prior work. The practical use of vertical handover rapidly grew as mobile devices like smartphones were widely used. References [4–7] provided various intelligence-based solutions for proper vertical handover between macrocells and supplement access points such as pico/femtocells and Wi-Fi APs. They used various parameters related to link quality of each access point, such as signal strength, bit error rate, bandwidth size, and jitter level. Some of them used device-centric method to reflect the status of user or device condition, but most of them used network-centric method that can give more accurate feedback regarding link quality of access points. Although the tendency can be a problem in terms of scalability issue (in case Wi-Fi access points and macrocells cannot cooperate each other due to different service provider), such solution is practical since most of mobile carriers have their own Wi-Fi access points installed in many regions to make those cowork with macrocells that might be overloaded by unlimited traffic. Reference [8] proves that such tendency still exists. Reference [8] uses SDN controller to centralize AP Load information and lead the mobile device to the most appropriate access network.

However, aforementioned prior work take no care on user preference that differs by mobility state. Even if the location is the same, the user preferred AP for each mobility state may differ a lot, as shown in Figure 1. Moreover,

movable APs such as portable access points [9–12] are used in many places with various purposes, making consideration of user preference even more important. Portable access points usually have Wi-Fi for downlink traffic and other bandwidth with larger coverage for uplink traffic, which makes the attached mobile devices access Internet without frequently changing IP address while moving fast. Some people carry the portable AP which is mostly cheaper than regular LTE service as a data plan. Moreover, most of city buses in South Korea carry such dual link APs to provide wireless Internet service to passengers. In this case, the user in fast move would prefer those rather than static public APs.

Ad hoc networks [13, 14] are beginning to be deployed in various areas such as poor wireless support due to man-made or natural disasters in developing countries [15] and opportunistic network composed of multiple cars in roadways [16, 17]. Even flying drones can be used as an ad hoc based movable AP as we tested in previous work [18]. For better connections to aforementioned movable access points as well as static APs, we believe considering mobility state of the user can enhance both low cost Internet connectivity and high wireless service quality. We expect these movable APs will become general in the future of IoT and smart cities that have quite enhanced network availability. Considering such tendency, we believe it is necessary to find the user mobility state and match this with Wi-Fi usage characteristic to derive the list of APs that best suits to the current user situation. Simple speed or velocity information without deep understanding of relationship between user mobility and Wi-Fi usage pattern cannot satisfy user on wireless service quality.

For proper Internet connectivity, we designed mobility based user preferred Wi-Fi connection scheme. However, unlike traditional navigation systems, we created our own mobility state estimation system for several reasons. Previous work like [19–21] require external signal sources such as GSM and GPS, which means the estimation can be affected by out-sources greatly. Errors due to personal motility characteristics can be reduced by machine learning of the user behavior; it is difficult to correct errors due to device hardware by programming. It is true that there are several solutions to distinguish mobility state based on only in-built sensors such as an accelerometer [22, 23]. However, those methods do not cover exceptional cases of acceleration outliers due to abnormal device movement unrelated to user mobility. High acceleration does not always mean high activity of the user regarding mobility. Indeed, the highest acceleration change occurs when the user actively moves body, rather than driving a car with much greater speed. We found out that the level of acceleration is linearly related to the activity degree of the user within the same mobility state rather than to the list of mobility states in ascending speed order. To accomplish accurate mobility state estimation, something else is required other than an accelerometer, and we found that a barometer mostly attached in today's smartphones is the key parameter for the solution.

Prior to our work of having user movement as a key parameter of connecting short range network, there are several research works done for better Wi-Fi selection which

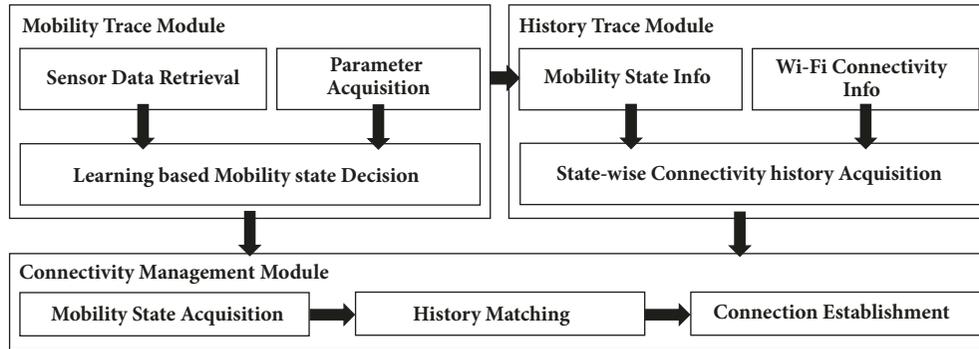


FIGURE 2: SIMLESS overview: components and operation flows.

can provide effective Wi-Fi offloading from overloaded cell towers. Reference [24] proposed a fast AP selection mechanism based on beacon reception delay in perspective of a mobile device. The authors believed that the beacon would stack on AP's buffer and more data on buffer would eventually result in sending beacon in longer delay. However, current IEEE 802.11 protocol specifies that control frames including a beacon frame skip the process of buffering in order to send control messages within appointed time. Therefore, Wi-Fi selection based on beacon reception delay cannot be accomplished currently. Instead of evaluating link quality of APs before connection, [25] proposed a system that connects every accessible AP to evaluate the link quality using short test on data frames. The system can provide high resolution of link quality map composed of surrounding accessible Wi-Fi APs, but the association attempt to every AP results in high delay which is usually not acceptable to users that need to access Internet within short period of time. We also had attempted to solve this issue, trade-off between quality evaluation accuracy and delay, and proposed a link-aware AP selection mechanism [26] which uses the beacon collision rate and signal strength variance among consecutive beacon receptions from surrounding APs. We used Galaxy Nexus to implement such solution into a real device, and experiments using the device showed extensive performance on selecting the best AP out of others.

There are also multiples of prior work [27–29] that proposes learning based network selection. Reference [27] proposes Markov chain process based network selection. The work uses several network statistics to derive optimal solution; however, these statistics require the solution to be applied at network side, not on the user device itself. For instance, blocking probability of each access network cannot be collected by the device within a short period of time. Reference [28] is another learning based work that uses a modified version of genetic algorithm. This work still requires multiple network statistics that are difficult to be collected by device users. Reference [29] is also a work that suggests a learning based mechanism but applies network state information in order to fit user's status in real time. However, the idea of changing access network while transmitting brings other challenges to be solved in reality. In order to change access network according to the network state information

and traffic type information, the traffic type should be defined in real time and connection overhead should be minimal. If there are multiple network providers a device can use, handover between access networks of different providers will not be easy.

Although aforementioned solutions can be used to connect a proper Wi-Fi AP that seems to have the least interference and the highest wireless speed, all those works are unaware of the position changes of the user. With high-speed movement of a user, a highly qualified Wi-Fi AP is useless unless it is carried by the user. To avoid awkward selection of an AP without human intervention, proper mobility recognition is required. Therefore, we propose SIMLESS that derives such relationship from the user mobility state and corresponding Wi-Fi history, and properly connects to an AP which is most likely wanted by the user. SIMLESS can be extended to Wi-Fi selection criteria with link quality evaluation methods described above.

3. SIMLESS Overview

SIMLESS aims to establish stable connection to low cost access points such as Wi-Fi without explicit link quality evaluation. The connection should be stable in terms of data rate and connectivity time, which means frequent handover is not preferred unless necessary. To achieve the goal, SIMLESS is composed of three modules as shown in Figure 2: Mobility Trace Module (MTM), History Trace Module (HTM), and Connectivity Management Module (CMM).

MTM traces the mobility state of the user, using series of sensor variables that are retrieved from a three-axis accelerometer and a pressure sensor. The module contains parameter acquisition phase which retrieves set of parameters derived from machine learning. Machine learning is based on the mixture of Bayesian probabilistic model and decision tree. Using machine learning parameters and series of sensor data, the module decides the current mobility state of the user and provides the result to other two modules. HTM traces the connectivity state of the device, to discover whether or not the device is connected to low cost network and what access point is in use. Each mobility state contains such the connectivity statistics and will show the tendency of low cost network

usage that differs by a mobility state. Based on connectivity history of each mobility state and current mobility state derived by aforementioned two modules, CMM orders the device to connect the most preferred AP on the current mobility state. Since MTM traces the mobility state based on estimation, frequent state transition can occur without assuring whether the transition is really happening or not. CMM handles the error case based on a stability management method authors designed. The details of each module are described later in this paper.

Based on aforementioned modules and their interactions, SIMLESS was implemented upon Android mobile platform and installed on various smartphones compatible to the platform. Most of analysis and evaluations were done using Nexus 5, but applicability of SIMLESS on other smartphones were proved to be fine through an empirical survey on multiple users. Each user is required to collect sensor data and network usage history for prelearning of the user behavior, which could be done by using mobile application implemented by authors. The experimental smartphones contained both accelerometers and pressure sensors for proper mobility estimation and supported both Wi-Fi and LTE communication. Since any LTE service should be given by a certain mobile carrier, each smartphone could also use Wi-Fi access points supported by the mobile carrier. Moreover, smartphone users also possessed portable Wi-Fi devices that supported Wi-Fi service by the user carrying the device. Based on aforementioned characteristics of the users and the smartphones, later sections describe the functionality of each module step by step along with the performance evaluation of each module.

4. Mobility State Estimation Using IMU

Inertial Measurement Unit (IMU) is installed in most of mobile devices nowadays, as a form of microelectromechanical system (MEMS) which is designed to be assembled into a small device such as a smartphone. IMU refers to a set of inertial sensors such as an accelerometer, gyroscope, and a digital compass. Many smartphones also contain barometers that are used to estimate altitude and position. To accomplish stable mobility state estimation performance using above sensors, it is necessary to analyze the characteristic of each mobility state in terms of sensor output. Therefore, we used Nexus 5 which contains all the sensors needed for mobility state estimation. Figure 3 shows the acceleration difference among mobility states that are defined as *Walking*, *Stationary*, and *Driving*. Data logging for each mobility state was held more than 30 minutes independently, without interruption of other mobility states within. The graph shows just some part of the result to visualize fluctuations of acceleration values clearly. To measure the change of acceleration, the acceleration value is defined as an Euclidean distance between two consecutive data points and differs by the time period of 0.5 second. The resulted Euclidean distance is the square root of a sum of three-axis differences, which can be casted as follows: $E_{distance} = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$. As shown, the Euclidean distance of *Walking* state is much

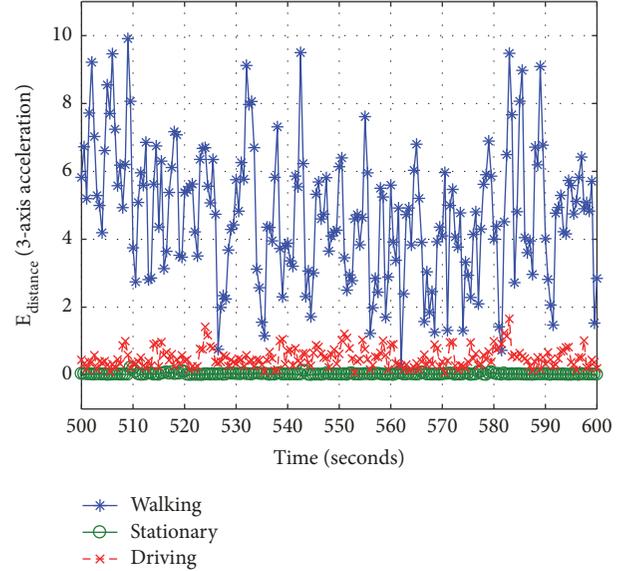


FIGURE 3: Acceleration patterns for three mobility states: Walking, Stationary, and Driving.

higher than those of other two states, whereas *Stationary* and *Driving* states contain similar data points. Although *Driving* state shows much more fluctuation than *Stationary* which constantly shows 0 acceleration value, there are too much data points overlapping each other, making classification between two states difficult by solely using the accelerometer of the device.

Therefore, a barometer is used additionally to create distinct classification between *Stationary* and *Driving* states. The previous work called DUTA [30] indeed used orientation variable to classify these two states. However, we found out that there are frequent cases of orientation fluctuation even if the user is not in *Driving* state, along with the cases of no orientation fluctuation when the user is actually in *Driving* state. For instance, a usual high-speed railway rarely has curve section that provides enough orientation change on a mobile device, making classification inaccurate for long time. Therefore, a barometer was used instead, which makes *Driving* state have distinct difference compared to other two states as shown in Figure 4. The absolute pressure value in terms of *hPa* is almost constant for *Stationary* and *Walking* states because the location has not changed a lot for 100 seconds. Indeed, the pressure changes during *Walking* state very slowly, but the change is incomparable to the change of *Driving* state. There are various reasons of air barometric pressure change such as temperature, weather, time, and location. The change of location due to fast movement sufficiently changes pressure because temperature, altitude, and weather differ by regions.

Based on simple analysis derived by aforementioned statistics, a 3-axis accelerometer and a barometer equipped in a smartphone called Nexus 5 are used for the mobility classification. Although the mobility states can be classified into more several states, we decided to use only three states because they are the most usual states that can possibly

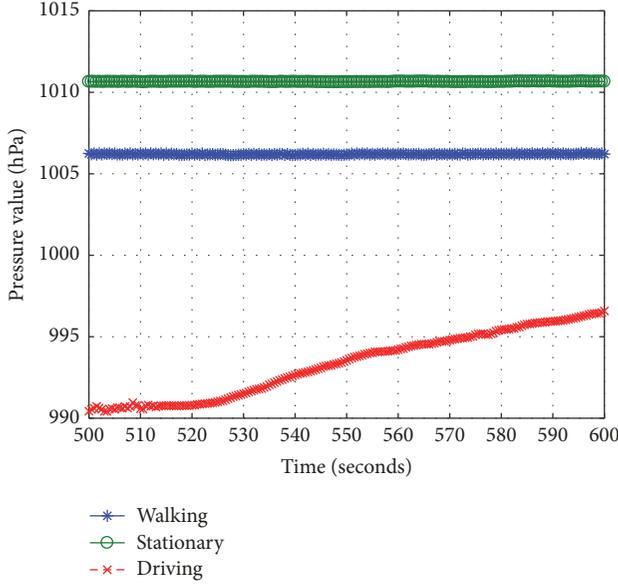


FIGURE 4: Pressure patterns for three mobility states: Walking, Stationary, and Driving.

include other substantial mobility states, such as move upstairs as *Walking* and taking a bus as *Driving*. Indeed, classifying only three states quite improved the network connectivity of the mobile device, which will be shown later in this paper.

Based on two sensor variables, Bayes' rule is applied to derive acceleration and pressure tendency of each mobility state in the following equation:

$$P(S | x) = \frac{P(S) p(x | S)}{p(x)} \quad (1)$$

where $P(S | x)$ indicates the probability of becoming mobility state S when input is x . Input x can be $E_{distance}$ for either an accelerometer output or a barometer output. $P(S)$ refers to the probability of mobility state S out of all the sample data, which can be alternatively described as a prior probability. $p(x|S)$ is the class likelihood of having input x . $p(x)$ is the probability of having input x for any mobility states in this system. Except for $P(S | x)$, all the variables can be derived from a trained data that is composed of device recorded mobility states and corresponding sensor outputs. The recorded mobility state is the ground truth state variable, S , and each sensor output is expressed as relative variable rather than absolute variable, to be used as x on (1). x is an Euclidean distance variable between two sensor values with different time domain, but in our case, it is not necessarily consecutive two values. We defined the sensor output, or classification input x as follows:

$$\begin{aligned} x &= \max \left(E_{distance} (s_i, s_j) \mid i > 0, j \right. \\ &= \left. [i - T_f, i), j \bmod t = 0 \right), \end{aligned} \quad (2)$$

where s_i refers to the current sensor output and s_j refers to prior sensor output within time range $[i - T_f, i)$ where

T_f refers to predefined time period. Instead of having time j simply adjacent to time i , T_f is defined to provide some flexibility for better judgment of classification. For instance, pressure value does not abruptly change within time of 0.5 seconds even if the user is driving quite fast. Since it takes some time to have distinguishable change of pressure value to decide mobility state as *Driving* state, T_f is set to 30 seconds for pressure value and 3 seconds for acceleration value. These values are also used for later evaluations throughout this paper.

With given definition of $E_{distance}$ for each sensor output, $P(S | x)$ is derived for a training set that has about 2-hour long record containing all three mobility states and corresponding sensor results. The user brought Nexus 5 and played an application we made in order to collect data and manually set ongoing mobility state that best describes the user's movement. The user frequently changed his mobility state as accurately as possible whenever his mobility state had changed. Figure 5 shows the resulted probability values for each mobility state with different sensor input ranges. Since sensor results are float variables, we made 20 sensor data ranges for each sensor, to make the probabilistic model easily applied to the mobility state estimation. As shown in the graph, the probabilistic model seems unusable especially when the sensor output is high. Traditional machine learning based on Bayesian probabilistic model mostly decides a class that has the highest $P(S | x)$ for given input x . However, applying the method directly can severely downgrade the performance of mobility estimation since there are many sensor data ranges that are not dominated by a single mobility state, causing the performance degradation of classification. In case of having acceleration distance value between 15 and 20, the probability of three states are similar one another, as shown in Figure 5. In reality, high acceleration usually means high activity of the user, and the probability of *Walking* state should dominate the other two in these data ranges. The reason of this classification difficulty is that there are some exceptional cases caused by the user touching device without any intention to move. For instance, the user can accidentally drop the phone, or the user may toss the phone to a nearby person. Exceptional cases also happen to the pressure data similarly. Therefore, the proposed mobility estimation applies the concept of decision tree to aforementioned Bayesian probabilistic model, to cut off the exceptional cases that lower the estimation accuracy.

Basic decision tree based classification requires some threshold variable that classifies two or more classes (or states). The decision tree structure of proposed mobility state estimation is designed as shown in Figure 6. A threshold variable is given for each decision phase, and those variables can be defined in various ways using machine learning. In this paper, Bayesian probability based classification method is used to decide these thresholds. As shown by previous observation on mobility states and sensor outputs, it is easy to find that there is a turning point of dominating probability. In Figure 5(a), $E_{distance} = 3.75$ is such a turning point, where the Bayesian probability of *Walking* state overcomes that of other two states. According to the basic rule of Bayesian probability based classifier, the mobility state at each sensor

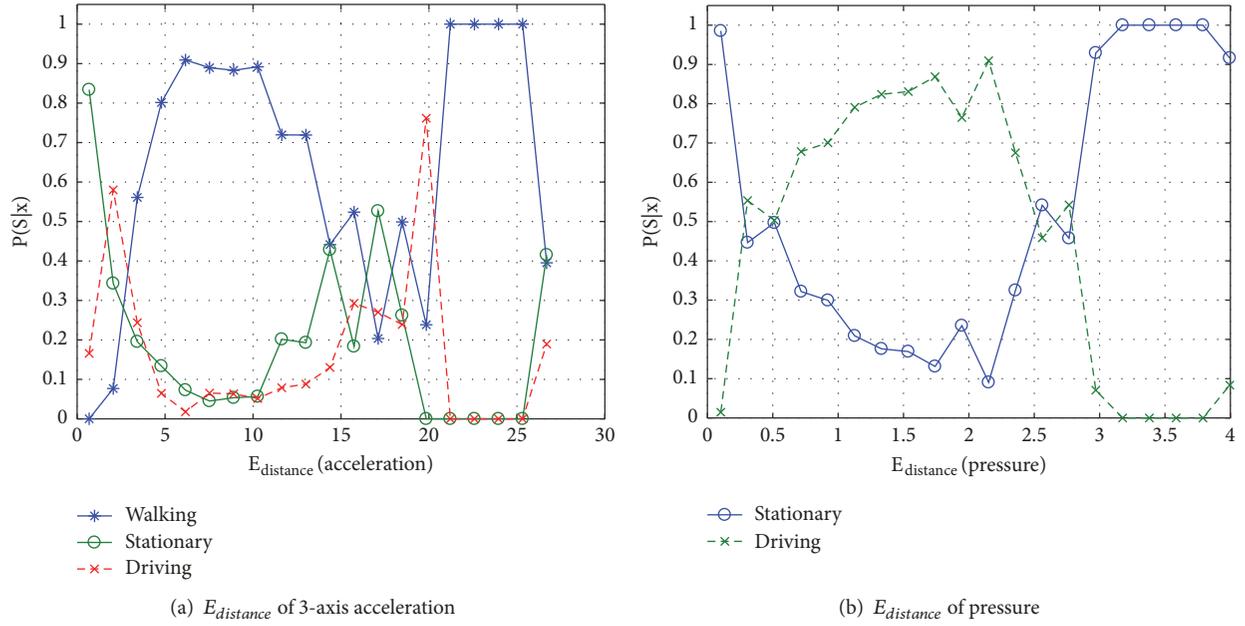


FIGURE 5: Probability distribution of mobility states based on sensor distance input: Bayes' rule results.

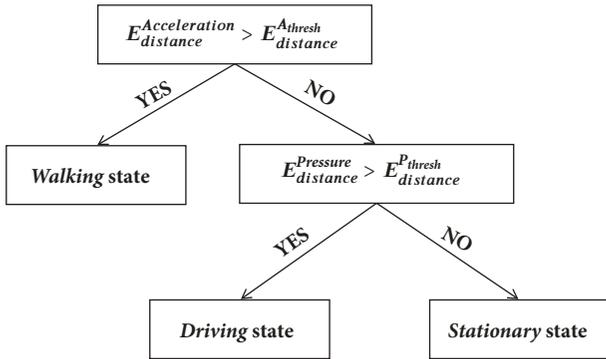


FIGURE 6: Mobility state estimation based on decision tree.

data range should be the one with the maximum probability. However, the proposed mobility state estimation searches aforementioned turning point and forces the rest of higher sensing data range to have the same mobility state as that of the turning point. Therefore, the threshold of acceleration becomes 3.75, and the threshold of pressure becomes 0.33 based on the tendency in Figure 5. Given the threshold by Bayesian probabilistic values from a set of training data, the mobility state estimation mechanism runs the decision tree to finalize the current mobility state out of three states.

In addition to the aforementioned decision tree, we added several simple logics to the estimation mechanism to raise precision. Firstly, we removed mobility state transition occurring by short acceleration bump, which can be caused by various external factors such as a jumping vehicle due to a speed bump, a handheld smartphone, and instant leg shaking. To remove such exceptional cases, the mechanism prevents transition to *Walking* from other states when the

high acceleration appears for short period of time. Secondly, we removed a direct transition from *Driving* to *Stationary*. This is because there is no way a person moves out of a vehicle without walking or other behaviors to go out of a car. There is always an intermittent mobility state in between two mobility states which means direct transition between *Driving* and *Stationary* is conceptually impossible. However, the mechanism allows the transition from *Stationary* to *Driving* because the mechanism needs some time to judge *Driving* state using the shift of the pressure value. When a person takes a transportation vehicle, the mobility state will change from *Walking* to *Stationary* since the vehicle stays until all the passengers get on. Even if the vehicle starts moving right away, the change of pressure value takes sometime, which makes the mobility state estimation misidentify *Driving* state as *Stationary*.

By using the proposed mobility state estimation based on decision tree with aforementioned two exception handling models, 4 datasets are derived from Nexus 5 to perform cross-validation among different datasets. Each training set is held for more than 2 hours but less than 7 hours and contains all three mobility states. Figure 7 shows the result of the mobility state estimation compared to simple decision tree based approach. As explained before, the decision tree is constructed by automatically generated threshold values, where the values are the turning points of Bayesian probabilities among mobility states. The white accuracy bar in Figure 7 is the result of the decision tree, whereas the higher black bars are the results of mobility state estimation mechanism that has included additional exception handling models. By handling exceptional cases, the performance of estimation mechanism improved up to 9%, resulting in 93% accuracy in average, whereas pure decision tree showed an average of 89% accuracy. Out of 4 datasets, the one with the highest

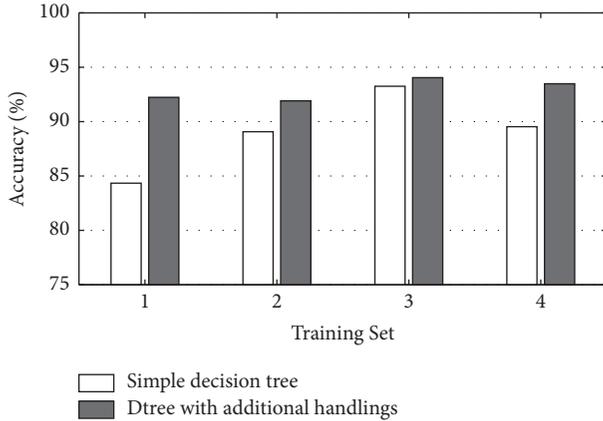


FIGURE 7: Performance analysis of the mobility state estimation.

accuracy (training set 3) was used for later evaluations in order to perform mobility state estimation for automatic Wi-Fi connection handling without any training step. Output thresholds for both acceleration and pressure values were built into Nexus 5 for fast classification.

5. Wi-Fi History Analysis for Each Mobility State

While MTM acquires the device mobility state in real time, HTM collects Wi-Fi usage information corresponding to the estimated mobility state. Although we specified the low cost access point as a Wi-Fi AP, this can be altered to any other low cost communications that can be used in near future. Assuming Wi-Fi is the only low cost access point we can use for now, the purpose of tracing Wi-Fi history is to discover the user preferred Wi-Fi for each mobility state. The user preference indeed possesses many meanings, since there are many aspects of what user prefers some APs than others. For instance, the user might want more secure access point even if it provides less speed than other APs. Since this paper focuses on cost-effectiveness and majority of performance analysis given in this paper compares the quantitative measure of cost-effectiveness, we assume the more user preferred AP hereby means better speed and less cost. By discovering the list of user preferred APs for each mobility state, the mobile device can attempt the most preferred AP by priority and continue the attempt to next preferred AP until the device succeeds in connection. To make these trials applicable, it is necessary to have another training phase that collects sufficient amount of Wi-Fi history, which includes the time of user pain from manual Wi-Fi connection due to device malfunction.

HTM and MTM were designed and implemented into Nexus 5, which contains both an accelerometer and a barometer and supports up to 802.11ac Wi-Fi specification. An experimenter carried Nexus 5 with aforementioned 2 modules and used the phone for 24 hours whenever he wanted to. USIM card is inserted to Nexus 5 in order to make the experimenter use Nexus 5 as a cellphone without feeling a

sense of incompatibility. The experimenter indeed carried the phone anywhere he went, such as his work place, his home, his car, and other public places he visited. At the same time, the device recorded both Wi-Fi history and corresponding mobility history in real time, using constant parameters such as T_f and machine learned threshold parameters for proper mobility state estimation.

Assuming that mobility state estimation is 100% precise, Figure 8 shows the Wi-Fi history result of each mobility state in terms of connection time. The history shows that there are several access points that are frequently used, whereas there might be hundreds of access points of which the device has scanned before. In each mobility state, the mobile device has used 6 to 7 access points normally and dominantly used 1 to 2 access points out of them. Moreover, the connection tendency is different for each mobility state. When the device is in *Stationary* state, the device usually uses Hans5G or wine515_hyun. This is because Hans5G is an AP installed on the experimenter's house, and wine515_hyun is an AP installed on his research office. The third frequently connected AP of *Stationary* state is No Wi-Fi, which defines the device with no Wi-Fi connection. In this case, the device uses LTE alternatively which is not preferred for users with low price data plan. However, the rate of LTE usage for *Stationary* state is much less than that of *Walking* state, since the user mostly stays at the research or home where a stable AP is provided almost all the time. About 40% of *Walking* state experiences the device with no Wi-Fi connection, and this is because most of Wi-Fi access points are static and have very short coverage range, about 20 to 30 meters in reality, which results in too frequent Wi-Fi connection attempt while walking. Although a portable AP named Gustns is frequently used to prevent frequent Wi-Fi handover and reduce LTE connection, the experimenter often does not carry the portable AP. We found that this is because the experimenter carries the portable AP only when he goes to some place that is far away from his current location. Therefore, the device in *Driving* state dominantly uses Gustns, since the experimenter always carries the portable AP whenever he takes a train or drives a car.

Another interesting observation from Figure 8 is the rare use of public APs provided by an affiliated mobile carrier. o1lehWiFi and other versions of the AP are rarely used even though it is connectible to the experiment device without any cost. This is because these APs are the most crowded APs used by many subscribers, which results in serious degradation of Wi-Fi communication quality. Even though these APs are movable APs installed in public transportation vehicles such as buses and trains, they are not preferred by most of people due to too much congestion and packet dropping. However, a mobile device without any intelligence does not remove these APs from its candidate list, which results in accidental connection to a not preferred AP, causing low wireless communication performance. The motivation of proposing SIMLESS is to prevent such accident and provide autoconnection to low cost network for sustaining Internet connection as long as possible. Using the Wi-Fi history in terms of the device mobility state, SIMLESS can avoid the connection to less preferred APs and preemptively connect to

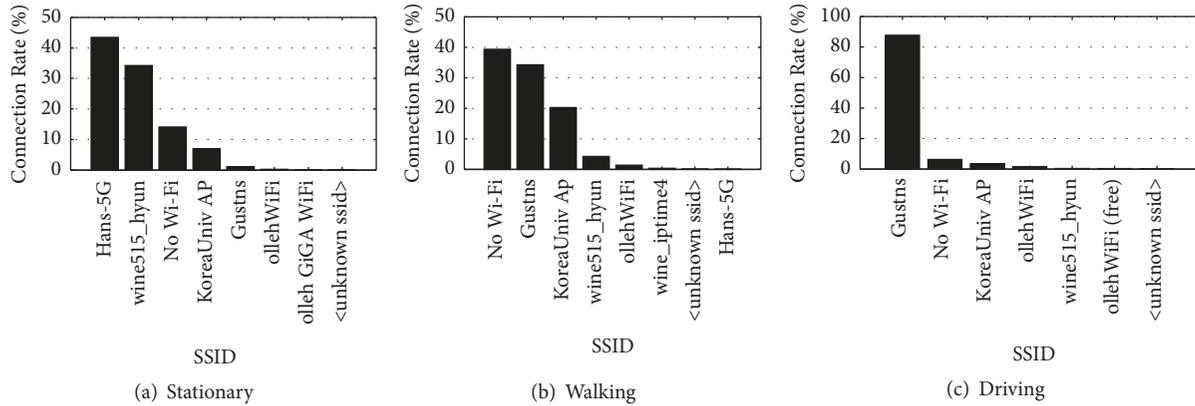


FIGURE 8: Probability distribution of mobility states based on sensor distance input: Bayes' rule results.

the most used one out of all. The performance enhancement of utilizing the history data is shown on Section 7.

6. Low Cost Wi-Fi Connection Mechanism for SIMLESS

Based on trace information retrieved from MTM and HTM, SIMLESS operates CMM to have appropriate connection to Wi-Fi AP which is the most preferred one out of other in-range APs. The basic idea of connection is series of connection trials in descending priority order. An AP that has been used the most possesses the highest priority level, which means CMM will try to establish connection to the AP first. If the connection fails, the device attempts connection to an AP with the next highest priority level and continues the iterative process until the device succeeds to connect to Wi-Fi or runs out of AP candidates. However, there are several precautions to prevent the algorithm malfunction which can degrade the device performance in terms of throughput and delay.

- (i) If a mobility state has high *No Wi-Fi* preference, the device should know when to use Wi-Fi or not. In case the device insists on Wi-Fi connection as long as possible, throughput degradation can be severe, causing the communication quality be graded in great degree.
- (ii) If a connection attempt occurs only when mobility state is changed, constant connection to a low priority AP connection would be a problem. The problem often occurs when a passenger takes a public transportation that has a set of access points that move with the vehicle.
- (iii) Frequent change of mobility state mostly creates unnecessary scan and connection trials of the device. The behavior downgrades both the service quality and energy efficiency of the device.

First problem occurs mostly when the user is in *Walking* state. Unlike other mobility states that have clear preference of Wi-Fi, a *Walking* person does not have chance to get a proper Wi-Fi connection unless a movable Wi-Fi AP follows the

person. When the person walks, the device cannot properly communicate through static Wi-Fi APs that will lose signal of the person's mobile device sooner or later. Even if the person possesses a portable device, the possibility of using a portable device decreases distinctively if the user mostly has short *Walking* behavior. The short *Walking* behavior includes instant walking around one's room, moving to kitchen, or moving to nearby market to buy some daily necessities. In these cases, most of people do not carry a portable access point, because there is barely no chance of using Internet. If SIMLESS ignores these *No Wi-Fi* cases, the device will experience both power waste and low wireless quality due to unnecessary connection attempt to low quality APs. Therefore, SIMLESS needs to find appropriate preparation to such the case.

The second problem often happens when the list of surrounding APs change due to device mobility. If a currently connected AP is a static AP, the phenomenon is not a problem since the mobile device will be disconnected as soon as it gets out of the AP's communication range. However, if the device is originally connected to a movable AP, such as an AP installed on the ceiling of a bus, the device will never change its current connection until the user changes one's mobility state. Periodical scanning is required to hold the chance of handover to more preferred AP. The last problem due to frequent mobility state change occurs because instant walking behavior of the user shakes the device and this happens frequently in everyday life. For instance, a person always enter *Walking* state when the person tries to take a vehicle. The duration of *Walking* state will not be so long if the person was waiting a vehicle at a bus or a subway station. Also, if the person has stayed at room and starts moving toward kitchen to bring some food, it will be unnecessary to make mobile device change its Wi-Fi AP even though the mobility state has changed. Estimation of instant mobility state shift is required to decide a connection attempt for the recently changed mobility state.

SIMLESS is designed in the way to consider aforementioned problems, resulting in stable connectivity performance in terms of throughput, low cost, and energy consumption of the device. The overall operation flow of CMM is

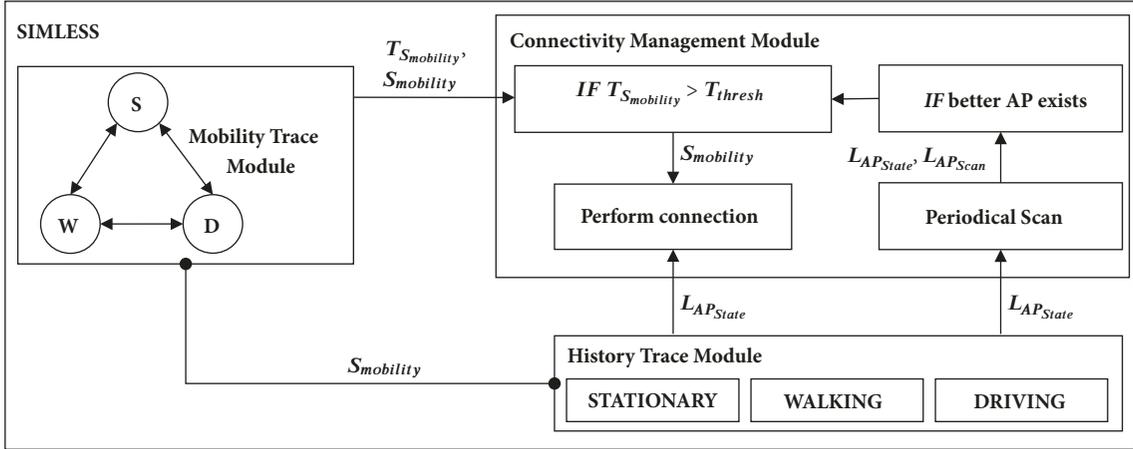


FIGURE 9: Overall behavior of CMM.

described in Figure 9. The device attempts Wi-Fi connection only when the current mobility state has sustained for more than time T_{thresh} or a more preferred AP appears to exist due to the device mobility. As soon as mobility state is changed, the module will perform scan to search through the most preferred AP in that mobility state. These conditions are given to solve aforementioned second and third problems. Before CMM operates, MTM tracks both current mobility state, $S_{mobility}$, and duration of the state, $T_{Smobility}$. On the other hand, HTM gathers Wi-Fi history information by collaborating with MTM. Given the current mobility state and the list of preferred APs of the mobility state, CMM performs Wi-Fi connection in descending preference order. Once connection process is done, the device maintains Wi-Fi connection unless the mobility state is changed or a better AP is found after numbers of periodical scans.

When the module makes a Wi-Fi connection, it iteratively attempts connection until it succeeds in Wi-Fi connection or it reaches *No Wi-Fi* as the next high preferred one. When it reaches *No Wi-Fi*, the module overseas the less preferred APs and finds if there is any major access point. In this paper, we define a major AP as an AP that has been distinctively used by the mobile device. The value of distinctiveness is defined as 10% of the state's overall Wi-Fi history. Therefore, CMM attempts to connect such the APs even if their preference lies below the case of *No Wi-Fi*. This behavior can solve the aforementioned second problem, which results in resolving continuous *No Wi-Fi* for *Walking* state due to given Wi-Fi history in Figure 8. As shown, *No Wi-Fi* has the highest preference in state *Walking* which can prevent the device attempt to try to establish connection to *Gustns* or *KoreaUniv* AP access points.

By resolving aforementioned problems, we found that SIMLESS operates appropriately in terms of balancing the network cost and performance. To prove well behavior of the system SIMLESS, we proceeded with several experimentations. For every experimentation, we used a real mobile device named Galaxy Nexus or Nexus 5 that are reference phones for Android mobile platform. Also, we used real-life access points installed on campus, subway stations, public

trains, and buses and several buildings that an experimenter visited quite often. In the aforementioned important parameters, scanning period is set to 2 minutes and T_{thresh} is set to 30 seconds. Details of experimentations are described in Section 7.

7. Performance Evaluation

To perform SIMLESS on a real mobile device, we used pregenerated classification parameters for mobility state estimation and Wi-Fi history collected by HTM. As aforementioned in Section 6, T_{thresh} is set to 30 seconds, whereas Wi-Fi scanning period is set to 2 minutes. Based on given parameters, the device operates SIMLESS by estimating mobility state in real time and manage Wi-Fi Connectivity according to the estimated mobility state. To show the merit of SIMLESS, this section lists 3 evaluations, each targeting different advantageous features of SIMLESS. First evaluation shows the QoE enhancement in terms of network throughput, and second evaluation shows cost-effectiveness of SIMLESS in terms of data amount per cost, where cost lies on price of using cellular data. Third evaluation measures the power overhead of SIMLESS, which is expected to be much more energy efficient than a system that enables Global Positioning System (GPS) module. In forth evaluation, we performed empirical study on multiple smartphone users in order to verify applicability of SIMLESS in reality. Finally, we compared the performance of SIMLESS with other device-centric solutions such as [25, 26] to emphasize strength of our solution among device-centric approaches.

7.1. QoE Enhancement in Terms of Network Throughput. To prove QoE enhancement achieved by SIMLESS, we found a place that makes a mobile device with traditional Wi-Fi capability malfunction most of the time. As shown in Figure 10, a user that holds a mobile device moves to his research office, whereas the device was originally connected to *KoreaUniv* AP since it had nothing else to connect. By the time he arrives to the research office, the device should connect to *wine515.hyun* which is an AP

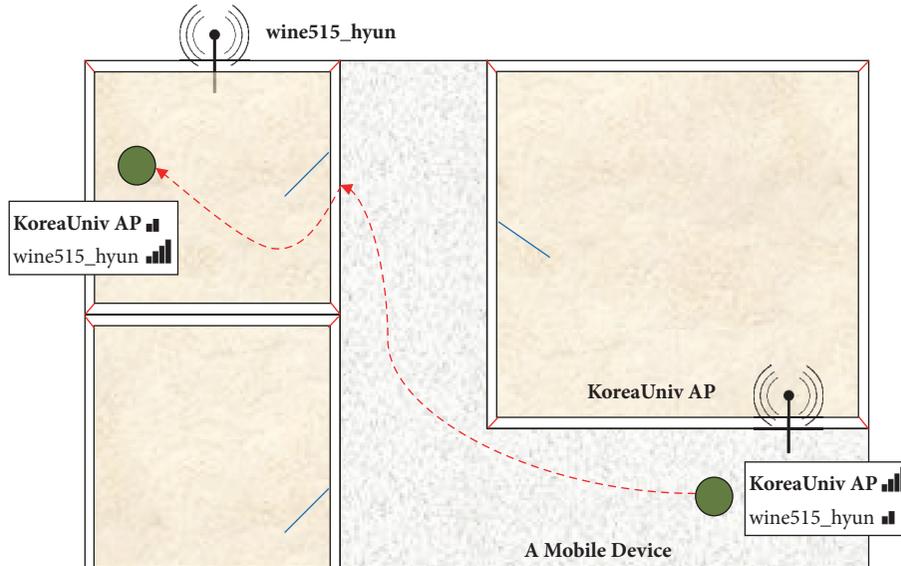


FIGURE 10: Experimental setup: a malicious connection to a bad AP.

installed on the research office and supposed to perform much better than the public AP, KoreaUniv AP. However, traditional mobile devices will sustain its connection to KoreaUniv AP since the connection is not terminated yet. Therefore, we can expect there is going to be severe QoE loss for the moment of staying at the research office without adjusting Wi-Fi connection to wine515_hyun. To show QoE degradation due to the behavior along with QoE enhancement achieved by SIMLESS, we performed sets of experiments that two devices were both carried by an experimenter who moved along with the trajectory shown in Figure 10. One device installed SIMLESS, whereas the other one did not. While the experimenter was moving, he executed three consecutive webpage clicks for each mobility state.

Figure 11 shows the resulted data flow of clicking webpages, which are assumed to incur http segments. Figure 11(a) shows overall data flow containing both *Walking* and *Stationary* states. and below two figures show parts of the data flow for each mobility state more closely to visualize differences. As shown in Figure 11(b), while *Walking*, two devices with different connectivity management had almost similar data flow pattern since they were connected to the same Wi-Fi AP, KoreaUniv AP. However, Figure 11(c) shows completely different data flow pattern between a SIMLESS device and an ORIGINAL device. SIMLESS device always finished its data reception sooner than the device without SIMLESS, and this is because the SIMLESS device quickly switched to wine515_hyun AP which was highly preferred AP when the device was in *Stationary* state. To visualize throughput enhancement more clearly on variety of Internet services, we performed additional evaluations on different Internet services similar to the way we did for webpage service. After experimentations, we computed the average throughput of each service in use for both *Walking* and *Stationary* states. We used Android's TrafficStats class to derive total amount of

bytes received, which includes both UDP and TCP streams. The results are shown in Figure 12.

The services named PlayMovie and PlayStore are a streaming service and an application download service provided by Google. Interestingly, SIMLESS performance enhancement gets greater for these services. The main difference between these two services and other two (webpage and YouTube) is the continuity of wireless communication. When PlayMovie and PlayStore require continuous wireless communication until the end of each service, webpage and YouTube download contents in discontinuous manner. The performance difference between these two service types occurs because services like webpage and YouTube stop wireless communication ahead of full utilization of bandwidth. From aforementioned set of experiments and results, we can tell that SIMLESS outperforms a mobile device with traditional Wi-Fi capability in terms of throughput in great degree for certain cases.

7.2. Cost-Effectiveness in Terms of Average Throughput with Expense. To show cost-effectiveness of the proposed system called SIMLESS, we performed a long-term experiment, one to two hours each, and evaluated the total communication cost in terms of dollar and the average throughput in terms of Mb/s. Two Nexus 5 devices were used to make comparison between an ORIGINAL device and a SIMLESS device, and all three mobility states are contained within each experiment to make evaluation realistic. Additionally, instead of fixating an experimenter's behavior into certain route, we made the experimenter put two devices into a backpack and made him carry along with wherever he wanted to go. The given liberty of behavior and mobility was expected to show the scalability and utility of SIMLESS, and the results of experiments proved that the expectation was right.

The cost-effectiveness of each device is derived by how much data expense is used by each device and how many data

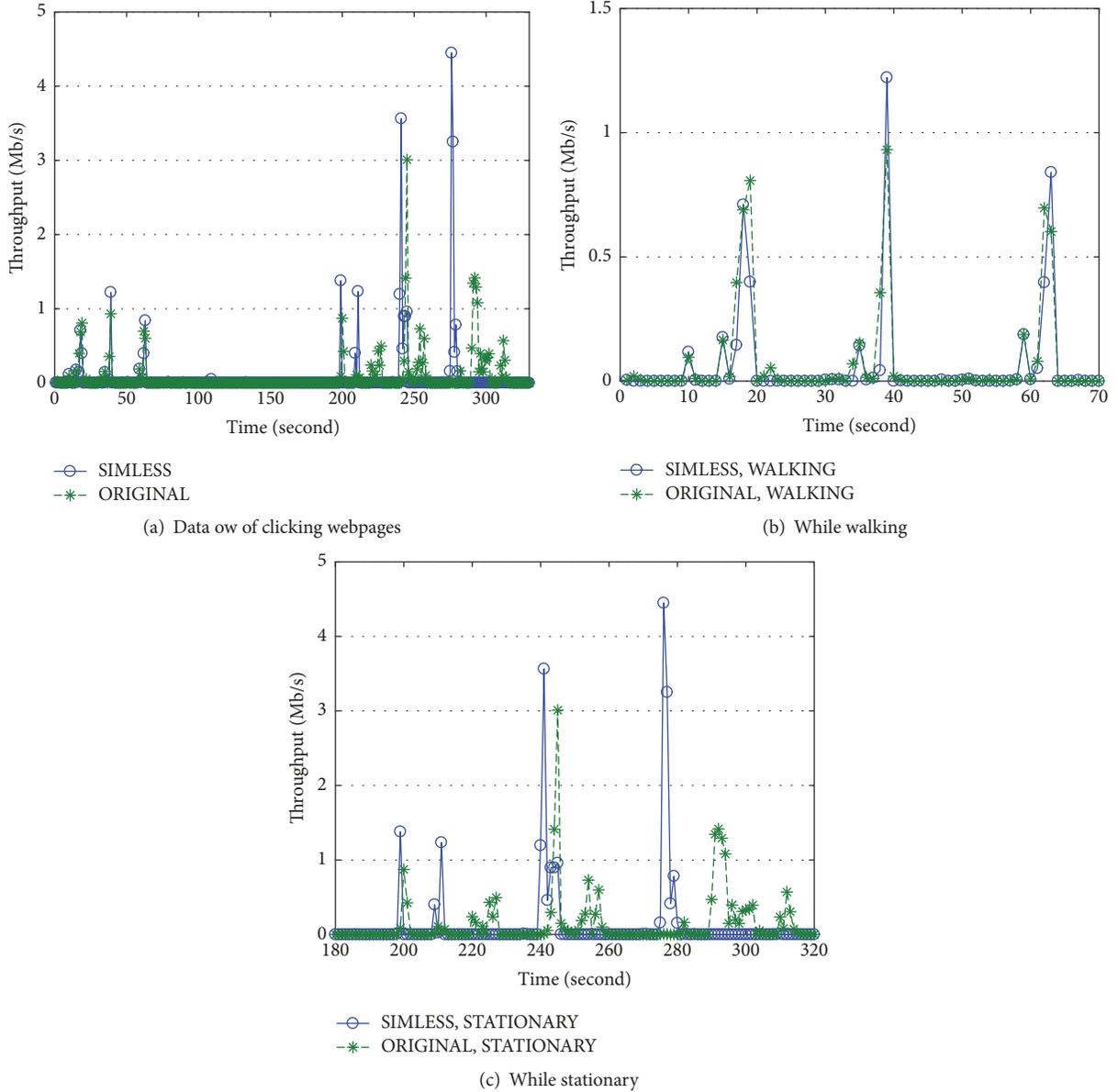


FIGURE 11: Experimental result: comparing two devices that consumes same Internet traffic through user clicks on webpages.

the device has communicated through any wireless interface during the evaluation. The cost of using Wi-Fi is free, whereas the cost of using LTE is in accordance with what data plan the experimenter is using for the mobile device. Assuming he pays \$50 for 15GB of data per month, which is real in South Korea, he can use up to 300MB for each \$1 payment. The better cost-effectiveness comes from the fact that the device uses more Internet services while paying less data cost. Therefore, the cost-effectiveness of wireless communication is described as follows:

$$E_{\text{cost}} = \frac{\text{Total}_{\text{data}}^{\text{Wi-Fi}} + \text{Total}_{\text{data}}^{\text{LTE}}}{\text{Total}_{\text{data}}^{\text{LTE}}/300\text{MB}/\$}, \quad (3)$$

where $\text{Total}_{\text{data}}^i$ indicates the consumption of data during connection to a specific access network i . The consumption

indicates the multiplication of Wi-Fi (or LTE) data rate and the period of using Wi-Fi (or LTE), and the data rate differs by what access network is in use. These can be interpreted as follows:

$$\text{Total}_{\text{data}}^{\text{Wi-Fi}} = \left\{ \sum_{i=1}^n r_i t_i \mid i \subseteq L_{AP} \right\}, \quad (4)$$

$$\text{Total}_{\text{data}}^{\text{LTE}} = r_{\text{LTE}} t_{\text{LTE}}, \quad (5)$$

where L_{AP} , r , and t refer to the list of APs in Wi-Fi history, data rate, and connection duration, respectively. To derive data rate, we evaluated the performance of LTE and every AP listed on Wi-Fi history. The performance test is based on speed test application which used UDP or TCP/IP to measure data rate of both uplink and downlink. Measuring data rate

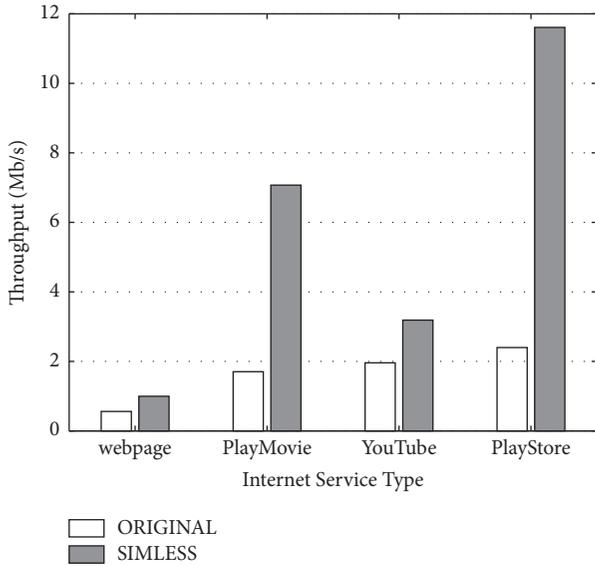


FIGURE 12: Experimental result: throughput enhancement on variety of services.

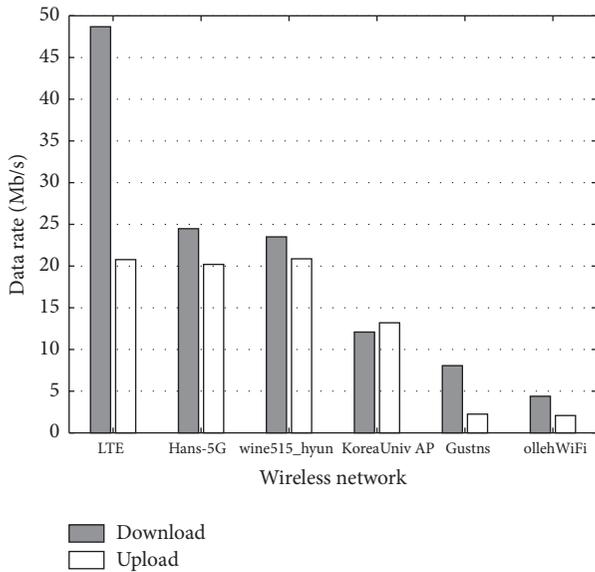


FIGURE 13: Wireless network performance analysis: download and upload speed.

of both direction is possible because the application provides backend server to collect statistics.

Figure 13 shows the performance of each AP and LTE data network in terms of data rate. We performed 50 speed tests to measure average data rate of each access network. Since the application we used to derive speed test had an external server, it was possible to derive the data rate of both uplink and downlink separately. LTE outperforms all other Wi-Fi APs in terms of download data rate, whereas its upload performance is similar to highly preferred Wi-Fi APs such as Hans-5G and wine515_hyun. However, Gustns, the mostly preferred Wi-Fi AP in *Walking* and *Driving* states, shows

very low performance than many others. However, most of other Wi-Fi APs are static APs that will be disconnected soon due to device mobility. Although total of 9 access points were collected to form a Wi-Fi preference list, four APs from them are not included in this performance analysis because they could not even succeed on establishing end-to-end connection for proper test. Therefore, we set data rate for those as 0 Mb/s.

Based on aforementioned cost information and data rate information of each access point, we evaluated cost-effectiveness of two devices, an ORIGINAL device without any intelligence on Wi-Fi connection and a SIMLESS device. Total of 5 test cases were done to prove generality of SIMLESS regardless of time. As shown in Figure 14, SIMLESS outperforms devices with original Wi-Fi connection behavior most of the time, except for test case 2. The reason of the ORIGINAL device having higher throughput than SIMLESS is because the device used LTE more often than SIMLESS. As shown in Figure 15, the cost spent by ORIGINAL device for test case 2 exceeds \$45, which extremely overpasses that of the SIMLESS device, which is less than \$20. Therefore, the result of test case 2 rather shows that the cost-effectiveness of SIMLESS since using less LTE resulted in low data rate but cost was down much less. The overall cost-effectiveness of both devices in various cases is shown in Figure 16. Cost-effectiveness of test cases 3, 4, and 5 clearly shows performance enhancement achieved by SIMLESS. Although the cost-effectiveness of test cases 1 and 2 is not visible from the figure, we have checked that at least 20% of improvement is shown except for upload cases of test cases 1 and 3. This is because the preference of the device user is focused on download rather than upload, which results in SIMLESS connecting to an AP with better downlink speed, but sometimes worse uplink speed. For instance, the user carrying portable device called *Gustns* shows 4 times higher data rate for download compared to upload, whereas usual public access points have 2 times higher data rate for download than that for upload. Since the user prefers the portable device that shows much higher download data rate, the possibility of connecting public access points decreases, which causes higher download quality and lower upload quality. We believe that a user that has equal preference on both download and upload will have a preference list with APs that have both higher data rate on download and upload, which results in cost-effectiveness results different from aforementioned results. But for this user, lowered upload speed will not matter since download speed was enhanced.

7.3. Power Savings Compared to GPS Based Systems. The main advantage of SIMLESS is the use of already-on sensors for mobility state estimation, unlike other mobility tracking systems that use GPS for high accuracy. Since SIMLESS does not require absolute location of the device, using GPS is rather a burden to the mobile device. Also, receiving GPS signal is not very reliable in case that the device is located inside building or between skyscrapers, which is very usual in urban areas. Unlike unstable and energy draining GPS enabled systems, most of IMU sensors and a pressure sensor

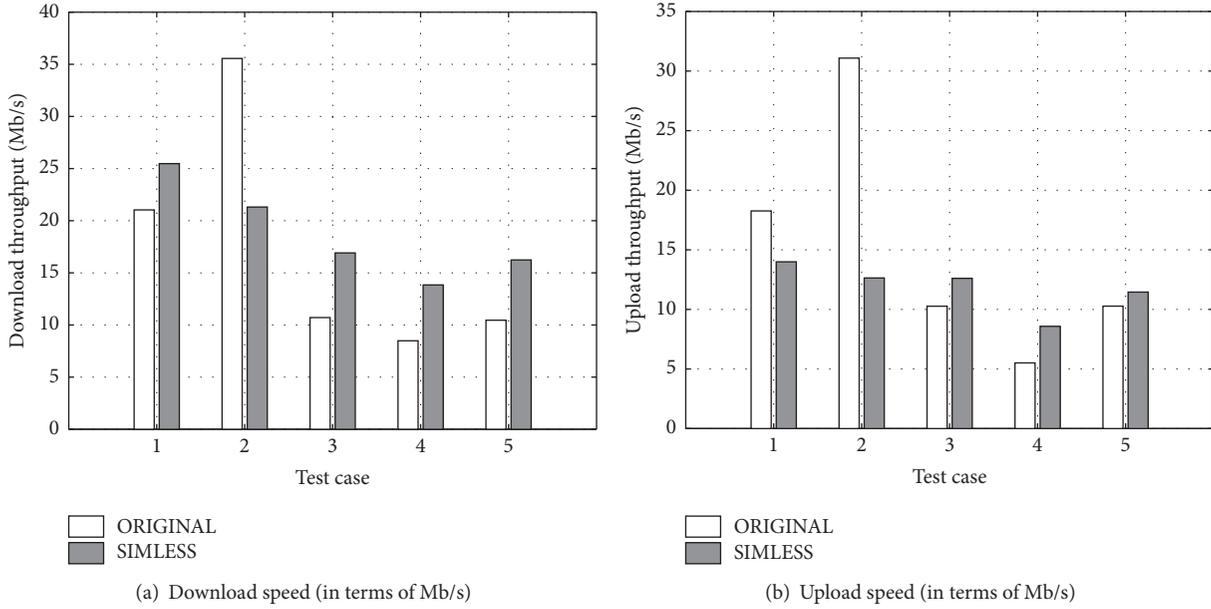


FIGURE 14: Throughput comparison between ORIGINAL and SIMLESS devices.

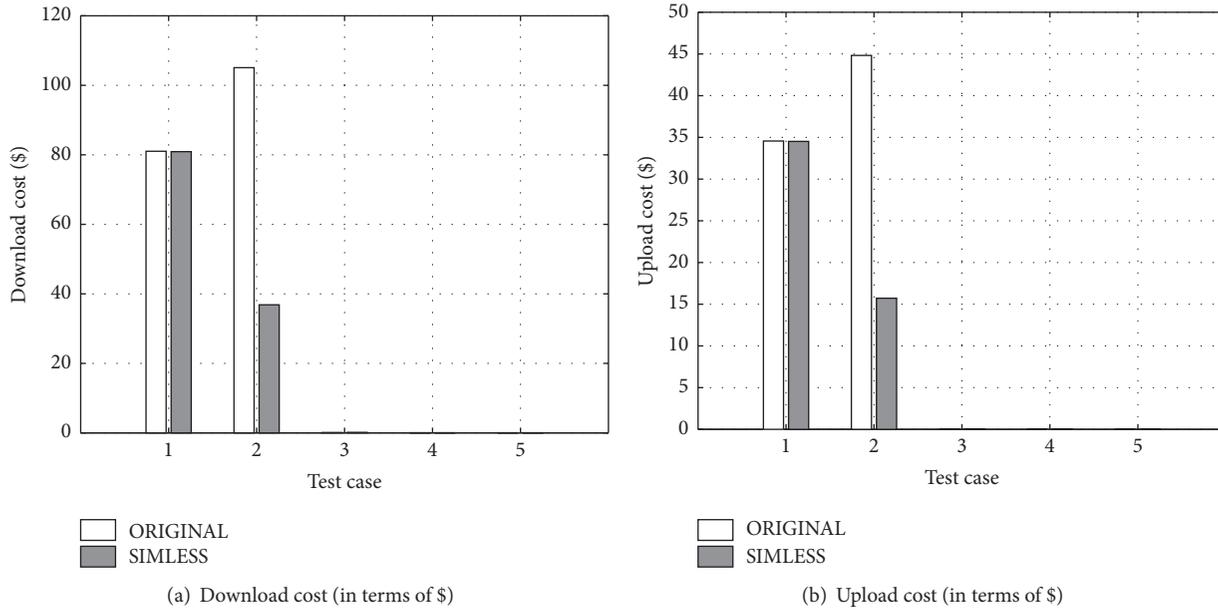


FIGURE 15: Cost comparison between ORIGINAL and SIMLESS devices.

are always-on whenever the device is enabled, and applications only need to use receiver function to receive output variables from these sensors. The power overhead only lies in computing sensor variables rather than the power overhead by enabling attached sensors. To visualize power efficiency of using SIMLESS compared to GPS enabled systems, we performed energy consumption analysis on mobile device named Galaxy Nexus using PowerMonitor tool as shown in Figure 17. Since PowerMonitor tool supplies power to the mobile device, the evaluation does not include any data about battery status and alteration.

Although we performed various experiments using Google Nexus 5, we used Google Galaxy Nexus for power evaluation because Nexus 5 is designed as solid bottom type. A user cannot separate battery from Nexus 5, which means we cannot wire magnetic pole inside the device without taking Nexus 5 into pieces. Although PowerMonitor tool supports USB mode power monitoring, we found that the power reading was too unstable to derive credible data. Therefore, we used Galaxy Nexus which had detachable battery that made power reading easy. We prepared three types of devices with three states of the devices, total of 9

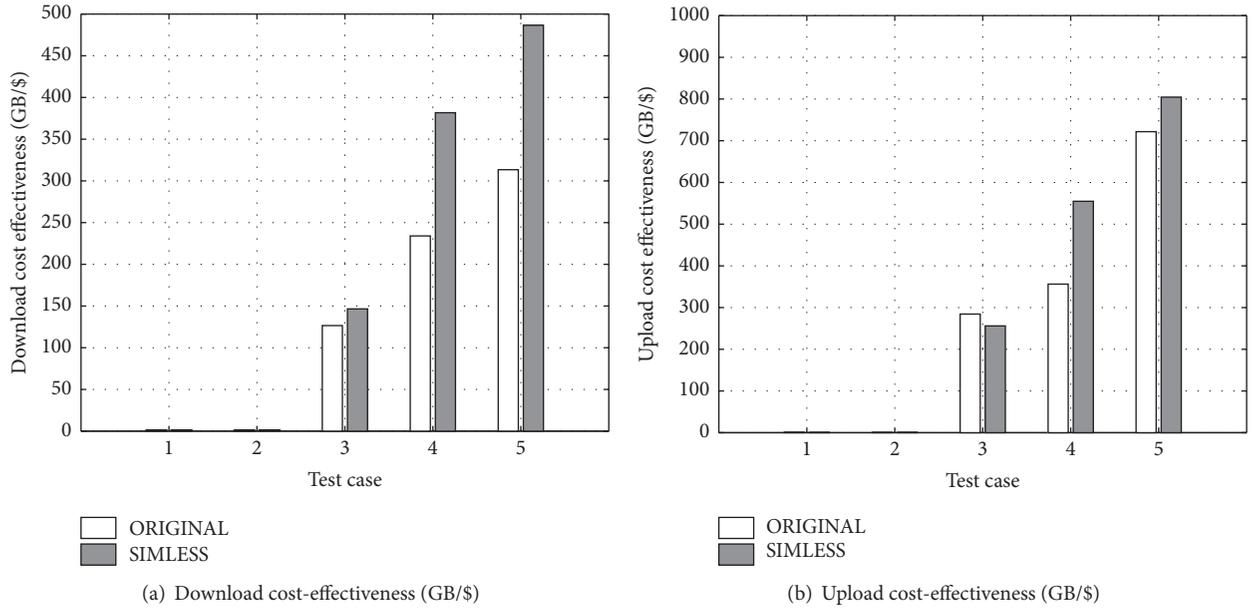


FIGURE 16: Cost-effectiveness comparison between ORIGINAL and SIMLESS devices.



FIGURE 17: Experimental setup: power analysis of ORIGINAL, SIMLESS, and GPS enabled devices.

experiments, to visualize the power efficiency of SIMLESS compared to GPS enabled device, and a device without any Wi-Fi intelligence applied. Each experiment was held for 10 minutes, and the results are shown in Figure 18. The bar graph shows average power consumption of three devices that are classified as ORIGINAL, SIMLESS, and GPS enabled. Total of three device states are measured, categorized as screen off, screen on, and video streaming. We used YouTube as a video streaming service. From the experiments, we can see that GPS enabled device consumes much more battery than the other two device types, whereas the difference is quite marginal between ORIGINAL and SIMLESS. However, the power overhead of SIMLESS becomes distinctive when the device’s screen is off. We found out that the power consumption of SIMLESS is almost 3 times that of ORIGINAL device. This is because SIMLESS requires computation for mobility state classification and periodical monitoring of Wi-Fi environment to sustain or even improve the wireless communication quality. However, we found that quite a lot portion of power is drained by frequent I/O operations due to file writing, which was required to collect experiment

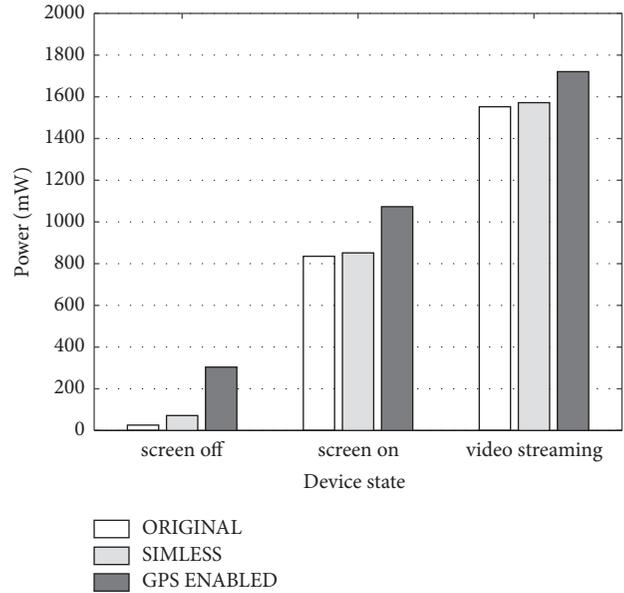


FIGURE 18: Experimental result: comparison among ORIGINAL, SIMLESS, and GPS enabled devices.

data for performance evaluation of SIMLESS. If SIMLESS disables such I/O operations, we could see that average power consumption decreases from 71.64 mW to 48.2 mW. We believe additional reducing computation on mobility state estimation and increase in the term of periodic scanning will reduce the power gap between ORIGINAL and SIMLESS.

7.4. Suggesting SIMLESS to Multiple Users. To prove applicability of SIMLESS in the real world, we suggested multiple smartphone users to use the system. Total of 5 users volunteered, using different device models released from 2011 to

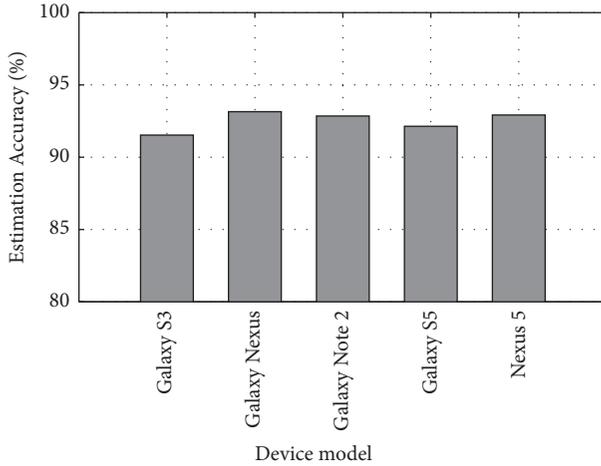


FIGURE 19: Mobility estimation accuracy achieved by 5 different mobile devices.

2014. Prior to installing SIMLESS on all these devices, we applied aforementioned mobility learning phase and Wi-Fi history learning phase in order to make SIMLESS properly operate. Also, the battery remaining from all experimenters was sufficient enough to finish up experimentation without disruption. We collected training set of 5 mobile devices from each user and applied machine learning to derive decision tree thresholds for proper mobility state decision. Figure 19 shows the accuracy result by using appropriate threshold derived by our machine learning method. Through multiple cross-validations, all mobile devices show more than 90% accuracy of estimating mobility state of their users.

Using mobility state estimation, Wi-Fi history of each mobile device was collected 24 hours long to analyze overall network access pattern of each user. Since each person might use different private, public, and office APs, we categorized tens of APs to 4 types, Home, Office, Movable, and Public. Figure 20 shows the Wi-Fi usage tendency of each mobile device, in terms of percentage of each access point type. The result shows variability of Wi-Fi usage tendency, which differs by life pattern and preference of users. For instance, Galaxy S3 user barely used LTE, whereas Galaxy S5 user used LTE quite often. This is because Galaxy S5 user took more time to travel back and forth between office and home because the user lived far away from his working place. Moreover, Galaxy S3 used movable AP more often than Galaxy Note 2 user even though their travel time, *Driving* state, was similar. This is because Galaxy S3 user had his own portable AP that performed much better than carrier APs installed on subway trains. Galaxy Note 2 user was annoyed by low quality movable APs, which made the user switch to cellular for better Internet service.

Based on Wi-Fi history result of each mobile device, we created AP preference lists and applied SIMLESS to each device to evaluate the cost-effectiveness. Experimenters ran another 24 hours long experiments for these 5 devices that interactively searched for more preferred APs using AP preference lists for each mobility state. Figure 21 shows the cost-effectiveness comparison between ORIGINAL and

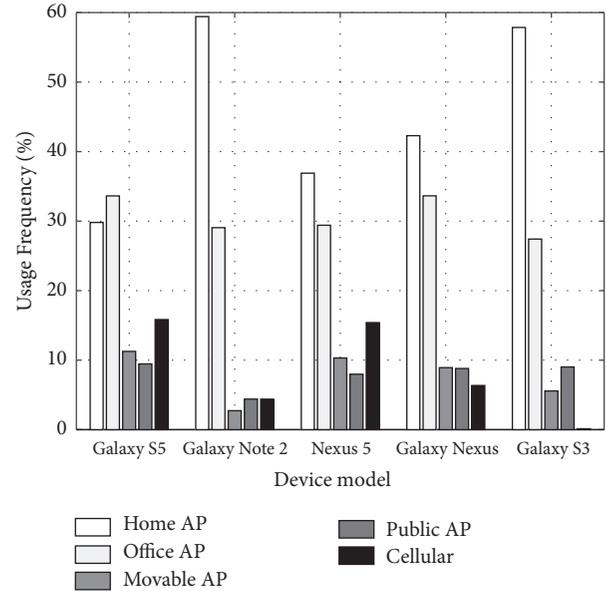


FIGURE 20: Mobility estimation accuracy achieved by 5 different mobile devices.

SIMLESS. Except for Galaxy Nexus, the rest of the devices show performance enhancement achieved by SIMLESS. For Galaxy Nexus, the cost-effectiveness is downgraded because the day SIMLESS solution was operated had more *Driving* state than the day without SIMLESS solution (about 1.5 times), which caused more usage of LTE due to no proper Wi-Fi AP to choose. Indeed, the cost of SIMLESS was about 10% larger than that of ORIGINAL device, resulting in lower cost-effectiveness. We believe that performance of SIMLESS would have been better if two cases were run simultaneously on the same day.

7.5. Comparing with Other Device-Centric Approaches. In this section, we show another set of experiments to show the strength of SIMLESS compared to other device-centric approaches that require some extra time to evaluate link status ahead of network usage. We excluded state of art learning based network selection mechanisms for this experiment because they all require feedback from network side. Requiring feedback from network side lacks scalability, which is one of the major strength of our solution SIMLESS. Therefore, this section only provides performance analysis of device-centric solutions.

First device-centric approach called *connection loop* [25] requires the device to connect every connectible AP and invest additional time to evaluate performance. Another device-centric approach called *wi-fi seeker* [26] requires consecutive number of scanning to estimate link quality of surrounding access points. On the other hand, our solution SIMLESS would require single scanning to list up available APs and choose one that is frequently used. To compare connection delay among three approaches, we went to a cafe, shown in Figure 22(a), that has been frequently visited by an experimenter with a mobile device that has been

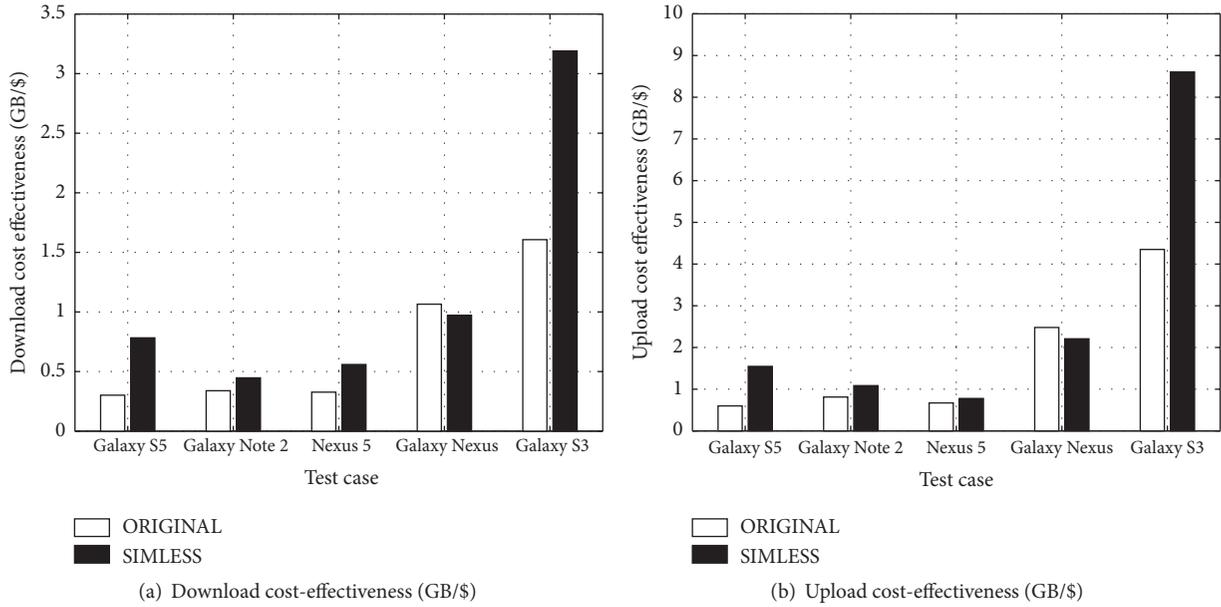


FIGURE 21: Multiple users’ survey: cost-effectiveness comparison between ORIGINAL and SIMLESS.

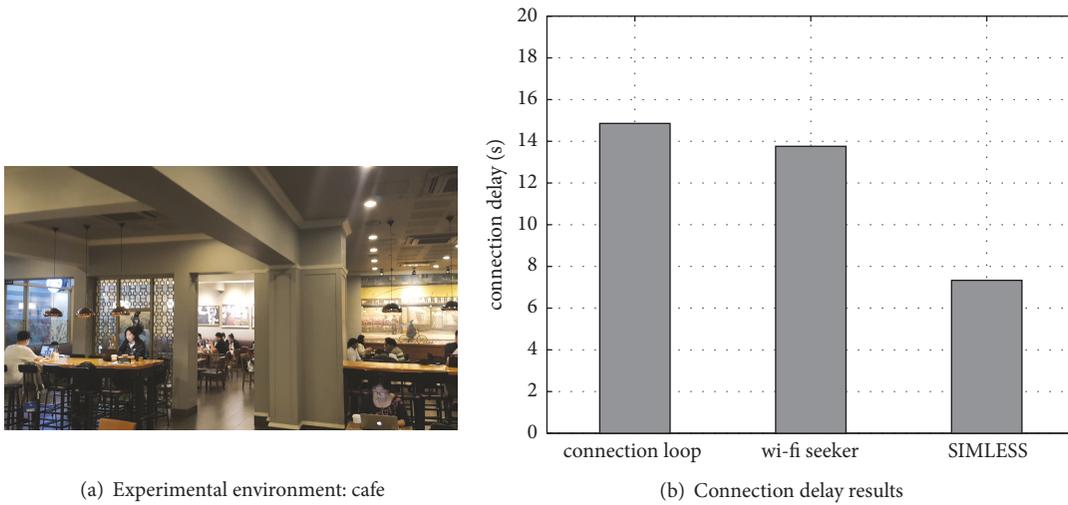


FIGURE 22: Comparing with other device-centric approaches.

trained by SIMLESS. The experimenter could access two Wi-Fi network, one of which has better performance than another. Through several experimentations, 3 device-centric solutions succeeded to connect the better one but showed different connection delay as shown in Figure 22(b).

As shown in the figure, SIMLESS shows much faster connection compared to others because it only requires one scan and one connection attempt whereas other device-centric solutions require extra time to measure the link quality. `connection loop` requires two connection attempts and extra link test by ping 3 times, whereas `wi-fi seeker` requires no more than one connection delay but still needs to perform multiple scanning to estimate link quality. If there were many connectible APs, the delay difference between these two solutions could be much larger.

It is true that many connectivity solutions based on receiving link quality statistics from backend servers would require much less delay than above two device-centric solutions. However, device-centric solutions have incomparable strength of having scalability. Since device-centric solutions can be used anywhere by anyone, including ones that do not subscribe services of network providers, we performed above delay comparison experiments excluding centralized solutions with less scalability.

8. Conclusion

In this paper, an intelligent Internet connectivity management called SIMLESS is proposed, which follows the concept of having different Wi-Fi preference based on different

mobility states. To make operable prototype, SIMLESS uses mobility tracing based on machine learning that utilizes sensor data from an accelerometer and a barometer and collects Wi-Fi history of the mobile device. Based on Wi-Fi preference which differs by mobility states, SIMLESS intelligently chooses the most preferred access point out of other selectable APs, resulting in enhanced user satisfaction with low Internet access cost. To prove usability and scalability of the proposed system, SIMLESS is tested on variety of smartphones that are released at different years, 2011 to 2014. From set of valuable experiments, SIMLESS is proven to enhance Internet service quality or reduce cost without requiring user intervention. Automation and intelligence on Internet connectivity management are brought by SIMLESS, and we believe the solution can be applied in various fields of handheld devices that require steady Internet connection with low cost.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Unmanned Vehicles Advanced Core Technology Research and Development Program through the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, Republic of Korea (NRF-2016M1B3A1A01937599) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01059151).

References

- [1] Ericsson, "Ericsson Mobility Report," 2014, <https://www.ericsson.com/en/news/2014/6/ericsson-mobility-report-june-2014>.
- [2] A. Sikora and V. F. Groza, "Coexistence of IEEE802.15.4 with other systems in the 2.4 GHz-ISM-band," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC '05)*, vol. 3, pp. 1786–1791, Ottawa, Canada, May 2005.
- [3] S. Lee, M. Bae, and H. Kim, "Future of IoT Networks: A Survey," *Applied Sciences*, vol. 7, no. 10, p. 1072, 2017.
- [4] K. S. Anupama, S. S. Gowri, B. P. Rao, and T. S. Murali, "An Intelligent Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks," in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I*, vol. 248 of *Advances in Intelligent Systems and Computing*, pp. 331–339, Springer International Publishing, Cham, 2014.
- [5] K. Radhika and A. V. G. Reddy, "Network selection in heterogeneous wireless networks based on Fuzzy Multiple criteria Decision Making," in *Proceedings of the 2011 3rd International Conference on Electronics Computer Technology, ICECT 2011*, pp. 136–139, ind, April 2011.
- [6] A. G. Mathias, M. S. Klahn, V. B. Yerrabommanahalli, and A. Nadathur, *Methods and apparatus for intelligent wireless technology selection*, uS Patent App. 13/768,508, 2013.
- [7] P. Dhawan, A. Mukhopadhyay, and C. Urrutia-Valdés, "Macro and small cell/Wi-Fi networks: An analysis of deployment options as the solution for the mobile data explosion," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 59–79, 2013.
- [8] C. S. H. DongKyu Lee, *Access point selection algorithm for providing optimal ap in sdn-based wireless network in*, APNOMS, 2017.
- [9] T. Hao, T. Papadopoulos, F. Tarcoci et al., "Portable access point," *uS Patent App. 12/785*, p. 154, 2011.
- [10] D. Križaj, M. Baloh, R. Brajkovič, and T. Žagar, "Design and Development of a Portable WiFi enabled BIA device," *Journal of Physics: Conference Series*, vol. 434, p. 012060, 2013.
- [11] R. K. Khedouri, J. N. Axelrod, and H. E. Price, "Wireless portable device for sharing digital content items," *The US Patent and Trademark Office*, vol. 8, 160, p. 495, 2012.
- [12] K. Fukuda and K. Nagami, "A Measurement of Mobile Traffic Offloading," in *Passive and Active Measurement*, vol. 7799 of *Lecture Notes in Computer Science*, pp. 73–82, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [13] X. Cheng, X. Huang, and D.-Z. Du, *Ad hoc wireless networking*, vol. 14, Springer Science & Business Media, 14, 2013.
- [14] A. S. Cacciapuoti, M. Caleffi, and L. Paura, "Reactive routing for mobile cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 803–815, 2012.
- [15] F. Legendre, "30 years of ad hoc networking research," in *Proceedings of the the 1st International Conference*, p. 217, Amritapuri, Kollam, Kerala, India, December 2011.
- [16] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380–392, 2014.
- [17] B. Brik, N. Lagraa, A. Lakas, H. Cherroun, and A. Cheddad, "ECDGP: extended cluster-based data gathering protocol for vehicular networks," *Wireless Communications and Mobile Computing*, vol. 16, no. 10, pp. 1238–1255, 2016.
- [18] A. Y. Chung, J. Jung, K. Kim et al., "Poster," in *Proceedings of the the 13th Annual International Conference*, pp. 477–477, Florence, Italy, May 2015.
- [19] T. Sohn, A. Varshavsky, A. LaMarca et al., "Mobility detection using everyday GSM traces," in *UbiComp 2006: Ubiquitous Computing: 8th International Conference, UbiComp 2006 Orange County, CA, USA, September 17–21, 2006 Proceedings*, vol. 4206 of *Lecture Notes in Computer Science*, pp. 212–224, Springer, Berlin, Germany, 2006.
- [20] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Determining transportation mode on mobile phones," in *Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC '08)*, pp. 25–28, Pittsburgh, Pa, USA, October 2008.
- [21] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with senseless," in *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds (MobiHeld '09)*, pp. 61–62, ACM, Barcelona, Spain, 2009.
- [22] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based Positioning for smartphones," in *Proceedings of*

- the 8th Annual International Conference on Mobile Systems, Applications and Services, MobiSys 2010*, pp. 299–314, usa, June 2010.
- [23] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær, “EnTracked: Energy-efficient robust position tracking for mobile devices,” in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services, MobiSys’09*, pp. 221–234, pol, June 2009.
- [24] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley, “Facilitating access point selection in IEEE 802.11 wireless networks,” in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, IMC 2005*, pp. 293–298, usa, October 2005.
- [25] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, “Improved access point selection,” in *Proceedings of the MobiSys 2006 - Fourth International Conference on Mobile Systems, Applications and Services*, pp. 233–245, swe, June 2006.
- [26] H. Kim, W. Lee, M. Bae, and H. Kim, “Wi-Fi Seeker: A link and Load Aware AP Selection Algorithm,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2366–2378, 2017.
- [27] Y. Liu, S. Cheng, and Y. Hsueh, “eNB Selection for Machine Type Communications Using Reinforcement Learning Based Markov Decision Process,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11330–11338, 2017.
- [28] N. Ul Hasan, W. Ejaz, N. Ejaz, H. S. Kim, A. Anpalagan, and M. Jo, “Network Selection and Channel Allocation for Spectrum Sharing in 5G Heterogeneous Networks,” *IEEE Access*, vol. 4, pp. 980–992, 2016.
- [29] Q. Wu, Z. Du, P. Yang, Y.-D. Yao, and J. Wang, “Traffic-aware online network selection in heterogeneous wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 381–397, 2016.
- [30] H. S. Kim, E. Kim, H. Kim, and H. Kim, “Microscale analysis on sensing devices and its impact on assessing user mobility,” in *Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications, AINA 2013*, pp. 910–917, esp, March 2013.

