An Intelligent Car Park Management System based on Wireless Sensor Networks

Yuan Zheng, Jiannong Cao, Vanessa W.S. Tang, Yi Lai

Internet and Mobile Computing Lab
Department of Computing,
The Hong Kong Polytechnic University, P.R.China

Abstract

Wireless sensor networks (WSNs) have attracted increasing attentions from both academic and industrial communities. A WSN can be deployed in various kinds of environments to monitor events and collect information. In this paper, we describe a WSN-based intelligent car park system. Low-cost wireless sensor nodes are deployed into a car parking lot, with each parking space equipped with a sensor node that detects and monitors the occupation events of the parking space. The occupation events detected by the sensor nodes are delivered to a database via the deployed wireless sensor network and its gateway connecting to the server. The database can be accesses by the upper layer management system to perform various management functions, such as finding vacant parking spaces, auto-toll, security management, and statistics report. We have implemented a prototype of the system using Crossbow Motes. Testing in a campus car park has been conducted, and the evaluation results demonstrate the effectiveness of the WSN-based system.

Keywords: Pervasive Computing, Wireless Sensor Network, intelligent car parking system.

1. Introduction

In recent years, wireless sensor networks have attracted a great amount of attention [1]. A wireless sensor network consists of a large number of low-cost sensor nodes which can be self-organized to establish an ad hoc network via their wireless transceivers. A sensor node is also equipped with various kinds of sensors, computation units, and storage devices. These functional parts enable sensor nodes to be easily and rapidly deployed to various kinds of environments to cooperatively collect, process, and transmit information.

WSNs have a great potential to be used in future pervasive computing systems. Taking the advantages of sensing and wireless communication, wireless sensor networks have already found many civil and military applications, such as smart home [2], intelligent buildings [3], health-care [4], wild environmental monitoring [5], and battle surveillance [1]. With the approaching of the automobile epoch, the demand on intelligent parking service is expected to grow rapidly in the near future. This emerging service will provide automatic management and high security measures, as well as convenience to the customers.

A few existing studies focused on the applications of car park management system using sensor technologies. For example, the system described in [6][7] adopts cameras to collect the information in car parking lot. However, video sensors have two disadvantages: a video sensor is energetically expensive, and it can generate a very large amount of data which has high cost in energy and bandwidth for transmission in a wireless network. These disadvantages greatly limit the application of video sensors.

In this paper, we describe the design and implementation of an intelligent car park management system based on a low-cost WSN. The system is event based. Wireless sensor nodes are deployed to the parking lot; each node is equipped with light, sound and acoustic sensors. The WSN can monitor and detect the occupation events and cooperatively process and deliver these events to a management system. By using the management system, the human managers and administrators can obtain the events occurring in the car park, including real-time and statistics information. In addition, the management system can alert the illegal mobility of a car in the car park. The system can also record the duration a car parks in the parking lot and automatically do the accounting of the parking fee.

We have built a prototype of the system using the Crossbow Motes [8]. We have deployed the prototype system in a university campus car park. The testing results show that our prototype system can achieve 100% accuracy and, meanwhile, a low delay of event reporting.

The remaining part of this paper is organized as follows. Section 2 briefly describes the related works. Section 3 analyzes the requirements of an intelligent car parking management system. Section 4 discusses the generic architecture of a WSN-based application system. Section 5 describes the implementation of our system, including hardware platform, system architecture, software architecture, and the event-driven processing mechanism. Section 6 describes our experiments with the prototype system and reports the testing results. Finally, Section 7 concludes the paper with a discussion of our future work.

2. Related Work

Developing WSN-based applications for transportation management faces many challenging issues, such as high speed mobility detection and prediction, mobile object identification and tracking, etc. Several systems have been developed to deal with different aspects. In this section, we briefly review some of these systems related to our work.

Irisnet [6][7] proposed a wide-area architecture for pervasive sensing networks used by drivers to retrieve the information about car parking spaces via wireless access. In this system, video cameras (Webcams), microphones, and motion detectors are employed to detect and recognize the automobiles in the parking lot. The sensory data, such as parking field images captured by Webcams, is processed in network rather than only at the backend server. The processed data will be published on the web, and the user can acquire the interesting information by using the web access technologies. However, as we have mentioned in the first section, the video cameras will generate a large amount of data. The transmission and process of these data will consume a great deal of resources, including communication bandwidth, processing cycles, and energy, which are very limited in a wireless sensor network.

Traffic Plus Technology [9] and MIT Intelligent Transportation System [10][11] are also transportation applications based on WSNs. Automobile sensors are deployed on both sides of a road, as well as embedded into a road bed, to detect the relevant information about automobiles. Although both systems can be effective for traffic and road condition monitoring, they are not designed for car park management. In addition, the hardware used in the systems is quite expensive and complicated.

An important problem in designing a car park management and transportation system is how to accurately detect the mobility of automobiles, especially when the vehicles move at a high speed. Magnetic sensors have been employed [12], but operations on this kind of sensors are energy intensive [13]. The widespread deployment of these sensors is still a challenging problem in the wireless sensor networks with the constraint on energy consumption.

3. Requirement Analysis

In this section, we discuss the requirements of designing a WSN-based intelligent car park management system. Although the requirements of a conventional car park management system can be easily satisfied, we still need to address the challenging issues raised by wireless sensor networks.

From the business consideration, the common goal for all car park management systems is to attract more drivers to use their facilities. Thus, the basic facilities in the systems need to fulfill the following conventional requirements:

- The location of the car park should be easy to find in the street networks.
- The entrance of the car park should be easy to discover.
- The number of parking spaces should be abundant and a parking space should be large enough for a car to park in.

However, an intelligent car park system should provide more convenience and automation to both the business and customers. It should minimize human interventions, so as to effectively reduce the cost of manpower and the lost resulting from human mistake. More specifically, it should satisfy the following requirements:

- The system should provide informative instructions and guidelines to help drivers find an available parking space.
- The system should provide effective security measures to prevent the cars from being stolen or damaged.
- The system should provide drivers appropriate automatic payment methods.
- The system should provide functions to facilitate administrators and managers to manage a car park.
- The system should provide higher accuracy, reliability, and flexibility in operations, and low cost of maintenance.

Wireless sensor networks are a promising technology for building the infrastructure of a car park management system that satisfies the above requirements. They have the advantages of rapid deployment, on-site sensing and processing, wireless communication, and low-cost maintenance. These advantages allow us to develop effective system solutions as required.

4. Architecture of WSN-based Applications

Although WSN-based systems are application dependent, they have similar architectures and share some common system components. In this section, we describe the generic architecture of a WSN-based application system and the corresponding software architecture.

An application system based on wireless sensor networks typically adopts a 3-layer architecture, as shown in Figure 1. The first layer is the layer of wireless sensor networks, where the major hardware components are distributed nodes. The second layer is the Server layer. The server receives the collected sensory data from the sink node (gateway) via serial port, Ethernet, or wireless link and stores these data for real-time analysis or later use. The third layer is the application layer, which mainly consists of the application servers and client programs with user interfaces.

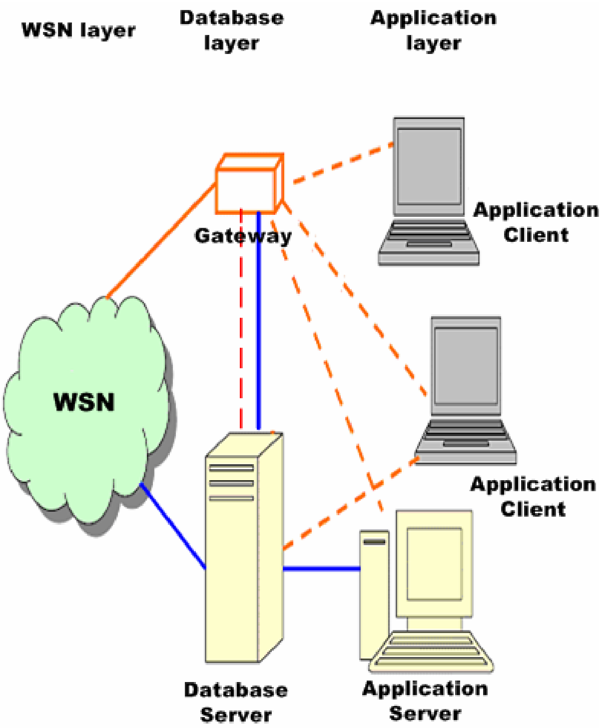


Figure 1. 3-layer architecture of a WSN-based application system

Accordingly, a WSN-based application system also has a 3-layer software architecture, as shown in Figure 2. In the WSN layer, Sensor nodes are programmed with operating system and application tasks, which mainly contains the sensor drivers and networking protocol stack for data sensing and transmission, respectively. In the database layer, the software is developed for sensory data logging, filtering, transforming, etc. The software at the application layer provides tools for visualization, monitoring, and analysis. The tools are used to display and interpret sensory data. Also, the software at this layer can process the sensory data according to the specific application business logic.

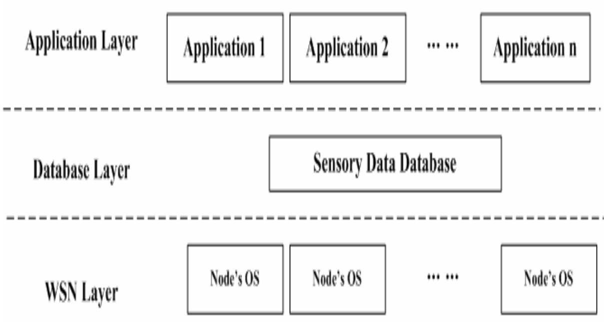


Figure 2. 3-layer software architecture of a WSN-based application system

5. System Implementation

In this section, we describe the implementation of our intelligent car park management system. First, we introduce the hardware platform. Second, we discuss the 4-layer system architecture, which is based on an event-drive model and is an extension of the 3-layer generic architecture described in the last section. We also introduce the functional modules in the system. Finally, we describe the implementation of the event detection and processing mechanisms in our system.

5.1. Hardware Platform

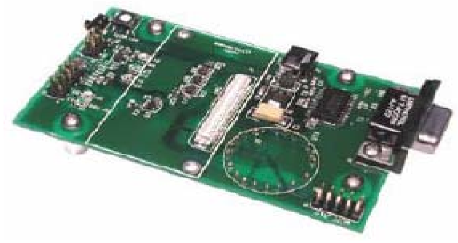
In our system, the wireless sensor nodes and the gateway used as the underlying hardware infrastructure are Berkeley's Motes and the relevant computer devices from Crossbow Technology Inc..

A Mote is composed of two parts, a Mote processor radio board (MPR) and a Mote sensor board (MTS). An MPR consists of a processor and a wireless transceiver. The Motes are battery-powered and pre-loaded the open-source TinyOS [14] operating system which provides the applications development and operation platform and the stack of physical, media access control, and routing layer protocols.

Our system uses the Mote of MPR2400 (see Figure 3(a)). The Motes have routing capability and can communicate wirelessly with each other in an ad hoc, multi-hop manner. They operate compatibly with IEEE802.15.4 and can be extended to connect to different sensor boards.

The MTSs can directly mate to MPR2400. The MTS boards employed, namely MTS300 (see Figure 3(b)), is equipped with the light, temperature, and acoustic sensors and a sounder. In practical systems, the combination of MPR2400 and MTS300 provides a full functional sensor node. In our car park management system, we use light sensors to monitor the statuses of parking spaces, because the car "move-in" and "move-out" will change the lightness of the parking spaces.

(a) MPR2400 **(b) MTS300**



(c) MIB510

Figure 3. Crossbow Mote products[8]

The sink node used in the Mote-based WSNs consists of two parts, a Mote and a Mote interface board (MIB). The MIB, MIB510CA (see Figure 3(c)), can attach to a Mote. It also possesses a standard RS232 serial port. Thus, the sink node can receive the sensory data from other Motes via wireless communication and can also forward the sensory data to a database server via wired serial communication.

However, directly deploying the Motes to the parking spaces may face some problems. First, the Motes are likely to be smashed by the moving cars. Second, the wireless communications among the sensor nodes tend to fail due to the hindering of the cars' metal bodies. Third, the condensed water droppings generated by water-cooling systems of cars' engines will damage the circuits of the sensor nodes. To tackle the problems, we attached an additional light sensor to every Mote via a wire, as shown in Figure 4, and place the attached light sensor in the park space. The Motes are installed in safe places, for example, pillars or walls of the car park.

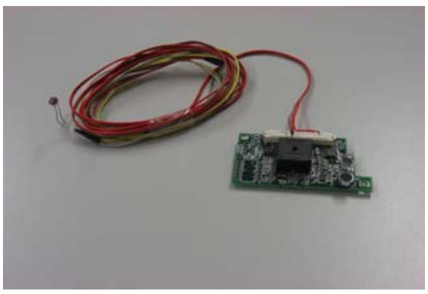


Figure 4. an adapted MTS300

5.2. System Architecture

As shown in Figure 5, our system adopts an event-driven architecture consisting of 4-layers, namely the event detection layer, the event delivery layer, the storage layer, and the presentation layer. At the event detection layer, the Motes use their sensors to monitor the changes of lightness and detect the occurrences of parking events, including car "move-in" and "move-out" events. At the event delivery layer, the deployed Motes are self-organized to form an ad hoc network and cooperatively and reliably deliver the detected events to the sink node. At the event storage layer, a database server records the events reported from the sink node. The presentation layer has the user interfaces showing information updated according to the newly coming events in the database server. It also provides statistics reports to facilitate the management of the car park.

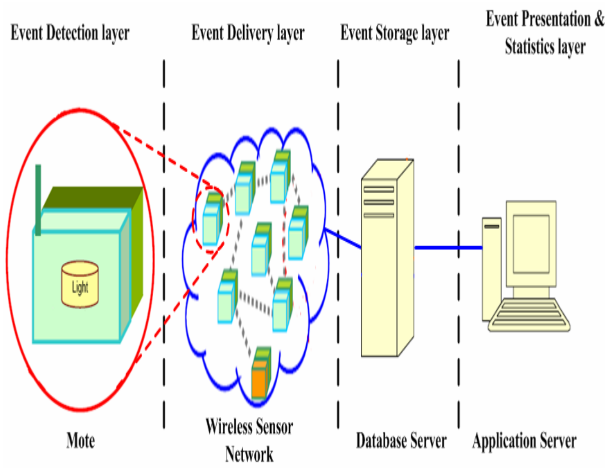


Figure 5. 4-layer event-driven architecture of our system

This 4-layer architecture can effectively decouple the upper layer applications from the underlying WSNs by using the event-driven mechanisms. Thus, the change in the underlying WSNs will not cause problems to the upper layer application systems. Also, the extension to the functionality of the upper layer applications need not to consider the details and will not affect the operations of the underlying WSNs.

In the event-driven system architecture, the software of our system is also divided into 4 layers, as shown in Figure 6.

In the bottom layer, based on the TinyOS operating system, we developed the on-site event detection algorithms on Motes. In the event delivery layer, we developed a reliable event delivery protocol based on Mint routing protocol [15] and B-MAC media access control protocol [16], which have already implemented in the TinyOS operating system. The event storage layer uses the Postgres [17] database system, where the events from Motes are installed by a daemon program at the serial port, namely SerialForwarder [18]. At the top layer, we developed four functional modules:

1) Parking space management module which presents the information about the occupation of parking spaces.

2) Auto toll module which manages the payment of parking fee.

3) Security management module which alerts the abnormal leaving of cars in the parking spaces.

4) Statistics and reporting module which generates reports to help managers or administrators understand the business status of the whole car park.

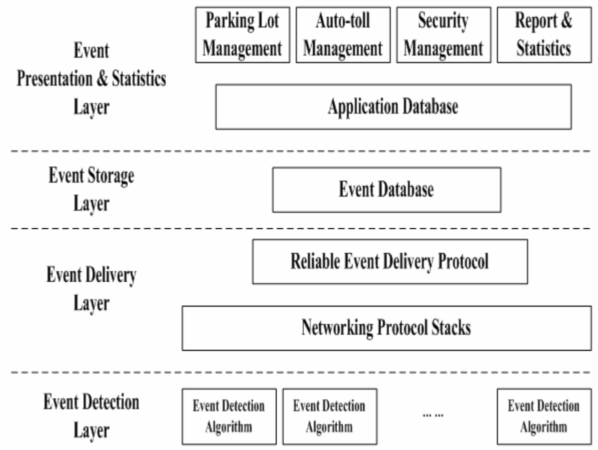


Figure 6. software architecture of our system

5.3. Event-driven Processing

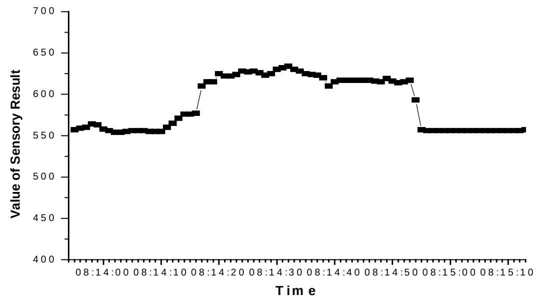
The event-driven model is widely accepted for developing many WSN-based applications because WSNs are a kind of resource-constrained networks [1]. Event-driven mechanisms can take the advantages of on-site computation and processing of sensor nodes to avoid frequent transmission of sensory data, thus save the resources in WSNs. In this subsection, we describe event processing in our system, which includes event detection, delivery, storage, and statistics. Since the event storage just uses a set of typical database operations, we will focus on event detection, delivery, and presentation and statistics.

5.3.1. Event Detection

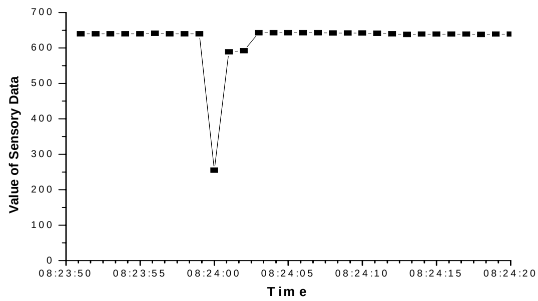
In our intelligent car park management system, there are two major kinds of events: car “move-in” and “move-out”. However, the detection of these two events is not trivial. A node can accurately determine the occurrence of an event only if two conditions are satisfied.

The first condition is that the change in the average sensory data value before and after the event should be sufficiently large. A threshold is used to detect such a change. The second condition is that the change in the average sensory data value should last for a period of time. For example, a sensor node can determine the occurrence of car “move-in” or “move-out” events only if a certain number of consecutive average values significantly decrease or increase over the pre-defined threshold. According to the testing in the campus car park of our university, the threshold should be set no less than 50.

The length of the period for which the new average value lasts cannot be easily determined, especially in the following two scenarios. First, a car often moves to its parking space by passing the parking spaces nearby. This can result in a significant decrease of the sensory data value collected by the nodes in the passed spaces. Second, the headlight or tail lamps of cars often cause the increase in the sensory data values on the nodes in the surrounding spaces.



(a)



(b)

Figure 7. the sensor data continuously generated by two sensor nodes

Figure 7 shows the sensory data continuously generated by two sensor nodes deployed in the campus car park of our university. According to the testing record, the variation in sensory data values shown in Figure 7(a) is caused by the first scenario, while the variation shown in Figure 7(b) is caused by the second one. The first scenario can result in false detection of car “move-in” events, whereas the second scenario can result in false detection of car “move-out” events. To avoid the interferences that can occur in the two scenarios, we use a “temporal window” to filter the “noises”. According to our testing results, such kind of noises never last for more than 30 seconds. Thus, we set the window size for event detection to be 60 seconds, which is the double of the size of the “noise window”.

We let each sensor node perform one sensing operation in each second. In every 10 seconds, the sensor node calculates the average of the 10 values it has obtained. Each of the average sensory data is compared against the previous one. If a node detects a variation greater than the threshold value of 50 and lasting for more than 60 seconds, the node can determine the occurrence of an event. If the sensory data change from high value to low value, the event is a "move-in" event; otherwise, the event is a "move-out" event. Once a sensor node detects an event, it will transmit this event and its node ID to the sink node using the reliable event delivery protocol which is introduced next.

5.3.2 Reliable Event Delivery

In our system, every sensor node just reports the detected car "move-in" and "move-out" events but not the sensory data collected. This event-driven mechanism can effectively reduce wireless communication so as to save the resources in the network. However, the low capability of wireless communication of sensor nodes makes it unreliable to deliver events [1]. Furthermore, the mobility of cars can also considerably affect the packet delivery in the wireless sensor networks.

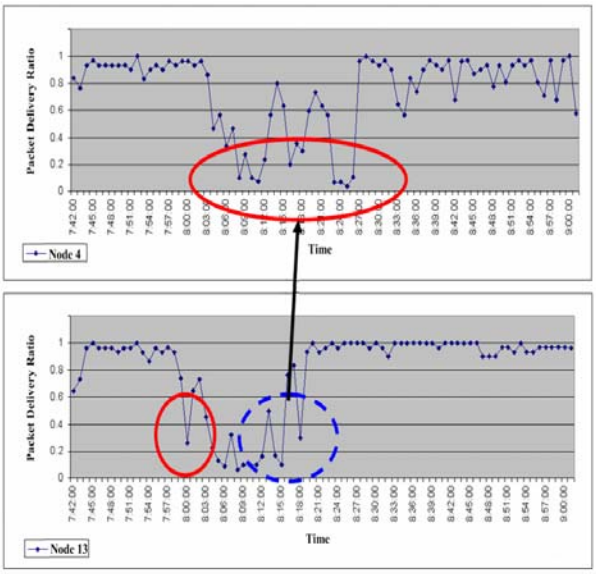


Figure 8. packet delivery ratio of two nodes without using reliable event delivery protocol

Figure 8 shows the test results of packet delivery without using reliable event delivery protocols. In the test, every node sends a packet every second and the packet delivery ratio (PDR) is measured for each node. PDR is the ratio between the number of the packets successfully delivered and the total number of packets transmitted by a node within one minute time. Curves inside the red circles show that a bypassing car can greatly affect the PDRs of the two nodes. The circle with blue dashed line shows that the drop in PDR at Node 4 leads to a great decrease in the PDR at a downstream node, Node 13. In a practical system, event delivery failures can directly invalidate the operations at the event presentation and statistics layer.

To tackle the problem, we have developed a reliable event delivery protocol to solve the problem of event delivery failures. In our protocol, every event has a unique ID, which is composed of the ID of the event's source node and a sequence number assigned by the source node. The sequence number is increased by one when the source node sends out a new event. Using the unique event ID, every node on the transmission path can uniquely identify an event and provide the reliable event delivery using hop-by-hop acknowledgement. The pseudo-code of the protocol is given in Figure 9.

Basically, the protocol maintains an event queue and a corresponding acknowledgement queue, which temporarily buffer the events and acknowledgement waiting to be sent. The protocol uses a timer to periodically trigger the sending operations. A buffered event will not be discarded until its corresponding acknowledgement is received.

Reliable event delivery protocol:

1. Initialization:

Create an event queue, namely *E_QUE*;
Create an ack queue, namely *A_QUE*;
Start up a re-transmission timer;

2. When receiving an event, namely *E*:

If $E \notin$ *E_QUE*
Then
Append *E* to *E_QUE*;
Generate ACK_E corresponding to *E*;
Append ACK_E to *A_QUE*;
Else
Discard *E*;
EndIf.

3. When the re-transmission timer expiring:

Send out *E_QUE.first_event*;
Remove *E_QUE.first_event* from *E_QUE*;
Send out *A_QUE.first_ack*;
Remove *E_QUE.first_ack* from *A_QUE*;

4. When receiving an ack, namely *A*:

If *A* is corresponding to an event *E*, s.t., $E \in$ *E_QUE*
Then
Remove *E* from *E_QUE*;
Else
Discard A;
EndIf

Figure 9. the reliable event delivery protocol

Using hop-by-hop acknowledgement could increase communication overhead, but in the context of resource-constrained WSNs, however, we believe that this approach is still a better choice, comparing with end-to-end acknowledgement or transmission via multiple disjointed paths.

5.3.3 Event Presentation and Statistics

In the event presentation and statistics layer, there are four major types of events, namely *timer*, *car-in*, *car-out*, and *Field Management Events.*

- ***Timer event.*** This kind of events is generated by the system clock and is used to periodically retrieve the new events stored in the event database.
- ***Car-In Events***. This kind of events indicates that cars have checked into the system.
- ***Car-Out Events***. This kind of events indicates that cars have checked out from the system.
- ***Field Management Events***. This kind of events is issued when a manager of the car park performs the management operations.

These events trigger the operations in the system. For example, as shown in Figure 10, the operations triggered by "car-in" event include updating car record database, assigning a nearest available parking space, and confirming this car parked, while the operations triggered by "car-out" include recording the parking time of the leaving car, calculating parking fee.

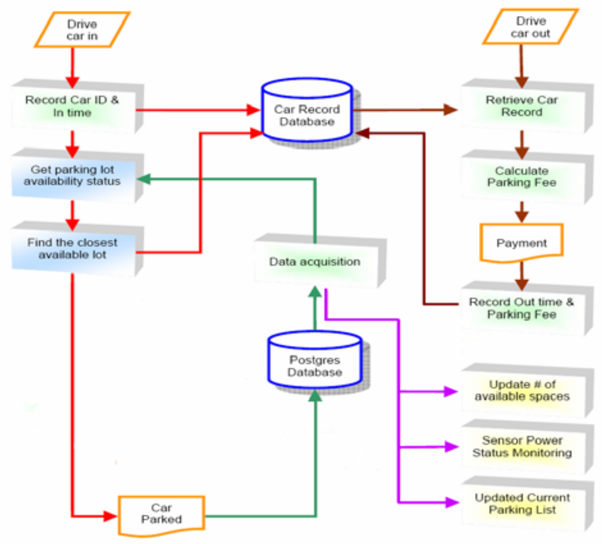


Figure 10. the flowchart of the system operations for car move-in and move-out

6. Experiments and System Evaluation

We have deveoped a demo of our intelligent car park management system. Figure 11 shows the user interface of the demo. We used remote-controlled toy cars to validate the functions of our system (see Figure 12.)

We have also deployed our system in the campus car park of our university to test the system performance in real world scenarios. Figures 13 and 14 show the floor plan of the campus car park of our university and the corresponding customized user interface. Figure 15 shows the deployment of the sensor nodes in the campus car park. In our experiments, we deployed 28 sensor nodes, each node is attached to a pillar. To reduce the intervention of the metal car bodies, all the nodes are deployed at the higher ends of the pillars. We conducted the experiments with different window sizes of 20, 30, 40, 50, 60, 70, and 80 seconds and threshold values of the window size 20, 30, 40, and 50 for event detection on these nodes.

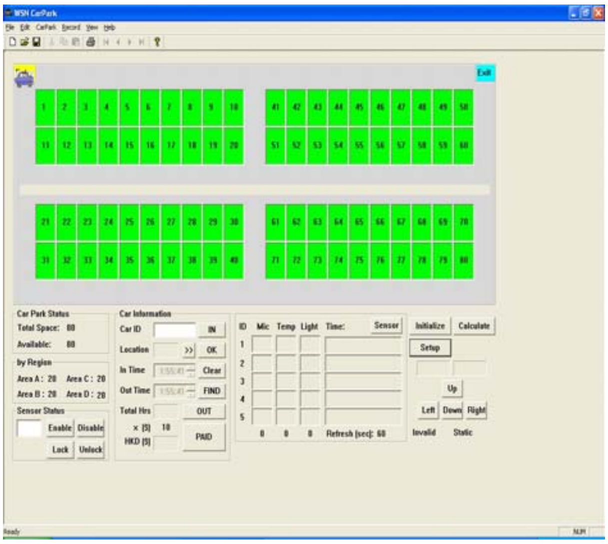


Figure 11. the user interface of our demo system

Figure 12. the testing scenario of our demo system

Two metrics are used to evaluate the performance of our system. One metrics is *event detection accuracy* which is defined as the ratio between the number of correct event reports and the total number of event reports generated. The second metric is the *average event detection delay* which measures the duration from a car "move-in" or "move-out" to the event report received by the event database server.

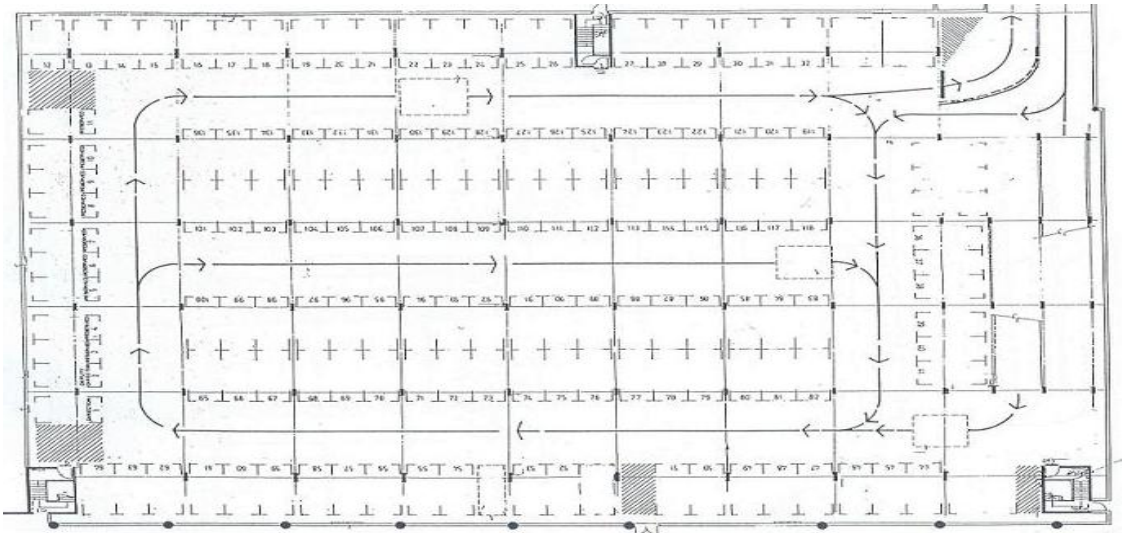


Figure 13. the floor plan of the campus car park in our university

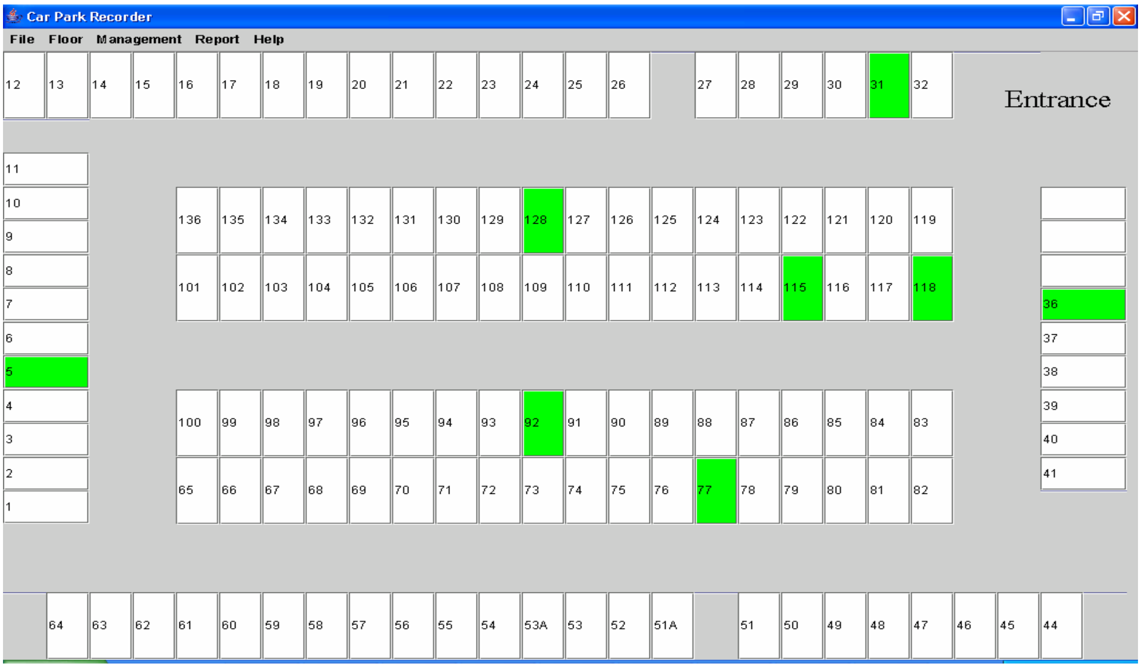


Figure 14. the customized user interface

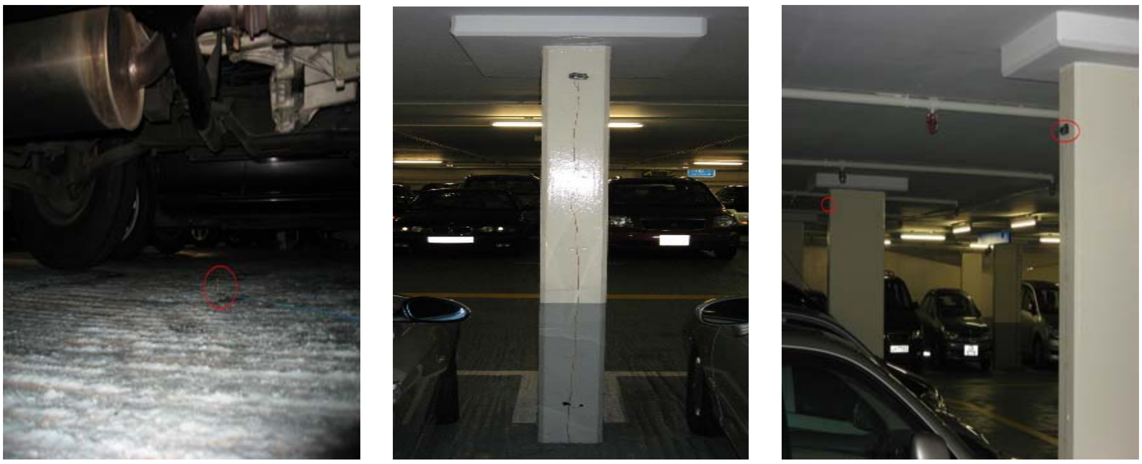


Figure 15. the deployment of our WSN

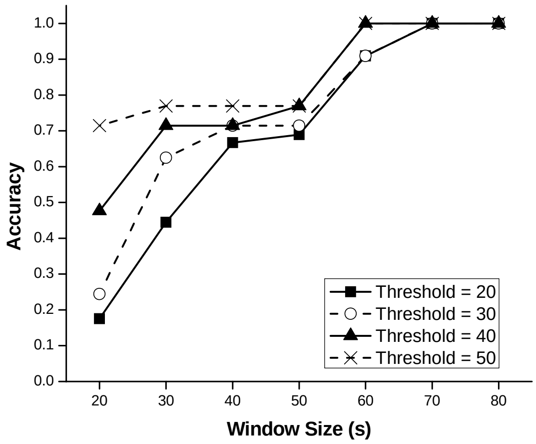


Figure 16 accuracy of our system

From Figure 16, we can see that the accuracy is very low, i.e., sensor nodes tend to generate quite a lot of false event reports, when window size is small or the threshold is low. However, the accuracy can be obviously improved when we increase the window size and threshold. The accuracy reaches 100% when the window size is larger than 60 seconds and the threshold is 40 or 50.

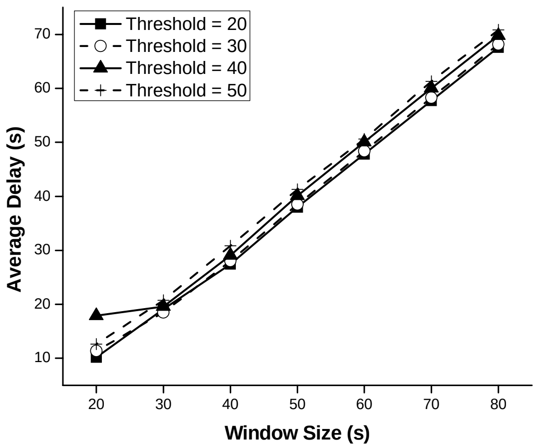


Figure 17 average delay of our system

As shown in Figure 17, the value of average delay is almost linearly increased along with the window size. More specifically, every 10-second increase in window size resulted in almost the same amount of change in delay. This figure shows that the delay in the system is stable and not much affected by the network traffic load or the mobility of cars. Considering both figures, we can find that the minimum average delay for our system to achieve 100% accuracy is no larger than 70 seconds, which is reasonable to the requirements of a real car park management system.

In our experiments, we manually record the time of the duration between car “move-in” or “move-out”. We did not put down any time record until the car is completely settled down. The sensor node, however, has already begun to detect the event before the car is completely settled down. Therefore, as shown in Figure 17, the average delays are usually smaller than the corresponding window sizes.

7. Conclusions

In this paper, we described an intelligent car park management system based on a wireless sensor network. We analyzed the requirements of real car park management systems. Based on the analysis, we proposed the main system functionalities and designed the system architecture. We also implemented a prototype system to realize the designed functionalities by using the adapted crossbow products of Motes. We deployed our prototype system in the campus car park of our university. The testing results demonstrate that our system is able to achieve 100% accuracy and, meanwhile, a low delay of event reporting. Therefore, our system can effectively satisfy the requirements of a real intelligent car park management system.

We believe that wireless sensor networks can be a very promising technology for building the infrastructure of a car park management system. We will improve our existing work in the following aspects.

First, we will establish an information disseminating platform for this system to publish the interesting and helpful information to the users of the car park management system.

Second, we will combine the car park management system with our on-going projects of mobile agent-based mobile computing platform and intelligent transportation system to provide the users more effective and efficient transportation services based on pervasive computing technologies.

Acknowledgement

This work is supported by Hong Kong Polytechnic University under the ICRG grant A-PF77 and *the large equipment fund G.61.27.D01B.*

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. *A survey on sensor networks*. In IEEE Communication Magazine, 40(8): 102-114, 2002.

[2] http://www.intel.com/pressroom/archive/releases /20040316corp.htm

[3] J. P. Lynch, K. Loh. *A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring.* Shock and Vibration Digest, Sage Publications, 38(2):91-128, 2005.

[4] http://www.soumu.go.jp

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson. *Wireless sensor networks for habitat monitoring.* In ACM international workshop on Wireless sensor networks

and applications, September 2002
[6] http://www.intel.com/technology/techresearch/research/rs11031.htm
[7] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, S. Seshan. IrisNet: an architecture for a worldwide sensor Web. IEEE Pervasive Computing, 2(4):22 - 33, 2003
[8] .http://www.xbow.com
[9] http://corporate.traffic.com/ntdc/
[10] http://mtlweb.mit.edu/researchgroups/itrc/
[11] http://www.media.mit.edu/resenv/vehicles.html
[12] S.-Y. Cheung, S. Coleri Ergen and P. Varaiya. *Traffic surveillance with wireless magnetic sensors.* In 12th ITS World Congress, Nov. 2005.
[13] G. Anastasi, M. Conti, E. Gregori, A Falchi, A. Passarella. *Performance Measurements of Mote Sensor Networks.* In ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile System (MSWiM), October, 2004.
[14] http://www.tinyos.net
[15] A. Woo, T. Tong, and D. Culler. *Taming the Underlying Challenges of Reliable Multhop Routing in Sensor Networks.* In ACM International Conference on Embedded Networked Sensor Systems(Sensys), November, 2003.
[16] J. Polastre, J. Hill, and D. Culler. *Versatile Low Power Media Access for Wireless Sensor Networks.* In ACM International Conference on Embedded Networked Sensor Systems(Sensys), November, 2004.
[17] http://www.postgresql.org/
[18] http://www.xbow.com/Support/Support_pdf_files/Getting_Started_Guide.pdf