A Study of Middleware for Real-time Application with USN

Chun-Gun Yu*, Young-rag Kim*, Chong-Gun Kim*, Byung-Wook Kang**

*Graduate School of Computer Engineering, YeungNam University

**Dept of Computer Engineering, YeungNam University, Korea

Abstract

The current middleware, managing the bridges equipped whit sensor networks on real-time basis, merely processes the data collected from the sensor networks.

In this study, we develop a very versatile middleware which mediates between sensor nodes and exclusive applications or web browsers. It collects and processes the data from the network. On database failures, it stores the data in a file and sends them to the database at a proper speed considering the rate of the database system when the database is restored.

Lastly, in order to notice abnormality of the bridge to the operators when some problems occur in the bridge, we propose a mechanism for producing alarm signals.

1. Introduction

These days, broadband networks and technology fusion are accelerated for ubiquitous computing environment in the whole industries as well as telecommunication. Solutions on wireless sensor networks are being developed under ubiquitous environment. These solutions have computing functions like propagation identification of all objects and are paving the way for an era of ubiquitous sensor networks where all objects are connected[1-3] Sensor networks technology where outside events are detected and delivered on sensor networks is actively researched [4]. In a field of road maintenance, sensor networks technology is being utilized and it is partly considered in intelligence traffic information system. Checking condition of bridges or roads on the spot has many problems in an aspect of efficiency, reliability, rapidity and stability. Using sensor networks, it is possible to examine condition of every remote spot on a bridge on a real-time basis and promptly take actions when problems occur through exclusive applications or web browsers. Previous studies suggested a fact that sensor networks technology can be employed in bridge management but they didn't embody standards and functions of middleware. Because most of the previous studies don't provide enough information for essential functions and efficient architecture of bridge management system, people might have some difficulties when implementing the system.

In this study, we specify functions and architecture of real-time middleware for bridge management system using sensor networks to solve the problems above. As a result, it can contribute reliability, efficiency, stability and accuracy to the system.

This study consists of five chapters. In chapter 2, middleware and sensor networks are discussed. In chapter 3, middleware design and implementation of the proposed system are covered and chapter 4 deals with middleware implementation on sensor networks. Chapter 5, it ends with conclusion and future work.

2. Previous Works

2.1. Sensor Networks

Sensor networks is a networks which is comprised of small nodes with a processor, sensors and a pair of wireless transmitter and receiver. A node can sense a variety of data, manipulate them and transmit them to the base system. Figure 1, Figure 2 shows sensor nodes. Sensor networks features a bigger scale of densely distributed sensors than Ad-hoc networks. Sensor nodes often fail to operate because they work in extremes and topology of the networks can easily changed. Peer to Peer communication is main concerns in common networks but nodes on sensor networks

** Corresponding Author : Byung-Wook Kang

prefer broadcasting mechanism. They have very limited hardware resources like power, memory and calculation ability. They are compactly deployed in a small area. They cannot use international identifiers such as IP address [5].

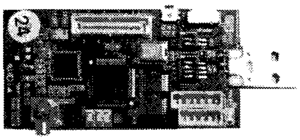

Figure1.MSP430MoteType2

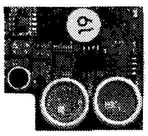

Figure2.US/MicSensors

2.2. Middleware

A middleware can be define as software which facilitates communication between computers and perform various functions needed in a system. In other word, it means software which supports communication among different protocols, hetero operating systems or database systems and applications. It helps applications run under any information system. Therefore, it can be describe as all kinds of products which can promote communication functions between client and server or between computers. Software which equipped with the above tasks can be categorized as middleware even if they are simple.

The middleware of this study collects and filters a big amount of data under heterogeneous sensor networks and it summarizes them as meaningful information and conveys it to the base system. In the early days of technology and market formation for sensor networks, wire networks where data can be transferred automatically was the main stream. But Sensor networks middleware in charge of collecting, manipulating and controlling data for integration of heterogeneous platforms is drawing attention.

3. Middleware Design

3.1. System Outline

The whole system of middleware is composed of sensor part, middleware part and application part as Figure 3. In the sensor part, data gathered from environment change, deformation, droop, slop, vibration sensor etc are forwarded to sink nodes. Data arriving at sink nodes move to Gateway, XPE (Windows XP: Embedded) through the serial channel. XPE forwards data in the form of TCP/IP packet to middleware on HSDPA (high speed downlink packet access) channel through cellular phones base station networks. Raw data conveyed to middleware are transformed by IMS Controller. The transformed data are stored in DB and some of them are sent to alarm manager for real-time monitoring of some critical conditions. IMS (Infra-structure monitoring system) Controller forwards real-time to exclusive applications or web servers and past data accumulated in DB are forwarded to them by DB server. Commands issued in applications are delivered to sensor nodes throughout IMS Controller.

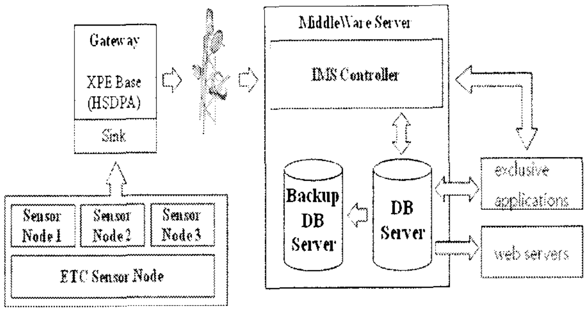

Figure3. Flowchart of the whole system

3.2. Design Middleware Architecture

The middleware architecture transforms raw data from as real unit and relay the transformed data to applications at the same time stores them in DB. And middleware is responsible for transmitting commands from exclusive applications to sensor nodes.

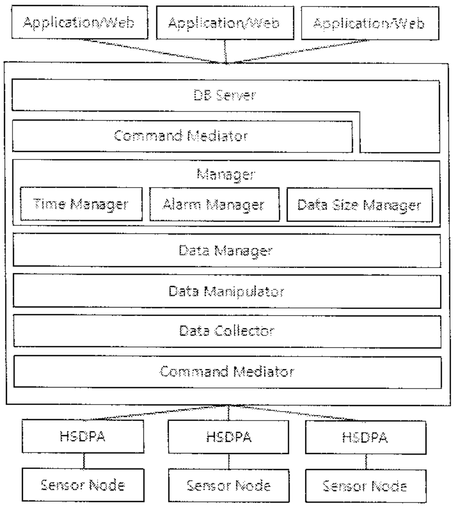

Figure4. Design Middleware architecture

The middleware takes care of data through many phases as Figure 4. Each phase is as the following.

- Data Collector - collects raw data from each node.
- Data Manipulator - Transforms raw data to engineering unit.
- Data Manager - Stores data transformed from data manipulator in DB and send data to applications and alarm manager for real-time monitoring. It watches and copes with DB Server failure. If DB server stops working, it bypasses data of high importance to file system on disk. While writing data on disk, it keeps checking whether DB server recovers by periods. When DB server recovers, it slowly moves data from files on hard disk to DB server considering load of DB server.

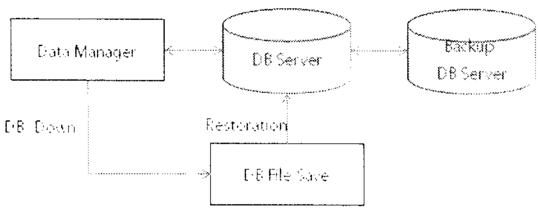

Figure5. Process of DB restoration

- Alarm Manager - located between IMS controller and DB server, monitors important data which operators requires to watch and create alarm messages when abnormal data occur. Figure 6 shows how alarm manager deal with alarm messages. Alarm manager receive data which operators previously register for monitoring from data manager. Alarm messages have 4 primary categories such as LL(Low-Low), L(Low), H(High), HH(High-High). Operators set normal ranges on data of interest. If the value of one item of data approaches to a lower bound of the normal range, a L alarm message is created. If it goes out of a lower bound, a LL alarm message is created. Messages for H and HH are created for a upper bound as above. Sounds and visual warning signs are accompanied with alarm messages to catch operators' attention. When operators recognize them, they take appropriate actions and acknowledge them. Command "CLR" can be used to get rid of all the acknowledged messages.

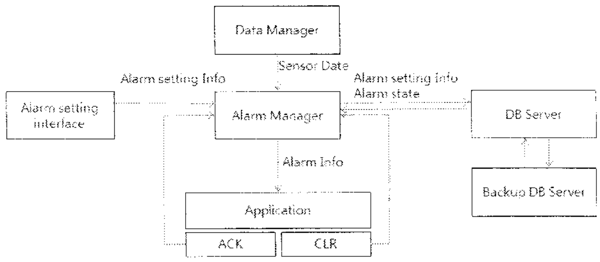

Figure 6. Process of Handling Aram Messages

- Command Mediator - relays commands from exclusive applications to sensor nodes. Issued commands are accumulated in DB for later search of history.
- DB server - stores data from data manager, alarm manager, and command mediator. It also sends data to exclusive applications or web servers without intervention of middleware. The reason why DB server sends data directly to them is that they ask data very frequently from DB server and load of middleware can be reduced in this way.

4. Implementation of the Middleware

Figure 7 shows the snap-shot of middleware presented in this study. Middleware has functions such as general monitoring, data transformation, data control, DB management, alarm control, SMS management and environment setting.

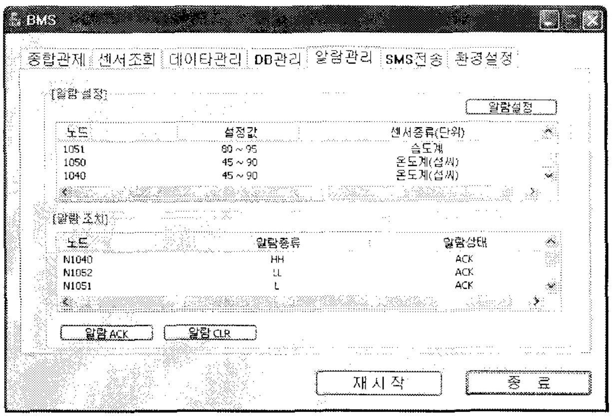

Figure 7. Implementation of Middleware

- General Monitoring - shows essential and important items in data control, DB management, alarm control, SMS management and environment setting.
- Data transformation - configures how to transform raw data according to sorts of data.
- Data control - provides a user interface for data collector and observes data from sensor nodes.
- DB management - confirms the state of DB server connection to middleware and provides configuration interface which direct what to do in case of DB server breakdown. In this section, operators can change configuration so as to store data of high priority in files.
- Alarm control - consists of a user interface where data of importance are registered to be monitored and their ranges are set and an alarm interface where operators take care of alarm messages.
- SMS management - has a user interface where operators' phone numbers are registered to send text messages to them in an emergency.
- Environment setting - is a user interface where basic options of middleware are set.

5. Conclusion and Future-work

In this study, essential functions and an efficient architecture of middleware for bridge monitoring system were considered and the basic standard of middleware architecture was presented. Basic functions performed between sensor nodes and applications were defined hierarchically and flow of data among all stages was proposed. As a result, this study gained a foothold in reliability, efficiency, stability and accuracy for bridge monitoring system based on sensor networks. Future work is that we must apply the architecture presented in this study to real bridges and verify the performance of middleware. Problems and shortcomings in the stage of application as above need to be solved.

6. Acknowledgements

This research was financially supported by the Ministry of Commerce, Industry and Energy (MOCIE) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

7. References

[1] U. Kaiser and W. Steinhagen " A low power transponder IC for high performance identification system", CICC, P.335, 1994.

[2] Kaiser , U. and Steinhagan W., "A low power transponder IC for high performance identification system", IEEE JSCC, Vol. 30, No. 3, p. 306, 1995

[3] R. M. Homby, "RFID solutions for the express parcel and airline baaggage industry", in Proc. IEEE Colloq. RFID Technology, London, U.K., P. 2/1 - 2/5. 1999.

[4] D.Culler, D. Estrin, M. Srivastava "Overview of Sensor Networkss" IEEE Commputer Society pp.41-49 August 2004.

[5] Akyildiz, I.F., W. Su, Y.Shankarasubramaniam, E. Cayirci" A Survey On Sensor Networks" IEEE Communications Magazine pp.102~114 August 2002.

[6] Junghwan Jang, Wanjong Kim, Hohyun Ahan, Seho Lee, Teayoung Jung, "Sensor based brigdge management system" Korean noise vibration society pp.602~607 2003.

[7] Jongwon Lee, Jeadong Kim " Bridge monitoring method for maesuring action stress" Korean structure diagnosis society pp31~36, 1999.

[8] C. Shen, C. Shisathapornphat, and C. JaiKeeo, "Sensor Information Networksing Architecture and Application." IEEE Personal Communications." IEEEPersonal Communications, Vol. 8, No. 4, pp.52~59, Aug. 2003.

[9] T. Nieberg, S. Dulman, P.Havinga, L.v. Hoesel, and j.Wu. Collaborative algorithms for communication in wireless sensor networks. In T. Basten, M. Geilen, and H. de Groot, editors, Ambient Intelligence:Impact on Embedded Systems Design, pp271~294, 2003.