Smart Campus Phase One: Smart Parking Sensor Network

H.M.A.P.K. Bandara, J.D.C. Jayalath, A.R.S.P. Rodrigo,
R.G. Ragel, A. Bandaranayake, Z. Maraikar
Department of Computer Engineering,
Faculty of Engineering, University of Peradeniya,
Peradeniya 20400, Sri Lanka.

Abstract—**Smart Parking Sensor Network project aims to develop a low cost sensor based parking system to map the usage of parking areas. This system consists of sensor nodes which can detect the occupancy of parking space; relay nodes to communicate between sensor nodes and the server; server application to get data from the relay nodes and send data to mobile application; and a mobile application to display the parking areas and the occupancy of the parking areas on a map. The mobile application was developed using Android and the server application is hosted in AWS (Amazon Web Services). The vehicle detection sensor node was designed with magnetic sensor and a distance sensor. The magnetic sensor detects the presence of the vehicle and the distance sensor clarifies it.**

I. Introduction

The ”Smart Campus” concept is an initiative to use ICT (Information and Communication Technology) within a University Campus to improve the quality and performance of the services, to reduce costs and resource consumption, and to engage more effectively and actively with its members. Smart Parking is a constituent of the Smart Campus, that looks to address this issue of parking within a campus. The difficulty in finding available parking spaces within a campus waste time and fuel, and create stress among many visitors and employees. In a regular park, there is no way to know where the available parking spaces are. By delivering a real time map of the availability of parking spaces, smart parking idea directly impacts on the sustainability of university, revenue and the level of the service.

This paper presents a pilot project to be at the forefront of this change at the University of Peradeniya. Phase 1 of Smart Campus introduces a Sensor Network/IoT-based Smart Parking System to map the usage of parking spaces within the University. This parking system consists of sensing devices to determine the occupancy at the parking space, a network to transfer data, a server application to process the data, and a mobile application to display occupancy of the parking spaces. Smart parking is a huge industry overseas, but not in Sri Lanka. Because of that, all the components should be bought from foreign vendors to implement a smart parking system, which includes the vehicle detection node, the relay node (to communicate between server and sensor) and the applications. The average price of a vehicle detection node is USD80 - USD160 (ROSIM-ITS) and the average price of a parking system is USD5000 per year (excluding taxes and delivery cost) [1]. Because of that, it is costly to deploy a smart parking system using imported components. Therefore, this paper suggest a low cost solution which can be developed within the country and deployed in mega scale.

II. Related Work

Commercially developed smart parking solutions are very popular in the global market. Streetline [2] is one of the leading smart parking company based in USA. Their solutions include; parking analytics platform, parking inventory management platform, parking map, real-time parking guidance application, parking data API, potential real-time violations for optimal workforce efficiency. Their vehicle detection sensor is surface mounted and primarily made with magnetic sensors. Smart Parking Limited [3] (ASX: SPZ) has been providing smart parking solutions for more than ten years. The company provides on-street, off-street and parking management solutions. Their vehicle detection sensor is made with infrared technology, and it is surface mounted. Libelium [4] designs and manufactures hardware and a complete software development kit (SDK) for wireless sensor networks so that companies can deliver reliable Internet of Things (IoT), and Smart City solutions. This platform is designed to be buried in parking spaces and to detect the arrival and departure of vehicles. The sensor node is based on magnetic sensors.

The above three are the leading companies in smart parking solutions. All three companies provide sensor nodes which are surface mounted and buried in the parking area and this paper discusses a surface mount sensor node, that can be applied even for already constructed parking areas without making lots of changes. The commercial solution providers are mainly providing their solutions in American, European and Australian continents. Applying the solution in Asian continent is expensive because of the reason discussed in the previous section. Our system is implemented in cost effective way to counter that problem. Our mobile application also contains similar functionalities to those which are in the solutions mentioned above.

There are some research based projects about vehicle detection sensors. In [5], Koszeteczky and Simson suggest a magnetic sensor-based system to identify the vehicle and estimate their powers with some algorithms. The proposed solution of this research is that a vehicle can be determined by using magnetic sensors. In [6], Casas, Sifuentes and Pallas-Areny suggest a Node made of a magnetic sensor and an Optical sensor for detecting vehicles. They choose the optical sensor due to massive power drain of the magnetic sensor. The magnetic sensor is woken-up only when the state of the optical sensor is changed. In [7], S. Ma, C. Xu, Y. Wang, F. Li and X. bao conducted their research on reliable wireless vehicle detection using a magnetic sensor and a distance sensor and they suggest a node mode of the magnetic sensor and a distance sensor to detect the vehicles. Magnetic sensors were used in all above researches because they are the best to detect metal objects and vehicles contain a significant amount of metal parts. This paper presents a solution with a magnetic sensor as one part of the vehicle detection node and a secondary sensor, which is selected from distance and optical sensors.

III. Proposed Methodology

The architecture of the solution is given in Fig. 1. This architecture consists of three main sections, Sensor network, Server, and Mobile application. The sensor network consists of vehicle detection sensor nodes and relay nodes. Sensor nodes connect to relay nodes through a wireless communication medium (wifi/MQTT). Relay nodes connect to the server through a wireless communication medium (wifi). Sensor nodes do not connect to the server through the outer network because these nodes designed to be low cost with limited communication capability and relay nodes will be more powerful. The server application will be on the internet and relay nodes will transfer data using the MQTT protocol. The server application will send data about the occupancy of parking areas when requests come from the mobile application.

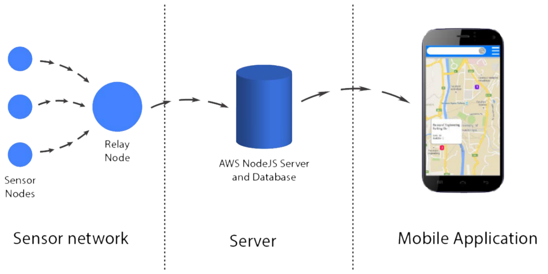

Fig. 1. System architecture

A. Vehicle Detection Sensor Node

The wireless sensor node detects the presence of a vehicle in a predetermined zone (parking space). The sensor node includes a magnetic sensor and a distance sensor. The magnetic sensor that drains a very small current is always on and detects the presence of nearby ferromagnetic material. The distance sensor is woken up by the magnetic sensor to measure the distance to the nearest object over it. Both sensors are directly connected with a microcontroller, which results in a compact, simple vehicle detector.

1) Magnetic Sensor: This sensor detects the changes in the Earth's magnetic field. The change can be happen because of a vehicle, which have significant amount of ferrous metals and their magnetic permeability is much higher than that of the surrounding air and soil. Different types of vehicles have distinct magnetic signatures A given magnetic anomaly can be produced by a small vehicle close to the sensor or a larger vehicle father away.

2) Distance Sensor: Due to the reliability in magnetic sensor readings mentioned above, we propose to add a second sensor. Distance sensor is able to determine the distance information when a vehicle is parked over it. If the magnetic sensor detects a vehicle, the distance sensor wakes up to detect whether the vehicle is over the sensor or not. For this measurement, the distance sensor should be small, low-power, less expensive and have a measurement range of 10cm-100cm. Among Laser, ultrasonic and infrared distance measurement techniques, infrared matches the above requirements well. This sensor can emit modulated light pulse and measure the distance through light transmission time accurately.

B. Relay Node

In the sensor network, sensor nodes sense the presence of vehicles and this data needs to be transmitted to the base station using wireless media (wifi). Due to various factors sensor network faces challenges such as limited power, scalability and connectivity in this task [8]. One of the solutions to overcome these challenges is to use relay nodes, whose job is only to relay data generated by other sensor nodes. It has been shown in the literature [8] that the introduction of relay nodes in sensor networks may result in prolonged lifetime as they can remove some burden from the nodes. The relay nodes may also shorten the transmission distance between a pair of distantly located nodes by acting as a hop between them [9].

C. Server and Mobile Applications

The base station which gathers data is the server application. It acts as the layer between the mobile application and the sensor network. The server application is also capable of storing the data for future analytics. The mobile application has the capability to show where the available parking places are. The application uses an event-driven method to communicate with the server [10][11]. It provides a bi-directional communication channel between a client and the server. That means the server can send data to all connected users at the same time.

IV. Design and Implementation

A. Sensor Node Architecture and Vehicle Detection Algorithm

The sensor node consists of a processing unit, a magnetic sensor, an IR sensor and at power supply. The sensor node architecture is shown in Fig.2

NodeMCU [12] is selected as the experimental processing unit. It is a low-cost development board with ESP8266 unit combining computing power with wifi. It has several sleep modes [13] in order to save power. The magnetic sensor

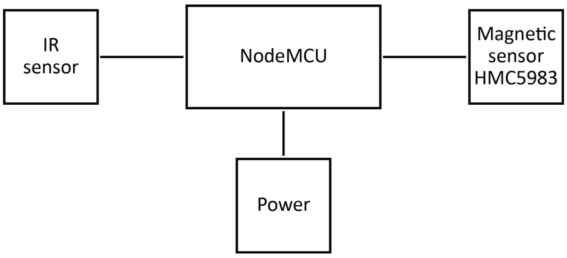

Fig. 2. Sensor node architecture

is HMC5983 [14], which is a sensitive, small and low-cost 3-axis anisotropic magnetoresistance (AMR) sensor with digital output, with a typical sensitivity range of 230 to 1370 LSb/gauss, and a field range of -8 to +8 gauss. The sensor needs extremely low current; 2 μA in idle mode and 100 μA in the measurement mode. The distance sensor is the sharp GP2Y0A21YK0F, low-cost IR sensor, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit. The distance measuring range is 10cm to 80cm and typical current consumption is 30mA, higher than the magnetic sensor but acceptable because it only requires to work several times when woken up by the magnetic sensor and works in 4.5V to 5.5V [15]. Fig. 3 shows the complete circuit diagram of the sensor node.

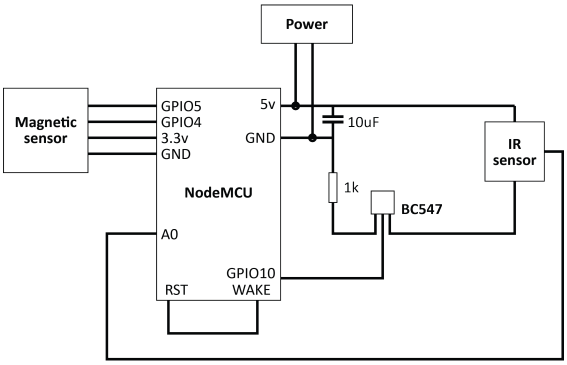

Fig. 3. Wireless vehicle detection sensor node circuit with NodeMCU, magnetic sensor and distance sensor

Vehicle detection algorithm: Fig. 4 shows the detection algorithm in detail. After the sensor node is placed on application site, the first procedure is the system initialization. The sensor node then opens the distance sensor and initialize the initial distance measurement. Then set the (to an interval of 1 minute) deep sleep mode of the NodeMCU. After each minute, the NodeMCU wakes up and measures the magnetic field strength. Then, the current measurement is compared with the previous magnetic field strength.

$$mag\_diff = |mag\_now - mag\_prev|$$

if mag_diff is larger than a predefined threshold_mag, the magnetic field is assumed to have changed. Then NodeMCU powers on the distance sensor to measure the distance for two times. Then the distance is compared with predefined distance threshold threshold_dis (the threshold_dis is between 20cm - 80cm). If the distance is within the threshold_dis, it is assumed that there is a vehicle, otherwise it means there is not. If there is, the node checks whether the variable ”occupancy” is 1. If it is, a vehicle was already there before and no need to transfer data again. If not, the node changes the variable ”occupancy” into 1 and transfer data. If the distance is not within the threshold_dis, it is assumed that there is no vehicle. Then the node checks whether the variable ”occupancy” is 0. If it is, no need of transferring data. But if not, the variable ”occupancy” changes into 0 and transfer data. Otherwise NodeMCU goes into deep sleep for a minute.

The Sensor Network was implemented using MQTT[16] (Message Queuing Telemetry Transport) protocol. This MQTT protocol follows the publisher/subscriber architecture. There is a central unit called the Message Broker in the middle of the publishers and subscribers in the network. As the Message Broker, Mosquito[[17] Message Broker was selected. It is a free and open source product. All the Nodes in the park are publishers in the network while all the mobile app users are subscribers. Subscribers can get the data(sent by the nodes) through the Message broker by listening to the relevant channel name. This Message broker deployed to the AWS Elastic Computer Cloud.

B. Server and Mobile Applications

The Server application was configured as follows. First, the application architecture was designed with a loosely coupled front-end and back-end. This architecture is preferred many developers and communities in node.js application development life cycle. Second, the express.js[18] framework was configured as the main framework in the application. In software development process it is important to have separate environments for development, testing, and production. Separate configurations were implemented for each environment. For server side rendering in the application, ejs[19](embedded JavaScript) template engine was used.

The Elastic Computer Cloud of Amazon Web service was selected to deploy the web application and database. As the database, MongoDB[20] was selected and database connection was implemented using third party tool called Mongoose[21]. Mongoose is a node.js library which is used to ORM (Object Relational Mapping) for MongoDB.

The Real-Time communication part was configures using socket.io[11] middleware. The Server behaves as a subscriber of the mosquito message broker. When a Node publishes data, the broker will send those data to relevant subscribers. Server application also receives those data. After receiving the data, Server will process those data, store them in the database and send them to connected mobile applications using socket.io middleware in real time.

The mobile application of the smart parking sensor network is called ”ParkMe”. It has the capability to show where the available parking places are. This mobile application is made

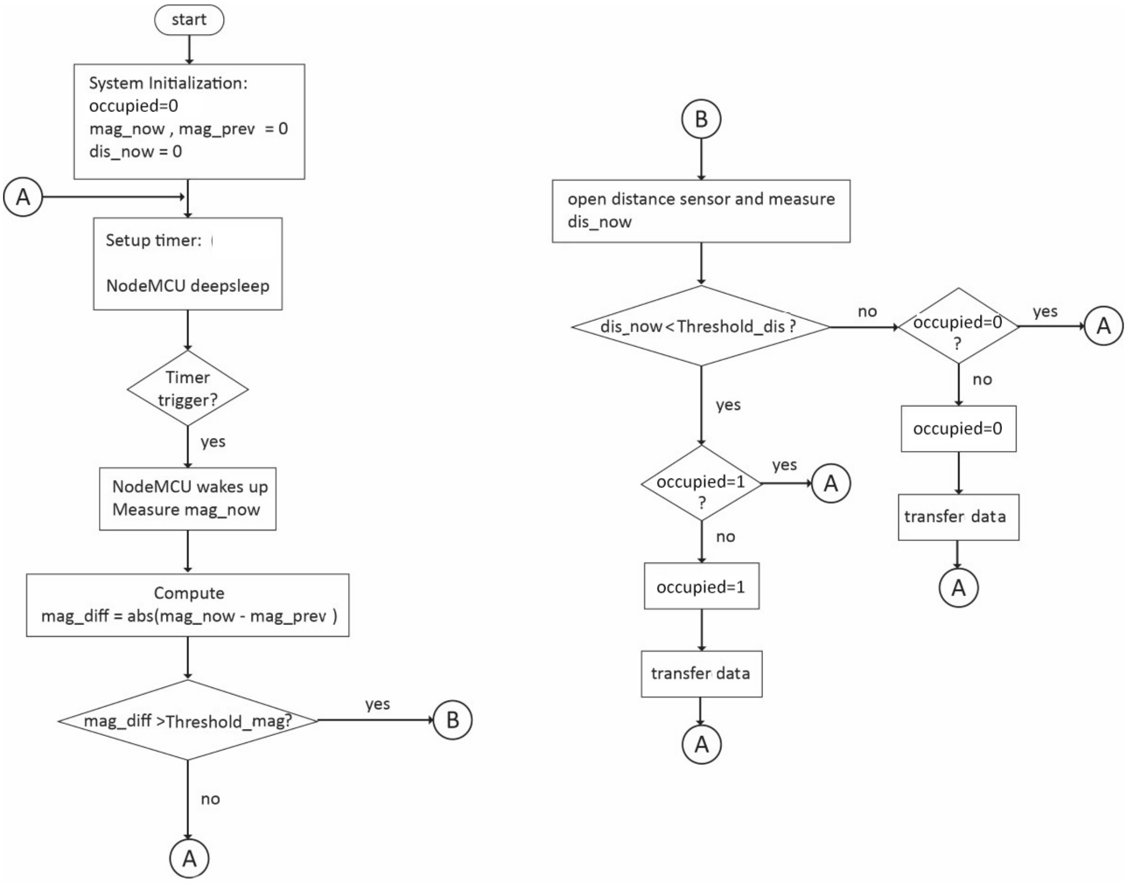

Fig. 4. Vehicle detection algorithm

available for Android users. Using this mobile application, users can check available parking places in the nearest Park, from where the user finds the parking slots and also after clicking the google map markers user can get directions to the selected parking area.

The ”ParkMe” Mobile application is developed as a native android application using Java. The Google Map API is used to show the google map in the mobile application. When using the mobile application user can change google map settings as well as the parking area types. The custom markers are used to show the available number of free parking slots and also two colors are used to show the parking area type which is indoor and outdoor. By clicking the markers, users can look into more details of the parking area, such as the available percentage of parking space of each floor if it is a multi-story park and parking chargers etc. For the communication between the server and the mobile application, the MQTT protocol is used because it is a lightweight protocol which is developed mainly for IoT application. In the mobile application for getting parking area data from the server to client mobile application, the Paho[22] Android Service was used. It is an MQTT client library written in Java for developing applications on Android.

V. Results & Discussion

To facilitate parallel development in all areas of the project the proposed system is tested as separated sections in the architecture. The mobile application was tested by setting up virtual parking areas on the server and sending data with different allocations of the parking areas. Fig.5 shows the parking area visibility on the map.

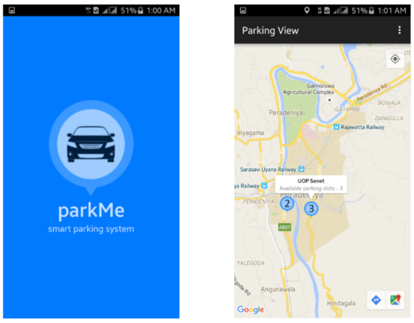

Fig. 5. Mobile application interfaces with parking areas

To test the proposed sensor nodes, we built the sensor node in Fig.3 and the test plan was to place the sensor nodes in the center of parking spaces as in Fig.6. The test was planned to conduct with a car and an SUV in different weather conditions and times of the day.

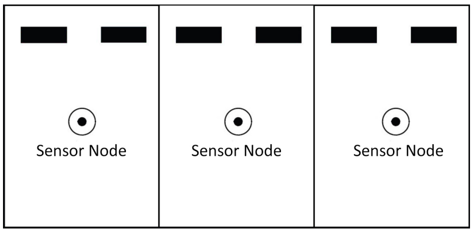

Fig. 6. Sensor node experimental area

As discussed, vehicles in nearby parking space can trigger the magnetic sensor to wake up the distance sensor, which wastes battery life. To minimize these false positives, a predefined threshold should be determined. The following scenarios were planned to carry out. Scenario 1: car enters the middle parking space, stays parked for 5 minutes (deep sleep time of the sensor node is a minute), then leaves. Observe the magnetic sensor behaviour and distance sensor behaviour of three sensor nodes. Scenario 2: SUV repeat the same as scenario 1. Observe the sensor behaviour. Scenario 3: car parks in the first parking space and the SUV enters the third parking space and repeat as scenario 1. This scenario will display the behaviour of the sensor node with different vehicles and the accuracy of the sensor node. The test plan on the sensor and the sensor network was not conducted by the time of this publication.

VI. Conclusion

In this paper, an inexpensive solution was proposed to design and develop a smart parking sensor network with a mobile application, server application, a relay node and magnetic and distance sensor based vehicle detection node. Mobile and server applications were developed. Vehicle detection node design is currently ongoing and node testing scenarios were proposed along with the vehicle detection algorithm. Continuation of this project can be done as the next phase by conducting the test scenarios and adding low-power consumption methods in vehicle detection node.

References

[1] *For Universities: streetline*. URL: http://www.streetline.com/manage-parking/for-universities/.

[2] *Streetline*. URL: http://www.streetline.com.

[3] *About Smart Parking Limited*. URL: http://www.smartparking.com.

[4] David Gascon Alicia Asin. *Smart Parking Sensor Platform enables city motorists save time and fuel*. 27 May 2011. URL: http://www.libelium.com/smart_parking/.

[5] B. Koszeteczky and G. Simon. *Magnetic-based vehicle detection with sensor network*. Tech. rep. Vesprem, Hungary: University of Pannonia, Department of Computer Science and System Technology, 2013.

[6] O. Casas E. Sifuentes and R. Pallas-Areny. *Wireless Magnetic Sensor Node for Vehicle Detection with Optical Wake-Up*. Tech. rep. 2011.

[7] Y. Wang F. Li S. Ma C. Xu and X. bao. *Reliable Wireless Vehicle Detection using Magnetic Sensor and Distance Sensor*. Tech. rep. Shanghai Institute of microsystem and Information technology, Chinese Academy of Science, 2014.

[8] Ataul Bari. *Relay Nodes in Wireless Sensor Networks: A Survey*. Tech. rep. University of Windsor, 2005.

[9] Errol L. Lloyd and Guoliang Xue. *Relay Node Placement in Wireless Sensor Networks*. Tech. rep.

[10] Drifty Co. *ionic framework*. URL: http://ionicframework.com..

[11] *socket.io*. URL: http://socket.io.

[12] *NodeMCU*. URL: nodemcu.com/index_en.html.

[13] *Reduce Power Consumption by Sleep*. URL: https://sourceforge.net/p/nodemcu/wiki/Sleep/.

[14] *HMC5983*. URL: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5983_3_Axis_Compass_IC.pdf.

[15] *Sharp GP2Y0A21YK0F*. URL: http://www.sharpsma.com/webfm_send/1489.

[16] *MQTT*. URL: http://mqtt.org/.

[17] *MOSQUITTO*. URL: https://mosquitto.org/.

[18] *ExpressJS*. URL: https://expressjs.com/.

[19] *EmbeddedJS*. URL: http://www.embeddedjs.com/.

[20] *MongoDB*. URL: https://www.mongodb.com/.

[21] *Mongoose*. URL: http://mongoosejs.com/docs/2.7.x/docs/model-definition.html.

[22] *paho*. URL: https://www.eclipse.org/paho/.