Web-supported Mesh Sensor Network Technology for Asset Management System

Min Goo Lee, Myung Hyun Yoon**, Jung Hoon Kang**, Jun Jae Yoo**,Yun Sik Lee**, Han Jong Ryu****

*Dept. of Ubiquitous Computing, KETI
**Dept. of Ubiquitous Computing, KETI
***Dept. of Information and Communications, Hanyang University
E-mail: emingoo@keti.re.kr, yoon@keti.re.kr, john1126@gmail.com

ABSTRACT

Even though many fields in Sensor Network have been proposed, Asset management system is the one of most important application. That is because this application requires the factor of low cost, scalability, simple monitoring operation that Sensor Network can provide very well.
But many asset management applications in Sensor Networks are not useful to use in real industry. Because asset management have to be more easily accessed at any time, anywhere by users. That means it needs to provide web browser services. So users can access to their asset and control it more efficiently.
The asset management application with web browser is not that simple just only merging 2 kinds of techniques. But also it means great extension to the Ubiquitous service. In this paper, we will show you the benefits and implementation of web accessible asset management system. Also we will propose the problems which could be occurred when we use web services in asset management system in Sensor Networks.

Keywords: Sensor Network, Mesh Network, Asset Management, Web Browser

1. INTRODUCTION

The words Wireless Sensor Network have been used in wide usage at technology sessions of various societies and in various industrial sectors for many years. This shows a stronger demand than ever of the diverse contributions by sensor networks in preparing for the ubiquitous computing age coming soon.
Since this paper is a proposal on developing more effective (any time, any where and any device) asset management applications by using web server services.
The wireless sensor network sector is being led by the NEST(Network Embedded Systems Technology) project, which is led by the team of Professor Culler at U.C Berkeley. Wireless sensor network is literally a technology giving wireless networking function to existing sensors once used with a function limited to sensing, and has the purpose of appropriately processing environment data obtained through sensors, and then effectively transmitting it to the destination through wireless RF[1].
The study on wireless sensor network technology is in progress in various sectors including study on diverse forms of sensor node development[2], study on efficient MAC(Medium Access Control) in terms of energy, study on strong and efficient routing algorithm, study on effective storage and processing of obtained sensing data, study on security of data transmission, and study on applications that can be linked to sensor networks.
Among the diverse sensor network study areas, this paper is about developing asset management applications using sensor network system. Our proposed asset management applications provides more practical use, and more effective approach in order to utilize asset management application more efficiently through web services.
The concept of our proposal about asset management application is very critical issue in practical usage of Sensor Networks even though it looks like quite simple idea.
Usually Wireless Sensor Networks are defined as a bridge which correlates the real world with a cyber world which implies Internet. So, providing web services are very important for the roll of Sensor Networks.
Furthermore we have to make sure that Sensor Networks are not similar to the existing Ad-hoc networks, which means that Wireless Sensor Networks has a nature of low power, resource and area limitation in contrast to the existing Ad-hoc network. Therefore it's not efficient for Sensor Networks to adopt the theory of the existing Ad-hoc networks and Sensor Networks are growing independently.
The purpose of this paper is to develop an asset management application which we can use it anywhere and anytime practically.
In chapter 2, we mention the platform and routing algorithm which are used in our implementation for sensor devices. In chapter 3, we explain the web application

system that anybody can access the information of remote sensor devices more easily without any area and time constraints. In chapter 4, we propose the problems which is possible to be occurred through our web accessible asset management system.

2. SENSOR NETWORK

2.1 Sensor Network Platform

The representative platform which is used most widely in the world for Sensor Networks is the MOTE from U.C Berkeley. So far, there are lots of mote series(DOT, MICA, MICA2, TelosA, TelosB, Micaz, etc) and their clones which are derived from the Berkeley's mote.

Our platform for this paper is TIP 700CM developed by Maxfor. As shown in Figure 1, TIP 700CM uses the inverted F-Type ceramic antenna using 2.4 GHz wireless RF band[3], and supports various sensor(light, temperature, humidity) functions.

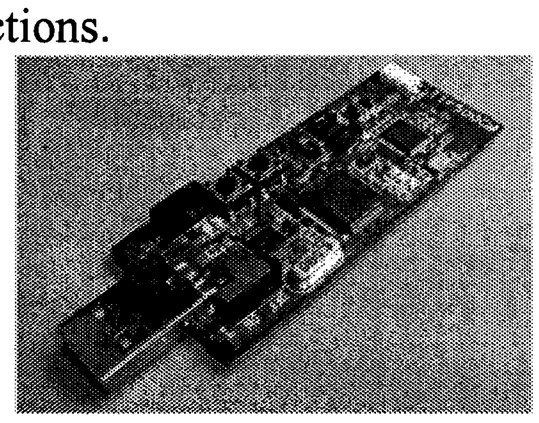

Figure 1: TIP 700CM by Maxfor

Also it uses MSP430 F1611 micro controller from TI and provides 48KBytes of flash memory and 10KBytes of RAM. TIP 700CM is AAA size battery powered system. Particularly this TIP 700CM has a very useful feature of USB program interface in contrast to the previous platform.

2.2 TinyOS and NesC

In this paper, the OS for sensor network node used TinyOS distributed by U.C Berkeley. TinyOS is a specially made OS for Network Embedded Systems like the sensor network. This was designed to create an ultra-mini capacity OS with small data memory with event-based application and mini core OS (about 400 byte code), and it only provides an event-based simple scheduler function.

As shown in Figure 2, TinyOS is an OS with limited functions, only supporting a scheduler function. TinyOS checks the existence of tasks to be handled in the task queue with FIFO structure, and processes them. When this process is done, it enters Sleep status, and when an event occurs, the event handler operates according to the interrupt vector based on the interrupt service routine (ISR). When this process is done, it returns to Sleep state.

As such, designed to be used in the sensor network, the time sleep state is maintained for TinyOS is over 99%, and since it is designed operate and maintain long-term continuous node time, it may be an ultra-mini OS appropriate for sensor network nodes with limitations in terms of resource.

NesC is a language which contains many features for TinyOS. As such, its application code is wired to components with bidirectional interfaces and it provides concurrent model with tasks and events.

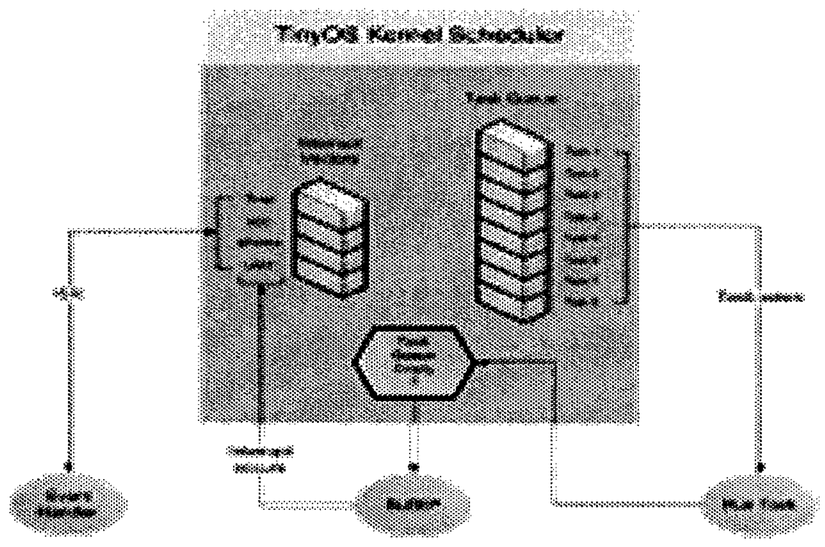

Figure 2: TinyOS Kernel Scheduler

Therefore NesC could support many various features of context awareness, concurrent modeling, and event message communication. Furthermore implementation of application is very effective with NesC due to it finds out data race conditions at compile time. Also it does reduce code size and remove lots of possibilities of bugs.

NesC seems to be look like extension of C language in large point of view. The reason why NesC is derived from C is that C is most effective language for micro controller. And the other reason is C is most widely used by programmers.

2.3 Routing Algorithm

The ultimate goal of the routing algorithm is to find a method for efficiently identifying the best performing route to reach the destination node for transmitting sensing data. For this, we divide routing algorithm into 4 components below: (1)Neighbor and Routing Tables to save link quality and routing information, (2)management component which manages these tables above, (3)Link estimation component for quality of link, and (4)Parent selection component.

2.3.1 Link Estimator

As shown in Figure 3, Link estimation generates when source node tries to search target node.

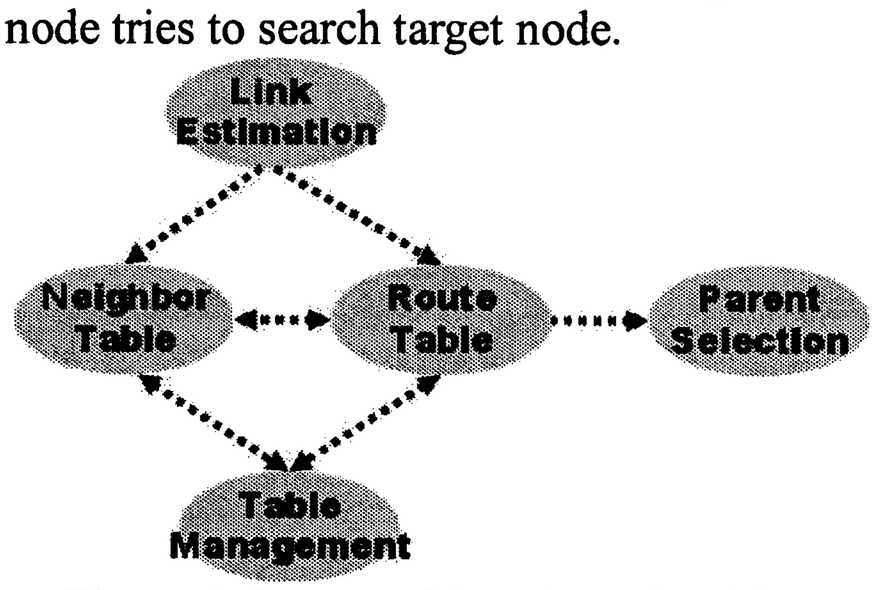

Figure 3: Proposed Routing Algorithm

Each of nodes estimates the RF link quality through the rate of success/failure of incoming/outgoing packets of the

link. The estimation methods are shown below.

1. Check the value of Neighbor Table is exist or not.
2. If it's exist, parent selection algorithm could be started.
3. whenever node gets the RSS value 5 times, update this value on the Neighbor Table.

2.3.2 Neighbor Table Management

Neighbor table management is a period of sequence that sorts the value of table data in order to figure out the best link quality and inserts a new value which is better than old values.

Table management consists of 3 operations. As such, "insert/delete/refresh". After analyzing the link quality through incoming packets, table manager decides whether insert it or not. When the quality of incoming packet is better than old one, it checks the source node of incoming packet is listed on the table or not. If it is exist, refresh operation will be followed. But if it's not exist, insert operation will be followed. Also whenever the table has to be exchanged with new neighbor's value, it operates deletion.

2.3.3 Parent Selection

Parent selection chooses the most efficiency Neighbor which is satisfied with the conditions. And it publishes this neighbor as a parent node. It implies more stable parent condition if the parent selection is not occurred many times, and stable parent is useful for routing because it's predictable.

The factor of cost which is used for parent selection is as follows.

1. Estimated values of incoming/outgoing packet
2. Number of hop count

3. WEB APPLICATION SYSTEM

In this paper, we propose the idea of web accessible asset management system in order to achieve more effective usage in practical way. As such, we can access and obtain the information from the sensor devices more effectively and easily, even wherever it's located or whatever platform it is.

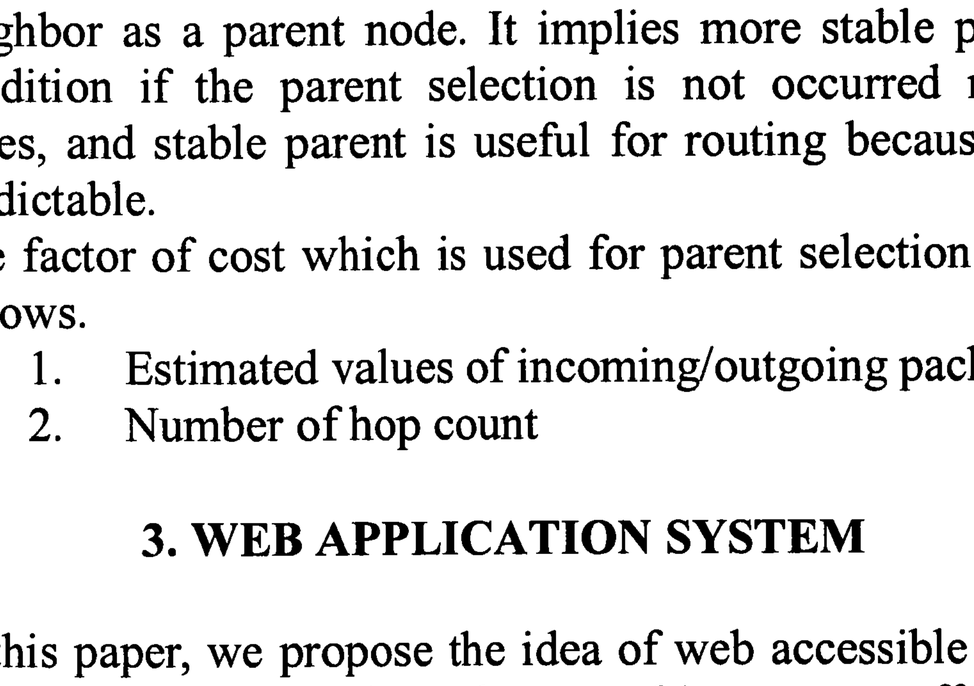

Figure 4: Architecture of Web Accessible Application

supporting Mesh Network Topology

In figure 4, it shows the architecture of our Web application system that supports Mesh network topology. As above, it consists of sensor nodes, received packet server, DB server and web browser.

Each sensor node sends both sensing data and routing messages toward base node. Then base node delivers these packets into received packet server.

The received packet server parses incoming packets when there is a new packet. Then it could figure out node's information, neighbor's quality, sampled sensing data from parsing the packets. These obtained information is loaded on the DB server.

The key point of this architecture is that there is no update without any change in DB server, because it's impossible for the web browser(client) to require the DB server data repeatedly. If client really requires the DB server data repeatedly, it causes huge load of data from the DB which is not necessary. So we need to make sure that the web browser(client) only requires the updated DB server data.

For this features, we have implemented our web accessible application system with a Java program. The reason why we select Java is that it supports xml format by Java script. And this function is very useful to make client only download updated data. So the browser of our web accessible application system can update the image information without reloading web browser.

As such, Java script requires only changed data from the DB server and draws the only changed images on the web browser.

4. EXPERIMENT

4.1 Environment of experiment

Figure 5 is the environment for experiment to analyze performances of the web accessible asset management system. For nodes, a total of ten TIP 700CM, laptop (SONY PCG-TR2: 1G CPU, 1GRAM) was prepared including the base node, and also another laptop(SONY PCG-TR2: 1.2G CPU, 1GRAM)is prepared for remote connection to the web server which implies DB server. Also in order to make the sensor node height identical, all nodes were allocated on top of office partitions.

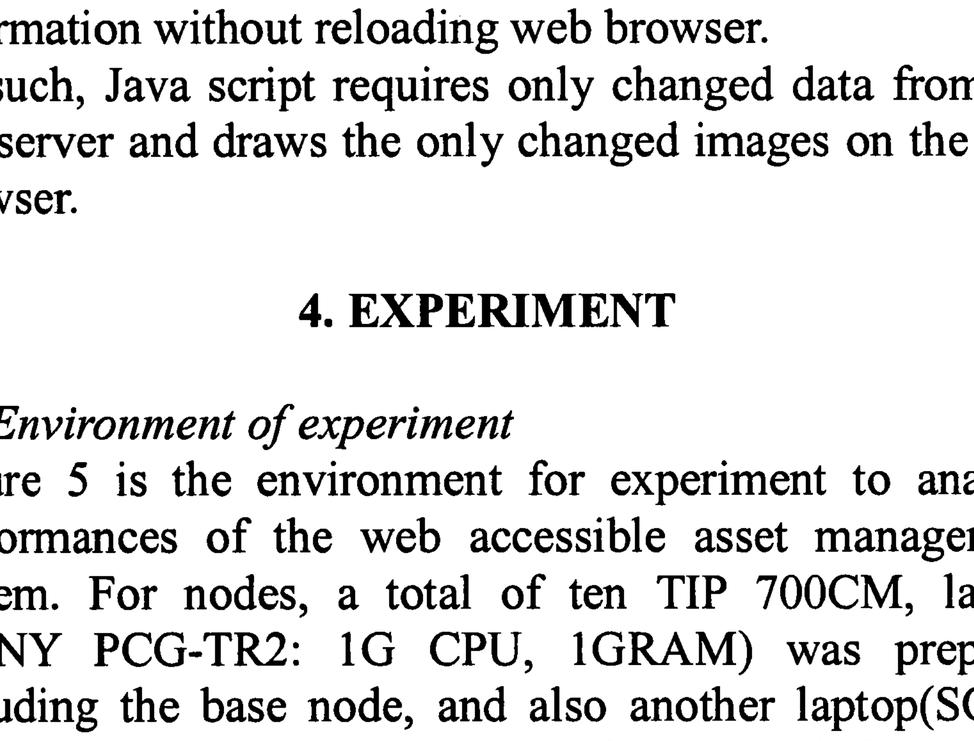

Figure 5: Environment of experiment

4.2 Result of experiment

We developed monitoring tool as shown in Figure 6 in order to calibrate performance of web accessible asset management. This tool is accessible through web browser in anywhere, anytime and supports monitoring Mesh network topology. Web accessible asset management tool let us be able to connect and obtain the sensor device's information wherever if it is possible to access internet.

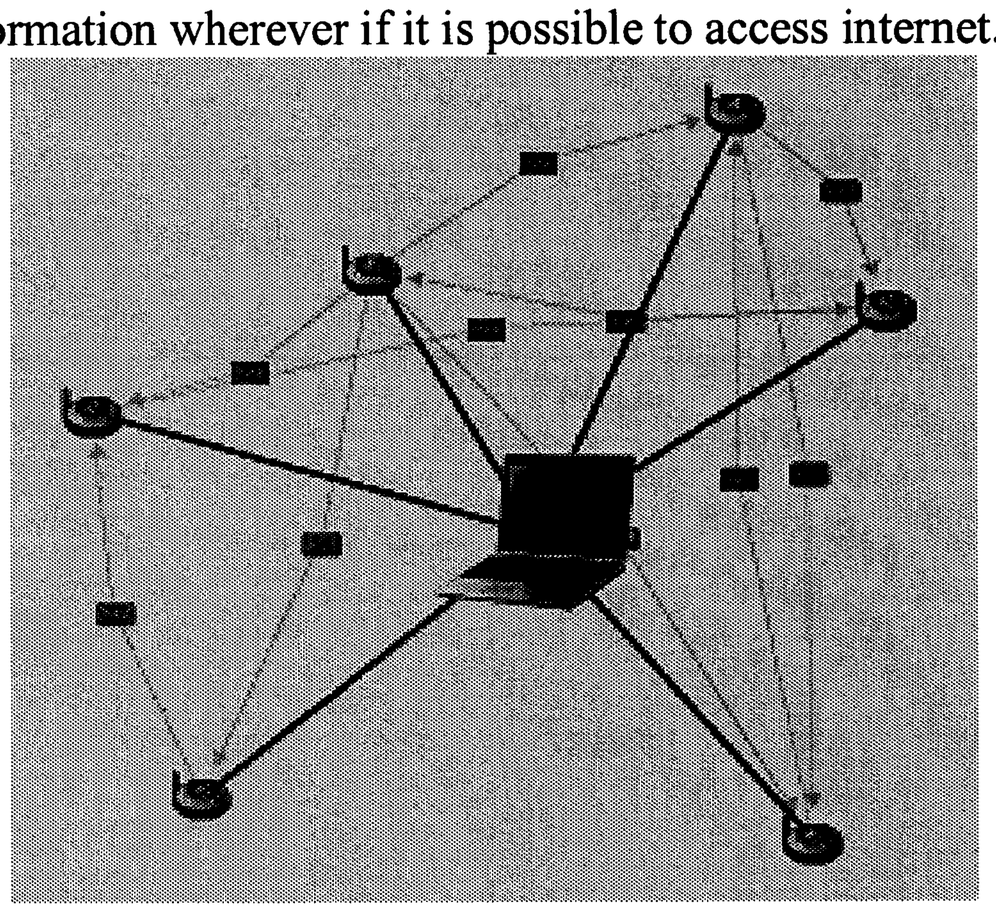

Figure 6: Mesh Network Topology

As shown in Figure 7. it's really easy to obtain sensor device's temperature with any terminals such as laptop, PDA, etc.

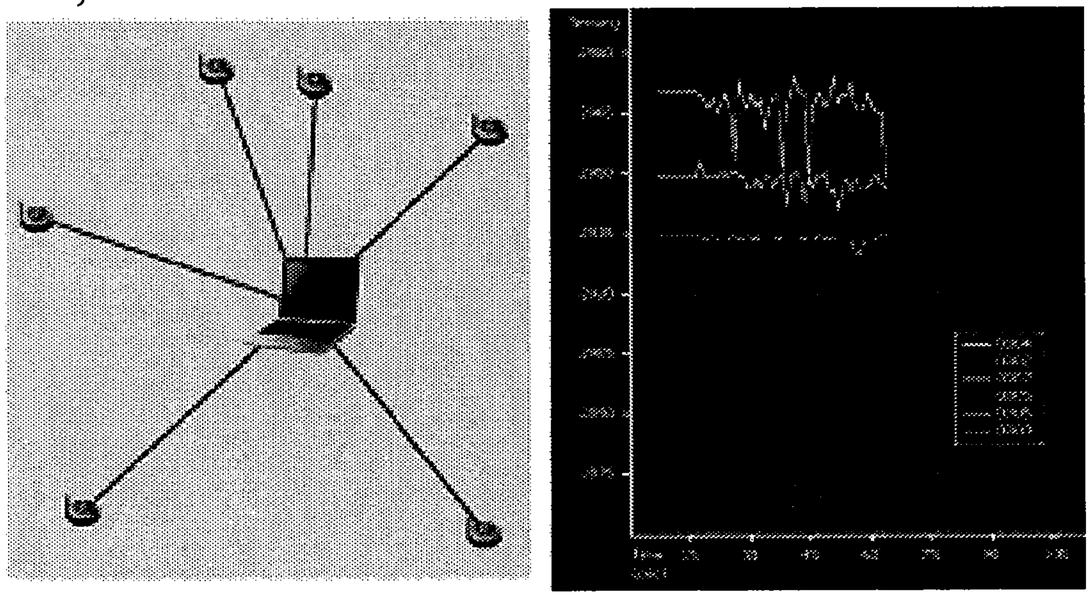

Figure 7: Monitoring Temperature of Sensor Nodes

There is analysis of 2 different routing algorithms in Figure 8. We can check the fact that time performance of create network topology degrades rapidly after the nodes are over 10. However we can also check our proposed routing algorithm(NEW) has better performance than previous algorithm(OLD) using RSS.

We found out the reason of decrease is that drawing lines on the web browser cost lots of load due to web browser doesn't have a drawing functions. And the other fact is Mesh network topology force to draw huge lines rather than general topologies. There was no problem when we tried to access to star or tree topologies.

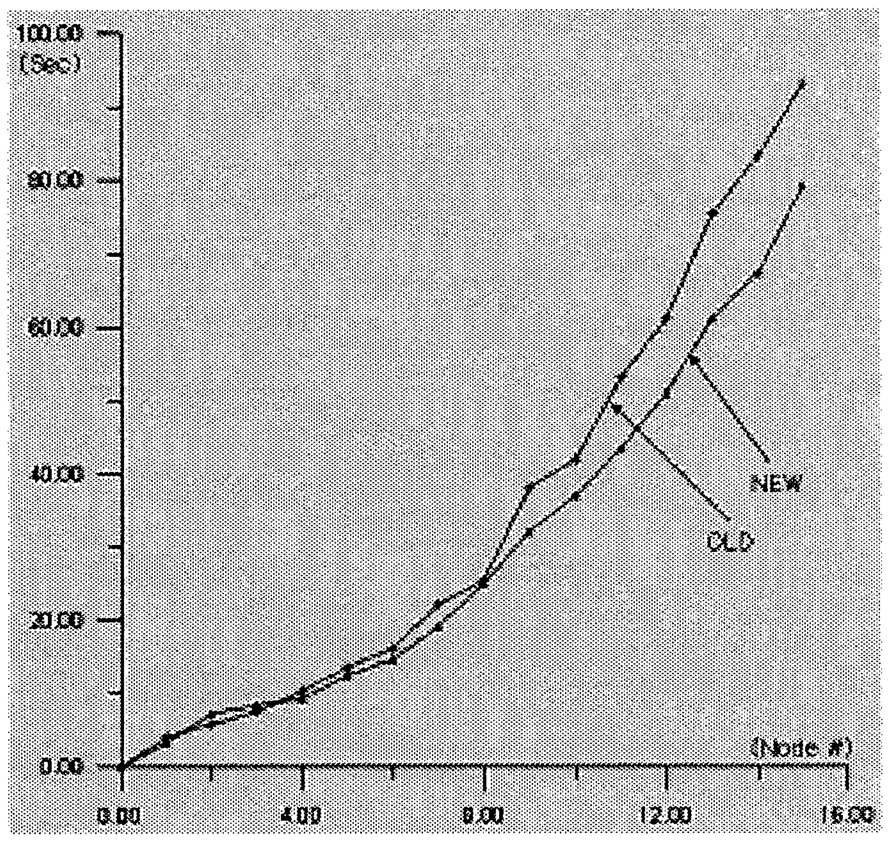

Figure 8: Experiment result of Network Topology

5. CONCLUSIONS

This paper proposes merging web services with wireless Sensor Network to achieve more effective usage. As a result of our experiment, it's possible and useful to connect to the sensor device's information through Internet.

But while we were implementing our web accessible tool, we found out very critical problem of scalability. When the sensor nodes are over 10, the performance of Web services degrades obviously.

Also we implement our web server on the laptop now, but the web server should be ported to low cost gateway or mobile platform.

As a conclusion, we suggest that web application service is very critical issue to make Sensor Network more widely. And also we have implemented web application service with asset management system. And we propose the possibility of problems that can be occur when merging web service with Sensor Network.

6. REFERENCES

[1] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister. " System architecture directions for network sensors " ASPLOS 2000, Cambridge, November 2000.

[2] Joseph Polastre, Robert Szewczyk, Cory Sharp, David Culler. " The Mote Revolution: Low Power Wireless Sensor Network Devices" Proceedings of Hot Chips 16: A Symposium on High Performance Chips. August 22-24, 2004.

[3] "ChipCon CC2420 Data Sheet," http://www.chipcon.com/les/CC1000 Data Sheet 2 1.pdf