Plug-configure-Play Service-oriented Gateway
For Fast and Easy Sensor Network Application Development

Lin Wu[1], YongJun Xu[1], ChaoNong Xu[2] and Fei Wang[1]
[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *Department of Computer Science and Technology, China University of Petroleum-Beijing, Beijing 102249, China*

Keywords: Sensor Networks, Plug-configure-Play, Resource-oriented, Gateway.

Abstract: Sensor Networks are typically designed for specific applications. We propose to transfer application-specific tasks from internal sensor networks to external applications, and making sensor networks universal infrastructure connected with gateways. This paper illustrates how we make the development of external sensor network applications fast and easy through our elaborately-designed gateway. Our “smart” gateway is resource-oriented other than just interconnects different networks transparently. It encapsulates the data, capabilities and information of heterogeneous sensor networks to homogenous resources. Besides, the gateway is plug-configure-play in the sense that it’s a common gateway for any type of sensor networks. A universal program and information related with specific sensor networks are separated in the gateway, and you can update the information (i.e., configuration) easily. In this paper, we realize and verify our ideas in practice.

1 INTRODUCTION

Sensor network applications are typically tightly coupled with certain sensor networks, such as: Monitoring Heritage Buildings (Ceriotti et al., 2009), Smart Logistics (Bijwaard et al., 2011), Sensor Browser (Wan et al., 2012), etc.

Juntunen et al., (2006) divided sensor network applications into two groups: external and internal applications. We use these two terms in this paper to refer to the applications on remote devices (PCs, Mobile Phones, etc.) and sensor nodes respectively. This paper focuses on the development of external applications and we propose to transfer application specific tasks from internal applications to external applications. Internal applications are universal, only responsible for simple tasks like sensing and transmiting data to sink nodes.

As sensor networks are heterogeneous, it seems that application developers must be experts about them. To ease the workload of application developers, we designed and implemented a resource-oriented gateway. But another problem arises: service-oriented gateways of sensor networks are typically tightly coupled with certain types of sensor networks like those in the previous literatures (Mueller et al., 2007); (Bimschas et al., 2010). We solve this problem by the plug-configure-play gateway presented in this paper, which is a universal one.

This paper is organized as follows. Section 2 describes two main features of our gateway: service-oriented and plug-configure-play. Section 3 is about the arch練ure of our system. In section 4, we’ll introduce the services our gateway provides. Section 5 describes the configuration of it to show that it’s universal other than dedicated for limited types of sensor networks. Section 6 describes the development of an application with it. After that, we compare our work with previous in section 7. At last, we conclude our work in section 8.

2 MAIN FEATURES

2.1 Service-oriented

Our gateway is service-oriented other than just transparently interconnects networks, making the development of sensor network applications fast and easy.

It encapsulates the data, capabilities and information of heterogeneous sensor networks to homogenous resources with universal resource

Wu L., Xu Y., Xu C. and Wang F..
Plug-configure-Play Service-oriented Gateway - For Fast and Easy Sensor Network Application Development.
DOI: 10.5220/0004271700530058
In *Proceedings of the 2nd International Conference on Sensor Networks* (SENSORNETS-2013), pages 53-58
ISBN: 978-989-8565-45-7

identifiers (URIs). And access of those resources is service it provides to external applications through friendly and uniform interfaces. These interfaces include data pulling, subscription and control.

With data pulling interfaces, you can pull resources identified with URIs in a HTTP-like way. With subscription interfaces, you can subscribe events (alarms, important data, status changes, etc.) from specific nodes. When an event occurred at your interested nodes, a notification message will be sent to you indicating it. Compared with pulling data, subscription is timely and resource saving. With control interfaces, you can request actuators in sensor networks to control the physical world. This kind of interfaces is needed for a smarter environment but rarely seen in litereatures.

With the resources encapsulated by our gateway, external applicaion developers can spend most of their time on innovation other than details of sensor networks.

2.2 Plug-configure-Play

The plug-configure-play mechanism makes our gateway universal.

"Plug-Configure-Play" is a variation of the term "Plug and Play". Like a plug and play device, you can plug our gateway between sensor networks and external applications, configure it and then it can work for you. Our gateway consists of a program and a database. The program interpretes messages with the help of sensor network specific information stored in database. The word "configure" means to annotate sensor networks with such information, like nodes' names, locations, functions, sensor types, etc. We make our gateway universal by separating program and sensor network specific information.

It's possible to implement a Plug and Play gateway: sensor nodes can publish their information to the gateway using SensorML (Mike and Alexandre, 2007) or MoteML (Ali et al., 2011). But we think this has some significant disadvantages in terms of bandwidth consumption, flexibility, etc.

3 ARCHITECTURE

In this section, we'll first give the overall architecture of sensor network application development and then show how our gateway works.

3.1 Overall Architecture

The proposed overall architecture of sensor network application development is shown in Figure 2. The "adapters" in this figure include sink nodes as they also connect sensor networks to gateways. The size of each layer represents that layer's intelligence level.

Take our smart home project for example, the overall architure is like that in Figure 2. It is in fact an instantiation of Figure 1.

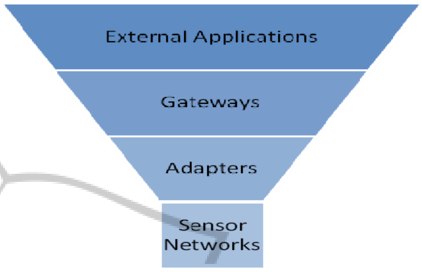

Figure 1: Proposed overall structure.

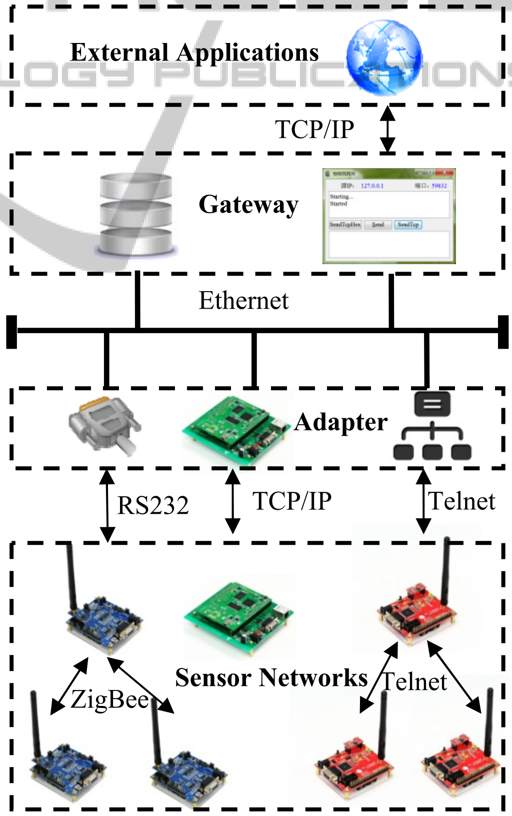

Figure 2: An implementation of overall architecture.

3.2 Gateway Architecture

The architecture of our gateway is illustrated in Figure 3. From this figure, you can see how our gateway works. Tasks of major components are explained below:

- **Database** – Information about sensor networks such as nodes IDs, sensors types and protocols are stored in database. Users can configure the gateway by updating related tables in the database.
- **Packets Analyzer** – Analyze the type of received packet (data, query or control), and deliver it to responsible components.
- **Protocol Translator** – Translate different *application layer* protocols.
- **Resource Manager** – Encapsulate data, capabilities and information of heterogeneous sensor networks to homogenous resources, and provide them to external applications.
- **Publish/Subscribe Manager** – Publish events to external applications who subscribed them.
- **External Applications Status Manager –** Manage status (active or not) of external applications.
- **Communication Scheduler –** Schedule the communication with sensor networks to smooth the traffic
- **Connectivity Manager –** Record IPs and UDP ports of sink nodes/adapters and IDs of active sensor nodes connected with each of these sink nodes/adapters.

4 SERVICES PROVIDED

In this section, we'll introduce the services our gateway provides to highlight its service-oriented feature.

4.1 Connectivity

Our gateway supposes that sink nodes connecting with it are all IP-enabled, and it can connect with sensor networks with any kind of network protocols

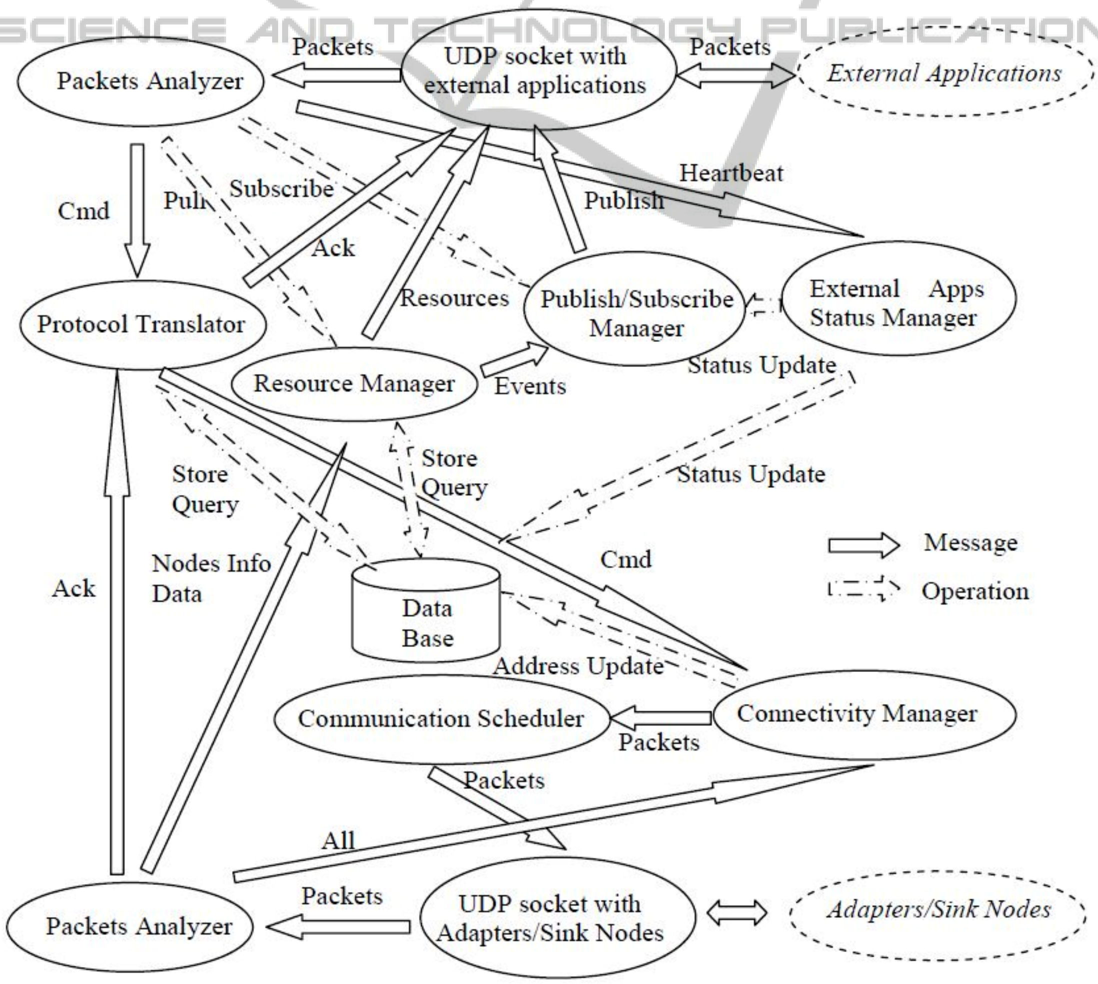

Figure 3: Gateway Architecture.

(Ethernet, ZigBee, GPRS, etc.) with the help of adapters. This greatly simplifies the design of its connectivity function and makes it universal.

For the gateway, adapter is just a sink node running the protocol shown in Table 1. For example, one of the sink nodes we use works as a TCP server and can't send messages to the gateway automatically. To connect it to our gateway, we write a program, connecting to the sink node to read data and send to the gateway. The gateway will then look on this program as a sink.

Most of the gateways which have the function of transport layer and application layer protocol conversion we surveyed (Bimschas et al., 2010); (Mueller et al., 2007) do the conversion themselves. As a result, they usually can only work for sensor networks running several specific kinds of protocols. In contrast, we achieve the goal of a universal gateway by transferring the task from gateway to adapters.

The application layer protocol between gateway and sink nodes/adapters is like that in Table 1. Table 2 is about the meaning of each item. Application layer protocols of sensor networks don't have to comply with this, you can use a sink node or adapter to translate messages from your sensor networks to those compatible with this.

Table 1: Protocol with sensor networks.

Header	Dest	Src	SN	Len	Type	Payload	Checksum

4.2 Data Pulling

The data and information of sensor networks are encapsulated and identified by Universal Resource Identifiers (URIs). Thus external applications can request the resource of sensor networks in a clear and easy way, independent of what the sensor network is like. The data pulling interface is like the request line of HTTP, as is shown below:

Get URI

This stytle of interface is easy to understand and can facilitate the development of Web of Things upon our gateway.

For example "Get node1/RealTime/query? SensorName=Temperature+Humidity" means to get the latest temperature and humidity value from node1.

4.3 Subscription

The service of subscription is necessary for two reasons: Firstly, external applications may expect to be notified immediately when some events occur (e.g., the window of a room is broken suddenly); Secondly, information should be shared when several external applications control the same sensor network. The subscription interface is as below:

Event Watch NodeID

This tells the gateway that an external application is interested in the events from the node with ID/physical address NodeID. If an external application wants to monitor the status of a window, then it can watch the node monitoring this window. When the window is broken, a notification will be sent to the external application immediately.

4.4 Controlling the Physical World

External applications should make our environment smarter by enabling it to interact with us. This brings the need to control the physical world. Our gateway provides such services:

Mac NodeID SN CmdType CmdName Params

For example:

Mac 5149012977665754 0 CmdNrm LED_SetLightIntensity 20

This command requests the node with ID/physical address 5149012977665754 to set the light intensity of LED to 20. The sequence number of this packet is 0.

5 CONFIGURATION

In our gateway, the program and sensor network specific information are separated. The former is universal, and the latter is stored in database. You can deploy the program in your computer or other smart devices, update the database (i.e. configuration) if needed, and then the gateway can work. We list some important kinds of configuration in this section to show how we achieve the goal of universal through our elaborate design.

5.1 Nodes Information Update

There are two tables in the database storing information related with sensor nodes: "NodeAddressMap" and "NodeSensorMap".

"NodeAddressMap" stores information about nodes in sensor networks such as names, physical addresses and IPs (of nodes themselves or adapters). "NodeSensorMap" stores the information about sensors on nodes. It tells the program on gateway how to interprete the ADC values in the messages from sensor nerworks.

5.2 Sensors Information Update

It's impossible to put all types of sensors into consideration when writing the program on gateway as the number is too huge. To make our gateway universal, users "tell" the gateway information about sensors through configuration. Tables "TypeMap" and "NodeSensorMap" store related information.

5.3 Control Types Update

You can "tell" the gateway what controls sensor networks support through configuration. There are three tables in the database storing information related with this: "AvailableCmdQuery", "ParamInfo" and "CmdEncode".

Table "AvailableCmdQuery" lists all the controls available. Table "ParamInfo" lists parameters of all coontrols one by one. Table "CmdEncode" lists the codes of packet types and data types.

Our gateway converts the commands from external applications to the messages understandable by adapters/sink nodes with the help of these tables. If your sensor network provides new types of controls and you want to provide them to external applications, then just insert values into these three tables.

6 CASE STUDY

We've developed various applications with our gateway. In this section, we take an Android application as an example, see Figure 4.

Figure 4: Curtain Opening/Closing.

This application is used for opening/closing curtains in the house. Commands are sent to our gateway when swiping a finger. The code of this application related with controlling curtains is listed below:

```
   udpSocket.connectToHost(GatewayIP,
GatewayPort);
   udpSocket.write("Get
Info/NodeAddressMap/NodeID/Query?NodeNa
me_like_'%curtain%'");
   udpSocket.waitForReadyRead();
   nodeID=getNodeID(udpSocket.readAll()
);
   udpSocket.write("Event Watch
nodeID");
   udpSocket.write("Mac nodeID SN
CmdNrm Switch_On");
```

First of all, the program queries the gateway about the ID of the node that controls the curtain. As nodes have been anotated through configuration (see section 5.1), database will tell the gateway which node is controlling the curtain. Then the program requests the gateway to make this node to turn on the switch which will control the curtain.

The gateway can translate messages like "**Mac nodeID SN CmdNrm Switch_On**" to those compatiable with the application layer protocol in Table 1. The table "CmdEncode" in database will tell the gateway how to encode items in the message like "CmdNrm" and "Switch_On", and the table "Protocols" in database will tell the gateway positions of each item.

It should be noticed that, there may be more than one person controlling the curtain at the same time. So the program subscribes the events about this node. When another one is controlling the curtain, the change of curtain's states (closed/opened) will be sent to the program. Then it can tell the user through GUI.

7 RELATED WORKS

In the literatures we surveyed, there are rarely gateways offering interfaces for controlling the physical world through sensor networks.

Some of the gateways are only for interconnection (Emara et al., 2009); (Qian et al., 2010). Those gateways connect networks with different network protocols such as ZigBee, GPRS and Ethernet transparently.

Some other gateways also have the service of protocols translation, but they are usually designed for specific kinds of sensor networks, such as the gateway proposed by Mueller et al. (2007) and Bimschas et al. (2010).

Juntunen et al. (2006) focused on the expandability of internal sensor network applications. Information about sensor nodes and services available are stored in nodes and transmitted to gateway when service discovery request is flooded to all nodes. The service discovery messages add extra transmission cost. Besides, it brings extra workload for sensor nodes to support the reply of service discovery messages.

The work of Jong-Wan et al., (2009) is the most similar one. Their system composes of a main server and several sensing servers connecting with different sensor networks, which resemble our gateway and adapters respectively. But in their system, most tasks are completed by network-dependent sensing servers other than main server. This means that most of the tasks are left to users.

8 CONCLUSIONS

This paper has presented a plug-configure-play service-oriented gateway, aiming to make it fast and easy to develop various external sensor network applications. The gateway encapsulates data, capabilities and information of heterogeneous sensor networks to homogenous resources. Access of those resources is service it provides to external applications through friendly and uniform interfaces independent of types of sensor networks. The gateway is not for specific kinds of sensor networks or applications. Its feature of plug-configure-play is achieved by separating the universal program and sensor network specific information. It's promising in stimulating the emergence and prosperity of external application markets.

ACKNOWLEDGEMENTS

This paper is supported in part by Important National Science & Technology Specific Projects under grant No.(2010ZX03006-002 , 2010ZX03006-007), the National Basic Research Program of China (973 Program) (No. 2011CB302803), and National Natural Science Foundation of China (NSFC) under grant No.(61173132,61003307). The authors alone are responsible for the content of the paper.

REFERENCES

Ali, F., Feaster, Y., Wahba, S. K., & Hallstrom, J. O. (2011). A metadata encoding for memory-constrained devices. *In Proceedings of the 49th Annual Southeast Regional Conference*, pages 191-196, Kennesaw, Georgia.

Bijwaard, D. J. A., Kleunen, W. A. P. v., Havinga, P. J. M., Kleiboer, L., & Bijl, M. J. J. (2011). Industry: using dynamic WSNs in smart logistics for fruits and pharmacy. *In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (SenSys'11), pages 218-231, Seattle, Washington, USA.

Bimschas, D., Hellbrück, H., Mietz, R., Pfisterer, D., Römer, K., Teubler, T. (2010). Middleware for smart gateways connecting sensornets to the internet. *In Proceedings of the 5th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks* (MidSens'10), pages 8-14, Bangalore, India.

Ceriotti, M., Mottola, L., Picco, G. P., Murphy, A. L., Guna, S., Corra, M., Zanon, P. (2009). Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. *In Proceedings of the 2009 International Conference on Information Processing in Sensor Networks* (IPSN'09), pages 277-288, San Francisco, California, USA.

Emara, K. A., Abdeen, M., & Hashem, M. (2009). A gateway-based framework for transparent interconnection between WSN and IP network. *In EUROCON '09*, pages 1775-1780, St.-Petersburg.

Jong-Wan, Y., Yong-ki, K., Choon-Sung, N., & Dong-Rye, S. (2009). Sensor Network Middleware for Distributed and Heterogeneous Environments. *In International Conference on New Trends in Information and Service Science* (NISS '09), pages 979-982, Beijing.

Juntunen, J. K., Kuorilehto, M., Kohvakka, M., Kaseva, V. A., Hannikainen, M., & Hamalainen, T. D. (2006). WSN API: Application Programming Interface for Wireless Sensor Networks. *In IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1-5, Helsinki.

Mike, B. and Alexandre R. (2007). *OpenGIS Sensor Model Language* (SensorML). Retrieved September 13, 2012, from http://www.opengeospatial.org/standards/sensorml

Mueller, R., Rellermeyer, J. S., Duller, M., & Alonso, G. (2007). Demo: A Generic Platform for Sensor Network Applications. *In IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems* (MASS'2007), pages 1-3, Pisa.

Qian, Z., Ruicong, W., Qi, C., Yan, L., & Weijun, Q. (2010). IOT Gateway: BridgingWireless Sensor Networks into Internet of Things. *In 2010 IEEE/IFIP 8th International Conference on the Embedded and Ubiquitous Computing* (EUC'2010), pages 347-352, Hong Kong.

Wan, J., O'Grady, M. J., O' Hare, G. M. P. and Colakov, T. (2012). Browsing the Sensor Web: Pervasive Access for Wide-area Wireless Sensor Networks. *In 1st International Conference on Sensor Networks* (Sensornets'2012), Rome, Italy.