A Parking Management System Based on Wireless Sensor Network[1)]

BI Yan-Zhong[1,2] SUN Li-Min[1] ZHU Hong-Song[1] YAN Ting-Xin[1,2] LUO Zheng-Jun[1,2]

[1](*Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

[2](*Graduate School of Chinese Academy of Sciences, Beijing* 100039)

(E-mail: yanzhong02@ios.cn, sunlimin@ios.cn)

Abstract This paper introduces a parking management system based on a wireless sensor network developed by our group. The system consists of a large amount of parking space monitoring nodes, a few parking guiding nodes, a sink node and a management station. All the nodes exchange information with each other through wireless communication. The prototype of the parking management system has been implemented and the preliminary test results show that the performance of the system can satisfy the requirements of the application.

Key words Wireless sensor network, application, parking management, parking guidance

1 Introduction

Wireless sensor network usually consists of a large number of low cost sensor nodes that are deployed in sensing area. These nodes can sense, sample and process information gathered from their surroundings. They form an *ad hoc* network to exchange information and transfer data to remote servers in a hop-by-hop manner. Wireless sensor network can be applied to many fields such as military affairs, environment monitoring, city traffic control, warehouse management, smart home, *etc.*

This paper introduces a parking management system based on a wireless sensor network, which has several advantages over both deployment and maintenance, compared with traditional parking management systems that are usually based on RS-485 bus. Furthermore, most of the nowaday parking management systems only have control at the entrance and exit of the park, while they do not deal with the management of parking spaces or vehicle guidance. Although some parking management systems adopt cashless parking payment scheme to improve efficiency, they ignore the demand of drivers to park their cars quickly and safely. Drivers can enter the parking lot without stopping their cars, but may spend much more time on looking for idle parking spaces. However, making use of the sensing ability of the nodes, the parking management system based on a wireless sensor network is capable of monitoring and managing each parking space and providing particular guiding services.

In the parking management system we developed, a sensor network is deployed in the parking lot. Three kinds of sensor nodes are installed, which are parking space monitoring nodes, parking guiding nodes and a sink node. The monitoring node is installed near each parking space to detect its occupied status. Guiding nodes control the LEDs display on the turnoffs of the road to help drivers finding an idle parking space. Sink node takes charge of collecting data from the whole network and transmitting the data to a management station, which is set in the monitoring room of the park. All of these nodes communicate through wireless channel, and self-organize into an *ad hoc* network. According to the topological structure of the parking lot and the occupied status of each parking space, the management station computes the guiding information and transmits it to proper guiding nodes.

The rest of the paper is organized as follows. In Section 2, we introduce the system architecture of the parking management system. In Section 3, we give some detailed explanations of the sensor network technologies in our system. Section 4 describes the mechanism of the guiding system. In Section 5, we present the software framework of the management station. Then, we give some preliminary test results and analyze the performance of the system in Section 6. Finally, we conclude the paper in Section 7.

2 System architecture

The entrance and exit of the parking lot in our system are similar to the traditional ones. A display screen at the entrance tells the number of remaining idle parking spaces. A management station, which runs parking management software, is set in the monitoring room. The wireless sensor network in the parking lot contains three kinds of sensor nodes, which are monitoring nodes, guiding nodes and sink node. Monitoring nodes are deployed upon or aside every parking space. Guiding nodes are deployed

1) Supported by National Natural Science Foundation of P. R. China (60373049), National Basic Research Program of P.R.China (2006CB 3030000)
Received November 11, 2005; in revised form March 10, 2006

at main turnoffs of the roads. In addition, sink node is in the monitoring room. The overview of the wireless sensor network in our system is shown in Fig. 1.

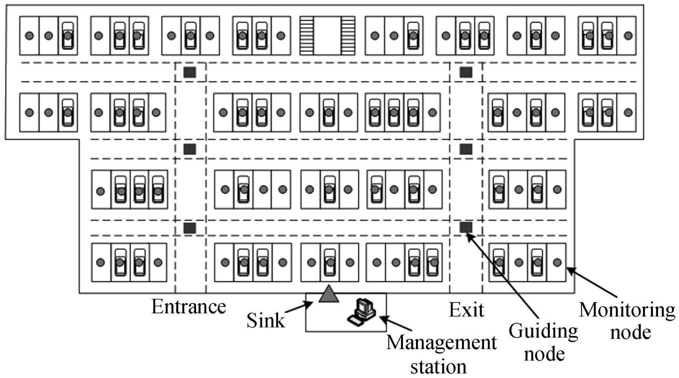

Fig. 1 The overview of the wireless sensor network inside the parking management system

The functions of the devices in the parking management system are as follows.

1) Monitoring node

The monitoring node detects the status of parking space with ultrasonic sensor and transmits or receives messages through RF communication module. It also receives commands from the management station to carry out some procedures such as time synchronizing, debugging, working status reporting and so on.

2) Guiding node

Equipped with an RF communication module and a display module, the guiding node receives guiding information from the management station and shows it on the LEDs display. This can help guiding vehicles to find idle parking spaces with less time. Furthermore, it can also transmit report messages and receive commands like monitoring nodes.

3) Sink node

The sink node collects parking status reports and delivers them to the management station. It acts as a gateway between wireless sensor network and networks outside. In our system, the sink node connects directly to the management station through an RS-232 interface.

4) Management station

The management station takes charge of managing and maintaining the whole system. It processes the data received from the sink node, counts parking fees, and displays necessary information on the monitor. The management station sends parking-space guiding messages to guiding nodes and updates the display screen at the entrance of the parking lot in time.

When the system is running, all sensor nodes form a tree-like topology autonomously for data gathering. The monitoring nodes check the availability of each parking space and transmit the report messages to the sink node hop by hop. The sink node collects the report messages and delivers them to the management station, and then user can get the visual status information of the whole parking lot on his monitor screen. The management station calculates the guiding information for each guiding node and sends the guiding messages to them through the sink node. The guiding nodes will receive the messages and display the guiding indication.

When a newcoming car parks in a parking space, the monitoring node would find that the parking space is occupied in a short time and send a report message to the sink node. Other nodes on its way to the sink node would forward the message as soon as possible. After receiving the report, the sink node would notify the management station of the change. Consequently, the management station would re-calculate the guiding information and send it to the proper guiding node.

3 Parking monitoring nodes

3.1 Hardware architecture

Fig. 2 shows the hardware architecture of sensor nodes on the left and an actual photograph on the right. The sensor node used for parking space monitoring is a micro device equipped with a microprocessor, a wireless communication module and a sensor for detection. The microprocessor is ATmega128L[1], which is an 8-bit MCU working on a 7.3728 MHz system clock and has a 128K bytes in-system programmable flash memory, a 4K bytes RAM and a 4K bytes EEPROM on the chip. The processor provides three interfaces, which are SPI, ADC and GPIO, and uses them to communicate with CC1000, ultrasonic sensor and temperature sensor separately. The sensor nodes also support both standard parallel and serial communication interfaces, thus we can use parallel interface for nodes programming and serial interface for debugging. The wireless communication module we used is CC1000[2], which is a low power RF module and can provide a 19.2Kbps communication speed of Manchester encoded data. The sensing module consists of an ultrasonic sensor and a temperature sensor.

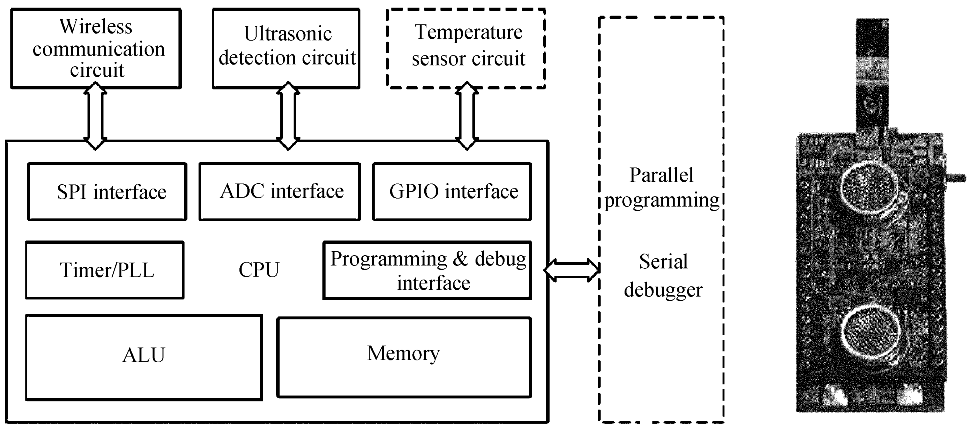

Fig. 2 Hardware architecture of parking monitoring nodes

3.2 Software architecture

The software in our sensor nodes is based on TinyOS[3], a widely used operating system for sensor networks developed by UCB. Fig. 3 shows the overall software architecture. The software system consists of two layers, which are application layer and system layer. The application layer has a serial of modules such as vehicle detection, data collection, routing and time synchronization to provide necessary services for parking place monitoring. The system layer contains some supporting modules for applications, such as MAC control, data dispatching and forwarding, local debugger, *etc.* The significance of extracting some functions to form a system layer is that these modules can be ported to other different applications easily.

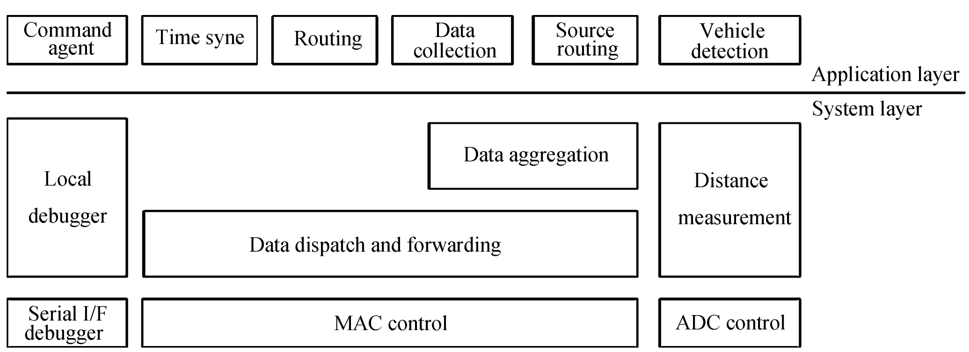

Fig. 3 Software architecture of parking monitoring nodes

3.2.1 Vehicle-detecting module

The function of vehicle detecting module is to read the measuring result of the ultrasonic sensor and provide the occupation information of the parking space with certain algorithms. Meanwhile, this module maintains a state machine to be aware of and records the state-switching events of the parking space it monitors.

The main principle of ultrasonic-based distance measurement is to record the time interval between sending an ultrasonic signal and receiving its echo, and then calculate the distance between the sensor node and the obstacle. As the external temperature has an enormous influence on the precision of distance measurement, we use an additional temperature sensor to give some feedback.

When vehicle-detecting module finds a state-switching event, it performs a checking algorithm periodically and reports the results to the management station.

3.2.2 Routing module

The routing protocol module provides a multi-hop data delivery mechanism for all the nodes in the network to transfer their data to the management station. We designed a novel routing protocol that can construct a cluster-based data aggregation tree to achieve such purpose. It has two main steps, topology construction and data transferring.

Fig. 4 presents an example of the network adopting our routing protocol. The topology construction process starts from the sink node by broadcasting a topology constructing command that contains the fields of node ID, hop count, the sink node and parent node ID. Each node will adjust its parent node as well as its own hop count when receiving this command and then rebroadcast it. Thus, the command will reach out to the entire network to form a hierarchical topology.

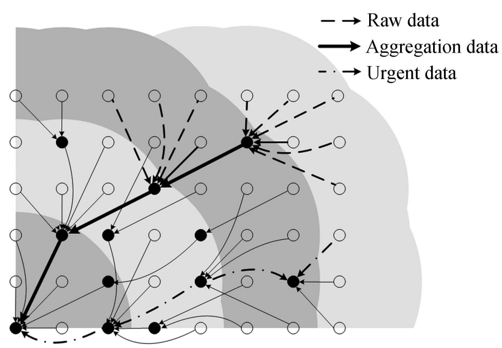

Fig. 4 Routing process of the aggregation tree

During the topology construction process, we encourage the adjacent nodes to choose the same node as their parent node. This mechanism would make the topology not only a sink-rooted tree, but also a cluster-based network, which is very helpful for data aggregation and other applications. As shown in Fig. 4, the solid nodes play the roles of cluster headers, and most nodes become leaf-nodes.

The link quality would change dramatically as external environment changes or link failures happen. If the link failure occurs between a parent node and its child, it would cause a rupture of the network, which will cause a dramatic delay in data transferring. We use an active checking mechanism to guarantee a reliable link quality of a node and its parent. Whenever the link quality has dropped below the threshold we can endure, the child node would re-select a parent node by broadcasting a message for help. However, the parent node would broadcast a notification message to force its children to re-select parents when its residual energy decreases below a threshold. The reason for using a threshold is to assure that the parent node can work as a leaf-node for a enough long time.

After establishing the topology, child node would send the sensing data to its parent node, which would forward it to the management station hop by hop. Sensor nodes would report two types of data, which are emergent data and periodical data. Emergent data is an immediate report of the state change of the parking space, and periodical data is a periodical report of the state. For the requirement of real-time processing, emergent data would be forwarded to the management station as soon as possible, but periodical data would be aggregated with other data on the cluster headers in order to reduce the traffic burden of the entire sensor network.

3.2.3 Data aggregation module

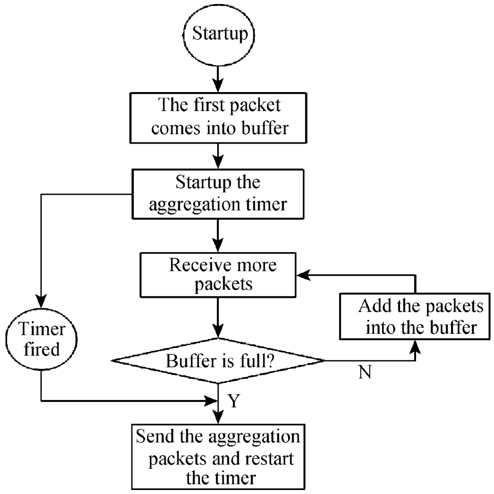

Fig. 5 The process of data aggregation

The purpose of the data aggregation module is to assemble several short packets into a long packet to reduce the traffic burden of the network. Data aggregation would bring a little more delay in data transmission so the packets, which need to be aggregated, are some non-urgent packets, such as periodical detection data, remote debugging data and topology report data.

The data aggregation module has a buffer that is big enough for any type of packets in the network and a timer to control the time of sending an aggregation packet. The length of the timer is related to the first packet that comes into the buffer, and the longer the packet is, the shorter the waiting time would be. If the buffer is full before the timer fires, the aggregation packet would be sent immediately and the timer would be reset. Fig. 5 gives the detailed operating process of the data aggregation module.

3.2.4 Time synchronization module

The time synchronization module provides a standard system time for sensor nodes and keeps it in a synchronized manner among all the sensor nodes. It acquires the standard time from the management station and broadcasts it to all the nodes in the network. Since absolute time is meaningless in TinyOS, the system time should be regarded as a relative time.

Sink node takes the responsibility of starting the process of time synchronization. Sink node gets the system time from management station, records it as the reference time and sets the counter of ticks back to zero. It counts the ticks that flows and adds it to the reference time to get its current time. When it broadcasts a time synchronization packet (SYNC packet), it writes a timestamp into the SYNC packet. When one of its neighbor nodes receives the SYNC packet, the neighbor node adds an experiential compensation to the timestamp in the packet and records the result as its own reference time. Then it uses the same mechanism as sink node used to diffuse the SYNC packet. This mechanism would guarantee a unique start point for all the nodes in the sensor network.

If some node failed to receive any SYNC packets in the above process, it would broadcast a request for help. Any neighbor node that has a synchronized time would send back a reply that contains the correct time. This would help the node to synchronize to the network.

3.2.5 MAC

The MAC mechanism is based on the implementation in TinyOS. This MAC mechanism is contention based and has a random back off when collision occurs.

The MAC module is the implementation of the state machine of MAC protocol, which is driven by the interrupts from both SPI and ADC interface. The SPI interrupt would occur only when MCU has exchanged one byte with CC1000 module and the ADC interrupt would occur only when channel detection is done. If the channel is clear and there is something to send, the state machine would be ready to send the first byte, or it would wait for a random time for next channel detection. The criterion of whether the channel is clear is determined by the signal gained from RSSI and should be further calculated according to certain formulae.

4 Parking guidance

4.1 Hardware architecture of guiding nodes

Fig. 6 shows the hardware architecture of the the guiding node. Except for adding a LED driver module and removing the sensing module, it is the same as monitoring nodes in architecture.

4.2 Source routing scheme

Guiding nodes acquire guiding messages from the management station by utilizing a source routing scheme. A guiding node sends a message, which is used to maintain source routing periodically, to its parent node selected together with the monitoring nodes. The message includes a source routing field that contains only the guiding node′s own ID at first. After receiving the message, the parent node adds its own ID into the source routing field and delivers the message to its parent node as well. Thus, the maintenance message will be delivered upstream to the sink node along the routing tree and will record all the nodes it passes by. Consequently, the sink node will know all the nodes along the path from itself to the guiding node when it receives the maintenance message. Once some node on the source route path changes its parent node, it sends an update message to the sink node to report the event, so the sink node will wipe off the stale route and record the fresh one.

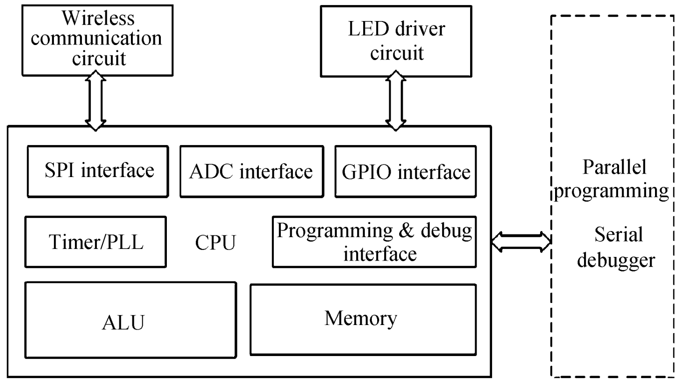

Fig. 6 Hardware architecture of guiding nodes

When the sink node needs to send a parking-space guiding message to a guiding node, it packs the message with the latest source routing information. The message will be delivered downstream to the guiding node. During the transmission process, the node, which receives the message, would remove its own ID from the source routing field of the message before transmitting it to the next hop to save communication energy.

4.3 Guiding information

Guiding nodes divide their managing areas into several sections according to the turnoffs of the roads where they are deployed. The management station takes the charge of computing the number of idle parking spaces in each section.

Fig. 7 shows an example of managing areas partitioned by the guiding node that is nearest to the entrance. In this example, sixteen parking areas and nine guiding nodes are in the parking lot, where guiding node A is nearest to the entrance. Locating at a crossroad, node A divides the whole parking lot into four managing areas, which are surrounded by dashed lines and marked with a number from 1 to 4. Obviously, the display of idle parking spaces on the four directions is different. For example, the number displayed for the up direction of node A is the sum of idle spaces in managing areas 1 and 2, but the number for the left direction is only the sum of idle spaces on both sides of the left road.

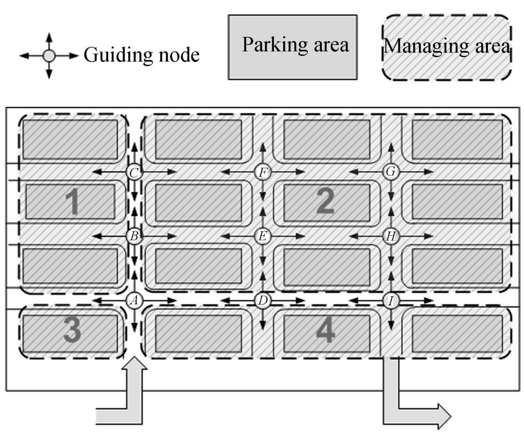

Fig. 7 An example of managing areas

5 Management station software

Fig. 8 shows the software framework of the management station. The modules in the framework are described as follows.

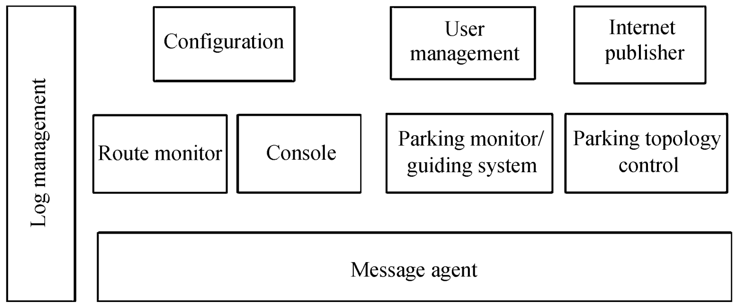

Fig. 8 The software framework of the management station

1) Message Agent

The message agent module communicates with upper layer modules by establishing socket connections, thus upper layer modules can run as stand-alone applications on remote computers. Message agent uses RS-232 interface to connect with the sink node and gathers data from it.

Once other modules establish connections with message agent, they will first register the data types they deal with. Message agent module segments the data received from sink node and dispatches data segments to the modules that have registered their interests in certain data types. On the other hand, message agent also receives commands from other modules and injects them into the sensor network.

2) Console

The console module provides a convenient means to control and debug the sensor network. It can send many kinds of control commands down to the network, analyze and display the messages gathered from the sensor network, and calculate the real-time statistic information for system testing and evaluating.

3) Log management

The log management module provides logging interfaces for other modules. The information logged includes run-time events and statistic data of the parking management system.

4) Route monitor

The route monitor module watches the topology status of the whole wireless sensor network. When the routing status changes, nodes will send changing messages with timestamps to the sink node. Route monitor module will deal with these messages and update the topology-display in time. The additional information provided by route monitor includes link quality, location of nodes and neighbor relationship of each sensor node, *etc*.

5) Parking monitor/guiding system

Parking monitor gathers all the parking status messages, including periodical ones and emergent ones. It records the current status of each parking space and visualizes the status of the whole parking lot on its GUI. The guiding system module calculates the number of idle parking spaces in each parking area and sends the guiding messages to the guiding nodes.

6) Parking topology control

The parking topology control module provides a GUI to generate the topology plots for parking management according to the architecture ichnography of the parking lot and map all the sensor nodes to parking spaces in the plots.

7) Configuration

All the modules need their own configurations. The configuration module provides a friendly GUI to generate, modify and dispatch these configurations.

8) User management

The user management module stores and manages the information of users. It can bind a user to a certain parking space and manage reservation requests.

9) Internet publisher

The internet publisher module publishes the parking status information on the Web in order that car drivers can know about the overview of parking status prior to their arrival and make reservations.

6 Preliminary experiments

We have implemented the prototype of the system and taken some preliminary experiments. In this section, we present some results of our two main experiments to describe the performance of the sensor network.

6.1 Experiment setup

The first experiment was taken on the sixth floor of our office building, where there are walls and pillars that would affect the communication quality of the network. Including a sink node and five guiding nodes, 30 nodes were deployed in this experiment, as shown in Fig. 9 (a).

The second experiment was taken in a dining hall with a grid-deployment of 67 nodes, which is similar to the deployment in a parking lot. Fig. 9 (b) shows the deployment scene of the experiment, and Fig. 9 (c) is a topology snapshot captured by the route monitor module when the network was running.

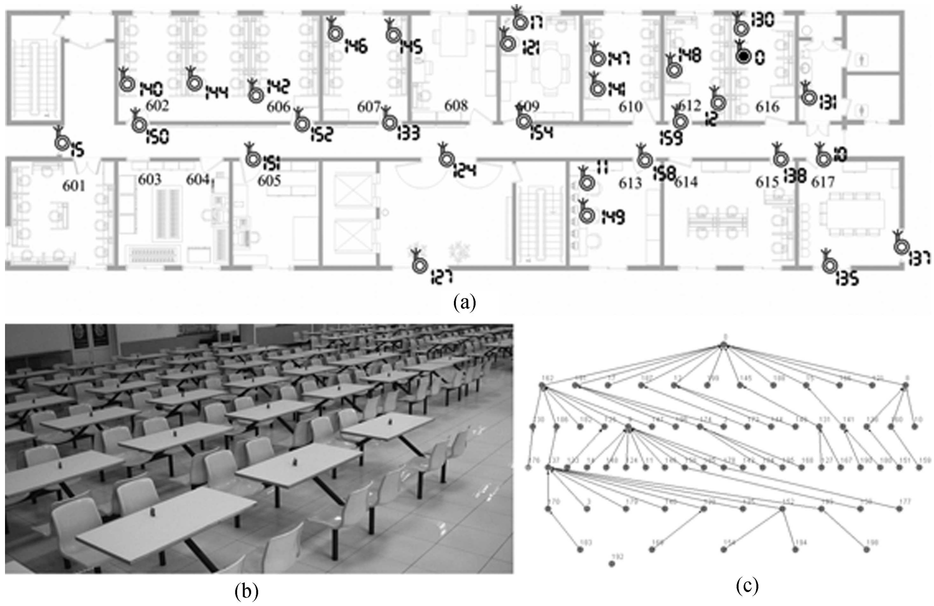

Fig. 9 The deployment of the two experiments
(a) Deployment of 30 nodes in an office building; (b) Deployment of 67 nodes in a dining hall;
(c) A topology snapshot of the network that has 67 nodes

In both experiments, all the nodes equipped with two AA batteries and the same software system. In order to form topology trees with four or five hops deep in the two experiments, the nodes in the first experiment used a -10dBm output power, but -20 dBm was used in the second experiment. Therefore, the nodes density in the first experiment was smaller than that in the second one.

The purpose of the two experiments is to evaluate the performance of the network in different circumstances for further improvement. The performance we studied includes packet loss rate, throughput, and packet delay, which are described in detail in the following sub-section. Instead of reporting the data sensed, sensor nodes in both experiments sent TEST packets every 20 seconds, which include timestamps and sequence numbers. The management station, which was connected with the sink node, sampled the data every 10 report-periods and took on the statistical work. Considering retransmission mechanism can guarantee a low end-to-end packet loss rate, we limited the nodes only to retransmit the TEST packet for three times on failure to observe the performance of packet loss rate.

6.2 Experimental results

1) Packet loss rate

As shown in Fig. 10, the packet loss rate in the first experiment is lower than that in the second one, which is due to that the higher the nodes density in the network, the more collision occurs during transmission. Fig. 11 shows that the packet loss rate increases with hop-count in both experiments. Since packets generated by the nodes that have big hop-counts have to take long trips to arrive at the sink, they would experience high probability to be dropped.

There are two main ways to improve the performance of packet loss rate in our system, one of which is to increase the retransmission times on sending failure, and the other is to mend our routing protocol by making use of the link quality information coming from the MAC layer.

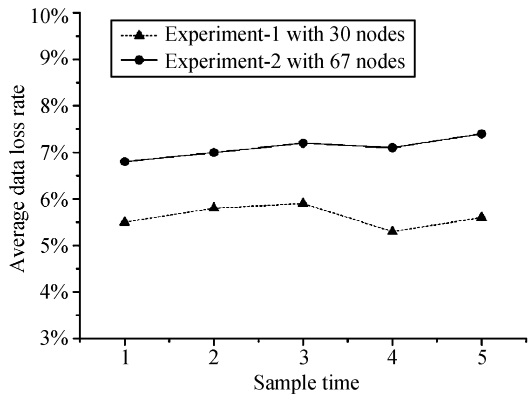

Fig. 10 Average packet loss rate at different sampling time

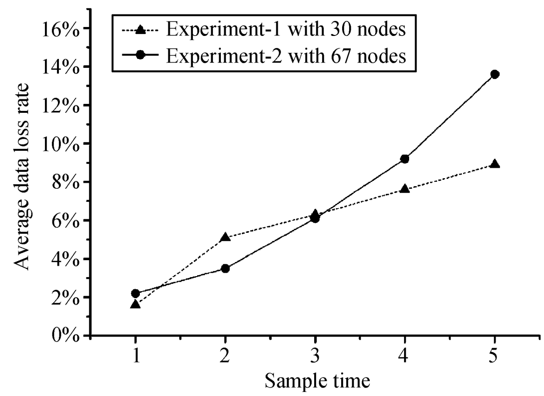

Fig. 11 Average loss rate at different hop-count

2) Throughput

The throughput we studied is historical, which shows the amount of data received by the management station during a period of time. As shown in Fig. 12, the throughput was very high on the first sampling point, which approached the burst rate, and it declined with time due to the packet loss. Finally, the throughput in the two experiments tended to stable levels at about 30 Bps and 60 Bps respectively.

3) Propagation delay

Fig. 13 shows that the increase of hop-count led to the increase in propagation delay. The propagation delay mainly consists of competition time and several dozen milliseconds that involved by the routing protocol to decrease collision. Therefore, the propagation delay in the second experiment was larger than that in the first one because of the worse collision caused by higher node density.

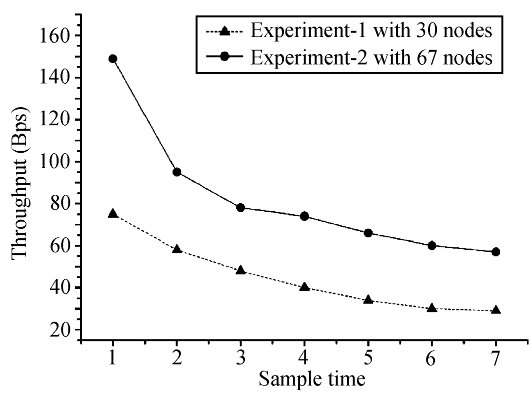

Fig. 12 Network throughput at different sampling time

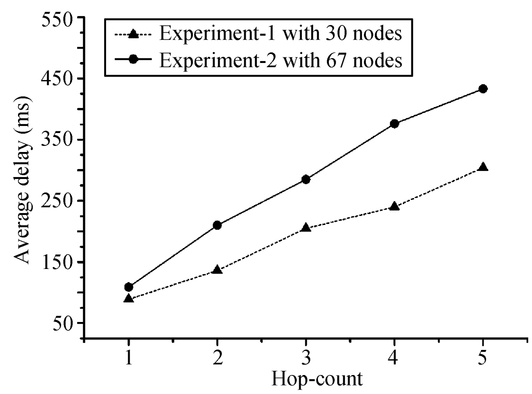

Fig. 13 Propagation delay at different hop-count

7 Conclusion

This paper introduces a wireless sensor network based parking management system, which provides functions of parking monitoring and management, and the parking guidance service as well. The system includes three kinds of nodes and a management station for central control. Each kind of nodes plays a different role in the system, and communicates directly or indirectly with other kinds of nodes. They collaborate with each other to accomplish topology formation, route establishing, parking space status sensing and reporting, command processing, and so on.

Based on the analysis of the preliminary experimental results, we found some aspects to improve, such as that the routing protocol could make use of the link quality information to decrease the packet loss rate. In addition, for this moment, the information is logged in formatted plain text files, so employing a MySQL database to store the users′ information and system logs is under considering. Furthermore, tests in real parking lots would be performed in the future.

References

1 Atmel AVR 8-bit RISC family, http://www.atmel.com/products/avr, Nov. 2005

2 CC1000 Single Chip very Low Power RF transceiver[Online], available: http://www.chipcon.com/files/CC1000_Data_Sheet_2_2.pdf, Nov. 2005

3 TinyOS[Online], available: http://www.tinyos.net, Mar. 2006

BI Yan-Zhong Ph. D. candidate at Institute of Software, Chinese Academy of Sciences. His research interests include routing and MAC techniques of wireless sensor networks.

SUN Li-Min Professor at Institute of Software, Chinese Academy of Sciences. His research interests include wireless sensor networks and mobile IP networks.

ZHU Hong-Song Ph. D. candidate at Institute of Computing Technology, Chinese Academy of Sciences. His research interests include embedded systems and wireless networks.

YAN Ting-Xin Master student at Institute of Software, Chinese Academy of Sciences. His research interests include routing and MAC techniques of wireless sensor networks.

LUO Zheng-Jun Master student at Institute of Software, Chinese Academy of Sciences. His research interests include ultrasonic techniques and embedded systems.