International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume – 3, ISSUE – I **ISSN (Online):2454- 1966**

Research Manuscript Title

DEVELOPMENT OF AN APP FOR A SMARTPHONE SENSOR-BASED BUS INFORMATION SYSTEM

A.Afrin[1], R.Nandhini[2],L.Shilpa[3], Dr.L.Malathi[4].

UG Scholar[1,2,3], Assistant Professor[4]
[(1,2,3,4)]Department of Computer Science and Engineering,
[(1,2,3,4)]Vivekanandha College of Engineering for Women, Tamil Nadu, India.

Corresponding author E-Mail-ID: azizjaibu.afrin@gmail.com

March – 2017

www.istpublications.com

International Journal of Future Innovative Science and Engineering Research (IJFISER)
Volume – 3, ISSUE – I ISSN (Online):2454- 1966

DEVELOPMENT OF AN APP FOR A SMARTPHONE SENSOR-BASED BUS INFORMATION SYSTEM

A.Afrin[1], R.Nandhini[2],L.Shilpa[3], Dr.L.Malathi[4].

UG Scholar[1,2,3], Assistant Professor[4]
[(1,2,3,4)]Department of Computer Science and Engineering,
[(1,2,3,4)]Vivekanandha College of Engineering for Women, Tamil Nadu, India.

Corresponding author E-Mail-ID: azizjaibu.afrin@gmail.com

ABSTRACT

As the number of vehicles increases, it is necessary to have an efficient and an intelligent transportation system (ITS) to reduce the congestion and emissions on the road. However, ITS setup and installation requires a huge budget. Therefore, only large metropolitan areas are covered by ITS services. This paper proposes a smartphone-based bus information system (BIS) that does not require any special devices designed for bus detection, and the installation cost is next to nothing. The BIS can be seen as a bus-oriented version of an ITS, and the BIS is adequate for small- and mid-sized cities, because the bus is their only form of public transportation. In the proposed BIS system, GPS device (the driver, for example) in a bus sends information about his/her current location to the BIS server using his/her smartphone. Therefore, the usefulness of the proposed BIS depends on the accuracy of the global positioning system (GPS) receiver installed in the smartphone. We performed experiments by collecting GPS data and analyzing the collected data in order to prove that GPS receivers installed in smartphones are accurate enough to determine the location of the bus, and that the proposed smartphone-based BIS is consequently valid. In order to perform these experiments efficiently, we developed a mobile app that allowed us to concurrently collect GPS values from many smartphones.

Keywords— ***Bus Information System, Intelligent Transportation System, Android App, GPS***

1. INTRODUCTION

Telematics is the convergence of various fields, including telecommunications, vehicular technologies, road transportation, road safety, electrical engineering (sensors, instrumentation, wireless communications, etc.) and computer science (multimedia, Internet, etc.). Practical applications for telematics include vehicle tracking, fleet management, satellite navigation, and wireless vehicle safety communications [1].

An (ITS) is defined as a system where information and communications technologies are applied in the field of road transportation. Applications for an ITS include emergency vehicle notification systems, automatic road rules enforcement, variable speed limits, collision avoidance systems, and dynamic traffic light sequencing.

As the number of vehicles rapidly increases, the importance of an ITs is also increasing, because it can help reduce congestion and emissions, and provides citizens with efficient and convenient services.

Therefore, many cities install and operate a bus information system (BIS) that provides bus users with important information, including bus arrival information, route information, bus service information,

allocation information, route arrival information, station arrival information, bus arrival intervals, transit times, and so on. Every BIS requires special devices to track buses; setting up and installing a BIS calls for a huge budget. To address this problem, we propose a smartphone-based BIS structure that does not require special devices to detect buses, and is therefore cost-effective.

2. RELATED WORK

The US National Intelligent Transportation System Architecture was defined by the United States Department of Transportation and consists of many subsystems, including center systems, field components, vehicle equipment, and traveler devices [4].

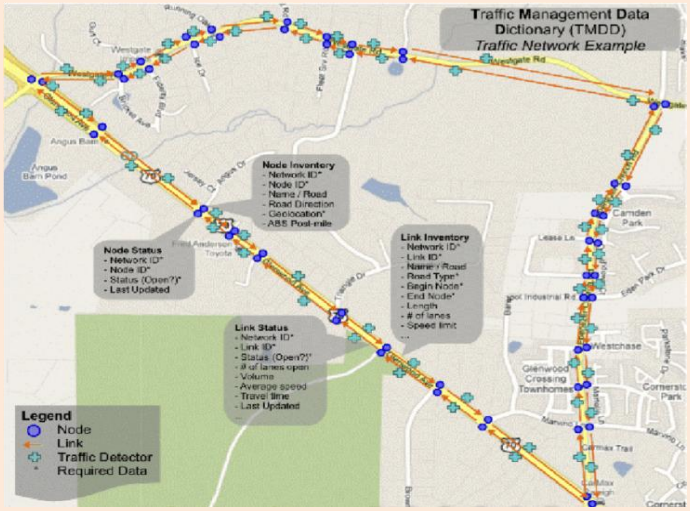

Fig 1: An example TMDD road network

IBM's Intelligent Transportation conforms to the National ITS Architecture. The Traffic Management Data Dictionary (TMDD) standard is the system through which the IBM Intelligent Transportation system gathers traffic and event data from other traffic center systems and field devices. In TMDD, nodes are defined as arbitrary points along a road, and links are defined as arbitrary parts of a road between two nodes. Figure 1 shows how TMDD nodes and links are mapped to a physical road network.

As shown in the figure, the National ITS Architecture requires a lot of traffic detectors. Examples of traffic detectors include inductive loop detectors, wireless vehicle detection systems [5], Doppler radar/microwave traffic detectors, and passive infrared traffic detectors. This implies that installing an ITS calls for an enormous budget.

Nowadays, most big cities, including Daejeon, have installed an ITS. The structure of the Daejeon ITS system is shown in Figure 2 [6]. A trunked radio system (TRS) is a complex type of computer-controlled two-way radio system that allows sharing relatively few radio frequency channels among a large group of users. Instead of assigning, for example, a radio channel to one particular organization at a time, users are instead assigned to a

logical group, or *talkgroup*. When any user in that group wishes to converse with another user in the talkgroup, a vacant radio channel is automatically found by the system, and the conversation takes place on that channel [7].

Each piece of road side equipment (RSE) contains automatic vehicle identification (AVI) and automatic vehicle location (AVL) systems. For example, an RSE can be based on a radio-frequency identification (RFID) reader. RFID is wireless non-contact use of radio-frequency electromagnetic fields to transfer data for the purposes of automatically identifying and tracking the tags attached to objects. Some tags are powered by, and read at, short ranges (i.e., a few meters) via magnetic fields (electromagnetic induction), and then act as a passive transponder to emit microwaves or ultrahigh frequency (UHF) radio waves. Others use a local power source, such as a battery, and might operate from hundreds of meters away. Unlike a barcode, the tag does not necessarily need to be within line of sight of the reader, and can be embedded in the tracked object [8].

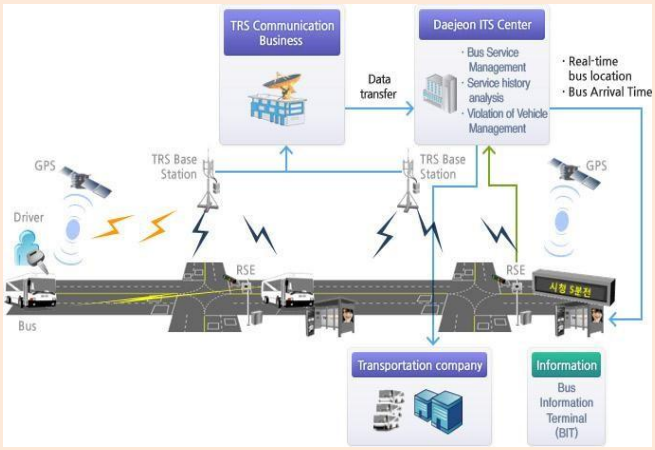

Fig 2: The structure of the Daejeon ITS

According to Figure 2, every bus has an attached RFID tag. The RFID reader inside the RSE reads the RFID tags and identifies the bus that arrives at the spot where the RSE is located. RSEs send collected data to a TRS base station in real time. TRS base stations, in turn, transmit the collected data to Daejeon's ITS center.

Based on the collected data, the ITS center then provides the following information to citizens: bus arrival information, route information, bus service information, allocation information, route arrival information, station arrival information, bus arrival intervals, transit times, and so on [6]. One of the most valuable services an ITS provides is bus arrival times.

Bus transports forms a significant part of public transportation in Singapore, with over 3.8 million rides taken per day on average of 2015.There are more than 300 scheduled bus services, operated by SBS transit, SMRT Buses and Tower Transit Singapore.

3. PROBLEM DESCRIPTION

Global positioning system is being actively employed in a variety of vehicular monitoring system such as car navigation and emergency assistance. Nowadays easily observed many passengers footed at the bus stops waiting for the buses. For the purpose of monitoring the movement of the bus, to report the location of the bus in the bus stops in its respective bus stop using Global Positioning System in collaboration with Transceiver module.

4. PROPOSED SYSTEM

A. Smartphone-Based BIS Architecture

In this paper, we introduce our design of a BIS that does not require any special equipment installed on buses or bus stops. Our system is a client/server system consisting of up-loaders, requesters and a server, as shown in Figure 3. An up-loader is a smartphone app that reads GPS values regularly, detects if the bus has stopped, and sends the current time and GPS values (latitude and longitude) to the server if the bus stops. Up-loaders are supposed to be run on bus passengers' smartphones. The server receives GPS values from up-loaders and saves them in Passengers Table in the database. The attributes in Passengers Table include User ID, Time, Latitude, and Longitude. The Bus Routes Table and Bus Stops Table are also part of the database. The attributes in Bus Routes Table include Bus Route ID, Start Stop, End Stop, and Bus Company.

The attributes in Bus Stops Table include Bus Stop ID, Bus Route ID, Latitude, and Longitude. With the sequence of GPS values from an up-loader, the server can recognize the bus route on which the bus is running. We call this bus route a bus running route. Note that we have one running bus route per up-loader. A requester is a smartphone app, and is supposed to be run by a user who is waiting for a bus. The user can be at a bus stop or at some other place. In the former case, the requester reads

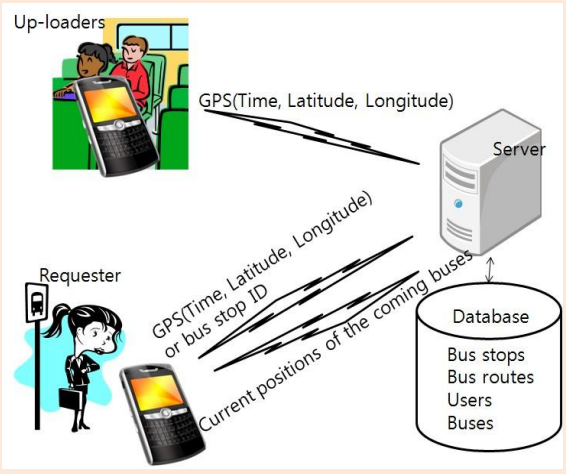

Fig 3: The structure of the smartphone-based BIS

The current GPS values and sends them to the server. In the latter case, the requester sends the bus stop ID (or name) to the server. Then, the server retrieves all the bus routes that contain this bus stop. We call these bus routes "related bus routes." The server returns the current positions of the passengers in the buses running on these related bus routes.

B. Feasibility Test

A BIS provides information on bus arrivals, routes, bus services, allocations, route arrivals, station arrivals, bus arrival intervals, transit times, and so on. Most of this information can be derived from bus location details. For example, if we know the bus locations at times t1 and t2, then we can easily find out the speed of the bus. If we know speeds of a bus at times t1, t2, …, tn at the same location, then we can figure out how heavy the traffic is at that location at times t1, t2, ..., and tn. A smartphone-based BIS depends entirely on smartphone sensors to gather bus location information. Therefore, we performed experiments to test the accuracy of smartphone sensors.

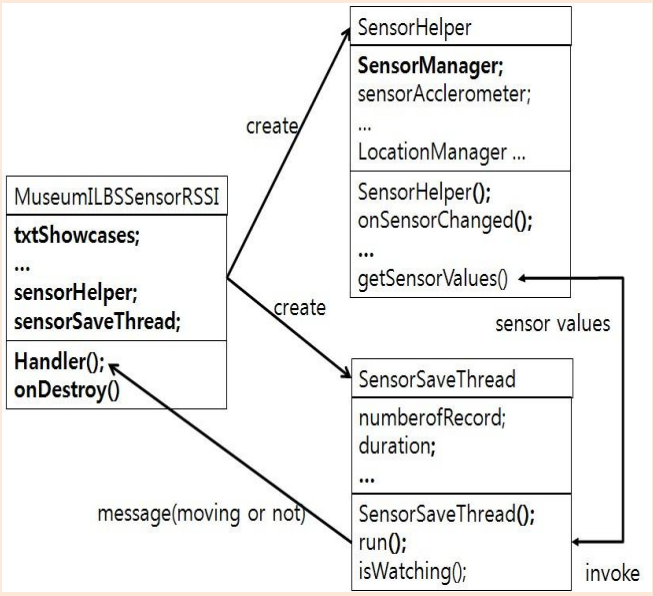

Fig 4: The structure of our Android app

The structure of our Android app, which determines whether the user is moving or not [19] The purpose of one of our tests was to identify if smartphone sensors can help to determine whether a bus has stopped or is moving. For this purpose, we implemented the Android app illustrated in Figure 4. The main process creates both Sensor Helper and SensorSaveThread, so that these two processes run concurrently. SensorSaveThread regularly collects sensor values by invoking getSensorValues(). is Watching() in SensorSaveThread determines whether the user is moving or not by analyzing the sensor values.

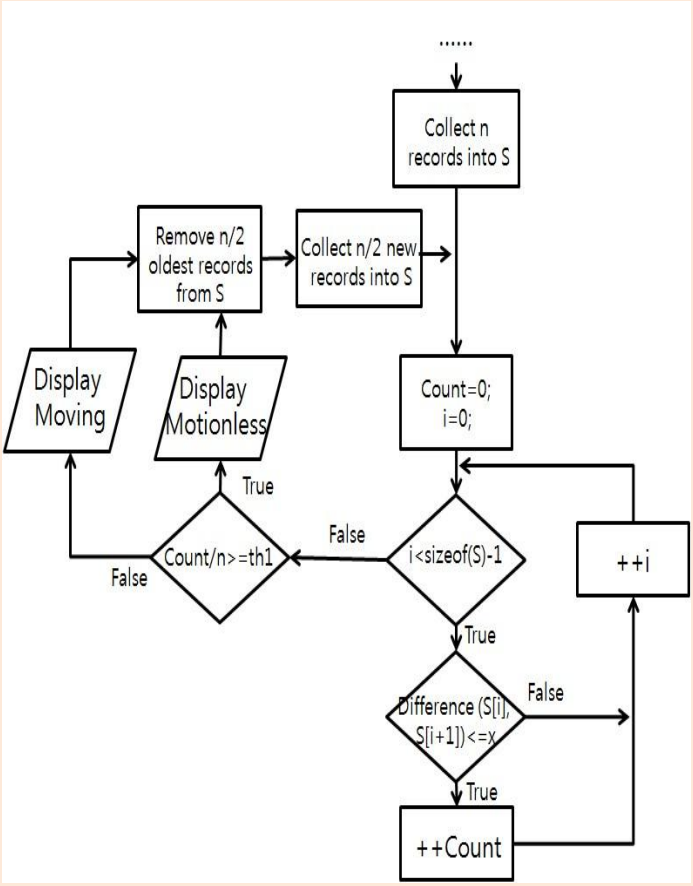

Fig 5: Our algorithm to determine whether the user is moving or not

The process is Watching() is shown in Figure 5. By investigating a sequence of accelerometer values, it determines whether the user is moving or not. In other words, it puts 10 consecutively collected accelerometer values in an array, S[]. For all i from 2 to 10, it compares S[i] with S[i-1]. If the difference is less than threshold x, it increases the counter. If the counter is higher than 5, it determines that the user is not moving.

Our test results showed that this app correctly detects whether a pedestrian is moving or not. However, the same app falsely indicated that a bus had stopped when the bus rapidly reduced its speed and continued at a slow speed. This app also provided false negative results. In other words, it determined that the bus was moving when the bus had moved slowly for a while before stopping. To address the above-mentioned issues, we slightly modified the app so it investigates speed values obtained from LocationManager instead of accelerometer values. It reads speed values every one second and outputs "The bus stopped" if the speed value is zero. This app displays buttons labeled "stop", "slow", "fast", and "curve," as shown in Figure 6.

We installed the app on a Nexus 7 with the Android 4.3 OS for the experiments. During the experiments, we touched the stop/slow/fast/curve button when the bus had actually stopped/ran fast/ran slow/ran on a curve. Our test results showed that the app correctly detects a stopped status 100% of the time, as shown in Table 1.

Table 1: A summary of testing the "stop status detection algorithm"

Fact / app	Stopped	Running	Running on a curve
Stop ped	c255	0	0
Moving	0	813	319

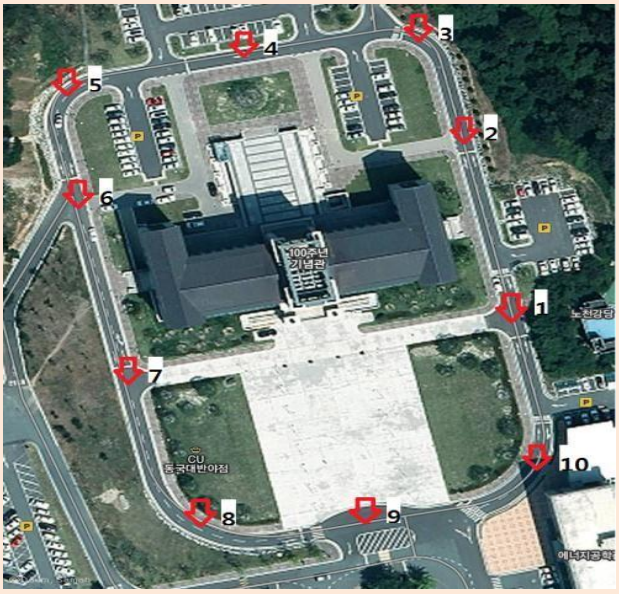

Fig 6: The test bed for estimating GPS deviation

Our BIS determines bus locations with GPS values. If the GPS is accurate, then our BIS correctly determines the bus location. Therefore, we performed experiments in order to find out how reliable the GPS values are. In this experiment, we made many mobile devices, (a Samsung Galaxy smartphone, tablet, Nexus, and others) read their GPS values at the same time in a running vehicle. If the GPS values were the same, then testing our BIS with only one mobile device was enough to validate the BIS. In order to make many devices read their GPS values at the same moment, we developed Android client and server apps so that all clients could return their GPS values as soon as they received a message from the server. If the collected GPS values are the same, then we can expect that the bus location provided by an up-loader is exactly correct. Therefore, we collected GPS values from many mobile devices in a car running the track shown in Figure 7. On the track were 10 landmarks.

Whenever we reached a landmark while moving, or stopped at a landmark, we collected GPS coordinates. The user interface of our server app has many buttons, as shown in Figure 8. A button label represents one of the landmarks shown in Figure 7. When the car reaches the landmark, we click the button associated with it, to indicate whether the car is running or stopped at a landmark. Then, the server sends the label to the clients. Each client reads its GPS values (coordinates) and saves the values with the label as soon as it receives a label.

Fig 7: A conceptual model of our client and server apps

The structure of our server app is described in Figure 9. DeviceListActivity gathers device IDs of all the smartphones around the server that are ready to communicate with it. onActivityResult() in MainActivity makes connection with one of the smartphones in the list made by DeviceListActivity and creates a ConnectedThread. SendObject() provides the button label to all connected devices when one of the 10 buttons (shown in Figure 8) is clicked.

We performed experiments with an Samsung J2. For each landmark, we computed a range of collected GPS values. Computed ranges for the F240L are shown in Table 2, for example.

Table 2: Test results for the samsung J2

Location	State	Range
1	running	5.655755497112620
1	stopped	20.422325377726800
2	running	12.004815335992400
2	stopped	20.552637001281600
3	running	19.413049688977800
3	stopped	39.312129270728700
4	running	98.866664248235300
4	stopped	84.579279547922100
5	running	23.379992164780400
5	stopped	20.864954390281200
6	running	17.216102873841100
6	stopped	36.656241710259900
7	running	8.578678240757560
7	stopped	28.219882639230400
8	running	15.818668138199300
8	stopped	99.456563770273400

9	running	9.747311910284890
9	stopped	108.176286109414000
10	running	16.043693259558300
10	stopped	48.694224808171400
Running average		22.672473135774000
Stopped average		50.693452462529000

Our test results showed that ranges of GPS coordinates were all less than 50 meters. The GPS installed on Samsung J2 is especially accurate, and the range of GPS values collected during the experiment (for both running and stopped states) was 6.1 and 3.1 meters, respectively. However, the range of GPS coordinates collected from an old model was significantly greater.

BISs provide various kinds of information. For instance, after a user supplies data on departure and destination bus stops, the system can return an efficient route that the user can take. The route can involve switching buses. Here are examples of services a BIS can provide: given a bus stop, the system can return a list of points of interest (POI) around it; given a pair of bus stops (A and B), the system will yield an estimated travel time between these points and the estimated arrival time of the next bus.

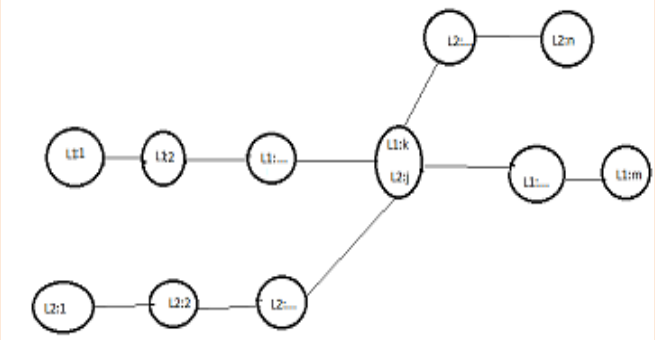

Fig 8: A typical bus routes graph

Therefore, a BIS should have a graph that represents all bus routes in the city. A typical bus routes graph is shown in Figure 11. The "L1:1" node represents the first bus stop of the L1 bus route. The "L1:k L2:j" node indicates that the k-th bus stop of the L1 bus route is also j-th bus stop of the L2 bus route. Given "L1:2" as the departure bus stop and "L2:n" as the destination bus stop, the BIS will recommend the following route:

"Take L1 bus at 'L1:2', get off at 'L1:k L2:j', take L2 bus, and get off at 'L2:n'."

By applying serial node fusion, we can reduce the number of nodes on the graph. For example, Figure 11 will be reduced to Figure 12 as the result of serial node fusion. The fewer the number of nodes, the faster the process of finding the bus route.

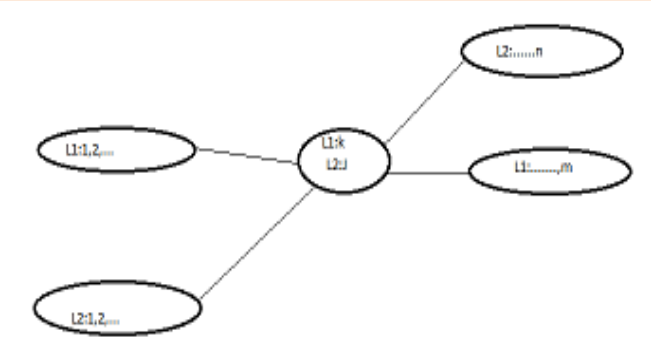

Figure 9: A result from serial node fusion

An element a[i, j] of an incidence matrix A of a graph can represent the weight of the arc from vertex i to vertex j. The weight of an arc usually represents the distance from vertex i to vertex j. We suggest filling a[i,j] with "GPS-time at bus stop j — GPS-time at bus stop i" to represent the elapsed time for the up-loader to reach bus stop j from bus stop i.

5. RESULTS AND DISCUSSION

Sensor value analysis allows us to determine whether the bus has stopped or is running. Therefore, we can make the up loader send its GPS value only when the bus has stopped, in order to save communications costs. Furthermore, we can make the up-loader send its GPS value only when the bus has stopped at a bus stop by referring to the list of pairs: (bus stop ID, bus stop coordinates).

6. SUMMARY AND CONCLUSION

The BIS has proven very useful, and can be implemented in many ways. However, BIS setup and installation requires a considerable budget. The smartphone-based BIS solution proposed in this paper does not require special devices dedicated to detecting traffic, and as a result, is more costeffective. Because the effectiveness of a smartphone-based BIS depends mostly on the accuracy of the GPS devices installed in smartphones, we ran several tests to prove the accuracy of the proposed solution. Our test results showed that GPS devices installed on most Android smartphones are accurate enough to be used for a smartphone-based BIS.

REFERENCES

[1] http://en.wikipedia.org/wiki/Telematics

[2] http://en.wikipedia.org/wiki/Intelligent_ transportation_system

[3] J. Yim, "Proposing a Collaborative Bus Arrival Time Estimation Method," Proc. The 2014 FTRA International Conference on Advanced Computing and Services (ACS-14), 2014.

[4] M. Christopher and M. Laffoon, "Build Intelligent Transportation Systems with the Traffic Management Data Dictionary Standard," IBM Developer Works, 15 July 2011.

[5] http://www.techtransfer.berkeley.edu/newsletter/083/vehicle-detection-with-wireless-sensor s.php

[6] http://traffic.daejeon.go.kr/eng/introduction/ intelligentTransportSystem.do

[7] http://en.wikipedia.org/wiki/Trunked_radio_system

[8] http://en.wikipedia.org/wiki/Radio-frequency_identi fication

[9] J. Gong, M. Liu and S. Zhang, "Hybrid dynamic prediction model of bus arrival time based on weighted of historical and real-time GPS data," Proc. 25th Chinese Control and Decision Conference (CCDC), 2013.

[10] R. Padmanaban, K. Divakar, L. Vanajakshi and S. Subramanian, “Development of a real-time bus arrival prediction system for Indian traffic conditions”, Intelligent Transport Systems, IET Vol. 4 Issue 3, 2010.

[11] M. Darbari, S. Medhavi, A. Srivastava, “Development of effective Urban Road Traffic Management using workflow techniques for upcoming metro cities like Lucknow (India)”, IJHIT Vol. 1, No.3, 2008. [12] R. Hou, Q. Wang, J. Wang, J. Wang, Y. Lu and J. Kim, “A Fuzzy Control Method of Traffic Light with Countdown Ability”, IJCA Vol. 5, No. 4, 2012.

[13] A. Czyzewski and P. Dalka, “Visual Traffic Noise Monitoring in Urban Areas”, IJMUE Vol.2, No.3, July 2007.

[14] S. Yoon, Y. Lim, H. Lim and H. Kim, “Architecture of Automatic Warning System on Urgent Traffic situation for Headphone Users,” IJMUE Vol.7, No.2, 2012

[15] R. Maravilla, E. Tabanda, J. Malinao and H. Adorna, “Data Signature-based Time Series Traffic Analysis on Coarse-grained NLEX Density Data Sets”, IJDTA Vol. 5 No. 1, 2012