ARTICLE IN PRESS
CSI-02776; No of Pages 12
Computer Standards & Interfaces xxx (2011) xxx–xxx
Contents lists available at ScienceDirect
Computer Standards & Interfaces
journal homepage: www.elsevier.com/locate/csi

On the design of beacon based wireless sensor network for agricultural emergency monitoring systems

Young-Duk Kim [a], Yeon-Mo Yang [b,*], Won-Seok Kang [a], Dong-Kyun Kim [c]

[a] Daegu Gyeongbuk Institute of Science and Technology (DGIST), Republic of Korea
[b] School of Electronic Engineering, Kumoh National Institute of Technology, Republic of Korea
[c] School of Computer Science and Engineering, Kyungpook National University, Republic of Korea

ARTICLE INFO

Article history:
Received 11 March 2011
Accepted 8 May 2011
Available online xxxx

Keywords:
Beacon frame
Mobile robot
Real-time agriculture monitoring system
Wireless sensor networks

ABSTRACT

In this paper, we proposed new sensor network architecture with autonomous robots based on beacon mode for real time agriculture monitoring system. The proposed scheme also offers a reliable association with parent nodes and dynamically assigns network addresses. For the large scale multi-sensor processing, the proposed system accomplished the intelligent database, which generates alert messages to the handheld terminal by means of the fire and air-based sensor data. Thus farmers can easily check out the current conditions of crops and farms at anytime and anywhere. Moreover, we also developed a robot platform with network based mobility function for mobile surveillance.

1. Introduction

Recently, WSNs (Wireless Sensor Networks) [1] and mobile robot technology in agriculture have become one of the most popular technologies for agriculture monitoring systems. In general, WSNs can be widely-used such as agriculture, military, medical and industrial purposes. Among these various fields, agricultural application is considered one of the most promising services for WSN realizations to enhance the food-crop production and reduce burdens of farmers. However, for the deployment of WSN systems for monitoring purpose in agricultural environments, a number of open problems remain. The most representative examples of such problems are as follows.

1) In general, most crops are very vulnerable to weather conditions such as temperature, humidity, intensity of illumination and etc., which is a significant burden for farmers to observe weather conditions every hour and every day. Furthermore, in indoor environments (e.g. vinyl green houses), the occurrence of fire is one of most fatal agricultural disasters. These facts imply that the agricultural WSN system needs to provide real-time monitoring services of whole environmental conditions for improving crop production, plant growth and preventing serious disasters in farms.
2) Recently, due to increased industrial developments, air pollutions and biologically noxious elements (e.g. carcinogenic substance and influenza virus) are prevalent worldwide. Thus, the agricultural monitoring system should not only detect various air pollutions but also report to farmer and related agency from the remote sites for further investigations. Moreover, this critical information needs to be dealt with more emergence processing, which means that the WSN system should provide low latency transmissions with high reliability.
3) The agricultural monitoring system with various sensors should record and store whole measured information to establish an agricultural database system which may provide not only analysis of crop growth but also harvesting prediction by using analyzed patterns of changing conditions in farms.
4) The farmer may use unmanned vehicles or robot systems in order to monitor and patrol the wide farm areas due to manpower shortage. In this situation, the mobile platform needs to communicate with the deployed sensor network to transmit measured data, which means that it becomes a mobile node to join and disjoint the networks. Thus, the monitoring system should support the mobility service for smooth handover and seamless data networking.

Besides the aforementioned requirements, when we deploy WSNs in an agricultural multi-hop environment, several problems can be observed because of limited bandwidth and packet collision by channel interference, and so on. To improve these performance limitations, the authors in [5–7] have proposed an

* Corresponding author.
E-mail addresses: ydkim@dgist.ac.kr (Y.-D. Kim), yangym@kumoh.ac.kr (Y.-M. Yang), wskang@dgist.ac.kr (W.-S. Kang), dongkyun@knu.ac.kr (D.-K. Kim).

0920-5489/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2011.05.004

efficient beacon based WSN protocols with BOP (Beacon Only Period) and LAA (Last Address Assignment) schemes. However, there is no implementation in real test-bed systems with mobile robots in agriculture and neither do they consider real-time data processing nor the nodes mobility. In this paper, we designed and developed an efficient agricultural WSN monitoring system with an autonomous mobile robot for real-time agricultural emergency message processing applications which can be used to monitor indoor/outdoor disasters such as with fire detectors and atmosphere observation in farms. All sensor and sink nodes are implemented on the TinyOS [2] system which is based on NesC. In addition, we have also developed a new agricultural disaster database system which can process real-time data for emergency applications.

The rest of this paper is categorized into four sections. In Section 2, we review TinyOS architecture for our operating system platform and IEEE 802.15.4 MAC protocol as well as its improved versions with beacon scheduling. In Section 3, we illustrate the detailed design architecture and implementation issues of our agriculture monitoring system. Performance evaluation through operations in real test-bed and simulation is presented in Sections 4 and 5, respectively. Finally, concluding remarks are given in Section 6.

2. Related works

2.1. TinyOS

TinyOS is a free software and open BSD-licensed operating system which is well designed for tiny low-power wireless devices, such as those used in sensor networks, ubiquitous computing, personal area networks, smart buildings, and other smart industrial purposes. It is written in the nesC programming language [3] as a set of cooperating tasks and processes and its programs are built out of software components, some of which present hardware abstractions. Components are connected to each other using interfaces. TinyOS provides interfaces and components for common abstractions such as packet communication, routing, sensing, actuation and storage. TinyOS is completely non-blocking: it has one FIFO (First In First Out) stack. Therefore, all I/O operations that last longer than a few hundred microseconds are asynchronous and have a callback. To enable the native compiler to better optimize across call boundaries, TinyOS uses nesC's features to link these callbacks, called events, statically. While being non-blocking enables TinyOS to maintain high concurrency with one stack, it forces programmers to write complex logic by stitching together many small event handlers. In order to support larger computations, TinyOS provides tasks, which are similar to a Deferred Procedure Call and interrupt handler bottom halves. A TinyOS component can post a task, which the OS will schedule to run later. Tasks are non-preemptive and run in FIFO based scheduling. This simple concurrency model is typically sufficient for I/O centric applications, but its difficulty with CPU-heavy applications has led to the development of a thread library for the OS, named TOSThreads. Fig. 1 shows the overall scheduling policy in TinyOS.

2.2. IEEE 802.15.4 and beacon based protocols

IEEE 802.15.4 [4] intends to offer the fundamental lower network layers of a type of wireless personal area network (WPAN) which focuses on low-cost, low-speed ubiquitous communication between tiny sensor devices. The emphasis is on very low cost communication of nearby devices with little to no underlying infrastructure, intending to exploit this to lower power consumption even more. Networks of IEEE 802.15.4 can be built as either peer-to-peer or star networks. However, every network needs at least one FFD (Full Function Device) to work as the coordinator of the network. Networks are thus formed by groups of devices separated by suitable distances. Each device has a unique 64-bit identifier, and if some conditions are met short 16-bit identifiers can be used within a restricted environment. Namely, within each PAN domain, communications will probably use short identifiers. For the wireless medium access, the MAC layer function, it manages access to the physical channel and network beaconing. However, it has severe limitations by the fact that it supports only 1 hop distance nodes from the FFD, which is not a good solution for multi-hop communication and multi-beacon enabled large mesh networks like agricultural areas. Thus, if we use the original IEEE 802.15.4 in the wireless sensor network with large multiple paths and heavy data traffics, the network may suffer from significant performance degradation such as severe packet collisions, path losses in routing procedure etc.

In order to overcome these limitations of IEEE 802.15.4, the authors in [5–7] proposed the BOP (Beacon Only Period) and the LAA (Last Address Assignment) algorithm for dynamic mesh networks. However, these schemes were not implemented in real test-bed environments and sensor nodes did not support a stable operating system like TinyOS. Another limitation of [5–7] is that they show poor network performance because they do not solve packet collision problems between flooding packets for route discovery and beacon frames. Moreover, they do not suggest actual solution of node mobility support when the application requires seamless data services. Besides, in order to successfully install the wireless sensor network system in real-time monitoring applications such as fire and air pollution detection services, we need an efficient monitoring system which is enabled to process real-time data and communicate with the all sensor nodes in networks. Thus, throughout this paper, we propose a new network architecture for the real-time WSN monitoring system.

2.3. Wireless sensor networks for agricultural purposes

There are many wireless sensor network implementation works which have considered agricultural monitoring requirements in large rural areas. The authors in [8] presented agricultural a WSN application for wine production chain along the different non overlapping areas. They used simple commercial sensor motes and GPRS (General Packet Radio Services) gateways to forward measured data to the remote database server. The user who has a laptop or a PDA device can query the data at anywhere connected to Internet. In [9], an efficient fertigation system based on ZigBee network is developed with sensor motes, mobile terminal, and SQL server. The sensors of this system calls heuristic functions to determine the amount time to open the water value. Then, the remote server can take into account historical runoff water data. The authors in [10] developed an experimental distributed sensor network based on the European ISM band and measured temperature, humidity, solar radiation, rain gauge and etc. This system also has an Internet based remote database system for biological and ecological research analysis. Besides these contributions, a lot of agricultural applications have been proposed to tackle various goals such as climate

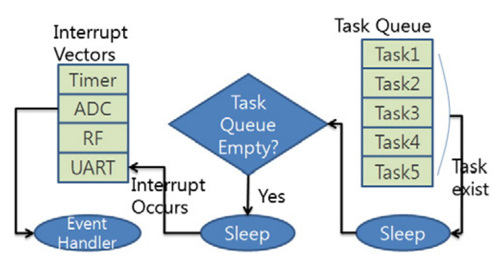


Fig. 1. The scheduling operation in TinyOS.

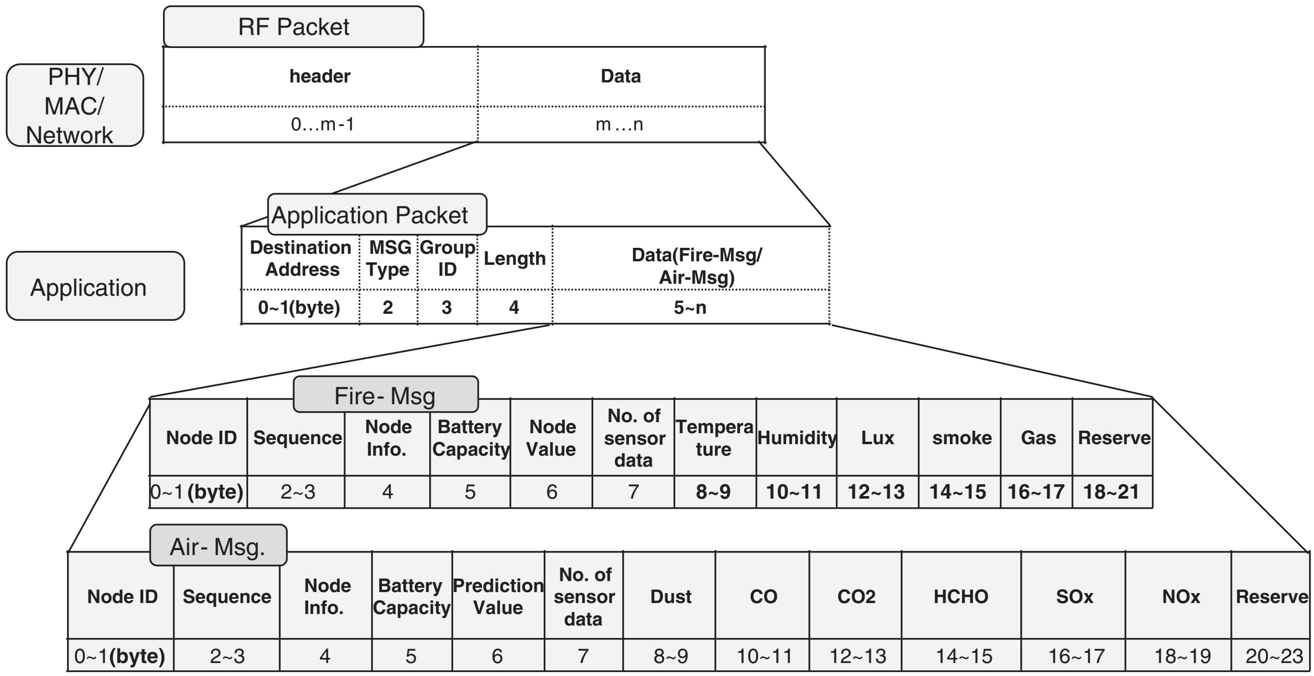


Fig. 2. Packet structure.

monitoring, soil and fruit assessment, control of insects and diseases, control of weeds, crop tracking, and so on. However, existing WSN monitoring system does not highly consider the real-time data processing network protocols for emergency reports and predictions like as: chemical air pollution and fire occurrence. Moreover, they neglect to support the node's mobility functions to perform active surveillance via mobile robot patrols in large field where the transmission range of sensor node does not fully cover.

3. Design and implementation

3.1. Packet structure and MAC layer functions

The packet structure for our agricultural emergency monitoring system is designed for detecting air pollution and the occurrence of fire, which is shown in Fig. 2. The application message is consisted of destination address, message type, group ID, data length, node ID, packet sequence, type and information of sensor node, battery capacity, prediction value of sensor node, the number of sensor data, and sensing data. By using this packet structure, the sensor node periodically gathers sensing data and transmits through RF channel.

In MAC layer operations, there are 4 basic operations, which are association, beacon scheduling, channel accessing with CSMA/CA and the route recovery process. To execute these operations, our protocol defines the WiBEEM_MAC component (WiBEEM_MACC) which is connected to the WiBEEM MAC module (WiBEEM_MACM) with the WiBEEM_MAC_SAP (SAP; Service Access Point) and the WiBEEM_MLME_SAP (MLME_SAP; MAC Layer Management Entity SAP) interface. The WiBEEM_MAC_SAP interface takes charge of the data transmission service and the WiBEEM_MLME_SAP interface has charge of connection management service such as association. Fig. 3 shows the

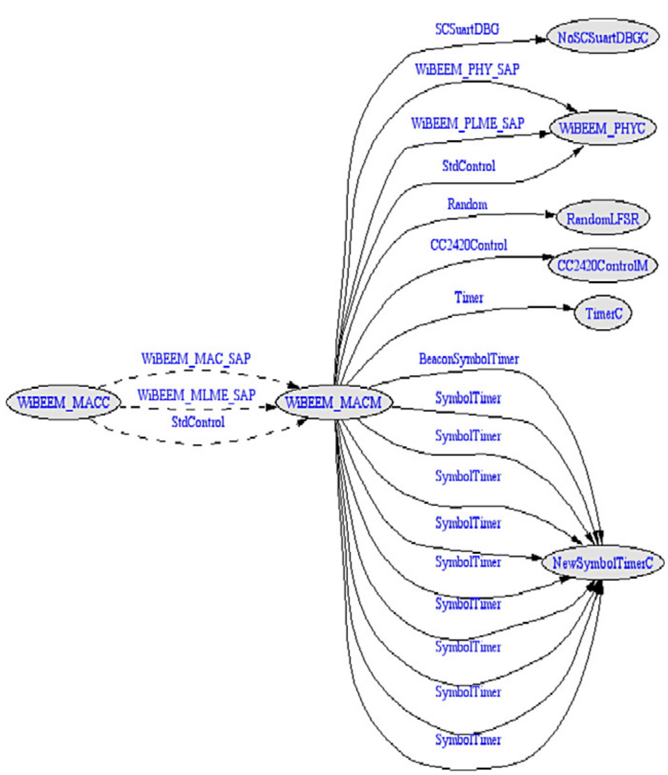


Fig. 3. MAC layer components in TinyOS.

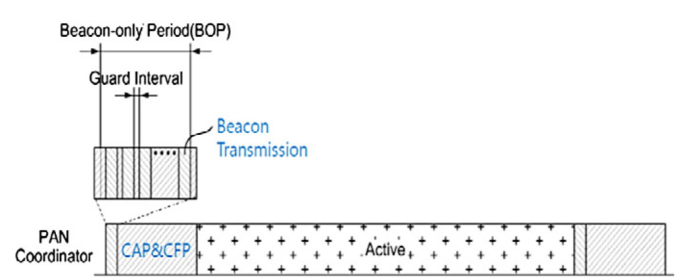


Fig. 4. Superframe architecture.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

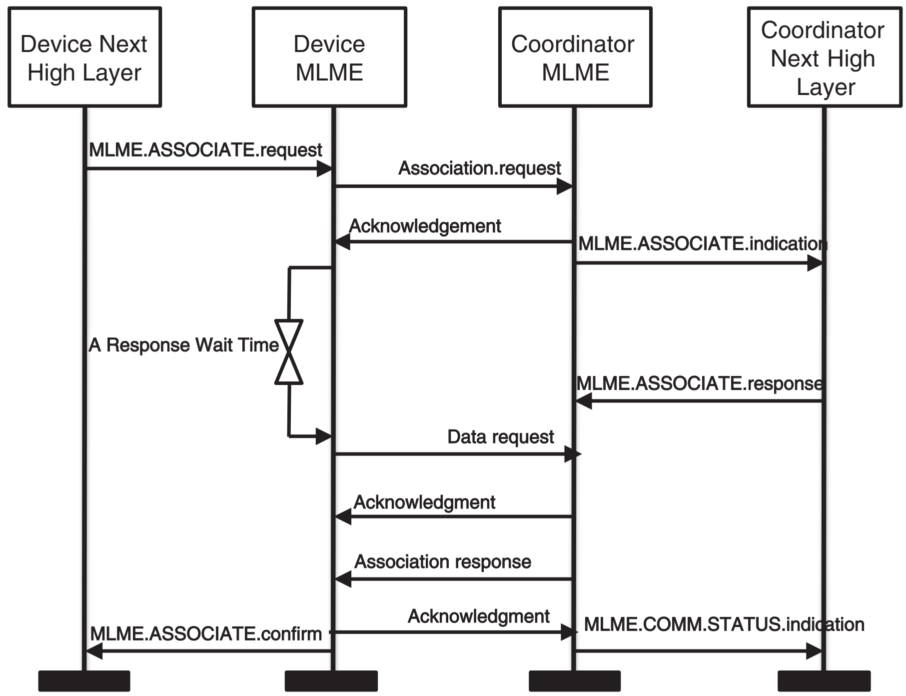


Fig. 5. Association procedure.

WiBEEM_MAC component and its related interfaces as well as these two interfaces. In the figure, the arrows between modules, components and interfaces denote the message sequence of each other.

3.2. Beacon scheduling

Although the superframe architecture of our proposed protocol is similar with that of IEEE 802.15.4, A BOP slot is added for dedicated beacon transmission. This BOP slot is divided into 16 single slots and the first slot is reserved for the PAN coordinator. The other slots are used by neighbor nodes. The overall structure is shown in Fig. 4. For the exact time synchronization, each routing node transmits BTTSL (Beacon Transmission Time Slot Length) messages to neighbor nodes. The BTTSL message describes the time information of beacon transmissions, which makes the network synchronized between the other nodes in the whole network. In this situation, neighbor nodes determine the slot time by themselves with an internal timer such as the BI (Beacon Interval), BOP and myBTTS (Beacon Transmission Time Slot) timer. These timers measure the time interval of each slot and notify the TinyOS to trigger the event handler. The overall scenario of each node for determining the time slot is as follows. At first, PAN coordinator transmits beacon frames with the execution of the MLME_START.request primitive after channel scanning procedure. When the coordinator confirms that the beacon frame is successfully transmitted, it runs the beacon timer during the beacon interval. Then other nodes should check their neighbor's BTTS as well as two hops distance neighbor's BTTS in order to avoid beacon collisions. If an end node overhears these beacon frames, it records the LQI (Link Quality Indicator) information in the PAN descriptor. Then the node requests the association process to PAN coordinator with the best LQI value [11].

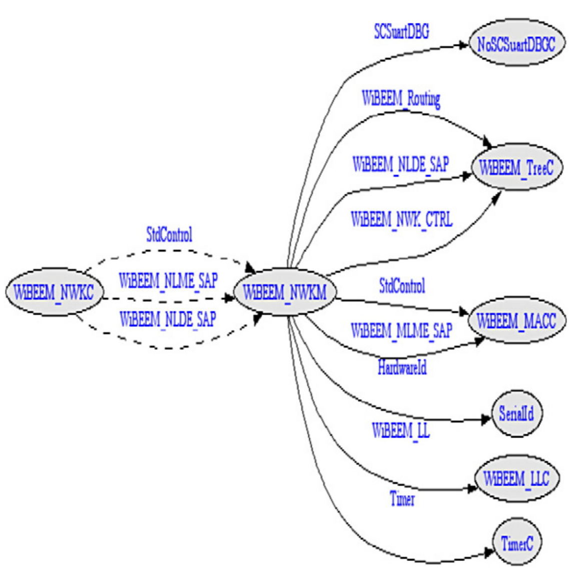


Fig. 6. Network layer components in TinyOS.

3.3. Association process

It is necessary that each end node starts in the association process to participate in the PAN communication. In order to make a hierarchy of architecture for association, we define 3 node types, which are the WC (Wireless PAN Coordinator), the WR (Wireless Router), and the WED (Wireless End Device). The WC plays the role of a sink node and gateway by transmitting periodic beacon frames, as well as, collecting data from the WRs and the WEDs. The collected data is forwarded to monitoring servers for more specific processing such as the management of alert messages to user terminals. The WR also periodically transmits beacons to neighbors and executes scheduling processes with neighbors by exchanging beacon frames. WEDs are logically located in the end of the network and generate packets containing sensing data. Each packet of WEDs is forwarded to the WC via WRs in every wakeup time of the superframe. In general, for IEEE 802.15.4 networks, there are only the FFD (Full Function Device) and the RFD (Reduced Function Device). However, in our work, we assume that the FFD is able to be not only a WR but also a WC. In

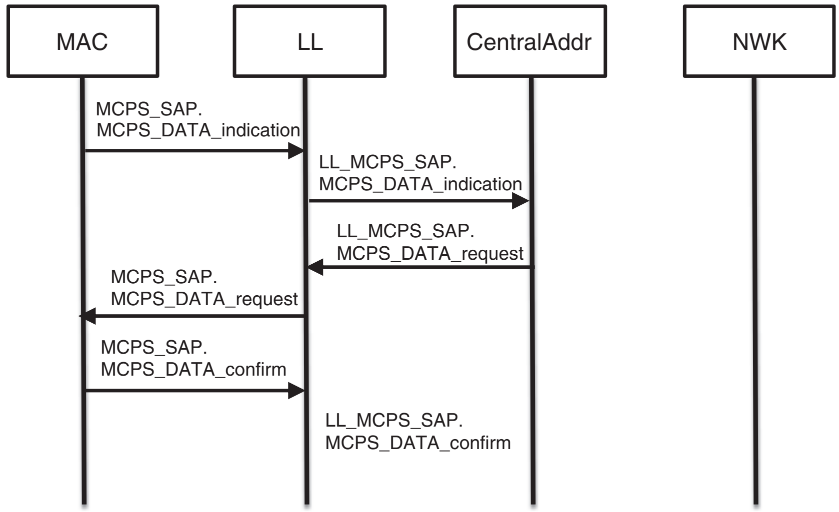


Fig. 7. Message sequence for LAA in coordinator.

addition, the FFD can manage the PAN or make its own network without participating in other PANs.

In order to organize and synchronize the network, the WC and WR transmit beacon frames to neighbors periodically. At first, one of the WRs becomes the WC if it does not hear any beacon frame. Then the WC starts to beacon with its network information such as beacon interval, identification, and information for its neighbors. When other WRs or WEDs try to scan the channel, they executes the MLME_SAP.MLME_SCAN_request() process to associate with the parent node. In this situation, the WR can also associate with another WR node and it calculates its own beacon schedule within the BOP length, which is executed by using received BTTSL information from its neighbors or parent node. The channel scanning process in MAC layer is invoked by calling Network_Discovery_request() command, then each node records beacon information of accessible channels which is between 11 and 26. The scanning information is delivered to upper layer by SCAN_Confirm() function. Finally, by using MLME_SCAN_request() and MLME_SCAN_confirm(), a node transmits Assocation_request() primitive to the parent node with maximum signal strength which, in turn, is derived from scanned beacons. This Assocation_request() is called by the Network_Discovery_confirm() from the network layer. Fig. 5 illustrates the overall procedure of association.

3.4. Network layer functions

The network layer manages network formation, network join and leave function, routing, LAA and path recovery. In network formation and joining procedure, WC scans the whole wireless channels between number 11 and number 26 in order to determine the channel with the best signal strength among candidates. Then, it starts to transmit beacon frames containing the PAN ID and address. The detailed operations of LAA are described in Section 3.5. Routing and path recovery procedures are explained in Sections 3.6 and 3.7, respectively.

For the component design, we define WiBEEM_NLDE_SAP and WiBEEM_NLME_SAP interfaces in WiBEEM_NWK (NWL; Network) component. WiBEEM_NLDE_SAP (NLDE; Network Layer Data Entity) offers data transmission services and WiBEEM_NLME_SAP (NLME; Network Layer Management Entity) supports join and leave procedures, respectively. Fig. 6 illustrates the overall interfaces of WiBEEM_NWK components as well as the aforementioned interfaces.

Table 1
An example of routing table.

Parent address		Child address	
Short address (16 bit)	Long address (64 bit)	Short address (16 bit)	Long address (64 bit)
6	0x0000000 000000006	7	0x0000000 000000007
		8	0x0000000 000000008

3.5. LAA (Last Address Assignment)

In ZigBee networks [12], network layer uses 16-bit address which is defined in IEEE 802.15.4 MAC. Although this scheme assigns a unique address by using a tree based approach, it significantly suffers not only address wastes but also insupportableness with dynamic topology changes. To solve these problems, we adopt the LAA scheme which dynamically assigns a new address to the end node with the last assigned identification.

When a sensor node turns on its power, it immediately tries to associate to its parent node. Then the coordinator or intermediate WR assigns an address for associating node. For this, WC or WR node receives an ASSOCIATE_indication message in network layer from the end node and transmits an ASSOCIATE_response message with dynamically generated address. After the association request is successfully done, if WR generated the response message, WR transmits a confirm message to WC whether it assigned correct address. Then finally WC updates LAA value with increased address in order to update the address for next assignment. Fig. 7 shows the sequence chart of WC when the address is successfully assigned to associating node. The message flow begins in the MAC layer and passes through the LL (Logical Link) entity Fig. 7. Finally, it receives the WC assigned address from CentralAddr entity.

In case of association in WR, the address assignment process finishes when it receives the confirm message from WC. However, in WC, it starts to comparison operation with its own LAA value. If the value is legal, WC updates its LAA with increased value and transmits the reply message to WR. Otherwise, if the value is illegal or mismatch with its own value, WC transmits a reply message with its own LAA + 1.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

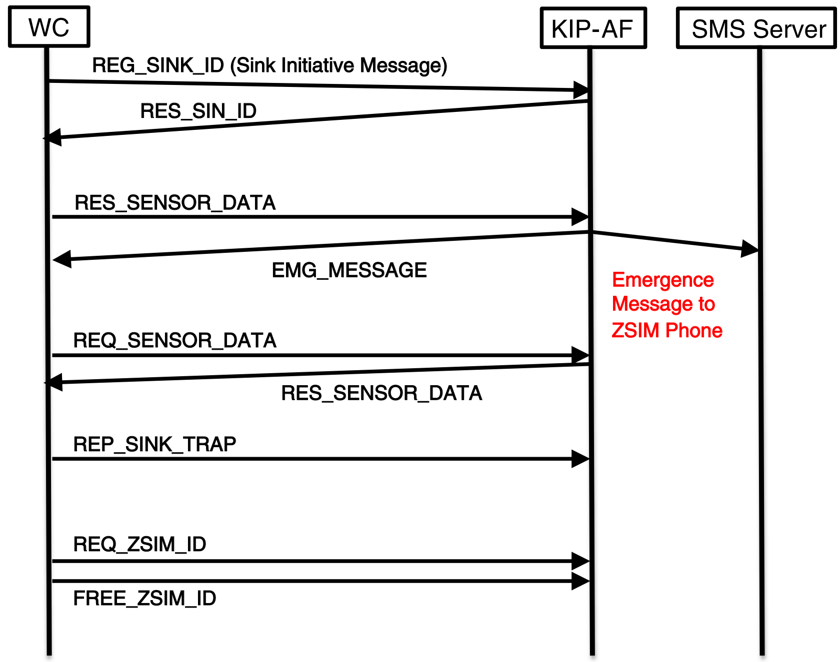


Fig. 8. Message flows between WC and KIP-AF.

3.6. Tree based routing with association information

In past decades, many on demand routing protocols such as AODV [13] and DSR [14] are proposed and they usually broadcast RREQ packets and receive RREP packets during the route discovery procedure. Although this broadcasting method using such control packets is well efficient for wireless mobile ad hoc networks, it is considered inappropriate to wireless sensor network which consisted of small sensor nodes, since, firstly, legacy on demand routing protocols significantly waste network bandwidth and battery

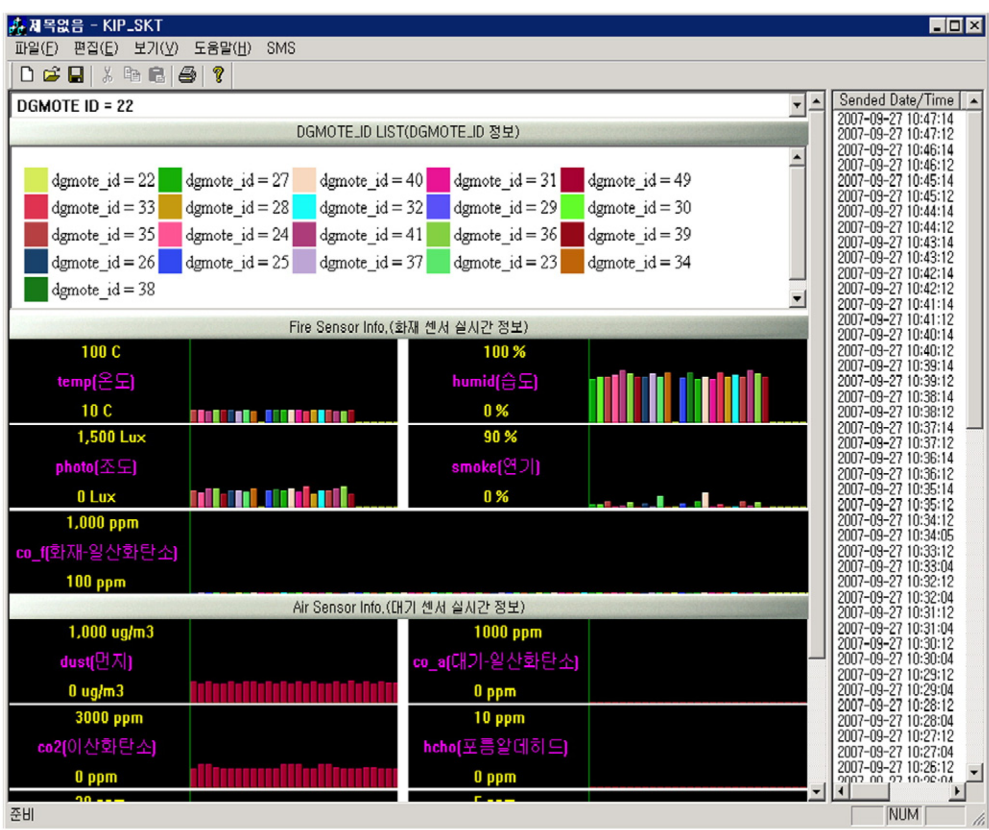


Fig. 9. Real-time monitoring system for detecting fire accidents and air pollutions.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

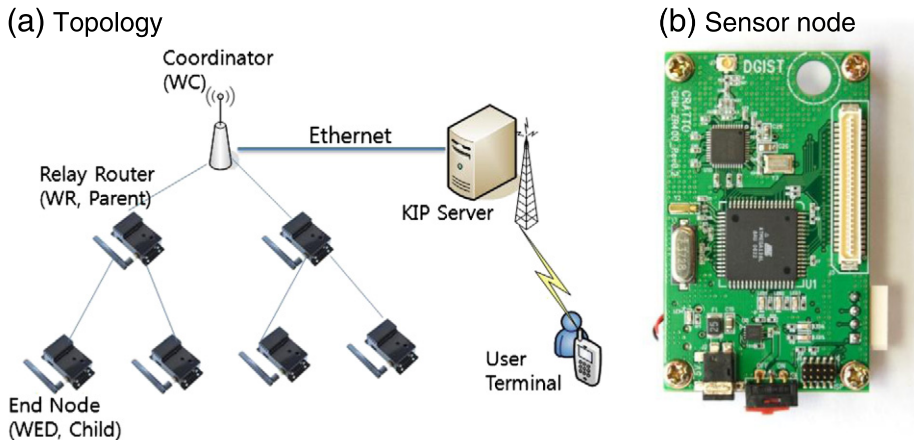


Fig. 10. Conceptual topology and developed sensor node.

power. Secondly, when relay nodes use the beacon frame for time synchronization, it suffers from significant packet collisions between beacons and other control packets. Then, when the duty cycle of each node increases, the flooding overhead also increases, which may result in network congestions or network partition. Thus, in order to reduce the control packet overhead for routing in network layer, we propose an efficient self-routing approach by using association information between parents and child nodes in MAC layer.

When a node wants to participate in network communications, it requests a new network address to its parent node (WR) or coordinator (WC). Then, WR or WC may dynamically assign the address by using beacon frames with the LAA scheme. After association procedure in this manner, the parent node can easily acquire the address of child node and the child node also can obtain the address of the parent node in tree based topology. The sharing address information between the parent and the child is stored in the simplified routing table which is shown in Table 1.

Then, when a node receives incoming packets from the network interface, it directly and immediately forwards to its parent node without using fatal RREQ flooding. Consequently, the source node and intermediate node on the path not only guarantee low latency packet forwarding but also mitigate additional control overhead. In addition, since each node does not need to maintain and exchange the routing table information of whole network, it also conserves physical memory capacity.

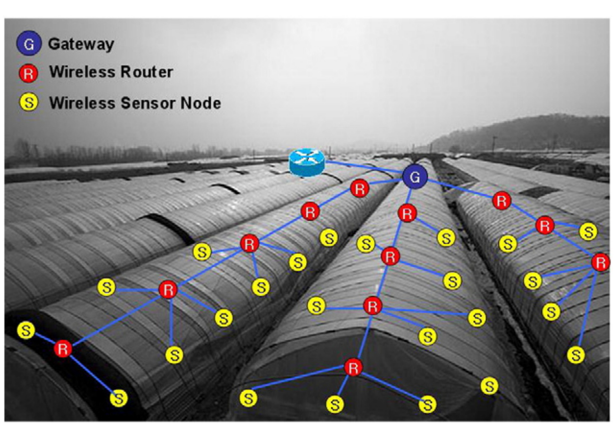


Fig. 11. Network topology for vinyl greenhouse in farms.

3.7. Mobility management and route recovery process

If a node detects any packet losses of expected beacon or data packet in a certain period, it considers that unexpected link error or handover is taking place in the MAC layer. In this situation, there are two desirable solutions to recover the data flows. The first one is that each node starts the re-association procedure to another parent node with the self-routing scheme mentioned in the previous section, which is simple and efficient approach by the fact that it does not require additional unnecessary route discovery procedure by using RREQ (Route Request) flooding. Thus, this re-association scheme prevents unnecessary bandwidth wastes and prolongs the battery life time of each node. The other method of path recovery is to send RERR (Route Error) packet to the source node along the active path in network layer. Although this scheme is one of the most common approaches in existing on demand routing protocols, it significantly suffers from more route rediscovery delay and more bandwidth wastes.

Thus, we adapted the MAC layer re-association scheme and left the network layer maintenance as optional function. When the node detects link failure during communication session, it immediately performs the re-association procedure. Then, MAC module of the node requests network layer to update route information in the routing table, which prevents broadcasting RREQ packets to the entire network. In this manner, WR and WC perform the route recovery operation with association based tree routing scheme mentioned as previous session. By using this cross-layer like re-association procedure, WR obtains the mutual information between child and parent nodes. Finally, the intermediate node which finished the re-association procedure just forwards the received data packets to the link of the parent node and this relay process is continued until it arrives in the coordinator. Consequently, the mobile node minimizes handover delay and it is believed that our proposed scheme accomplishes the self-routing from the fact that the intermediate node does not depend on other routing information.

3.8. Real-time monitoring system

The Real-Time Agriculture Monitoring System, called in KIP-AF (Knowledge Information Processor for Agriculture-sensors data and Fire-sensors data) accomplishes pre-processing and post-processing. The former functions are analyzing the payloads from the WSN environment. The latter functions are to predict the disaster situations from the analyzed payloads. If detected the events, it transmits the alert information about human-action guides to the ZSIM (ZigBee enabled

Fig. 12. Wireless sensor nodes for vinyl greenhouse.

SIM) phone. Also the user holding the ZSIM phone can query the current fire symptom and atmosphere status. The alert information is classified into tree status such as fire symptom, fire warning and fire occurrence.

Fig. 8 describes the communication protocol between WC and KIP-AF. When WC starts to associate with KIP-AF, it transmits REG_SINK_ID for registration. After the host responses with RES_SINK_ID, the initial process is done. Then, WC periodically transmits reporting message which is REP_SENSOR_DATA. If the reported message is considered as emergency, the host generates MEG_MESSAGE and forwards it to the WC and ZSIM phones. In order to support the query from mobile phone, the host and ZISM exchange REG_SENSOR_DATA and RES_SENSOR_DATA messages.

Fig. 9 illustrates the implemented KIP-AF system based on collected information through our proposed wireless sensor network. The GUI shows every node in the network order by node ID and their sensing information in sequence, which is also stored in the database systems.

4. Test-bed environment and performance

Our target application is the agricultural fire and air pollution monitoring systems in farms. Thus we have designed a simple farm topology which models grouped vinyl greenhouses for rural areas in Korea and the whole architecture is shown in Figs. 10 and 11. As shown in Fig. 11, the topology is linear in form because it reflects the long aisle in the greenhouse. Consequently, the WC is located in the center of the aisle and WEDs are scattered along the linear aisle. In addition, because the farm may have several greenhouses, the tree based network topology is reasonable for implementation.

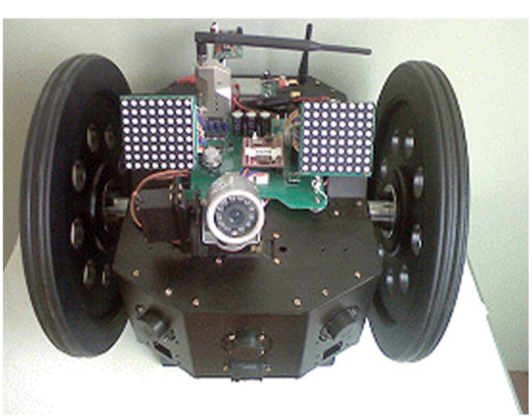


Fig. 13. Mobile robot platform.

We developed sensor modules with the CC2420 of TI Chipcon product as the RF transceiver and the ATMega128L as a main processor, which is shown in Fig. 10(b) and Fig. 12. The application was implemented for the fire and atmosphere monitoring services such as temperature, gas, smoke, humidity and illumination. The gas information is classified into CO, CO2, HCHO, SO2, NO2, and etc. This information is forwarded to the WC in every few seconds and the KIP-AF shows the measured values in real time. If any emergent data arrive in the KIP-AF, the server immediately transmits the alert message to the end user by using the ZSIM. For network organization, we used 1 coordinator node (WC), 8 routers (WR), and 50 sensor nodes (WED) and set tree based network topology with a maximum of 3 hops. The wireless radio transmission range was 30–40 m and the RF power control was set from 0 to −20 dBm. In our experiments, the length of data packet was set to 40 byte and the duty cycle was fixed to 100%, which means all sensor nodes transmit packets every transmission interval in order to process agricultural emergency data. In every interval, all sensor nodes transmit their sensing data to the WC and then gathered data is forwarded to the KIP-AF which maintains the agricultural database for intelligent decision and further analysis. When this KIP-AF detects emergence conditions such as fire occurrence and dangerous air pollutions, it immediately informs to the remote user with ZSIM terminal.

In order to verify the performance of mobility support, we used a mobile robot as a mobile sensor node with identical sensing module to fixed the nodes. This means that the robot may gather information of wide farming areas if WSN cannot cover the whole areas or some of the sensor nodes are out of order. Basically, the robot has two wheels with one assistance sub-wheel and can move with a maximum of 30 cm/s velocity. Fig. 13 and Table 2 describe the hardware platform of the mobile robot with wireless sensing module and summarized specifications, respectively. This robot platform is also designed and implemented on TinyOS software and equipped with an ATmega128L processor for the compatibility with the sensing module. When the robot takes a role of mobile node with mobility function, there is an

Table 2
The parameter of the mobile robot.

Category	Specification
Voltage range	6–9 V
Electric current	200 mA
Battery type	Lithium polymer
MCU	Atmega128
Flash memory	128 KB
Maximum speed	30 cm/s

Table 3
Loss rate measurements.

Performance measurement			
Node ID	Loss rate (%)	Node ID	Loss rate (%)
A	3.38	L	2.94
B	1.88	M	3.71
C	2.93	N	0.91
D	2.62	O	1.91
E	2.63	P	1.97
F	1.87	Q	1.93
G	3.44	R	0.72
H	2.92	S	2.93
I	1.87	T	3.01
J	1.92	U	1.87
K	2.50	V	2.26

inevitable problem of link failure caused by the network handover or loss of LOS (Line of Sight). In this case, the mobile robot tries to search another WR or WC with the best LQI value among the scanned WR candidates. For another approach to tackle the mobility support, we increase the RF power of the mobile robot by adopting power AMP and LNA (Low Noise Amplifier) functions, which enable the transmission range to be two times higher than ordinary conditions.

For the performance study of reliable transmissions, we measured the packet loss rate from WED to WC, which is logged and calculated in the remote monitoring server. The numeric expression for the packet loss rate is as shown in expression (1).

$$\frac{N_{\mathrm{Fail}}}{N_{\mathrm{Fail}}+N_{\mathrm{Rec}}}*100 \tag{1}$$

where N_{Fail} and N_{REC} are the number of failed packets to transmit to the coordinator and the number of successfully received packets in coordinator, respectively. In this expression, the N_{Fail} value includes not only the number of unreceived packets but also the number of packet losses by collisions. The measured results are shown in Table 3 and the maximum loss rate is less than 4%. The main reasons of these packet losses are packet collisions and retransmissions due to collisions. This packet loss rate may increase when the number of nodes increase or the network traffic also increases. However, when we use less duty cycles for data communication, our system will show better performance. Throughout the performance study, we can say that it is significantly reliable and the performance is well suited for real-time processing applications. Moreover, our system can support the fundamental architecture of QoS based applications such as multimedia service and VoIP traffic.

Table 4
Measurements of handover latency.

Trials	Handover starting time (min:s:ms)	Handover finishing time (min:s:ms)	Handover latency (ms)
1	10:48:083	10:50:174	2091
2	10:48:092	10:50:275	2183
3	12:36:048	12:38:168	2120
4	12:36:125	12:38:253	2128
5	15:23:116	15:25:518	2402
6	15:23:426	15:25:688	2262

We also executed the performance study for node's mobility support by using a “Sensor Network Analyzer (SNA)” of Daintree Networks [15], which is a commercial product for wireless packet analysis. Fig. 14 describes the route recovery scenario with two robot platforms as mobile nodes and the experiment is conducted as follows. At first, we set up two intermediate nodes, named WR1 and WR2, which are associated to the WC. After this, two mobile nodes associate with WR1. Finally, two WEDs move close to the transmission range of WR2, which means that they suffer the unexpected link error due to handover. After the route recovery process using cross-layer re-association, the mobile nodes have a new route to the WC and they can transmit data packets again.

In this experiment, when the mobile node executes the handover procedure, the average handover latency, T_{HO}, is calculated as expression (2).

$$\begin{aligned} T_{HO} = {} & T_{beacon\_loss} + T_{asc\_req} + T_{ack} + T_{data\_req} \\ & + T_{ack} + T_{asc\_res} + T_{ack} \end{aligned} \tag{2}$$

By using the following parameters,

T_{beacon_loss} Interval of beacon loss due to handover
T_{asc_req} Transmission time of association request frame
T_{asc_res} Transmission time of association response frame
T_{data_req} Transmission time of data request frame
T_{ack} Transmission time of ACK frame

According to expression (2), because association delay is relatively short, the average handover latency highly depends on the beacon loss interval during the link failure. Hence, in order to minimize the

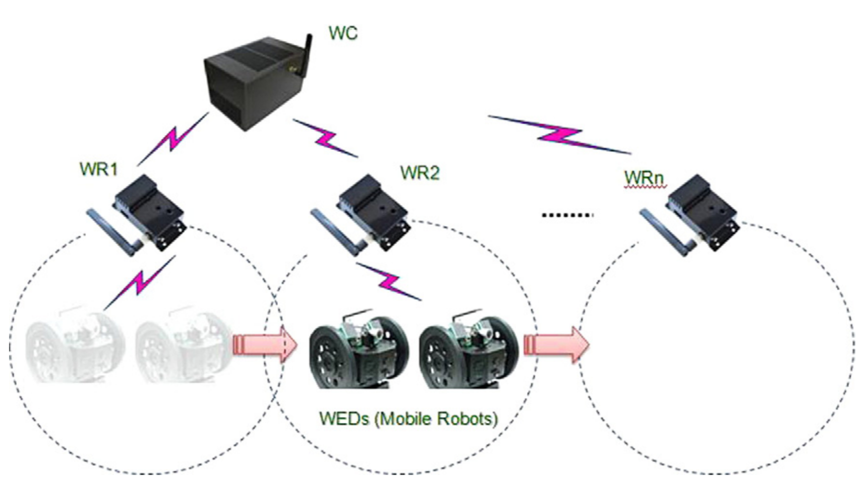


Fig. 14. Handover scenario for route recovery.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

beacon loss interval, we set the pending counter value in MAC layer as 2. This means that after the mobile node does not hear the beacon frame more than two times, it believes that the link is broken and performs the re-association procedure, immediately. In our implementation, we generated packets every second and set the beacon interval at 1 s. Thus, the T_{beacon_loss} value is approximately 2 s.

Table 4 shows the measurement results of handover latency for each trial. Around the time 10:48:08 (min:s:ms), both mobile robots which are associated to WR1, starts to move to WR2. After detecting the link failure by handover, an alternative route is established between WR2 and the WEDs in 10:50:174 and 10:50:275, respectively. As shown in the other results of trials, most average handover latency is under 2.5 s. Thus we can conclude that it is possible to successfully support mobility for communication between mobile nodes and the coordinator in agricultural emergency monitoring services.

5. Performance study with simulation

Beside evaluation of our real test-bed, we also have conducted computer simulations to verify our tree based self routing scheme comparing to the legacy AODV protocol. We used TOSSIM [16] with our beacon enabled MAC protocol and performed the simulation for 1000 s. The measured 4 performance metrics in whole simulations are as follows

Average throughput The total amount of packet bytes delivered to the coordinator during the simulation time.

Packet delivery ratio The ratio of the data packets delivered to the coordinator to those generated by source nodes.

Number of network command frames The number of command packets in network layer.

Number of frame collisions The number of packet collisions during the simulation time.

All performance metrics are measured as two functions such as the number of hops and the duty cycle values, respectively. For the network topology, we adapted the form of a binary tree. Thus, the number of nodes n in a perfect binary tree can be found using this formula:

$$n = 2^{h+1} - 1 \tag{3}$$

where h is the number of hops from the gateway node. For the network traffic generation, we set and adjust the duty cycle with a beacon order (BO) and superframe order (SO), which is presented as expression (4).

$$duty\_cycle = \frac{1}{2^{BO-SO}} \tag{4}$$

Fig. 15(a) illustrates the number of network command packets as a function of the number of hops when duty cycle is set to 50% with beacon order BO = 8 and superframe order SO = 7. When the topology is a simple tree type and the number of hops is smaller than 3, legacy AODV and our proposed scheme show similar network overhead performance. However, When the number of hops is higher than 4, our proposed scheme shows better performance since it does not require to flood the RREQ control packets and the only thing to do

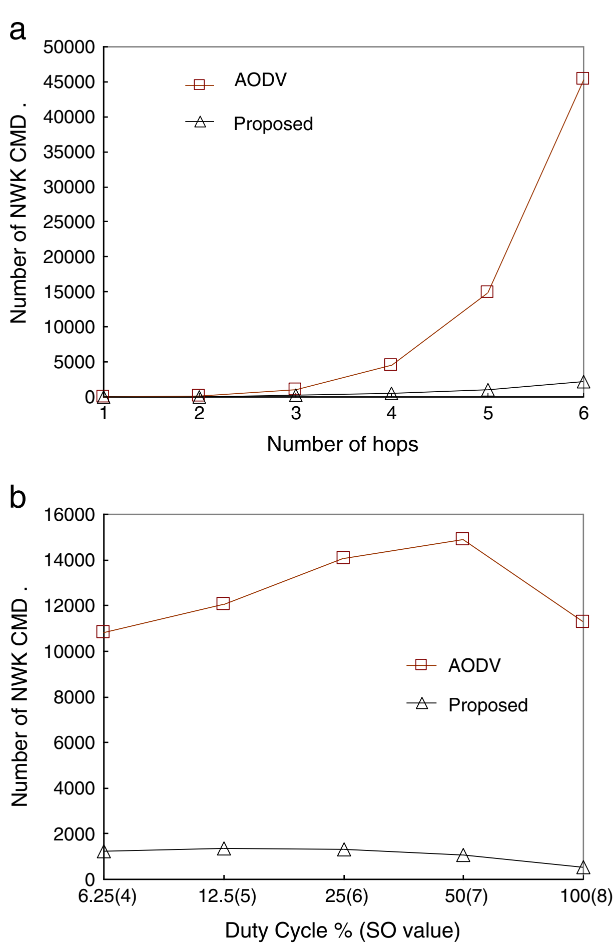


Fig. 15. The number of network commands.

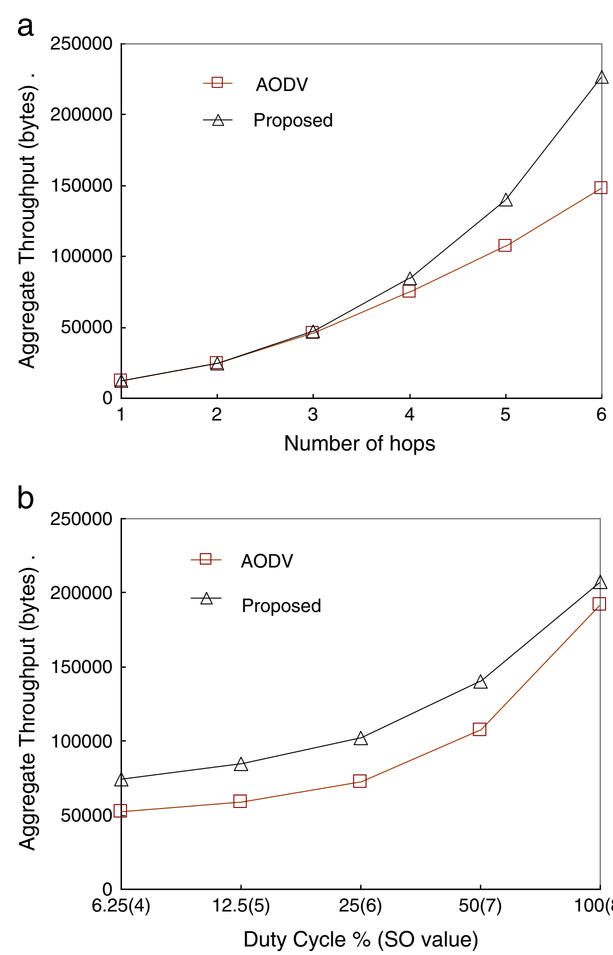


Fig. 16. Aggregated throughput.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

is to perform the association procedure. In Fig. 15(b), we also measured the number of network command packets as a function of network duty cycle. In this scenario, we adjust SO value between 4 and 8 while the number of hops is fixed for 5 with total 63 sensor nodes. Since we also set BO = 8, the duty cycle range is from 6.25 to 100. As shown in Fig. 15(b), proposed scheme shows significantly better performance regardless of duty cycle value when network has long hop counts. Thus, this result means that our proposed scheme has more chances to transmit data packets by mitigating control overhead in limited wireless environments.

Fig. 16 describes the aggregated throughput as the function of number of hops and duty cycle with same parameters in previous simulation of Fig. 15. Throughput performance is simply explained by comparing the network command performance mentioned in Fig. 15. Because the intermediate node on routing path does not need to relay control packets such as the RREQs and RREPs, it can transmit more data packets during the channel access time. On the contrary, in case of AODV, it should wait for another channel acquisition after unnecessary the route discovery procedure, which results in significant throughput performance degradation.

Fig. 17 shows packet delivery ratios according to the number of hops and duty cycle. This result also is correspondent to the previous simulation in Fig. 16. This means that when the number of hops increases, the performance gap between our scheme and AODV also increases. When AODV tries to establish the optimal route to the destination, it should use RREQ flooding. These flooding packets may collide with other data packets, which finally results in poor packet delivery ratio. Moreover, the collision problem is more serious when the number of hops increases because the number of flooding increases exponentially in the tree based network topology. The number of packet collision is illustrated in Fig. 18. The collision performance shows similar pattern with throughput results Therefore, in the large wireless sensor networks for wide farming areas, on-demand routing protocol such as the legacy AODV is not considered suitable for agricultural monitoring application with limited bandwidth.

6. Conclusion

The WSN technology with mobile robot systems is emerging for applications in agricultural monitoring services and WSN is assumed as one of the core components for developing the digital farms. However, farmers still have lots of difficulties to observe and analyze the whole measured information and cannot easily response to emergency condition such as occurrence of fire and fatal air pollutions.

In this, paper, we have designed and implemented a beacon mode based wireless sensor network system for agriculture applications in the TinyOS platform. Firstly, we define the message structure for agricultural emergency monitoring, then for energy efficiency and reliability, all sensor nodes can make a multi-beacon scheduling with BOP slots in the MAC layer. The network layer offers dynamic address allocations by using the LAA scheme and path recovery process for unexpected link failures as well as node mobility. In addition, we have developed a real-time agriculture monitoring server for whole sensor networks with various sensing modules. In order to verify the network performance, we manufactured various sensing nodes as well as a coordinator in real a test-bed. Throughout the packet loss rate measurement, we showed that our network architecture has the

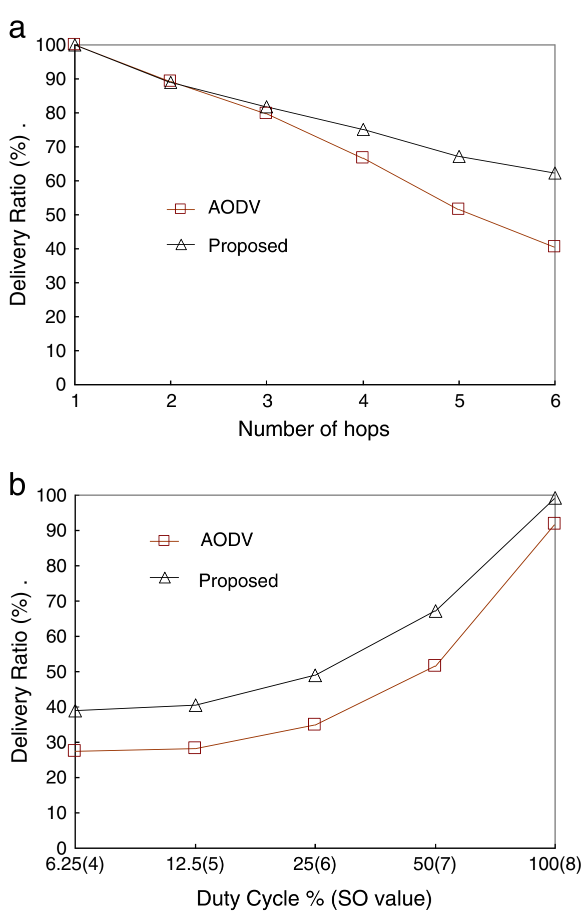


Fig. 17. Packet delivery ratio.

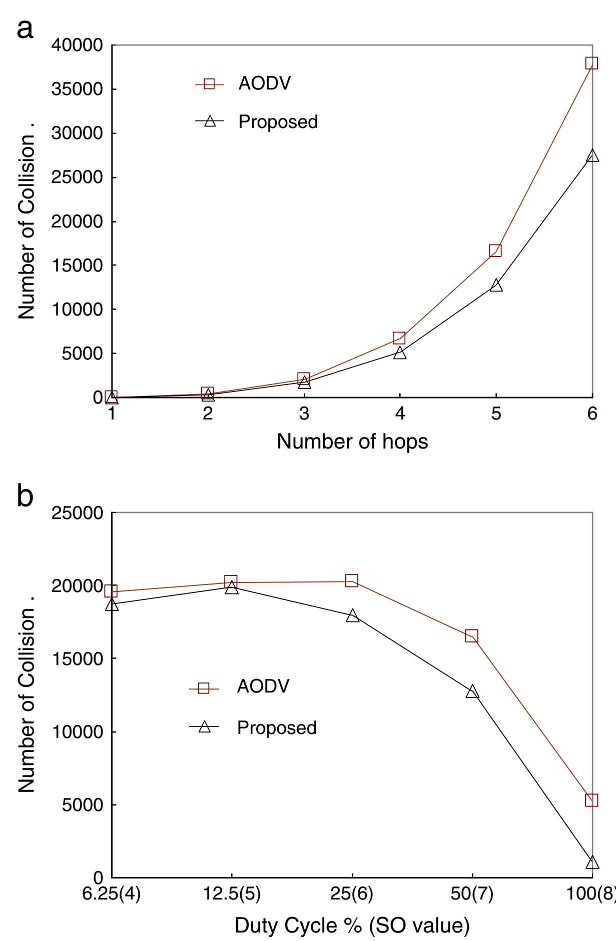


Fig. 18. The number of packet collisions.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004

reliable transmission ratio for a real-time processing service such as fire and emergency agriculture monitoring systems under heavy traffic environments.

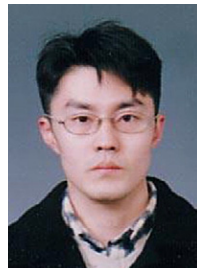
For the future, it is favorably recommended to design the seamless handover protocol for practical agricultural mobile nodes in more dynamic real environments. In addition, classifying the sensing packets according to QoS requirements for real-time multimedia traffic (e.g. voice and video) is suggested.

Acknowledgments

This research was supported by DGIST and funded by MEST (Ministry of Education, Science and Technology) of the Republic of Korea.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (March 2002).
[2] D. Gay, P. Levis, D. Culler, Software Design Patterns for TinyOS, LCTES, June 2005.
[3] D. Gay, P. Levis, D. Culler, The nesC Language: a Holistic Approach to Networked Embedded Systems, ACM SIGPLAN, 2003.
[4] Draft IEEE Std. 802.15.4, Part 15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), , September 2006.
[5] Ho-In Jeon, Y.S. Kim, BOP and Beacon Scheduling for MEU Devices, ICACT, Feb. 2007.
[6] Ho-In Jeon, Y.S. Kim, Efficient, Real-Time Short Address Allocations for USN Devices Using LAA Algorithm, ICACT, Feb. 2007.
[7] ISO/IEC JTC1/SC25, WiBEEM for Wireless home network services — part 1, 2, 3, ISO Standard Draft, March, 2008.
[8] B. Luca, C. Francesco, C. Giovanni, P. Davide, F. Romano, M. Antonio, M. Gianfranco, Agricultural monitoring based on Wireless Sensor Network technology: real long life deployments for physiology and pathogens control, Proc. 3rd International Conference on Sensor Technologies and Applications, 2009.
[9] R. Bradford, Krishna Shenai, Wireless sensor networks with embedded intelligence for agricultural applications, Proc. 27th International Conference on Microelectronics, May, 2010.
[10] P. Marino, F.P. Fontan, M.A. Dominguez, S. Otero, Deployment and implementation of an agricultural sensor network, Proc. 2nd International Conference on Sensor Technology and Applications, 2008.
[11] K. Srinivasan, P. Levis, RSSI is under appreciated, Proceedings of the Third Workshop on Embedded Networked Sensors, EmNets, 2006.
[12] ZigBee Alliance, URL:, http://www.zigbee.org.
[13] C.E. Perkins, E. Royer, Ad-hoc on-demand distance vector routing, Proc.2nd IEEE Wksp. Mobile Comp. Sys. App, 1999.
[14] David B. Johnson, David A. Maltz, Yih-Chun Hu, The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4, IETF RFC4728, Feb 2007.
[15] Sensor network analyzer (SNA), URL:, http://www.daintree.net/products/sna.php.
[16] Philip Levis, Nelson Lee, Matt Welsh, David Culler, TOSSIM: accurate and scalable simulation of entire TinyOS applications, SenSys'03, November, 2003.



Young-Duk Kim received the B.S. degree in Computer Education from SungKyunKwan University, Korea, in 2004 and the M.S. degree in Computer and Communication Engineering from POSTECH, Korea, in 2006. Since 2006, he has been with IT Convergence Laboratory, Daegu Gyeongbuk Institute of Science and Technology (DGIST), where currently he is a Senior Research Engineer. He is also Ph.D. candidate at Kyungpook National University, Korea. His research interests include robot communication systems, mobile computing, and intelligent embedded systems.

Yeon-Mo Yang received the B.S. degree in Electrical and Electronic Engineering from KAIST, Korea, in 1990. He received the M.S., and Ph.D. degrees in Mechatronics from GIST, Korea, in 1999 and 2006, respectively. From 2006 to 2008, he worked for embedded software research Laboratory in Daegu Gyeongbuk Institute of Science and Technology (DGIST). He is currently a Professor of Electrical and Electronic Engineering at Kumoh National Institute of Technology, Korea. His research interests include computer networks (PONs, WSNs), MEMs, robot control, and IT/MT fusion technology.

Won-Seok Kang received the B.S. and M.S. degrees in Computer Engineering from Yeongnam University, Korea, in 1998 and 2000 respectively. Since 2005, he has been with Robot Technology Laboratory, Daegu Gyeongbuk Institute of Science and Technology (DGIST), where currently he is a Senior Research Engineer. His research interests include distributed simulation algorithms, mobile robot systems, and intelligent embedded systems.

Dong-Kyun Kim received the B.S. degree in the Department of Computer Engineering, Kyungpook National University, Daegu, Korea. He also obtained the M.S. and Ph.D. degrees in the School of Computer Science and Engineering, Seoul National University, Seoul, Korea. He was a visiting researcher at the Georgia Institute of Technology, Atlanta, in 1999. He also performed a postdoctorate program in the Computer Engineering Department, University of California at Santa Cruz, in 2002. Currently, he is a professor in the Department of Computer Engineering, Kyungpook National University, Korea. His research interests include mobile computing, wireless network protocols, and next-generation all IP networks.

Please cite this article as: Y.-D. Kim, et al., On the design of beacon based wireless sensor network for agricultural emergency monitoring systems, Comput. Stand. Interfaces (2011), doi:10.1016/j.csi.2011.05.004