



Implementation of Network-centered Software Platform for Wireless Sensor Networks

Lujuan Ma* Deshi Li Yanyan Han Xueqin He Tao Sun Yuxi Hu

*School of Electronic Information
Wuhan University
Wuhan, Hubei 430079, China*

*Corresponding author's Email: malujuan1985@126.com

Abstract: Wireless sensor networks are data-centric, in which data management and representation is a crucial issue. In this paper, sensor networks deployed to monitor wetland and underwater environment by multi-hop networks are equipped with the Network-centered Software Platform to visualize the data and topology. Our visualization software platform adopts three-layer model as architecture. The three-layer model equips our visualization software platform with favorable flexibility, scalability, better security, higher reusability and good fault tolerance. The technologies involved in this paper include serial communication, database management and data visualization which is to visualize the data gathered by wireless sensor networks. The database can be accessed by both local users and remote users through Internet. Of course the users must have an authority to know the conditions of wireless sensor networks. Practical experiments have illustrated that our visualization software platform can get data from the wireless sensor networks and provide real time and historical information for users. It is convenient and efficient.

Keywords: WSNs; Visualization platform; Serial communication; Database; Data and topology visualization

1. Introduction

Nowadays, environmental protection is drawing more and more attention from scholars, governments and organizations all over the world. As a new monitoring method, wireless sensor networks are becoming more and more important in environmental monitoring because of its convenience and efficiency. After wireless sensor networks having been deployed in the preferred area, it is important for us to send orders to the networks to guarantee their normal execution and acquire real-time data about the environment conditions from them. Therefore, implementing a wireless sensor networks visualization software platform plays a significant role in the accuracy of environmental monitoring.

A common wireless sensor networks architecture is shown in Figure 1. Every node in this system can transmit data of its own and forward data from other nodes so as to monitor the environment. The environmental data are gathered by sensors embedded in the communication units. Each unit contains a micro-computer which is used to collect and transfer the environmental data, micro sensors and other accessories. Currently the unit is designed to measure temperature and humidity, both in atmosphere and soil. The units are deployed every few hundred feet to form a wireless ad-hoc sensor networks.

At the data center, data are stored for further manipulation and it can be accessed via the visualization platform. Data gathered from the wireless sensor networks are stored in a database. Because the

environmental data are recorded every few minutes, it is crucial for end-users to manipulate and translate the vast amount of data.

The main objective of this visualization software platform is to collect and convert raw environmental data into visual formats so that they can be easily understood and managed by users. Through the platform, users can get real time or historical views of environmental data in the format of data list and data curve. Users on the website client or other terminals can also get topology view of networks including cluster information based on our routing protocol.

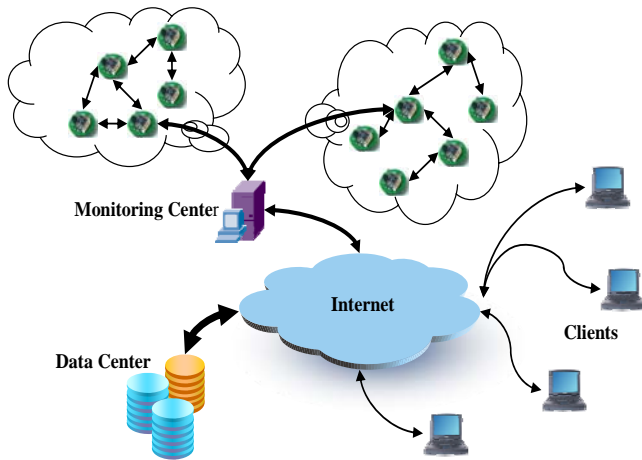


Figure 1 Wireless Sensor Networks Architecture

In the development of this visualization platform, we adopt the popular three-layer model as our software platform architecture. The three-layer model includes presentation layer, transaction layer and data layer, as is shown in Figure 2. Furthermore, on the basis of three-layer model, our visualization platform combines several technologies: serial communication, database management, data visualization and computer interactive design. This paper discusses the above technologies in more details as follows.

The following parts of this paper are organized as follows: First, we introduce the related works about wireless sensor networks visualization platform. Then we discuss the system architecture design of our platform. After it, the technologies used to implement the platform will be discussed in details. Besides, the website client based on our visualization platform will be shown to you. Then we discuss how we design the database. In the end, we mainly discuss how to visualize the network topology information.

2. Related works

At present, routing protocol, node localization and target tracking are the major research topics in wireless sensor networks. Among these topics, wireless sensor network visualization platform is relatively a minor issue. We take some typical platforms for analysis.

Spyglass is a modular and extensible visualization framework for wireless sensor networks. In Spyglass, data transmitted by individual sensor nodes is collected by one gateway and then passed on via TCP/IP to the visualization software on a potentially remote machine. Spyglass uses a multi-layer mechanism to visualize the wireless sensor networks information, to which the new visualization plug-ins for specific needs and applications can be added.

The Surge Network Viewer and the Mote-VIEW are Crossbow's products to visualize wireless sensor networks. The former features network topology visualization with logging and viewing of network statistics including yield, predicted yield, and link quality for every node, but is limited to these features. The latter covers essentially the same topics but presents a much cleaner user interface and more features. It is also capable to log sensing data to a database, analyze and plot sensor readings. Like Spyglass, both of them use a gateway to collect data from wireless sensor networks.

SNAMP is a multi-sniffer and multi-viewer visualization platform for wireless sensor networks developed by Chinese Academy of Sciences. SNAMP indicates network topology, sensing data, network performance, hardware resource depletion and other attributes in wireless sensor networks. It also allows developers to add application-specific visualization functions, which will facilitate the research and development of various sensor networks and shorten the time from laboratory to applications.

In most of the existing platforms, various data in wireless sensor networks are visualized to some degree. However, these platforms are lack of commonality due to their limitation to the hardware that they are based on. Therefore, such disadvantages decide that almost none of the existing platforms can be directly used for our own research and development of various wireless sensor network applications, so we exploit this multi-view visualization platform to implement data and topology visualization in wireless sensor networks.

3. System architecture design

Regarding to the software system architecture,

hierarchical model is the most common form. It can be described as follows. It starts from the lowest abstraction level, called the first layer, which is the base of the system. The key point of hierarchical model is to identify dependence relationship. Through layering, the dependence between subsystems can be restricted, which makes the system more loosely coupled and easy to maintain. Common hierarchical models are classified as Client / server model, Three-layer model (user presentation layer, logic layer and data layer), Multi-layer structure model (presentation layer, middle layer, and the data layer), Network system structure model (core layer, convergence layer and access layer) and Java-based browser / server model.

In this paper, the popular three-layer model is adopted as the software platform architecture, as is shown in Figure 2. It is divided into three parts: presentation layer, transaction layer and data layer. These three layers have a clear division of labor, and relatively independent in terms of logic.

The three-layer model has many advantages. One of the advantages is the convenience to change the system. It is because the transaction layer and the data layer are separate. For example, if we want to use Oracle to store our data instead of SQL Server, we do not have to modify the transaction layer. Similarly, if we want some change on the transaction layer, we do not need to reload the whole system but just update the components of transaction layer. Another advantage of the three-layer model is that developers can develop every layer independently and in parallel.

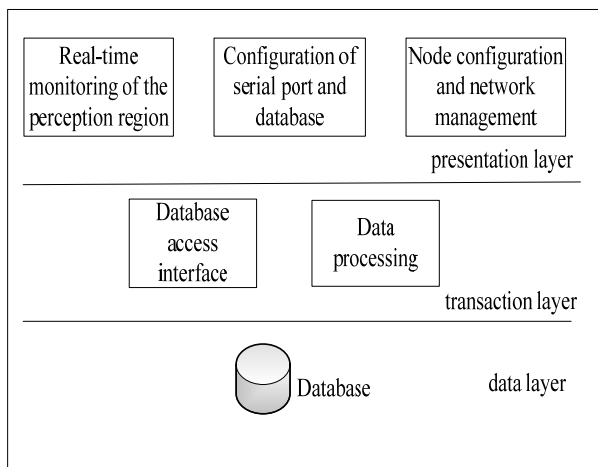


Figure 2 System Architecture

3.1 Presentation layer

Presentation layer, also named user presentation layer, provides interface of applications to the platform so the platform can interact with users. Users input information through this layer and get some relevant results. This layer can check out errors of input data without any transaction data processing and ensure inner confidential information is not transparent to users. In our platform, sensor data acquired from the monitoring area can be shown to users through data curve, data list, and network topology. Besides, users can configure serial communication and database connection to manage the whole wireless sensor network.

3.2 Transaction layer

Transaction layer, also named transaction logic layer, handles the transaction entity and the workflow. It is the bridge of the presentation layer and the data layer and provides transaction processing for presentation layer. It responds to users' request from presentation layer and completes the whole data processing through the interaction with data layer. We can say it is the main body of the architecture. In our platform, it satisfies users' request through interaction with the database using SQL. It enables users to monitor the sensing area in real time by transmitting real-time data collected by sensors according to the information users input into.

3.3 Data layer

Data layer provides access to external system (i.e. database or service). It maintains transaction data for a long time and makes data open to the transaction layer. Data layer consists of database server, data lists, the restriction of data list and some storage procedures. It defines the information of the platform and responds to the request of the transaction layer to access the data. In the platform, we use SQL Server 2000 to realize this function.

The three-layer model characterizes our visualization platform as follows: (1) It has favorable flexibility and scalability. Under the situation that the environment and application are changing frequently, we can achieve our goals by changing the application layer accordingly. (2) Different platforms share the same application server, which can provide services for client application program. This can reduce the development time and capital investment considerably.

(3) It has higher security. In this structure, the client application can not directly access the data. Application Server can control not only what the data is changed and accessed but also how the data is changed and accessed. (4) It strengthens the reusability. With the development of component technology, this model with reusable components is increasingly accepted by varieties of software developments. (5) Three-layer model becomes the true "thin client". Thus there is a higher stability, scalability, and implementation rate.

4. Serial communication

In order to communicate with the base station, we use serial communication between the base station and the platform. Thus, the visualization platform can get real time data from base station and send orders to networks via the base station.

First, a communication protocol regulating the two sides in communication must be observed by both sides. We design a uniform regulation including data format, data rate and so on as follows:

- Serial interface: RS-232
- Frame format: 8 data bits, 2 stop bits and 1 odd check bit
- Baud rate: 9600b/s
- Data packet: as shown in Figure 3

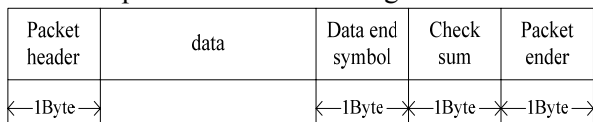


Figure 3 Data packet

As shown in Fig.3, each packet consists of five segments: packet header, data, data end symbol, check sum and packet ender. Among these segments, data is the main content transmitted in hex. Because of the possible effect of the distance and the spot condition, unexpected data errors may occur when transmitted between computer and the base station. So we use check sum to check errors in this paper. The check sum is the low byte of the sum of all bytes in data, which is used to make sure that the data received is correct in packets.

Before communication, the sender must add some bytes to the data to be transmitted to calculate check sum and then send them out in the form of data packet. After receiving this packet, receiver unpacks it and checks the data. If they are correct, receiver will

execute relevant functions according to the received data.

After declaring the communication protocol, we design a serial port class named CSerialport based on operation functions of serial port in Windows API to implement serial communication. This class is in charge of setting parameters of the serial port, deciding when to open or close the serial port and when to read from or write to the serial port. Once opening a serial port, the main thread creates a monitoring thread to deal with the information in the serial port. When some data arrive, a responding of serial port information function in the monitoring thread will take data packets out, unpack them and then convert raw data into regular physical data and store them in their own data tables in database. Figure 4 shows the flow chart of serial communication.

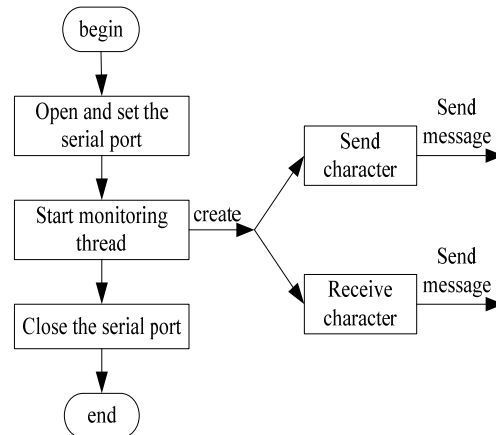


Figure 4 The flow chart of serial communication

5. Database design

In most situations, there is large amount of data collected by wireless sensor networks, so the traditional way of data storage using files will be inefficient when dealing with such kind of huge volume of data. In order to manage and access both the real-time and historical statistics more easily and accurately, database is the first choice to implement the data management. There are several popular databases, such as Microsoft Access, SQL Sever, Oracle, Dbase, Paradox, FoxPro, Excel and so on.

In our visualization platform, we use the database SQL Server 2000 to process data. Compared to other databases, the most obvious advantage of SQL Server 2000 is that it supports multi-users to access the database concurrently. The procedure to access SQL Server 2000 is shown in Figure 5. The database connection should be established firstly. After a

session is built, users can implement concrete operations. Then the session and connection will be closed until the access is completed.

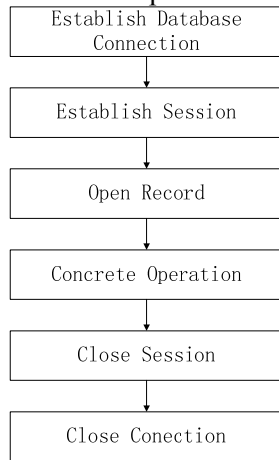


Figure 5 Procedure of database access

And we need design different data tables to store kinds of data and provide an access to the database and tables for data management. We mainly design three data tables as follows:

Node information table, which contains some basic information of nodes, the structure is:

Nodes=(A1,A2,A3,A4,A5,A6)

A1: Node ID

A2: Coordinate X

A3: Coordinate Y

A4: Coordinate Z

A5: Sampling rate of node

A6: Status of node

Sensor information table, which contains some monitoring data collected by sensors in the network, the structure is:

Sensors=(B1,B2,B3,B4,B5,B6,B7)

B1: Sampling time

B2: Node ID

B3: Temperature value

B4: Humidity value

B5: Sunlight value

B6: Temperature value of soil

B7: Humidity value of soil

Routing information table, which contains network topology information, the structure is:

RoutInfo=(C1,C2,C3,C4,C5,C6,C7,C8)

C1: Time

C2: Node ID

C3: Head Node ID

C4: Last hop Node ID

C5: The number of neighbor nodes

C6: Neighbor Nodes ID

C7: The number of nodes in a cluster

C8: Nodes ID in a cluster

All the data are stored in these tables. In order to get and use these data for visualization, database and records are needed to be connected when the visualization platform is running. It means that the database should keep open for other modules in the platform. So we design a class named CConnectionProps based on ATL/OLE DB library in Visual C++ to create a dynamic access to the database background.

6. Data visualization

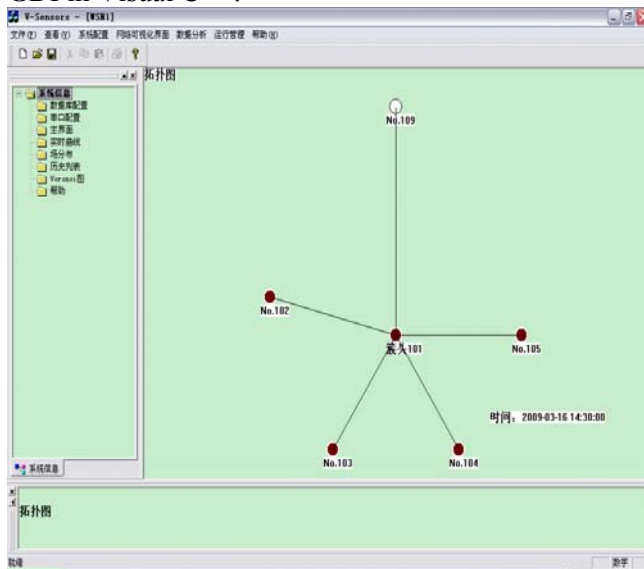
It has been discussed above that serial communication module enables the real-time communication between wireless sensor networks and visualization platform and provides data source to the visualization platform. And database management module implements the database connection and access function, providing an interface for the visualization platform to access and manage data sources. In our paper, three forms are applied to implement the data and topology visualization functions including data list, data real-time curve and network topology.

6.1 Network topology

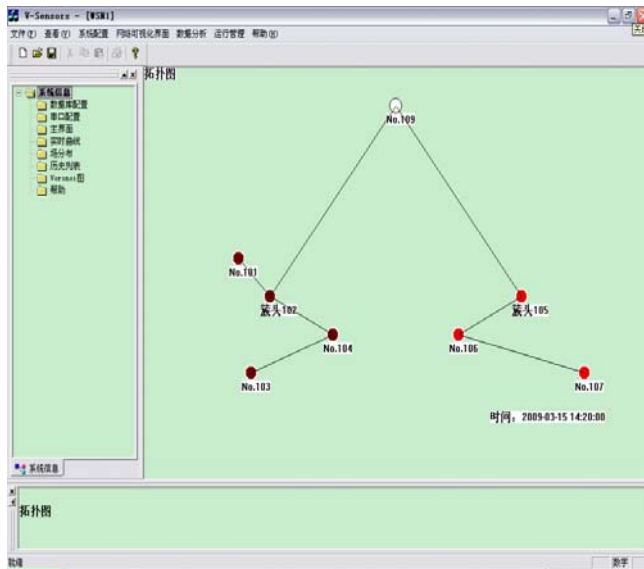
Routing data is the information of wireless sensor network topology. Topology visualization is an important way to analyze routing data, and its main objective is to illustrate the information of nodes and communication links between them. Thus it is easy for people to analyze the network status and then find out potential disciplinarians in the network. Network topology is the position distribution of nodes and their connection relationship. So topology visualization is required to display both of the two characters.

If a network contains N nodes, the network's connectivity information can be represented as an $N \times N$ adjacency matrix, in which entry a_{ij} is the number of links between nodes with the sorted indexes of i and j , where $i, j = 1, 2, 3 \dots N$. For a simple undirected graph (i.e. no self-loop and no multiple links between a pair of nodes), the adjacency matrix is a symmetric (0,1) matrix with zeros on its diagonal. Then the network's topology information can be stored in the sorted adjacency matrix and can be visualized as a planar graph in which nodes are represented as different shapes and connectivity between nodes is represented as a line.

In the implementation on computers, we take three steps to visualize network topology: topology discovery, topology storage and topology graph drawing. First, we get access to database to extract nodes and connectivity data from table RoutInfo. Then we use data structure to store this information. For example, nodes information is stored in arrays and the network's connectivity information is stored in an adjacency matrix. Finally we assign a logical coordinate for each node according to the physical deployments and draw shapes as well as lines to illustrate each node and their connectivity based on GDI in Visual C++.



(a) One-cluster network



(b) Two-clusters network

Figure 6 Examples of topology graph

We use different colors to denote information of different clusters. That is, all nodes including the head node in a cluster are assigned the same color automatically and different colors in another cluster. Thus, users can get the information of clusters clearly and quickly. Figure 6 shows two kinds of topology graphs: (a) one cluster network; (b) two clusters network and nodes join in a cluster in one-hop or two-hop.

6.2 Data list

In data list, the real-time data, which interest users, are extracted from the database and then displayed visually by means of list, as is shown in Figure 7. We also add auto-refresh function to the data list. Through the interface, users can view relevant information about varieties of historical and real-time data conveniently.

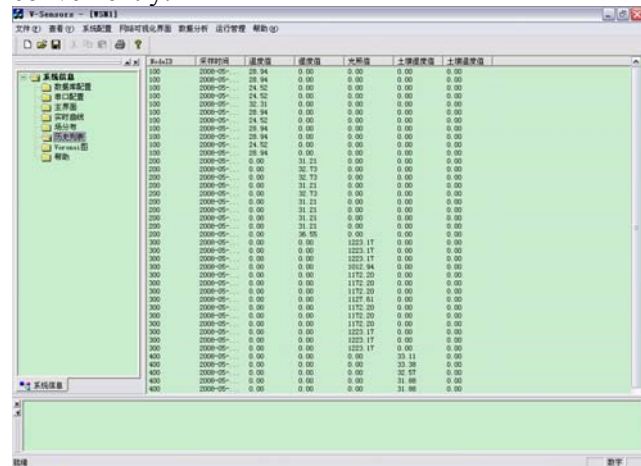


Figure 7 Examples of data list

6.3 Data real-time curve

Environmental data and routing data are collected by sensors, transferred by serial communication and then stored in the database. Environmental data is the information about the temperature, humidity in atmosphere and soil, illumination intensity and PH around the wetland monitored by the wireless sensor networks. In order to show the changing tendency of various data in the wetland, data curve is adopted to implement the data visualization.

As shown in Figure 8, the data curves illustrate the air humidity variation in the wetland in real time. In the data curve graphs, the X axis represents time, and the Y axis represents air humidity. The unit of the X axis is point and one point represents one minute. The unit of the Y axis is (g/m^3).

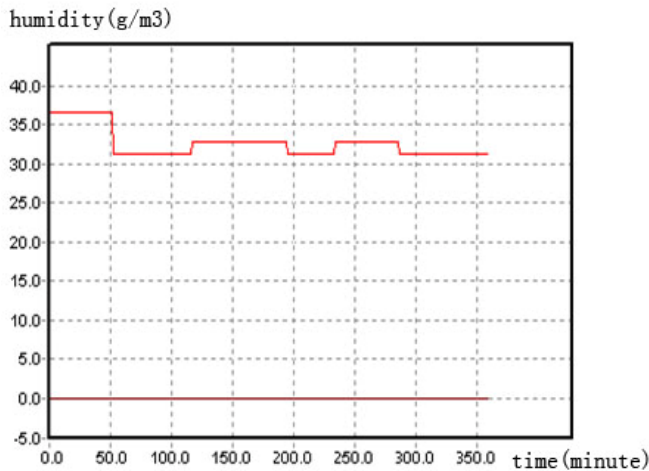


Figure 8 Examples of data real-time curve

7. Website client

Users online can interact with wireless sensor networks through website browser based on ASP dynamic website which is developed using DREAMWEAVER8.

Wireless sensor networks implement data collection from the monitoring area. In order to enable remote users to get access to the data which has been stored in the database, Internet will be used to transmit data. Figure 9 shows the FrontPage of our website.

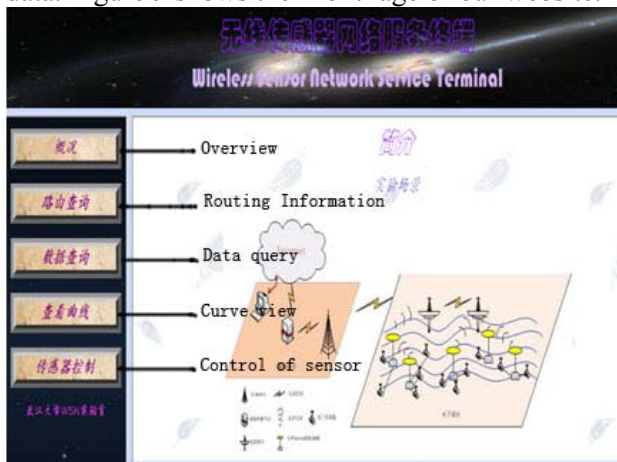


Figure 9 Frontpage of website

There are some types of accessing page used to present the real-time data of the wireless sensor networks. In the route inquiry page, after users have selected the interested node ID, the route information of this node will be derived. In the sensor data inquiry page, after users have selected the interested node ID, the sensor data about this node will be derived in the

form of data list. In the curve page, as is shown in Figure 10, after users have selected the interested sensor type, the changing curve of sensor data of this type will be created. In the sensor control page, if users have correctly logged in with authorized user name and password, they can enter the sensor control page and send command.

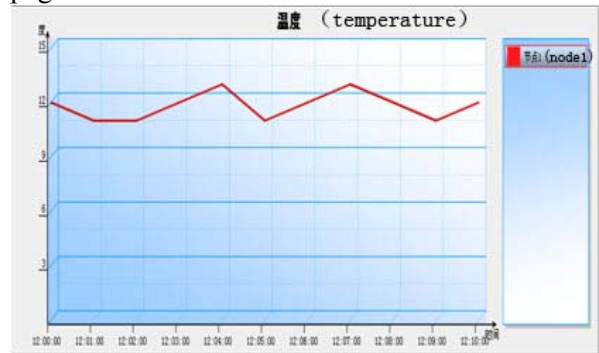


Figure 10 The curve page

8. Conclusion

In this paper, a visualization platform for wireless sensor networks is developed. Three-layer model are used to design our software platform, which equips our visualization platform with favorable flexibility and scalability, better security, higher reusability and good fault tolerance. Via serial communication technology, it successfully transmits the information gathered by the wireless sensor networks to data center. Both local users and users online can easily acquire both the real-time and historical data from database in the data center. And the most remarkable contribution of this platform is the data and topology visualization, which can provide users with more direct view of the environmental and routing data around the monitoring environment.

It is obvious that our visualization platform provides the chance for users to better control the execution of wireless sensor networks and get various data about the environment. Our platform can be used in both the onshore application such as wetland monitoring and underwater environmental exploration such as the underwater sensor networks. And the networks we design take remote access in thought, which has wide application and favorable scalability. In the future, the expansibility and compatibility for hardware are the prior research topics in this wireless sensor networks visualization platform.

Acknowledgements

The authors would like to thank the reviewers for

their detailed comments that have helped improve the quality of the paper. This research was funded in parts by China National High Technology Plan Grant 2007AA01Z225 and NSF China 60972044.

References

- [1] Fengxian Fan and Edoardo S.Biagioni, "An Approach to Data Visualization and Interpretation for Sensor Networks," *Proceedings of the 37th Hawaii International Conference on System Science*, 2004.
- [2] C. Buschmann, D. Pfisterer, and S. Fischer, "Spyglass: A Wireless Sensor Network Visualizer," *ACM SIGBED Review*, vol. 2, no. 1, 2005.
- [3] TUTON M. "MOTVIEW: A sensor network monitoring and management tool," *Second IEEE Workshop on Embedded Networked Sensors, EmNetS-II 2005*. 2005: 11 - 18.
- [4] Yu Yang, Peng Xia, Liang Huang, Quan Zhou, Yongjun Xu, Xiaowei Li, "SNAMP: A Multi-sniffer and Multi-view Visualization Platform for Wireless Sensor Networks," *IPN Progress Report*, May 15, 2005.
- [5] Y. Guo, C. Chen and S. Zhou, "Topology visualization tool for large-scale communications networks," *ELECTRONIC LETTERS*, Vol. 43, No. 10, 2007.
- [6] Layered Architecture Model: <http://book.csdn.net/bookfiles/653/10065320717.shtml>, 2009.
- [7] Yu Yang, Peng Xia, Liang Huang, et al. "SNAMP: A Multi-sniffer and Multi-view Visualization Platform for Wireless Sensor Networks". In: *Industrial Electronics and Applications. 2006 1ST IEEE Conference on*, 2006:1-4.
- [8] Rafael De T.Valle, Diego Passos, Celio Albuquerque. "Mesh Topology Viewer (MTV): an SVG-Based Interactive Mesh Network Topology Visualization Tool". In *ISCC 2008. IEEE Symposium on Computers and Communications.*, 2008: 292-297.
- [9] Topology visualization tool for large-scale communications networks: The Institution of Engineering and Technology, 2007.