EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems

Research Article **ICST.ORG**

Gatewaying the Wireless Sensor Networks

Ming Zhu [1], Zhaoshu Tang [1], Wenlong Yue [1],
Lei Wang [1,*], Zhenquan Qin [1], Lei Shu [2], Liang Sun [1]

[1] School of Software, Dalian University of Technology, China
[2] Guangdong Petrochemical Equipment Fault Diagnosis Key Laboratory, Guangdong University of Petrochemical Technology, China

Abstract

With the development of Internet of Things (IoT), bridging wireless sensor networks (WSNs) with other networks has become important. We divide bridging solutions into two categories: the hardware solutions and the middleware solutions. The hardware solutions have both low power short distance wireless interfaces and other types of transmission interfaces, e.g. GPRS, 3G/4G, via hardware implementations, which is more stable and applicable for the deployed sensor networks. In the middleware solutions, the whole system processes appropriate protocol conversion, and independence of hardware, making it easier to be reused in different applications and networks. This paper briefly presents the implementation details and key points of each solution. Particularly, we derive the most appropriate situation for each solution from our comparisons and discussions in terms of evaluation applied to different criteria.

Received on 22 March 2016 accepted on 23 May 2016; published on 01 February 2017

Keywords: Wireless Sensor Networks, Gateway, Middleware, Integration, Hijack, uSDCard, VIP Bridge, GSN, 6LoWPAN

doi:10.4108/eai.1-2-2017.152153

1. Introduction

In recent years, the IoT has attracted increasing attention from the academia as well as the industry and has been developing rapidly. It is regarded as the next generation information technology after the Internet and mobile communication networks revolution. The researchers believe that the IoT will change the way people live. When the IoT is integrated with the cloud computing or other information technologies, it can help the information system run efficiently and accurately.

The WSNs, as an important part in the IoT, is a popular research direction in this area [1]. A WSN consists of a large number of tiny nodes, which are embedded with a variety of sensors. The nodes have limited computational capabilities, communication capabilities and energy. However, when they are networked together, the aggregated computation and communication capacity are considerable. Determined by the characteristics of the nodes, the WSNs is mainly used in data-centric applications, as well as some application-centric networks. For example, the WSNs can detect real-time environmental data without manual operations, and transmit the collected data to people [2]. The data plays a vital role in environmental monitoring, scientifi research and disaster prevention.

As the data collected in WSNs is usually used in upper layer applications, the WSNs needs to be connected with other networks, instead of operating as a stand-alone network. Nervertheless, the WSNs uses the short distance wireless communication protocol and lacks uniform standardization in communication protocols. These characteristics make it difficult to transmit the data for long distance easily and conveniently without other devices. So how to transmit the data to remote servers of other networks, such as the Internet and the mobile communication networks (3G/4G, etc.), is an engineering challenge and one of the hot topics in this area [3], which can be solved to some extent by bridging solutions. Therefore, the bridging solutions play an important role in the integration of the WSN and other networks.

This paper presents bridging solutions and contributes as follows. First, this paper divides bridging

[*] Corresponding author. Email: lei.wang@dlut.edu.cn

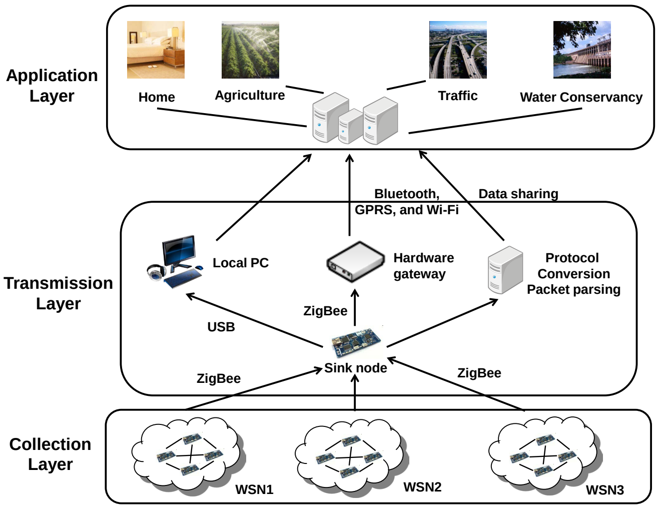

Figure 1. The different gateway architectures in WSN: the WSN1 is the normal WSN architecture without gateways; the WSN2 uses the gateway-based solutions; the WSN3 is the one using the middleware solutions.

solutions into two categories: the hardware solutions and the middleware solutions. And it presents the implementation details and key points of each solution of the two categories. For hardware solutions, we introduce the architecture design and the hardware configu ation of each solution; for middleware solutions, we introduce the models, components and software architecture. Second, this paper evaluates the advantages and disadvantages of each solution according to different criteria, metrics or dimension. Features used to evaluate hardware solutions are mainly mobility, ease of use, universty, bandwidth, and characteristics used to evaluate middleware solutions are mainly location, energy efficiency, consistency, transparency. Eventually, we derive the specific suitable scenario for each solution based on the evaluation and discussion. In addition, we summarize and make comparisons on two categories of these solutions. Generally, hardware solutions are more applicable for small-scale networks and the quick deployment in existing networks, while middleware solutions have better performance in large-scale networks and the centralized control.

The rest of this paper is organized as follows: Section 2 describes the architecture of the WSNs, as well as presents the need of the bridging solutions. Section 3 and Section 4 introduce the hardware solutions and the middleware solutionss respectively, and give a comprehensive comparison at the end of each section. Section 5 discusses the two kinds of bridging solutions mentioned in this paper and draws the conclusion, following with the future work.

2. The bridging solutions in WSN

According to the application system architecture, a typical WSN application includes the collection layer, transmission layer and application layer (Fig. 1) [4]. Each layer has different roles in a WSN application and the nodes of each layer have different characteristics. The collection layer is the basis of the entire WSN system, which consists of the most nodes of the system. It is responsible for the collection of environmental data. The critical research at this layer focuses on the signal detection, energy consumption and wireless communication technology. The transmission layer is mainly responsible for receiving the collected data and sending the data to the sink node. Then the sink node transfers the data to the upper application via the gateways using integration technologies. Therefore, the wireless sensor network routing protocols and integration technologies are very important in the transmission layer. The application layer contains the applications, which use the sensor data for further processing and analysis. At the application layer, the researchers concern about how to analyze the data better, and how to provide better user experience.

In the traditional WSNs, the sensor data from the collection layer will be sent to the sink nodes via the short distance transmission, normally the ZigBee. And the sink node sends data to upper layer devices through the wired transmission or USB interface. Without bridging solutions, the sink nodes cannot send the collected data to distant networks, and the valuable data can only be used in the local applications [5]. Therefore, the WSN applications need an application-specific bridging solution in the data transmission [6].

As shown in Fig. 1, the bridging solutions are mainly applied at the transmission layer [7]. Instead of transmitting data to the local base station via USB interface as traditional WSN applications, the WSN applications with bridging solutions send data to remote devices through the specific transmission interface after receiving the data collected from the sink node. These remote devices could be GPRS servers, smart phones or other intelligent terminals. With the use of bridging solutions, the real-time sensor data from the WSN can be shared with other networks, achieving the integration.

The main functions of the bridging solutions are data forwarding, protocol conversion. Data forwarding is the basic function, which concerns more about the efficient and accurate data transmission. Protocol conversion solves the engineering challenge of integrating WSNs with other types of networks. As the transmission protocol of the WSN is 802.15.4/ZigBee without fixed IP addresses, the gateway needs the ZigBee interface to receive the collected data and then processes the protocol conversion and the packet reassembling. Then it transfers the reassembled data to different types of networks through specific transmission interface, such as Wi-Fi, 3G/4G and GPRS. And the bridging solutions can be divided into two categories: the hardware solutions and the middleware solutions. The difference between the two categories is: the hardware solutions implement the specific hardware architecture designs physically to achieve these functions, while the middleware solutions achieve these functions by processing specific software procedures in normal WSN nodes.

3. Hardware solutions

The hardware solutions have both ZigBee interfaces and other types of transmission interfaces with hardware implementations (as the WSN2 in Fig. 1). After receiving the ZigBee packets from collection layer, the hardware solutions, acting as the gateway, process the protocol conversion, and then forward the reorganized data to a specific network through other types of transmission interfaces. This kind of solutions is stable and applicable for the networks that have been already deployed. These solutions require much hardware support , even specific hardware design, which makes them difficult to be reused in other networks. With the major burden of transmission on the hot pot hardware, this kind of solutions also has the problem of single point failure.

The hardware solutions we surveyed include: the MIB-510 and MIB-600, which are the products from Crossbow Company, and they can connect the node with the host via serial port or Ethernet port [8]; the GenOS, which is a WSN node with GPRS, and it can transfer data to the remote servers directly [9]; the Hijack, an interface for mobile phones to power and communicate with the peripheral [10]; the uSDCard, a peripheral for smart phones to access ZigBee and transfer the data via the standard SD slot [11].

3.1. MIB-510 and MIB-600

MIB-510 interface board is widely used in WSNs with MICAz, MICA2, MICA and MICA2DOT. It is one of the popular products of Crossbow Company [8], which usually serves as a sink node that not only aggregates data from nodes, but also transmits data to PC.

This board provides a serial interface, RS-232 (DB9 female), which is used to connect to PC or other standard platforms and provides the reprogramming function (Fig. 2). On the board, there are also a MICAz-series connector and a MICA2DOT connector that allow the MICA2 and MICA2DOT family motes to plug in, which constitute the base station together [12]. Moreover, MIB510 has an on-board in-system processer (ISP), which receives codes from the RS-232 serial interface and programs the mote plugged in. In addition, the incoming and the outgoing serial packets will be monitored by the ISP. The AC Wall-Power connector on the boards provides external power supply and the JTAG port allows the JTAG pod to connect for debugging. MIB-510 can transmit data, state information and any other types of message through bi-directional communication between clients and sensor nodes [13] .

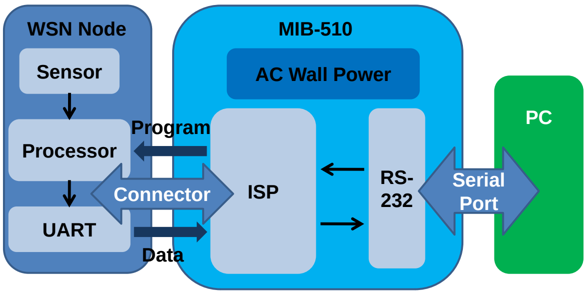

Figure 2: The architecture of MIB-510, and the MIB-600 is similar .

MIB-600 is another product made by Crossbow Company, which shares many common aspects with MIB-500 but has a greater popularity [14]. The compatibility makes MIB-600 widely applied in WSNs.

The most significant distinction between MIB-600 and MIB-510 is that MIB-600 connects with clients through Ethernet (10/100 Base-T) interface while MIB-500 through serial interface. With the Ethernet connection, TCP/IP protocol is used, so a unique IP address must be assigned to MIB-600, which allows it to connect with other network devices like hub, switch or PC, and we can access MIB-600 from remote servers to the sensor networks through TCP/IP protocol.

3.2. GenOS

As described above, the sensor data from WSN is collected to sink nodes. So it is the simplest way to integrate WSN with other networks to make sink nodes able to connect with other networks and deliver the data via other wireless communication technology [9].

The GenOS is a TelosB compatible wireless sensor node with GPRS. When it receives the sensor data from WSNs, it transfers the data to the buffer after encoding, which is for the GPRS module. Then it sends the data stream to the remote server through GPRS. Here the GenOS acts as the gateway, bridging the WSNs and other networks.

The GenOS is a kind of modified TelosB node [15]. It upgrades the processer by using MSP430F5438, instead of MSP430F1611 (Fig. 3). With this modification, the GenOS supports more UARTs, more GPIOs, and gets more storage at the same time. A standard GenOS has a variety of interfaces and can measure various parameters. Users can also integrate other sensors optionally.

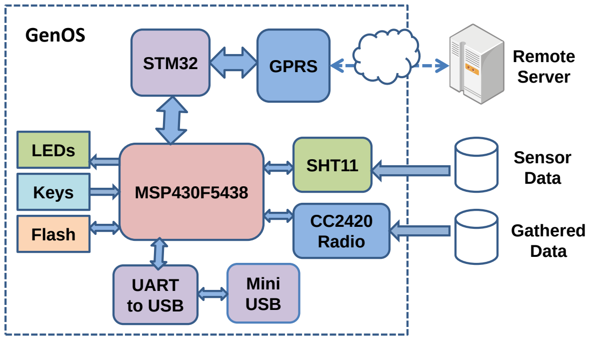

Figure 3: The structure of GenOS

As the gateway of WSN, the biggest innovation of GenOS is to attach the GPRS module to the standard TelosB node [16]. Since the GPRS module requires initialization, there is a proprietary module for the management of the GPRS module. In order to simplify the TinyOS program and manual operations, the GenOS extends the TelosB node with the auxiliary processor STM32 for the GPRS module initialization and control [17]. The main processor MSP430F5438 communicates with the STM32 through the serial port. The STM32 processor is responsible for the power supply and management information control of the GPRS module. In STM32, the low-power control is implemented, and users can control the GPRS module standby. When MSP430F5438 receives sensor data, it establishes a connection to STM32 initiatively, and then sends the commands and data to STM32 through the serial port. After receiving the data, the STM32 will control the GPRS module to transfer the data to the specific remote server. All the data transmission and control is transparent to the user.

3.3. Hijack

Hijack is proposed by Ye-Sheng Kuo *et al.*, [10]. It is an interface using the headset pot to power and communicate with external peripherals. The headset port is a truly open interface in mobile phones, making it easy to use. The Hijack supplies power delivery and data transmission, and allows the mobile phones to integrate with various peripherals.

The Hijack has two main parts: the energy harvesting circuit and the microcontroller. The energy harvesting circuit uses lever circuit to generate power [18], and microcontroller generates encoded signals according to the received data from UART connected with the sink node. And the mobile phones sample these signals and decode them to digital data stream. This whole process is bi-directional, which means the mobile phones can also send commands to the peripherals via the Hijack interface.

The standard four-core headset port consists of the left channel, the right channel, the common ground ring and the microphone (Fig. 4). In Hijack, the left channel transfers data from the mobile phone to the microcontroller, the right channel and the common/ground ring provides AC power to the energy harvester and the microphone provides the encoded data from the microcontroller to the mobile phone.

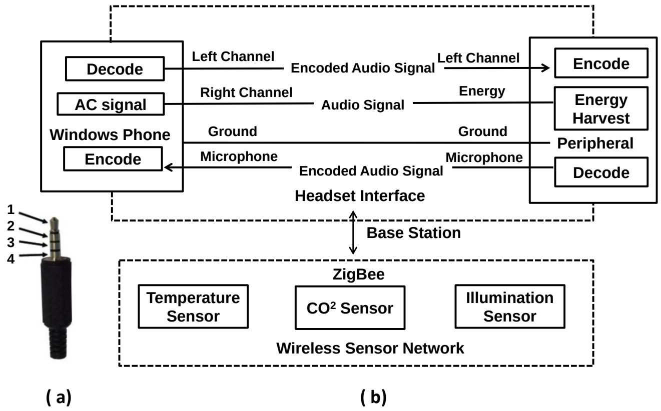

Figure 4: (a): The headset plug and pinout: (1) left earphone (tip), (2) right earphone (ring), (3) common/ground (ring), (4) microphone (sleeve). (b): The structure of Hijack system.

It is a big challenge to obtain energy from the headset port and convert it with high conversion efficiency [19]. The headset outputs a low voltage signal, which is even lower than the typical transistor's threshold voltage. So it is necessary to use energy harvesting circuit to convert it to a higher voltage signal. In Hijack, the energy harvesting circuit consists of a step-up micro-transformer, a FET-based rectification, some (parallel) blocking Schottky diodes and some filter capacitors [20]. The key design, the micro-transformer, leverages a recently introduced device for flyback and step-up in energy harvesting applications. With this design, the energy harvester can supply 7.4 mW voltage and the transfer scheme offers 8.82 kb bandwidth, which satisfies the interface in iPhone.

The other engineering challenge is to provide bi-directional communication between the mobile phone and the microcontroller [21]. To achieve this goal, the microcontroller needs to implement both the modulator function and the demodulator function. The Hijack uses the well-established Bell 202 signaling technique at a lower data rate of 0.3 kbps. At the digital level, the Hijack uses low-voltage RS-232 signaling to generate a virtual universal asynchronous receiver/transmitter (UART) abstraction over the audio serial bit stream.

The applications use the UART receiver and transmitter peripherals for the main communication interface. Other microcontroller peripherals connect with the UART via the pins of both the reception unit and the transmission unit for more processing.

3.4. uSDCard

The uSDCard is abbreviated from universal Sensor Data card, which is developed by the Research Center of Nokia [11]. It is used for accessing WSNs through smart phones with standard SD memory card slots. With the use of the uSDCard, users can monitor the states of the WSNs, collect the sensor data and even send commands to a specific node [22].

The uSDCard is an SD-based universal sensor data entry card with RF function. The uSDCard is completely compatible with normal SD memory card slots with standard physical and logical interfaces. So it is totally plug-and-play. Any device with an SD card slot can support a uSDCard directly without other drivers, regardless of the platform and the file system [23]. The SD Interface offers basic functions of SD card, such as reading and writing data, executing commands, and transferring the data. It is platform-independent and reprogrammable to adapt to various applications of WSNs. And the RF Part helps the uSDCard access WSNs as a gateway. The uSDCard can transfer the data received to the buffer for further process through various RF chips, such as ZigBee, Bluetooth, GPS, and Wi-Fi.

The hardware implementations of uSDCard consist of two data paths [24]. The first path transmits data between the WSNs and the host; the other is responsible for the memory control as normal SD card. The Data Path Selector transfers data between Radio transceiver/Memory and the host according to the address. Through the analysis of the address, the Data Path Selector chooses a data path to transfer data.

In the implementation of the uSDCard, a Special "File System " is designed to differentiate flash operation and RF operation, which appears to be the first data path between WSN and the host mentioned above [25]. The Special "File System " stores the file of WSNs in the FAT region with the table of names and addresses, without certain data. It is the key method of the communication between hosts and WSNs. When a host wants to get data from WSN, it sends the read operation to the virtual file system. According to the names and addresses carried by the operation, the file system gets the data from the specific node with the RF module. And then it transfers the data to the host through a FIFO Queue. With this method, the data operations all depend on the file system, which is performed independently.

3.5. Comparisons

We evaluate and compare hardware solutions mentioned above on the basis of the following criteria (TABLE 1): data source, network role, up-level device, mobility, ease of use, bi-direction, university and bandwidth. In the following discussion, we present a comprehensive discussion on the data source and the up-level device.

Data source/Up-level device. Data source and up-level device mean where the gateway receives the raw packets from and where the gateway sends the packets to. The MIB-510/600 requires to be connected with the sensor node, typically the base station, to receive the raw packets and send the packets to the serial port or the Ethernet port. Both of them are used for the data transmission without extra operations. However, the GenOS and the uSDCard can receive the raw data through the embedded module with RF function directly. They are responsible for both data collection and transmission. The Hijack offers the basic function of connection, so it is considered to be an "Interface". But all functions need specific device support and UART programming.

Mobility. In our discussion, mobility is the ability for remote data sharing, and it is the key criterion to the hardware solutions. As described above, the MIB-510/600 should be connected to the Serial/Ethernet port with the wired connecting, which limits its mobility. Nevertheless the other three solutions can be used to connect with smart phones or GPRS servers, which means the users can use these solutions for remote transmission though wireless connection.

Ease of use. Ease of use refers to the ability for rapid deployment and extension. The MIB-510/600, as a product of Crossbow Company, is totally plug-and-play, providing high-level ease of use. The GenOS can access the network and get the packets directly. Users can receive the packet in GPRS server simply. The uSDCard is basically the same, as it can receive the packets conveniently. But users need to program in the smart phone for parsing the packet. The Hijack, as an interface, requires UART programming operations, which causes some difficulties to users.

Bi-direction. Bi-direction communication concerns about whether the solution can send packets to the network in addition to receiving data from the network, and this is the key point for network management. The MIB-510/600 has the mature bi-direction communication. The data can be transmitted in two ways at the same time. The GenOS and the uSDCard are embedded with RF modules, supporting bi-direction communication function like normal nodes through peer to peer communication. The Hijack has the interface of bi-direction communication, which

Table 1. Comparison of the hardware solutions

	MIB-510	MIB-600	GenOS	Hijack	uSDCard
Data source	Terminal	Terminal	Embedded	Terminal	Embedded
Up-level device	Serial	Ethernet	GPRS server	Headset port	SD reader
Mobility	Low	Middle	High	High	High
Ease of use	High	High	High	Low	Middle
Bi-direction	High	High	High	Middle	High
University	High	High	Low	Middle	High
Bandwidth	Middle	Middle	Low	Low	High

needs extra UART connection and signal processing according to specific applications.

University. University makes the solution compatible with different wireless protocols, packets formats and client platforms. The MIB-510/600 and the uSDCard can select an optional terminal on the radio transceiver board as data source, which makes them adapt to different network environments. The Hijack can support various networks, but it needs more configuration and UART connection. The GenOS receives data from embedded modules, which limits its university.

Bandwidth. In our discussion, we use the bottleneck bandwidth as the metric to evaluate the entire gateway system. In the MIB-510/600, the interface bandwidth between RS-232 and PC can reach 115.2 kbps/10 Mbps. But the bottleneck is the bandwidth of data from a mote to the ISP via the connector, and its physical bandwidth is 115.2 kbps theoretically. The GenOS is restricted by the upstream bandwidth to the GPRS server, which only has 19.2 kbps physical bandwidth. In Hijack, the main bottleneck is smart phones' audio sampling rate, which makes its theoretical bandwidth is 22 kbps. The uSDCard supplies fast data transmission via a standard SD interface, so the bottleneck is the embedded RF module. It can theoretically provide the highest bandwidth, depending on wireless communication protocols.

From the comparison above, we can get the idea about the appropriate scene and application for each hardware solution. The MIB-510/600 and the GenOS are more applicable for the rapid deployment within existing networks. The MIB-510/600, as a product, provides convenience and university of the highest level. The GenOS can support remote data transmission and remote management of the network via GPRS module, but can only be used in the network compatible with its embedded RF module. The uSDCard and the Hijack can be easily connected with the smart phone, providing better mobility. The uSDCard has higher bandwidth and the Hijack is more universal as a peripheral device using standard headset port. However, the Hijack needs extra UART connection and signal processing, making it hard to use.

4. Middleware solutions

The middleware solutions are located between the transmission layer and the application layer theoretically (as the WSN3 in Fig. 1). In this solution, the entire system needs to process appropriate protocols, typically the IP protocol. After protocol conversion, the data packet has already been able to be recognized in networks of other types. So the sink node only processes the data transmission and sends the data to the designated device. The middleware solutions need more cooperation between nodes, instead of burden on the gateway. This kind of solution requires less hardware support and is easier for the reuse of different applications and networks, but the efficiency is relatively low.

The middleware solutions include: the VIP Bridge integrates the WSN and the traditional IP-based network using mapping table and virtual IP; the GSN is a flexible middleware for the rapid deployment and the network integrations with declarative specifications; the IP-based solution focuses on the integration of IP stack and WSN using the adaptation layer, one of whose famous implementation is 6LoWPAN; the data storage based solution uses software implementation to regard the host as the data source for data storage and data sharing.

4.1. VIP Bridge

VIP Bridge [26] is a bridge-based middleware to integrate WSNs with the traditional IP-based networks. Through this virtual network, users can obtain the data from specific sensor nodes directly and easily.

In the future network, each network device, with unique IPv6 address, should provide pervasive accessibility and mobility for users [27]. Internet users should be able to access and use the services provided by heterogeneous wireless networks transparently, which can be achieved easily by the VIP Bridge. The VIP Bridge maps the node ID in WSNs with IP address inside the bridge. And the IP address of each node will be stored in the bridge as a virtual IP address,

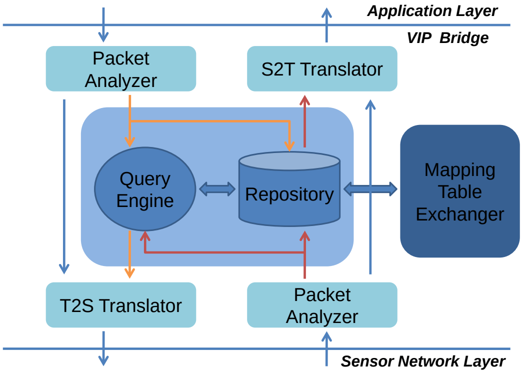

Figure 5. Components of VIP Bridge.

but not just be physically deployed on the sensor node. Packets that come from one side will be translated into corresponding packet formats and sent to another side via the VIP Bridge. The packet routing is based on the virtual IP address and the original IP address.

The VIP Bridge (Fig. 5) is composed of two Packet Analyzers, a Repository and a Mapping Table Exchanger. One Packet Analyzer is TCP/IP Network to Sensor Networks Packet Translator (T2S) and the other is Sensor Networks to TCP/IP Network Packet Translator (S2T). They analyze and translate packets from the applications and the WSN respectively. A Repository is physically located in the VIP Bridge, which stores all this information the packet analyzers processed. The VIP Bridge will map these different kinds of information. After packet analysis, query packets are sent to Query Engine to compose the new packet format. The Mapping Table Exchanger component exchanges the mapping tables between different VIP Bridges, and integrates all the VIP Bridges in this way. And here the XML is a proper way to express the Mapping Tables for its independence of any operating systems or protocols [28].

In the Z-IP project [29], Lei Shu *et al.*, implemented the VIP Bridge using the existing famous ZigBee as the routing protocol in WSNs. In this project, each sensor node has its own unique identity, the node ID. The VIP Bridge assigns global unique IPv6 address for each sensor nodes in Repository. By doing the mapping, the Internet users can easily find the specific sensor node and get the data through the corresponding IP address and ZigBee address. When the VIP Bridge receives the data query from the IP network, it will search the mapping table to get the node ID. And then it creates another packet for ZigBee routing in sensor networks. After querying, packets originally came from the WSN can also follow the same procedure to be sent back to users of IP-based networks.

4.2. Global Sensor Networks

The Global Sensor Networks (GSN) is a flexible middleware for the rapid deployment and network integration [30]. It abstracts from the underlying heterogeneous sensor network and enables the dynamic adaption of the system configuration during runtime with minimal effort. It provides the support for sensor node mobility and distributed query processing.

The GSN targets at flexible configurations and the integration with the existing approaches, through the way of abstraction and distributed query support. With the GSN, the sensor nodes can use the specific routing algorithms to deliver the sensor data to the sink node like normal WSN applications [31]. Then the sink node connects with the base computer via a software wrapper conforming to the GSN API. On the upper level of this physical layer, the GSN provides so-called virtual sensors. They abstract from implementation details of accessing to sensor data and the data stream received from sensors directly or other virtual sensors. The GSN takes a general view and provides API for query processing and management infrastructure with a declarative language interface. The GSN takes a peer-to-peer perspective, which supports relational queries using SQL.

Architecturally, the GSN adopts a service-oriented view on sensor networks [32]. In this view, the sensor networks are considered as abstract types of services, which perform a sensing task and provide a specific type of data. The sensor services are published through Zigbee networks based on their properties. Applications can discover sensor networks using the registry and access sensor networks by a standard data access interface.

The key abstraction in the GSN is the virtual sensor, which can be any kind of data producer. A virtual sensor have any number of input data streams and produces exactly one output data stream based on the input data streams [33]. The virtual sensor enables the user to declaratively specify sensors and combinations for the specific application complexity. To support rapid deployment, these properties of virtual sensors are provided in a declarative deployment descriptor, which is specified in the virtual sensor specification in XML. The input and output stream specifications provide various attributes for the control of the data processing, which is specified in SQL. Then the SQL query optimization and planning techniques can be directly applied, making the GSN easy to use and promote.

The GSN uses a container-based architecture to manage the virtual sensors [34]. With the declarative specifications, the virtual sensors can be deployed and reconfigured in GSN containers at runtime. Communication and processing among different GSN containers is performed in a peer-to-peer structure through standard protocols. The GSN dynamically instantiates the new virtual sensor based on this synthesized description when a new sensor node is detected by the GSN. And all local and remote processing dependent on the new sensor is executed in the runtime.

4.3. IP based solutions

With the Internet being widely used, resent studies make efforts on integrating WSNs with the Internet [35]. So the deployment of the IP stack within WSNs, which is the so-called IP based solutions, attracts more attention from the academia [36]. With the use of IP, WSN nodes can communicate with the IP-based devices directly. The end-to-end communication provides much greater flexibility and robustness in deployment. These features are important to bring the Internet into a new generation.

In order to implement IP over WSN, there are several technical challenges to deal with [37], which can be mainly identified as: large header overhead, lack of transport protocol limited energy, limited bandwidth. Among these challenges, the key point is the large header overhead, because it makes the payload too small and the transmissions inefficient. So it is necessary to use an adaptation layer to fragment the IP data packet and compress the IP packet header. In the layered architecture, the adaptation layer is between the MAC layer and the network layer as show in Fig. 6. And the two functions of the adaptation layer are the fragmentation of IP datagram and the header compression [38].

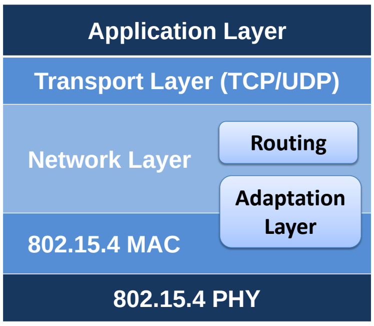

Figure 6: The IP stack with the adaptation layer

Fragmentation is the basic mechanism provided by the adaptation layer [39]. When the data packets cannot fit the 802.15.4 MAC frame payload size, the packets are fragmented into multiple link-layer frames. It simply provides the ability to encode a datagram, but not includes the end-to-end recovery of the lost fragments.

Since the flow path changes frequently, the header compression techniques using in the traditional IP stack are not applicable for WSNs, which requires to develop a new efficient header compression technique. In the IP stack over WSN, there are two main strategies: one is eliding the redundant information, which can be got in the link layer or the network layer, like the payload length and the IP version; the other way is assuming common values for header fields and defining compact forms of those values.

4.4. Data storage based

Generally, sensor data from the non-gateway WSN is usually used in the local application. In other words, it is difficult for other applications to use the data. However, a data set can support several different researches simultaneously. Therefore, it is intuitive to come up with an idea that the data source should be preserved as an intermediate result in a particular format for reusing.

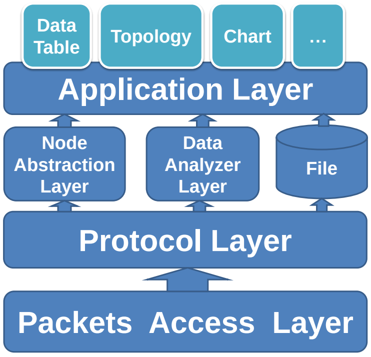

Figure 7: The model of the data storage technique in WSN.

The gateway in a network serves as the access point for the network, focusing on the network integration. But from the perspective of data sharing, the data storage in a specific device can also achieve the function of a gateway [40]. In TinyOS, Serial Forwarder is a tool which allows other applications to connect over TCP/IP stream to use the data source. Here the Serial Forwarder acts as the gateway, supporting the data reuse.

There have already been models to translate data to a specific format (e.g. XML) as the intermediate result, making the data platform independent. This model could be divided into five layers, each of which is independent (Fig. 7). The lowest layer, called Packets Access Layer, is mainly responsible for receiving the raw data from other devices. This layer is similar to the collection layer in WSNs. The upper layer is Protocol layer, which is usually used for packet parsing. Users can choose different protocols according to different purposes and conditions. No matter which protocol is used, it should complete several functions, like packets parsing, packets verifying, and error

Table 2. Comparison of the Middleware solutions

	VIP	GSN	IP-based	Data storage
Main Feature	Protocol conversion Mapping table Virtual IP	Virtual sensor Declarative specifications Network integration	Adaption layer IP overlay WSN 6Lowpan	Packet parsing Data fusion Data source
Location	Gateway	PC	Mote	PC
Energy efficiency	Middle	High	Low	High
Consistency	Middle	High	Middle	High
Transparency	High	High	High	Middle
Direct accessibility	High	Low	High	Low
Ease of use	High	Middle	Low	Middle

Table 3. Hardware solutions vs. Middleware solutions

	Hardware solutions	Middleware solutions
Implementation	Hardware	Software
Concentrate on	Data sharing and usage of data	Network integration
Platform independence	Platform independent	Depend on specific platform
Terminal access	Direct access via point to point communication	Needs intermediate device to process
Impact on original network	Almost no impact	Need to adjust to meet the requirement for th
Reuse in other networks	Hard to reuse for the embedded device	Protocol and communication transparent fo
Extra manual operation	Plug-and-play	Data format setting and specific software c
Efficiency	Directly raw data transmission	Package operation and more overh

packets discarding. Next layer is designed to store the intermediate result. Two tasks must be done in this layer: one is to decide which format is used for data storage; the other is to translate the data received from the lower layer and preserve the data. Other two layers are data processing layer and application layer, which we do not focus in this paper.

The Longhui Ma *et al.*, implemented this method in a software called NetViewer, which can save intermediate data and acts as the data source [41]. This tool uses the five layer model and data fusion technology. The NetViewer sets the format according to the demand of users and an XML file will be produced during this procedure. And then it receives the data from different sources and then the packets are translated with the format defined in protocol layer previously. the NetViewer displays the data to clients in various ways. User can get the topology graph and the status of the network through GUI.

4.5. Comparison

The same as above, we evaluate and compare the middleware solutions according to the following criteria (TABLE 2): location, consistency, transparency, energy efficiency, direct accessibility and ease of use. Since the energy efficiency is directly related to the location where the middleware is deployed, we comprehensively discuss these two criteria. And in the table, we also list the main features of each solution, aiming to present an intuitive distinction between them.

Location. Location refers to where the middleware is deployed, and it plays an important role in the energy efficiency. Here we consider more about and energy consumption in sensor nodes. The GSN and the data storage are usually applied in the PC client, guaranteeing them the best energy efficiency without extra energy consumption. The VIP Bridge is applied in the gateway, and the IP-based solution runs the IP protocol in normal sensor nodes. These two solutions will increase the energy consumption in the sensor nodes, which reduces the efficiency.

Consistency. Consistency is the ability to be compatible with different platforms and protocols. The feature makes the middleware easily be extended and modified The VIP Bridge and the IP-based solution are based on the IP stack, supporting all the IP-based networks. But beyond the IP stack, these two solutions need some changes. The GSN has high consistency in terms of the use of declarative specifications, providing different networks with a unified platform. The data storage solution gets the data stream from the terminal and parses the packets according to the manual setting, which makes it consistent with different network environments.

Transparency. Transparency is a key feature in the middleware. The users are able to use the services without knowing the underlying implementation. The data storage solution parses the packets according to the manual settings, and sends the processed packets to the users as data forwarder. It means all the processes need the participation of users. However, when the other three middlewares have been applied, they can provide service automatically to achieve transparency of a higher level.

Direct accessibility. Direct accessibility enables users to directly access a specific node and process necessary operations. Meanwhile, it requires every node to identify itself by a unique node ID or an IP address. The VIP Bridge and the IP-based solution are totally based on IP, supporting the direct access to every node in the network through IP addresses. Differently, the GSN and the data storage solution concern more about the network integration and the data collection, providing few operations for the direct node access. Users can only get the information of the node through GUI.

Ease of use. To achieve the ease of use, bridging solutions should be user-friendly. They should provide interfaces for users to deal with the early stage configuration and visit complex heterogeneous resources of the system. In the VIP Bridge, users only need to set the address mapping table in application layer. In the GSN and the data storage solution, it is required to configure the middleware by using XML for the rapid deployment. And in the IP based solution, the IP stack module should be added to the applications, which needs more manual operations.

Through the comparisons, it shows that the GSN solution is more like a network integration method instead of a data sharing method, so it is more applicable to provide large-scale networks with rapid deployment and convenient management. The VIP Bridge, as a traditional middleware for address mapping, supplies network integration and data sharing. Users only need to set simple configuration in normal scenes and applications. But now it only supports the IP stack, which limits its usage. The data storage solution, as a middleware based on the data storage and packet parsing, can be used as the data source for different kinds of clients. It is more applicable to provide service for the cross-platform applications, and the data needs to be shared in different platforms. However, the IP-based solution seems hard to use. For the reason of lack of address, limited computational capabilities and memory resources on nodes, it is unreasonable and unprofitable in recent applications. But it should be noticed that, with the widely use of IPv6, it is expected to integrate the WSN with the Internet in the future. This solution guarantees flexibility and scalability of a higher level. So it is open to discussion now, and needs more work to be done.

5. Discussion

As mentioned above, this paper focuses on the bridging solutions used in the integration, which fall into two categories: hardware solutions and middleware solutions (TABLE 3).

Generally speaking, hardware solutions are more applicable for the rapid deployment of small-scale network, or to be the supplement for the existing network. They can provide efficient data transmission conveniently without additional packet load. And they will not change the internal structure of the existing networks. Moreover, this kind of gateway can be connected with the mobile device easily, especially the smart phones. However, middleware solutions supply higher standardized and higher universal service, which makes them more applicable for the deployment of the large-scale networks in the early stage. In addition to achieving data sharing, this kind of bridging solutions supports the integration of different kinds of networks, providing unified solution for management. Each of the various solutions has its own characteristics in terms of the transmission mode, the data transmission rate and the cost. Users are supposed to select proper method depending on specific application according to the features described in the comparison part above.

6. Conclusion

This paper focuses on the solutions to integrate between the WSNs and other networks, which is a key challenge in IoT. Based on the role in the whole network and the way of implementation, we divide the main bridging solutions into two categories: hardware solutions and the middleware solutions. In this paper, we present the implementation details and key points of each solution and evaluate the advantages and disadvantages according to different criteria. We aim to describe the most applicable scenes and applications for each solution through the comparisons and discussions with our best efforts.

The future applications will contain more management component for the sensor nodes. This function requires the full-duplex bridging technology, which raises big challenges to the wireless transmission channels. At the same time, the researchers should concern more about the different ways to access WSNs. The more ways there are, the easier it is to integrate WSNs with other networks. It needs great concerns on universal interfaces used in the smart phones, such as Bluetooth and Wi-Fi, which will significantly enhance the mobility of WSNs application. In WSNs, the energy consumption problems have to be considered, so designing the energy saving bridging solutions is an important direction. Finally, with the use of bridging solutions, it needs to avoid single point failure problem. And it is necessary to guarantee the bridging security including the software security and the hardware security.

Acknowledgment

This work is partially supported by the Natural Science Foundation of China under Grant No. 61070181, Grant No. 60903153 and Grant No. 61170296 and is a research gift by Nokia Research Center. The whole uSD Research Group of Nokia Research Center gives our work great help, thanks for their technological supporting. Special thanks to Jin Cheng, Zhang Tianyu and Zhao Jing for their revision and insightful comments.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," in *Computer networks*, 2002.

[2] K. Crowley, J. Frisby, S. Murphy, M. Roantree, and D. Diamonda, "Web-based real-time temperature monitoring of shellfish catches using a wireless sensor network," in *Sensors and Actuators. A, Physical*, 2005.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 263–270.

[4] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Ad Hoc Networks Journal*, 2003.

[5] H. Dai and R. Han, "Unifying micro sensor networks with the internet via overlay networking [wireless networks]," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 571–572.

[6] P. Mohanty, "A framework for interconnecting wireless sensor and ip networks," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE, 2007, pp. 1–3.

[7] K. Emara, M. Abdeen, and M. Hashem, "A gateway-based framework for transparent interconnection between wsn and ip network," in *EUROCON 2009, EUROCON'09. IEEE*. IEEE, 2009, pp. 1775–1780.

[8] V. Tang, Y. Zheng, and J. Cao, "An intelligent car park management system based on wireless sensor networks," in *Pervasive Computing and Applications, 2006 1st International Symposium on*. IEEE, 2006, pp. 65–70.

[9] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices," in *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, 2009.

[10] Y. Kuo, S. Verma, T. Schmid, and P. Dutta, "Hijacking power and bandwidth from the mobile phone's audio interface," in *Proceedings of the First ACM Symposium on Computing for Development*. ACM, 2010, p. 24.

[11] C. Jiang, N. He, Y. Ren, C. Chen, and J. Ma, "usd: universal sensor data entry card," in *Consumer Electronics, IEEE Transactions on*, 2010.

[12] D. Jea and M. Srivastava, "Channels characteristics for on-body mica2dot wireless sensor networks," in *Proc. Mobiquitous*, 2005.

[13] J. Mache, C. Allick, J. Charnas, A. Hickman, and D. Tyman, "Sensor network lab exercises using tinyos and micaz motes," in *Proceedings of the International Conference on Pervasive Systems and Computing (Las Vegas, NV*. Citeseer, 2006.

[14] N. O'Donoughue, S. Kulkarni, and D. Marzella, "Design and implementation of a framework for monitoring patients in hospitals using wireless sensors in ad hoc configuration," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006, pp. 6449–6452.

[15] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. Ieee, 2005, pp. 364–369.

[16] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "Iot gateway: Bridgingwireless sensor networks into internet of things," in *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*. Ieee, 2010, pp. 347–352.

[17] J. ZHAO and Q. LI, "Design of gprs-based heating network remote control system," *Control and Instruments in Chemical Industry*, 2012.

[18] A. Sample, D. Yeager, P. Powledge, A. Mamishev, and J. Smith, "Design of an rfid-base battery-free programmable sensing platform," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 11, pp. 2608–2615, 2008.

[19] J. Gummeson, S. Clark, K. Fu, and D. Ganesan, "On the limits of effective hybrid micro-energy harvesting on mobile crfid sensors," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 195–208.

[20] Y. Ramadass and A. Chandrakasan, "An efficient piezoelectric energy harvesting interface circuit using a bias-flip rectifier and shared inductor," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 189–204, 2010.

[21] E. Koutroulis, K. Kalaitzakis, and N. Voulgaris, "Development of a microcontroller-based, photovoltaic maximum power point tracking control system," *Power Electronics, IEEE Transactions on*, vol. 16, no. 1, pp. 46–54, 2001.

[22] C. Chen, X. Zhang, J. Zhang, and Y. Tang, "usd card: a plug&play solution for mobile device to access wireless sensor networks," *Wireless Algorithms, Systems, and Applications*, pp. 354–365, 2011.

[23] W. Yue, L. Wang, M. Zhu, Z. Qin, L. Shu, and C. Chen, "Poster: A green solution for intelligent metropolitan heating system with usdcard," in *MobiSys*, 2011.

[24] S. Group *et al.*, "Sd memory card simplified specifications, part 1, physical layer specification," *2th version edition*, 2006.

[25] M. Corporation, "Microsoft extensible firmware initiative fat32 file system specification," *1th version edition*, 2000.

[26] S. Lei, H. Xu, W. Xiaoling, Z. Lin, J. Cho, and S. Lee, "Vip bridge: Integrating several sensor networks into

one virtual sensor network," in *Internet Surveillance and Protection, 2006. ICISP'06. International Conference on*. IEEE, 2006, pp. 2–2.

[27] G. Legg, "Beyond 3g: The changing face of cellular," 2005.

[28] F. Curbera, F. Leymann, T. Storey, D. Ferguson, and S. Weerawarana, *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR, 2005.

[29] J. LeiShu, S. Lee, M. Hauswirth, and L. Zhang, "Vip bridge: leading ubiquitous sensor networks to the next generation," *Journal of Internet Technology*, vol. 8, no. 3, p. 2, 2007.

[30] K. Aberer, M. Hauswirth, and A. Salehi, "The global sensor networks middleware for efficient and flexibl deployment and interconnection of sensor networks," *Swiss Federal Institute of Technology, Lausanne (EPFL), Tech. Rep*, 2006.

[31] L. Shu, M. Hauswirth, L. Cheng, J. Ma, V. Reynolds, and L. Zhang, "Sharing worldwide sensor network," in *Applications and the Internet, 2008. SAINT 2008. International Symposium on*. IEEE, 2008, pp. 189–192.

[32] M. Sgroi, A. Wolisz, A. Sangiovanni-Vincentelli, and J. Rabaey, "A service-based universal application interface for ad hoc wireless sensor and actuator networks," *Ambient Intelligence*, pp. 149–172, 2005.

[33] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, Y. Xing, and S. Zdonik, "Scalable distributed stream processing," in *Proc. Conf. on Innovative Data Syst. Res*, 2003.

[34] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Mobile Data Management, 2007 International Conference on*. IEEE, 2007, pp. 198–205.

[35] M. Durvy, J. Abeill, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne *et al.*, "Making sensor networks ipv6 ready," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 421–422.

[36] M. Islam and E. Huh, "Sensor proxy mobile ipv6 (spmipv6) ąła novel scheme for mobility supported ip-wsns," *Sensors*, vol. 11, no. 2, pp. 1865–1887, 2011.

[37] J. Rodrigues and P. Neves, "A survey on ip-based wireless sensor network solutions," *International Journal of Communication Systems*, vol. 23, no. 8, pp. 963–981, 2010.

[38] H. Kim, "Protection against packet fragmentation attacks at 6lowpan adaptation layer," in *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on*. IEEE, 2008, pp. 796–801.

[39] A. Ludovici, A. Calveras, and J. Casademont, "Forwarding techniques for ip fragmented packets in a real 6lowpan network," *Sensors*, vol. 11, no. 1, pp. 992–1008, 2011.

[40] K. Ahmed and M. Gregory, "Techniques and challenges of data centric storage scheme in wireless sensor network," *Journal of Sensor and Actuator Networks*, vol. 1, no. 1, pp. 59–85, 2012.

[41] L. Ma, L. Wang, L. Shu, J. Zhao, S. Li, Z. Yuan, and N. Ding, "Netviewer: A universal visualization tool for wireless sensor networks," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. IEEE, 2010, pp. 1–5.