Design and Implementation of Multi-Protocol Fusion Wireless Sensor Network

Zheng Liu [a)] and Guangcong Liu

[1] *School of computing, Guangdong University of Technology, Guangzhou, Guangdong 510006, China*

[a)] Corresponding author: 545701966@qq.com

Abstract. Using WSNs (Wireless Sensor Networks) and Internet to realize the unification addressing, collection of big data era need huge amounts of information and establish a wider range of universality to reduce differences design complexity is the IoT (Internet of Things) development of new ideas in recent years. This paper proposes and implements a fusion based on IPv6 gateway protocol of wireless sensor network system, ZigBee, Wi-Fi, Bluetooth sensor can communicate with each other under the IPv6 gateway, to meet the wireless sensor network in a variety of needs of different under different application environment, compared to a single communication protocol of WSNs more flexibility and universality. First, we design a gateway architecture based on IPv6, implement different wireless devices, all adopt IPv6 protocol to the format of the communication layer is unified, and Internet address. Second, to achieve the ZigBee, Bluetooth, Wi-Fi devices IPv6 transplantation, relative to the adoption of IPv6 wireless sensor, not avoid the gateway for complex application layer protocol conversion, reduce the gateway design complexity and power consumption.

Key words: WSNs; ZigBee; IPv6; Wi-Fi; flexibility; design; application environment.

INTRODUCTION

Wireless sensor remote monitoring and control technology is a new research hotspot. Managers connect to the wireless sensor less network equipment to be controlled through the Internet to collect, monitor and modify the wireless sensor node data.

However, complete IPv6 is difficult to implement on resource-constrained sensor nodes. Many people have reduced the IPv6 protocol to fit the use of sensor nodes [1-3]. However, there is no single norm. In order to solve this problem, the IETF (Internet Engineering Task Force) group set up a 6LoWPAN working group to formulate the corresponding standards. At present, the "6LoWPAN Overview", "IPv6 Packet Format for Transmission over IEEE 802.15.4" and "IPv6 Header Compression Specification" have been completed [4-5]. The new wireless Internet of Things technology uses the sensor network system of the 6LoWPAN protocol stack. The application layer program on the Internet host only needs to know the IP address of the sensor node to communicate with it end-to-end without knowing the gateway and the aggregation node. Existence greatly simplifies the network programming model of the sensor network system, reduces the system complexity and increases the efficiency [6].

The communication technology of the wireless sensor network is not only based on the Zigbee of 802.15.4 protocol, but also Bluetooth and Wi-Fi communication technology. Each of these communication technologies has its own advantages and disadvantages: The Wi-Fi signal is strong, the control is sensitive, it can support the audio and video transmission flow of instant messaging, but the power consumption is large; Zigbee has low cost and low power consumption, but the transmission rate and distance are higher than that of Wi-Fi. To be low; Bluetooth has a wide range of mobile applications but has a short transmission distance and low bandwidth. The investigation and research found that in the fields of smart home, industrial automation, and agricultural automation, it is difficult for the wireless sensor network to use a single communication technology to achieve the optimization of benefits and efficiency. In order to expand the WSNs communication range, and flexibility. [7] proposed a basic Wi-Fi Internet monitoring solution, but still used the gateway as an application layer protocol and did not implement end-to-end interconnection. Literature [8] proposed an IPv6-based low-power Bluetooth node architecture that enables end-to-end connectivity between Internet and Bluetooth sensor network nodes. However, the communication protocol is single, and the range of wireless Bluetooth wireless communication is too narrow. Applicability is not. Strong. [9] designed a ZigBee-Wi-Fi gateway to extend the communication range and realize the protocol conversion between ZigBee network and Wi-Fi network by solidifying the code in the chip. Although this kind of network can realize the fusion and interaction, the disadvantages are very high. Obviously, the first two microprocessors are used, which are costly, and secondly, it is difficult for such a gateway architecture to continue to expand and add other wireless communication networks.

In order to achieve end-to-end communication between Internet and wireless sensor networks based on IPv6 and various communication technologies such as Zigbee, Bluetooth, and Wi-Fi can be applied interactively in the wireless Internet of Things, this paper proposes an IPv6-based multi-protocol wireless sensor network. Fusion method. Subsequent chapters will introduce the architecture of this scheme, the detailed design of the system and the experimental results.

ARCHITECTURE

This section addresses the needs of the Internet and Zigbee, Bluetooth and Wi-Fi wireless sensor network interconnection [10], designed a multi-network fusion system software framework, proposed an IPv6-based gateway architecture and detailed introduced the gateway communication principle and protocol stack level.

1.System Software Work Framework:

The IPv6 intelligent gateway integrates the three major modules of 802.15.4 routing, WI-FI AP and Bluetooth Master. It is equivalent to the boundary sensor connecting the wireless sensor network and the gateway, provides routing functions, and supports the data connection of four types of wireless nodes. Incoming, through the gateway Linux operating system under the IPv6 protocol conversion process, providing external application service interface. The upper application can develop Internet of Things applications under Windows, Android, and Web based on the CoAP library. In this architecture framework, the IPv6 intelligent gateway integrates three types of networks, including 802.15.4, Bluetooth network, and WIFI network. Three different sensor nodes communicate with the gateway in different ways through the CoAP protocol.

2.IPv6 Gateway Architecture:

Three different node modules communicate with the gateway in different ways. The 802.15.4 border router module communicates with the gateway Linux kernel through a serial connection, and the Bluetooth and Wi-Fi modules connect via Bluetooth to the Bluetooth network and Wi-Fi and gateway Linux. The kernel communicates. In order to realize multi-protocol converged communication, the IPv6 gateway will switch the three different communication channels to the TUN/TAP virtual network device. The TUN/TAP driver implements the function of the virtual network card. TUN means that the virtual is a point-to-point device, and TAP means virtual Ethernet devices, which implement different encapsulation for network packets [11].

With the TUN/TAP driver, the network processed by the TCP/IP stack can be subcontracted to any process that uses the TUN/TAP driver, which is then reprocessed by the process and sent to the physical link. The TUN/TAP virtual network device ensures that the ZigBee subnet, Bluetooth subnet, and Wi-Fi subnet based on 802.15.4 are communicating on the IPv6 gateway at the same time.

SYSTEM DESIGN

Based on the protocol specification defined by 6LoWPAN, this chapter introduces the communication principles of IPv6 gateways and ZigBee nodes. At the same time, it designs the migration of Bluetooth and Wi-Fi nodes to IPv6, and the communication principles between IPv6 gateways and Bluetooth and Wi-Fi nodes, and implements the gateway hardware according to these principles. design.

1. IPv6 porting of Bluetooth and Wi-Fi sensor nodes:

Node 1, the sensor structure design based on the IPv6 protocol uses three parts: a sensor module, a chip based on the Contiki system, and a wireless module. The Contiki operating system is used in the sensor chip because the Contiki system is a highly portable multitasking operating system for wireless sensor networks. The platform has a simplified 6LoWPAN protocol stack: uIPv6 [12], which occupies only 11KB of ROM and 1.8KB of RAM, using uIPv6 protocol stack, so that the data transmitted by the sensor node can be

To meet the IPv6 packet format. The uIPv6 protocol stack communicates with upper layer applications and underlying hardware device drivers by defining a set of interface functions [13]. The processing of the header compression, decompression, and fragment reassembly of the datagram transmitted by the uIPv6 protocol stack is performed by the 6LoWPAN adaptation layer.

At the same time, the acquisition and control of the wireless sensor node is carried out through the CoAP protocol and gateway integrated in the Contiki system, and the Contiki system chip is connected to the sensor module and the wireless module through the serial port. The Contiki system-integrated CoAP protocol is an application layer protocol in the 6LoWPAN protocol stack. It is a network-oriented protocol that can be easily translated to HTTP. Provides convenience for gateways to connect to web servers [14]. The CoAP protocol provides services such as header compression, multicast support, and asynchronous message exchange. The transport layer uses the UDP protocol, which greatly reduces overhead.

ZigBee, Bluetooth, and Wi-Fi sensor nodes need to implement IPv6 migration and must have an IPv6 address for the end-to-end connection with the host. These three communication protocols cannot achieve automatic IPv6 address acquisition because the method of address acquisition needs to be configured here. Stateless autoconfiguration, specifically configured as follows: When the length of the MAC address of the wireless module is 8, it indicates that the address is the ZigBee wireless module Mac address. At this point, the ZigBee wireless module network address is generated as: ZigBee gateway network address + ZigBee wireless module Mac address the first byte is XORed with 0x02 + ZigBee wireless module Mac address is 2-8 bytes. When the length of the wireless module's Mac address is 6, it indicates that the address is a blue/Wi-Fi wireless module address, and the blue/Wi-Fi wireless module network address is generated as: blue/Wi-Fi gateway network address + 0x02+0xfe+ blue/Wi-Fi wireless module Mac The first 1-6 bytes of the address.

2. IPv6 Gateway and Sensor Node Communication Software Implementation

This section details the communication principles of the ZigBee wireless sensor network, Bluetooth, and Wi-Fi wireless sensor networks and the gateway.

Figure 5 shows that 802.15.4 border routers, namely ZigBee border routers, use the serial port driver to send the Slip packets to the Tunslip6 service, and the Tunslip6 service converts them to IPv6 packets and forwards them to the TUN virtual device. The TUN virtual device reports the IPv6 packets to Linux. The IPv6 layer of the kernel is sent to the user's APP for processing through the TCP/UDP protocol.

On the other hand, the user APP sends the IPv6 data packet request to the TUN virtual device. The Tunslip6 service converts the IPv6 data packet sent from the TUN virtual device into the Slip data packet and then sends it to the ZigBee border router through the serial port driver, and then sends the Slip data packet through the border router. Forward to other ZigBee node modules on the Internet.

The communication principle of the Bluetooth and Wi-Fi module gateways is similar to that of ZigBee, as shown in Figure 6, which is the schematic diagram of Bluetooth module and gateway communication. The user APP sends the IPv6 data packet request to the TUN virtual device. The Tunslip6 service converts the IPv6 data packet sent from the TUN virtual device to the Slip data packet. After the Tunslip6 service establishes a TCP connection with the BT server service, it forwards the data to the BT server through port 60001. The BT Server forwards the Slip data packet to the underlying BT driver through the port number, and then forwards the Slip data packet to other Bluetooth module nodes on the network through the Bluetooth module.

The work flow of the WIFI network on the IPv6 gateway is similar to the Bluetooth network. Only the listening port is different from the gateway service program.

3. Multi-protocol Fusion System Implementation:

Hardware implementation of IPv6 intelligent gateway: CPU uses high-performance Samsung ARM Cortex-A8 processor, clocked at 1GHZ, memory 512MB, Flash4GB, 7-inch true color LCD screen (AT070TN92), resolution 800*480, with 7-inch multi-point Capacitive touch screen. The connection between the gateway and the Internet network uses serial ports and 3G communication modules. Xidao Canal et al. proved that radio frequency access instead of serial communication can improve the communication speed in the communication process [15]. IPv6 gateway design On-board 3GW CDMA (MW100G) communication module enables the gateway to connect to Ethernet as long as it has a 3G network. Compared to the 4G communication module used by Yongdaoqu et al., the 3G network covers a larger area and is more universal. The gateway integrates the wireless WIFI AP module, the wireless Bluetooth Master module, and the wireless Zigbee module, which is equivalent to the border router in the traditional wireless sensor network. Integrated rich peripheral expansion interface, 4-way RS232 serial port, integrated USB2.0 HUB, 4-way independent USB HOST interface, 1-channel USB OTG interface (Mini USB interface), support for up to 32G 2-way SD card. The use of the embedded mainstream operating system Android+Linx on the IPv6 gateway software facilitates application development at the Internet of Things application layer and gateway service layer. The format of the route packet received by the Android server is shown in the following table, where the address is in the size of 1 byte in addition to the other parts of the address, and the size of the address part is (np*2+nc*2+nn*2) byte.

TABLE 1. Routing packet format

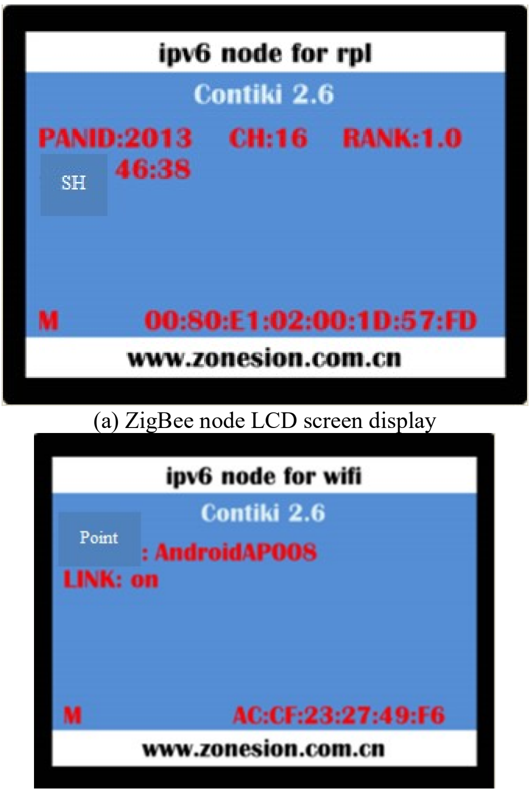
node	Number of parent nodes np	Number of child nodes nc	Number of neighbors nn	Address Sensors	Types of

Wireless sensor hardware implementation: In order to simplify the reuse, a modular wireless sensor is designed, including STM32 embedded backplane, wireless core module, sensor module, LCD display panel and serial port jumper. There are three types of wireless core modules: Zigbee core board with CC2530 processor, WIFI core board with LPA transparently transmitting serial WIFI module, Bluetooth core board with HC05 transparent serial port Bluetooth; Sensor interface board with 12PIN interface, including SPI/IIC / / GPIO / PWM / power / ground and other signals, made nine kinds of sensor modules, including: three-axis acceleration, temperature and humidity, fan, air quality, ultrasound, human infrared, relays, Hall, photosensitive. Among them, the wireless core module, the sensor module, and the LCD display board are all embedded in the bottom board with pins and can be replaced and disassembled at any time.

EXPERIMENTAL RESULTS AND ANALYSIS

This multi-protocol wireless sensor network test system consists of an IPv6 gateway, three ZigBee wireless sensors, three Wi-Fi wireless sensors, three Bluetooth wireless sensors, an IPv6 emulator, a PC, and a serial cable.

After connecting the test system, check the LCD display of one of the ZigBee wireless sensors. As shown in Figure1 (a), the MAC address of the wireless node is: 00:80: E1:02:00:1D:57: FD, Figure.1 (b) shows the MAC address of a Wi-Fi node.



ipv6 node for rpl
Contiki 2.6
PANID:2013 CH:16 RANK:1.0
SH 46:38
M 00:80:E1:02:00:1D:57:FD
www.zonesion.com.cn

(a) ZigBee node LCD screen display

ipv6 node for wifi
Contiki 2.6
Point : AndroidAP008
LINK: on
M AC:CF:23:27:49:F6
www.zonesion.com.cn

(b) Wi-Fi node LCD screen display

FIGURE 1. ZigBee wireless node screen display information

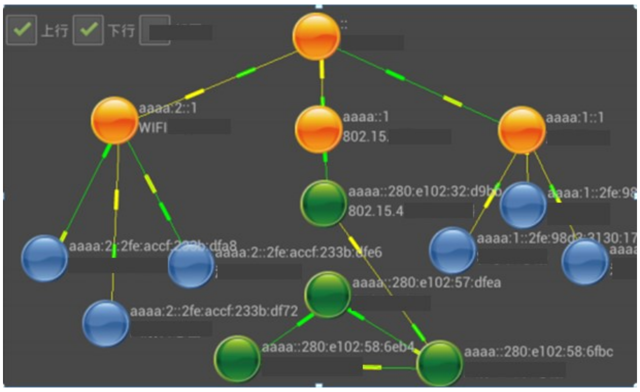
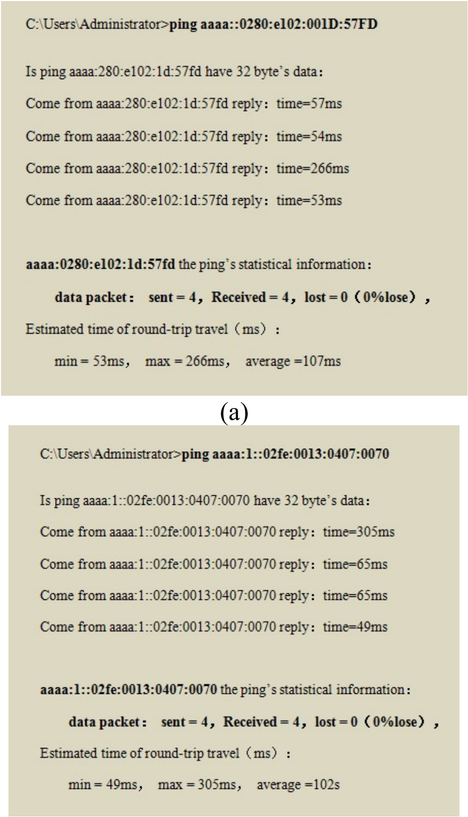


FIGURE 2. Multi-network fusion TOP diagram

The network address of the node is calculated by the method of calculating the address of the ZigBee wireless module in Section 3.1: aaa:0280: e102: 001D:57FD. The network address of the Wi-Fi node: aaaa:2::02fe: accf: 2327:49f6.



```
C:\Users\Administrator>ping aaaa::0280:e102:001D:57FD

Is ping aaaa:280:e102:1d:57fd have 32 byte’s data:
Come from aaaa:280:e102:1d:57fd reply: time=57ms
Come from aaaa:280:e102:1d:57fd reply: time=54ms
Come from aaaa:280:e102:1d:57fd reply: time=266ms
Come from aaaa:280:e102:1d:57fd reply: time=53ms


aaaa:0280:e102:1d:57fd the ping’s statistical information:
    data packet:   sent = 4, Received = 4, lost = 0 (0%lose) ,
Estimated time of round-trip travel (ms) :
    min = 53ms,   max = 266ms,   average =107ms
```

(a)

```
C:\Users\Administrator>ping aaaa:1::02fe:0013:0407:0070

Is ping aaaa:1::02fe:0013:0407:0070 have 32 byte’s data:
Come from aaaa:1::02fe:0013:0407:0070 reply: time=305ms
Come from aaaa:1::02fe:0013:0407:0070 reply: time=65ms
Come from aaaa:1::02fe:0013:0407:0070 reply: time=65ms
Come from aaaa:1::02fe:0013:0407:0070 reply: time=49ms


aaaa:1::02fe:0013:0407:0070 the ping’s statistical information:
    data packet:   sent = 4, Received = 4, lost = 0 (0%lose) ,
Estimated time of round-trip travel (ms) :
    min = 49ms,   max = 305ms,   average =102s
```

(b)

FIGURE 3. PC command line terminal display

On the PC, the ping command query results of the two IPv6 network addresses are calculated for these two. As shown in FIG. 8, the result shows that the end-to-end communication between the PC and the sensor network is successful.

Since the LCD screen is installed on the gateway, the Linux Android system is loaded on the gateway. The Android system server can establish an intuitive TOP map according to the routing packet information sent by the wireless node. After the test system is successfully connected, the resulting multi-network convergence TOP diagram is shown in Figure 9. The results show that the IPv6 gateway is successfully connected to the three wireless sensor networks ZigBee, Bluetooth, and Wi-Fi.

CONCLUSION

This paper designs and implements a multi-protocol wireless sensor network system based on IPv6 gateway, which has great application value. First of all, the IPv6 gateway unified the three different communication channels to the TUN/TAP virtual network device for processing. One module handles the tasks of multiple protocol conversions. The ZigBee, Bluetooth, and Wi-Fi communication protocols are used only for the conversion of the IPv6 protocol; The gateway is based on the 6LoWPAN protocol stack. The protocol format is naturally uniform. Sensors of different company products can support IPv6 gateway networks as long as they support the 6LoWPAN protocol stack, or transplant embedded operating systems that support the 6LoWPAN protocol stack, such as Contiki; again, based on TUN. The working characteristics of the multithreading of the TAP virtual network equipment and various wireless nodes ported by the IPv6 are communicated through the 6LoWPAN layer-level CoAP protocol and the IPv6 intelligent gateway, ensuring that the 802.15.4 subnet, the WIFI subnet, and the Bluetooth subnet are simultaneously on the IPv6 gateway. On normal communication; Finally, wireless node access

By using the CoAP protocol and the IPv6 intelligent gateway for communication, the client can directly access resources on the node.

REFERENCES

1. Kim J,Hong C,Shon T. A Lightweight NEMO Protocol to Support 6LoWPAN [J]. ETRI Joumal,2008,30 (5):685-695
2. Shin Myung-Ki,Kim Hyoung-Jun. L3 mobility supporting Large-Scale IP-Based Sensor Networks (6LoWPAN)[C]//11th International Conference on Advanced Communication Technology. 2009.New York: IEEE press,2009, 941-945
3. IETF RFC 4944.Transmission os IPv6 Packets over IEEE 802.15.4 Networks [s], IETF,2007
4. IETF RFC 6775.Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [s], IETF,2012
5. N. Patwari, J.N. Ash, S. Kyperountas, III Hero, A.O., R.L. Moses, and N.S. Corral. Locating the nodes: cooperative localization in wireless sensor networks[J]. Signal Processing Magazine, IEEE, 22(4):54 – 69, July 2005.
6. Bin Xiao, Lin Chen, Qingjun Xiao, and Minglu Li. Reliable anchor-based sensor localization in irregular areas. Mobile Computing, IEEE Transactions on, 9(1):60–72, 2010.
7. R. Stoleru, Tian He, S.S. Mathiharan, S.M. George, and J.A. Stankovic. Asymmetric event- driven node localization in wireless sensor networks[J]. Parallel and Distributed Systems, IEEE Transactions on, 23(4):634–642, 2012.
8. H. Wymeersch, J. Lien, and M.Z. Win. Cooperative localization in wireless networks[J]. Proceedings of the IEEE, 97(2):427 –450, feb. 2009.
9. Bram Dil, Stefan Dulman, and Paul Havinga. Range-based localization in mobile sensor networks[J]. In Wireless Sensor Networks, pages 164–179. Springer, 2006.
10. A. Savvides, W.L. Garber, R.L. Moses, and M.B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization[J]. Mobile Computing, IEEE Transactions on, 4(6):567 – 577, nov.-dec. 2005.
11. A. Savvides, W.L. Garber, R.L. Moses, and M.B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization[J]. Mobile Computing, IEEE Transactions on, 4(6):567 – 577, nov.-dec. 2005.
12. V. L. Mironov. Spectral dielectric properties of moist soils in the microwave band [J]. In Geoscience and Remote Sensing IEEE International Symposium, 2004.