The Design of Multi-Hop Routing with Asset Monitoring Application Using Wireless Sensor Networks

Hojung Lim, Jeong-Hoon Kang, Min-Goo Lee, Joon-Jae Yoo, Myung-Hyun Yoon, and Myung-Soo Lee
Korea Electronics Technology Institute
{hlim,budge,emingoo,jjyoo,yoon,leems}@keti.re.kr

Jeong-Hun Choi
Korea Polytechnic University
jhchoi@kpu.ac.kr

Abstract

The sensor network supports data delivery from physical world to cyber space. The sensors get physical events then wireless network transfers sensor data to service server. We use sensor network technology to manage location information of an asset. In ubiquitous computing environment, user localization is basic context for intelligent service. A lot of research groups have made efforts to develop low cost localization technology. In this paper, we propose the design of multi-hop routing with asset monitoring system using wireless sensor network. It is implemented using ad hoc network technology which can be adapted to smart home and this system can monitor asset location and movement. We also test the performance of the routing engine with asset management system in a real office.

1. Introduction

The sensor network is a network to consist of sensor nodes which aware and measure physical data such as light, sound, temperature, movement in physical space and transfer them to a base-station or sink. The sensor network is generally composed of several distributed sensor nodes such as multi-hop wireless network. The sensor nodes make up more than one sensor (temperature, sound, light, acceleration, location, magnetic, etc.), actuator, micro controller, several tens of EEPROM, several KB of SRAM, several hundreds of flash memory, and local wireless network modules. The sensor network technology transforms analog data which is measured using sensors and wireless network functions in physical space to digital signal and transfers to root node to connect electric space such as internet. The sensor network can be applied for intelligent services of new concepts such as intelligent environment monitoring, location-awareness service, smart medical system, intelligent robot system because all have the characteristic to connect physical world to be cyber space.

The advantage of sensor network is that sensor network uses low hardware specification and it is composed of ad hoc network. For example, wireless network technologies so far developed, such as Bluetooth and wireless LAN need high computing devices like PDA but sensor network node forms network independently. It can be used for base technology in ubiquitous computing environment due to the ease of constructing network. Intel expects technology roadmap based on sensor network [1].

Recently, chips and systems related to sensor network started to be commercialized but there are some problems to solve for distribution to industry. We can also predict that it is hard to commercialize in short time because of absence of network protocol, appropriate application service, etc.

TinyOS is a system technique to use in new computing paradigm of low small computer system, viz., network embedded system or sensor network. UC Berkeley mainly researches and corresponds fast to implement and apply sensor network area. Intel and Crossbow continuously invest to spread and Microsoft tries to possess related technologies.

TinyOS superficially looks different from ZigBee specification but main factor techniques aim to implement the same functions. For this reason, many people are highly interested in companies related to two technologies as two technologies try to implement seriously and industrialize.

In this paper, we analyze TinyOS and multi-hop sensor network technology to implement asset monitoring application service using wireless sensor network and propose system architecture and implementation method for application. Also, we summarize real implemented results and future considerations.

2. Related technology trend

2.1. TinyOS and NesC

There are three aims to develop TinyOS [2]. In the first, it designs software architecture to use sensor network node in the future. In the second, it designs to implement service application and operating system for diverse hardware platforms and software libraries. In the third, it designs to satisfy, with limited resources, simultaneousness, robustness, and requirements for different applications.

TinyOS consists of modules and implements using event-driven with limited resources to reach above aims. The operating system is able to apply on diverse hardware depending on compounding of modules and to maintain same abstraction of service application.

TinyOS does not separate system and user part. But there are frequently used modules such as timer, data receiver, power control, and network module in most of the applications. The usage of these modules is for acting node with limited power in sensor network, for gathering data periodically, for processing simple data, and for sending data to near nodes. Other sensor network platforms such as MANTIS [3], SENSORSIM [4], EMSTAR [5] have same characteristics.

TinyOS is open-source and has same advantages as open-source software. The principle of open-techniques such as Linux or BSD may be industrialized and extended techniques faster.

TinyOS has just single thread but uses Task and Event concepts to implement Multi thread. The following Figure 1 shows the entire operations of TinyOS.

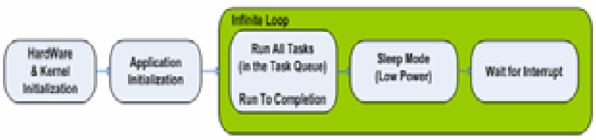

Figure 1. Entire operations of TinyOS

NesC is a programming language extended C for NES (Network Embedded System). That is, it consists of components and bidirectional interfaces and supports simultaneous model based on Task and Events. The following Figure 2 explains the concept of NesC.

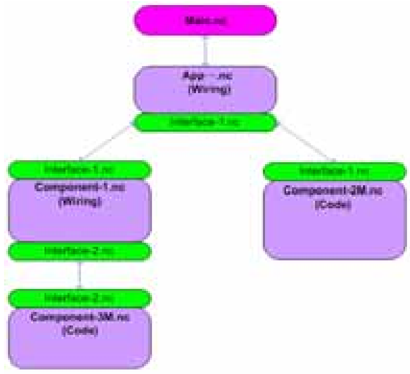

Figure 2. the concept of NesC

NesC consists of components and there are two kinds of components, Module and Configuration. Module provides application code or performs more than one interface. Configuration performs to wire interfaces which are used by component.

Components *provide* or *use* interfaces. Interface has interactive and use *command* and *event*. That is, Provider implements command but user implements event.

2.2. Multi-hop Network

The multi-hop network algorithms of MultiHopRouter, TinyDiffusion, GPSR, and BYR search next nodes which may be decided path and maintained table. This information is used in re-establishing path when routing path decided at the first time and created or deleted nearby nodes when network environment changed. Next node table includes routing metadata such as node address, link state, the number of hop. The link state is used in deciding path and the number of hops is used in managing table.

The implementation of the multi-hop routing protocol in the beginning of TinyOS had many errors to transfer [6] but MultiHopRouter, BYR, TinyDiffusion, and TinyOS DSDV using link state estimator solve above problems [7]. Also, this multi-hop routing starts to use *Send* and *Intercept* interface and it made nodes resend data before data processing because it includes *getBuffer* command to create efficiently packet Intercept event called when resent packet sends. Also, it gathers more accurate information of near node because it adds function to send packet of next node in lower network stack. It adds output queue, creation queue, transmission queue and increases transmission utility.

The implementation of the routing protocol in TinyOS overcomes disadvantage of the first version such as delay time, failure of forming network, and packet loss because it creates table about near nodes, sends packets to other node, and uses message queue.

3. The asset monitoring system design using multi-hop wireless sensor network

The aim of the asset monitoring system design in this paper is to find location of wireless sensor network node. To find location is decided to depend on base station location included in wireless sensor network node.

3.1. System components

Wireless sensor node forms networks to connect to one base node. The composed network transfers data to server to installed application through gateway. The gateway connects wireless sensor network to Ethernet and the location of the gateway is registered to the application when it is installed. Figure 3 is the proposed monitoring system in this paper.

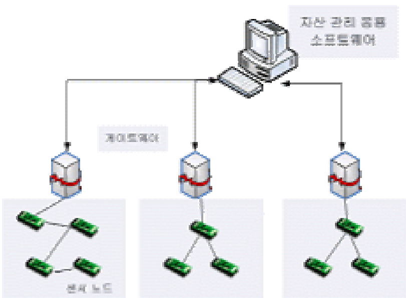

Figure 3. Asset monitoring system components

KETI developed sensor node, TIP-30C, with 916MHz to implement the system to execute efficient by tracking assets. It has humidity, temperature, and light sensors and is one board included in sensing module, processing module, and communicating module. Also, it made senor network platform tiny with Inverted F-Type ceramic antenna. TIP-30C includes in TinyOS, ATMEGA 128 (8 bits processor) by Atmel, 128KB flash memory, 38K baud data transmission rate, UART interface to communicate to computer, 3.0V power (1.5V AAA connected in series), and performance of 15M RF range. Figure 4 is TIP-30C for sensor node.

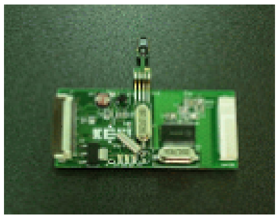

Figure 4. TIP-30C for sensor node

TIP-50G supports for connecting sensing data transferred by TIP-30C sensor network node formed by multi-hop wireless network, downloading diverse applications developed by user using TIP-30C, and forwarding sensing data connected to related server through Ethernet in TIP-50G.

Especially, TIP-50G is Mother/Daughter Board type and PoE (Power of Ethernet) type which supports power by Ethernet without power adaptor. Figure 5 shows TIP-50G for sensor network gateway.

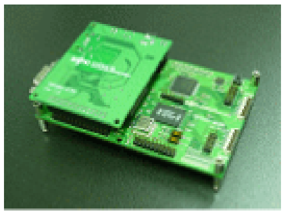

Figure 5. TIP-50G for sensor network gateway

3.2. Routing core component

In this paper, we aim to design the routing engine with fast response to move/change object based on DSDV (Destination Sequenced Distance Vector) to implement an efficient monitoring system. For above this, we need link estimator component enabled efficient link estimation, neighbor/routing table to save link state and routing information data, management component to maintain/operate table periodically, and selection component to choose the appropriate parent.

Figure 6 shows routing engine diagram to connect components proposed for the routing engine in this paper.

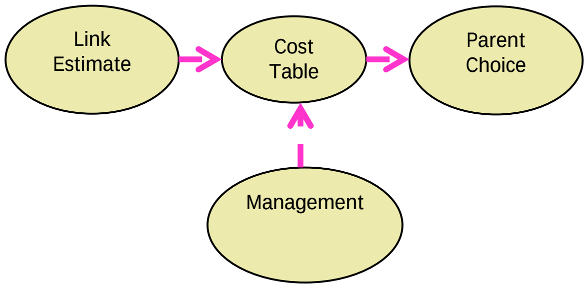

Figure 6. Proposed routing engine diagram

Figure 7 shows the data message and the route message structure used in the proposed monitoring system in this paper.

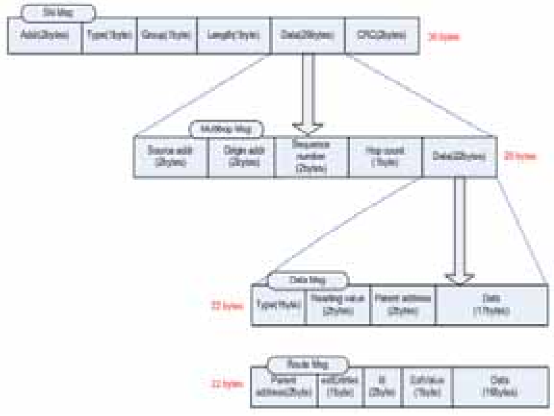

Figure 7. Data/route message structure in proposed monitoring system

In Figure 7, SN Msg header is processed in Data Link Layer and Multihop Msg header is processed in Network Layer. And data message or route message header is processed in Application Layer. Data payload of Route Msg to insert related data and to send and receive among sensor nodes completes Neighbor table.

Management component maintains/operates six good quality Neighbors in the table. We copy Neighbor table to within routing table and sort by Link value and the number of Hop count then setup Parent if Neighbor is the best quality Neighbor.

Figure 8 shows used Neighbor table in this application. Seq. No in Neighbor table such as DSDV (Destination Sequenced Distance Vector) which distinguishes new route from old route of sensor nodes and uses Loop-free of routing at the same time. And Hop Count and EstValue used to item to select Parent.

Neighbor Table

NodeID (2 byte)	parentID (2 byte)	Seq. No (2 byte)	Hop Count (1 byte)	failCount (2 byte)	rcvCount (2 byte)	EstValue (1 byte)
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue
NodeID	parentID	Seq. No	Hop Count	failCount	rcvCount	EstValue

Figure 8. Used neighbor table

The designed sensor network routing engine in this paper executes periodically by timer interrupt and executes to create and send/receive data message when routing message arrived.

Detailed operations are expected to following four cases:

- periodic execution by timer interrupt
- arrival routing message
- After got data, transferring data message
- Receiving data message

Figure 9 shows detailed process flow chart in this paper.

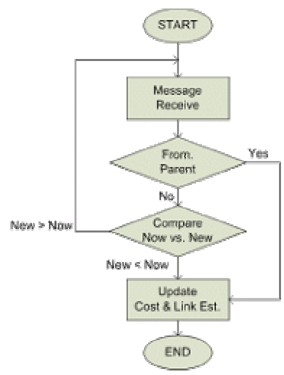

Figure 9. Detailed process flow chart

3.3. System implementation and test

Each asset attaches wireless sensor node and inserts itself information about itself per unique ID. The asset monitoring application software tracks asset's contents, location, and movement. Figure 10 shows monitoring display normally formed with real network topology with sensor network nodes using experiments in this paper.

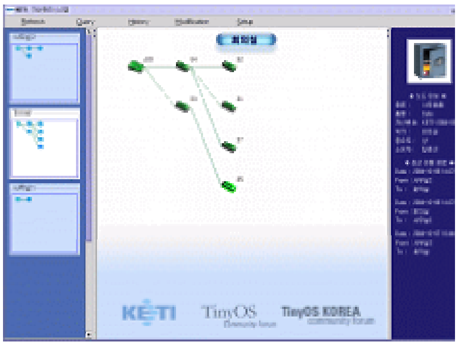

Figure 10. Network components per location

Figure 11 shows moving location by arrow in this application software when the location of assent is changed. New moved node has changed color to indicate it is moved.

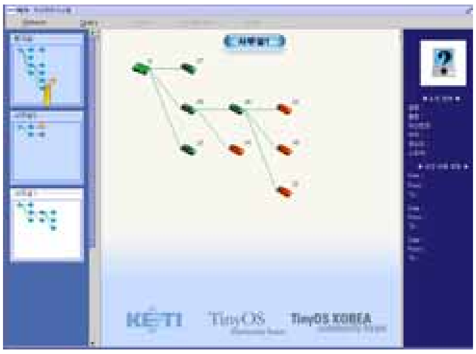

Figure 11. Sensor network node moving

Efficient assent management has to provide to integrate asset location monitoring and search. Figure 12 shows search interface to support for application software. When we search objects in the office, we fill out asset ID, owner, importance, etc. in the search window and push Query button.

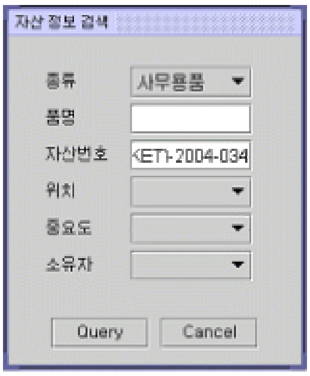

Figure 12. Sensor network node search window

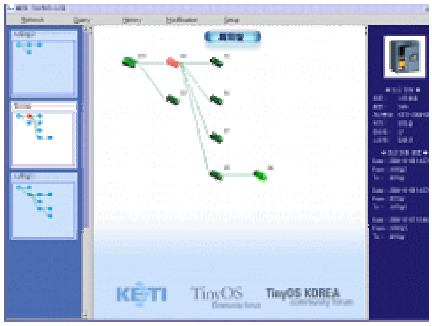

Figure 13. Node indication of searched sensor network

Figure 13 explains searched sensor network node and location of asset. The searched asset ID is 34.

To compare performance with routing engine provided in this paper, we have the following results with Surge application by UC Berkeley. Figure 14 compares and analyzes time of forming network topology.

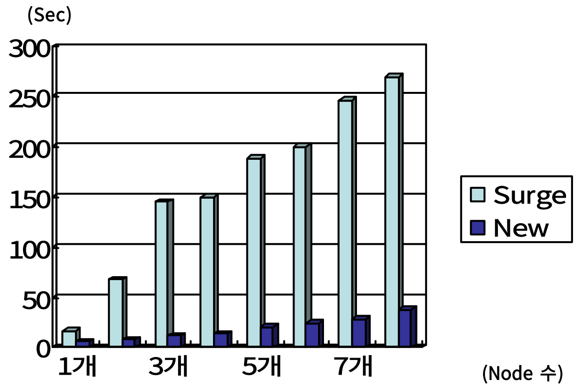

Figure 14. The result of forming network topology with Surge and New

In this experiment we use eight nodes and measure time to form entire network topology in ten times. In routing algorithm of this paper, it takes 40 seconds to form network topology with eight sensor nodes but it takes more than 5 minutes by Surge. That is, response time is improved compared to existing Surge.

4. Conclusions

This paper proposed design and implementation methodology of the asset monitoring system using sensor nodes to input event of physical space in ubiquitous computing environment. It consists of multi-hop network per base station. And to find location can be used in home and office environment. We also verify that our design of sensor network routing engine is more efficient. It is possible to manage asset efficiently because it saves current location and past moving path of asset.

It can happen that response of location-awareness becomes slow when assets move a lot in the proposed system components. We need to research further to use additional information to aware accurate location for asset management which moves a lot.

In the future, we need to try to deduce efficient commercial service by developing technology to find new technical requirements through implementation of diverse application service environments using wireless sensor network.

5. References

[1] http://www.intel.com/research/exploratory

[2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors", *In Proceedings of ASPLOS*, , Boston, MA, USA, Nov. 2000, pp. 93-104.

[3] H. Abrach, S. Bharri, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Suucker, J. Deng, and R. Han, "MANTIS: system support for MultimodAl NeTworks of In-situ Sensors", *In Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, ACM Press, 2003, pp. 50-59.

[4] S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: a simulation framework for sensor networks", *In Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ACM Press, 2000, pp. 104-111.

[5] J. Elson, S. Bien, N. Busek, V. Bychkovskiy, A. Cerpa, D. Ganesan, L. Girod, B. Greenstein, T. Schoellhammer, T. Stathopoulos, and D. Estrin, "Emstar: An environment for developing wireless embedded systems software", *Technical Report 0009*, CENS, Mar. 2003.

[6] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks", *Technical Report 02-0013*, UCLA Computer Science Division, Mar. 2002.

[7] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges for reliable multihop routing in sensor networks", *In SenSys '03*, Los Angeles, California, Nov. 2003.