

Design and Implementation of XMPP for Wireless Sensor Networks Based on IPv6

Ping Wang

Key Laboratory of Industrial Internet of Things and
Networked Control
Chongqing University of Posts and
Telecommunications
Chongqing, China
wangping@cqupt.edu.cn

Heng Wang

Key Laboratory of Industrial Internet of Things and
Networked Control
Chongqing University of Posts and
Telecommunications
Chongqing, China
wangheng@cqupt.edu.cn

Wenbo Wu

Key Laboratory of Industrial Internet of Things and
Networked Control
Chongqing University of Posts and
Telecommunications
Chongqing, China
wbcqupt@163.com

Abstract—XMPP (Extensible Messaging and Presence Protocol) is a widely used protocol for real-time information exchange in the Internet. In this paper, we extend it to the area of WSN (wireless sensor networks) based on IPv6 and present a lightweight XMPP client protocol software optimized for sensor nodes. Considering the resource-limited characteristic of the sensor nodes, we use UDP as the underlying transmission mechanism for XMPP instead of traditional TCP mechanism. With the purpose of decreasing the implementation overhead, we develop the XMPP client software through finite state machine method. The state design and frame parsing process of the software are presented. Finally, the proposed XMPP scheme and the client software are tested in an IPv6-based WSN platform. The experimental results show that using XMPP over UDP is feasible. It also demonstrates that XMPP can be used in the sensor networks as a unified method for information delivery among different kinds of sensors and networks.

Keywords-XMPP; WSN; UDP; IPv6; low power consumption

I. INTRODUCTION

Wireless sensor networks develop very rapidly today. In such circumstances, wireless sensor networks attract more and more people's attention. And wireless sensor networks are widely applied in many fields at present. However, because of the existing problems of wireless sensor networks, the practical large-scale application is still limited for the following reasons. Firstly, the standards of wireless sensor networks are not unified in different fields. It is not conducive to deploy wireless sensor networks in large scale. Besides, different fields have different characteristics, the interaction of data among different fields maybe difficult. It is also not conducive to deploy wireless sensor networks in large scale. The IPv6

sensor networks become a rising field recently. It introduces IPv6 technology into wireless sensor networks. The combination of these two popular technologies makes the IPv6 sensor networks a tendency.

XMPP (Extensible Messaging and Presence Protocol) is an instant messaging specification which is based on IP technology and XML. As an application layer protocol, XMPP contains the characteristics of real-time and good scalability. It is widely used in the Internet. With the IP technology and XML in wireless sensor networks going a step further, XMPP is designed to build a bridge that can easily associate the wireless sensor networks to the Internet. It also can solve the problems referred above. Besides, wireless sensor networks need to keep its protocol lightweight. So it is necessary to research and develop lightweight XMPP specification for wireless sensor networks which are based on IPv6.

II. XMPP FRAMEWORK

Our research emphasis of XMPP is on its XML stanza. As described in the RFC 6120, there are three kinds of stanzas. They respectively are presence stanza, message stanza, and I/Q stanza. The communication mechanism of presence stanza is a push mechanism. The communication mechanism of message stanza is a publish-subscribe mechanism and the communication mechanism of I/Q is a request-response mechanism. The presence stanza and message stanza are the research emphasis of this paper. Except the XML stanza, methods of XMPP are also very important. The implementation of XMPP referred in this paper includes five kinds of methods. They respectively are read method, write method, execute method, publish method, and report method. For details see attached TABLE I. However, not all these methods are realized to demonstrate the XMPP in this paper.

This finite state machine shows the XMPP sensor client how to work. All of these states are necessary for the process of authentication. Without the successfully authentication between sensor client and XMPP server, it is hard to extend the XMPP. After the successful authentication, the idle state and the state of sending data become the main states because the interaction of data between XMPP client in wireless sensor networks and XMPP client in Internet simply need the periodic skip of these two states.

B. Parsing The Received Data Module Design

The XMPP client in wireless sensor networks which is based on IPv6 technology needs to handle the response frame from the XMPP server and it also should parse the application layer strings. The XMPP client matches a specific string of the response frame and then sets corresponding state for the XMPP client. The appropriate flag needs to be set as a sign of successful reception as well. If the XMPP client matches the special string successfully, the state of finite state machine of XMPP client jumps to the corresponding state and then sends corresponding data to its XMPP server. However, at this state, the finite state machine also needs to judge whether the flag is true. If it is, the finite state machine needs to set this flag false again. The Fig .3 is the overall design for the module of processing the received data.

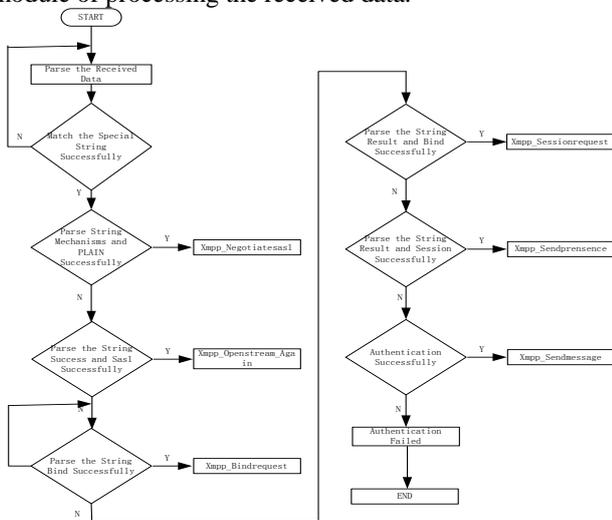


Figure 3. Parsing the received data design

In this module, it is not reasonable using nested loop to judge whether the received data right or not. That is to say, all of these judgments are parallel. The parsing module respectively sets different state of finite state machine for different response to guarantee respond the correct response to XMPP server. Besides, response data returned back is very big and there is probably the same string among different data from XMPP server. So matching the special string in the response is particularly important. When matching single string in the response does not work, we should match two or more strings in the response for more exact match.

IV. TEST RESULTS

The designing scheme of XMPP (Extensible Messaging and Presence Protocol) for wireless sensor

networks referred in this paper uses the 6LoWSN protocol stack as its development platform and re-develops the application layer protocol. To test XMPP, the test system uses CC2530 nodes as the XMPP clients in wireless sensor networks and uses source insight as the developing tool and uses the C51RF-CC2530-PK development board as the border route and uses Linux system as the running environment for XMPP server and uses pidgin software as the Internet client. The test system verified the XMPP via the successful interaction of data between Internet client and sensor client in wireless sensor networks. The test platform is shown in The Fig .4.

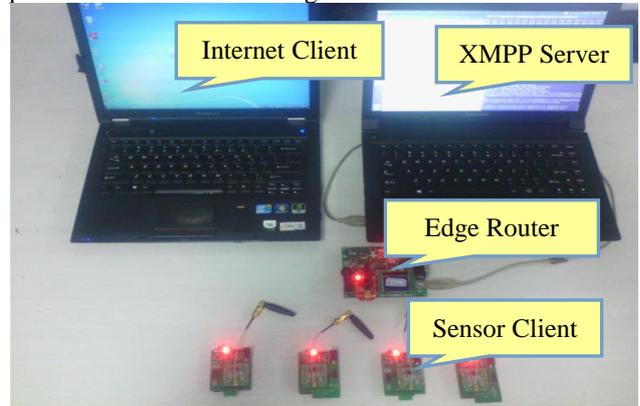


Figure 4. XMPP test platform

Pidgin client is installed in windows environment and XMPP server runs in Linux environment and sensor node plays a role as XMPP client in wireless sensor networks. Both of the pidgin client in Internet and the sensor client need the authentication to XMPP server and then they can communicate each other. The JID of the sensor clients, shown in Fig 4, respectively are humtemp@openfire, smog@openfire, dust@openfire, ch4@openfire and the JID of pidgin client is client@openfire. The XMPP clients in wireless sensor networks will send presence data to XMPP server after the successful authentication from XMPP server. The XMPP server will route the received data from sensor clients to the pidgin client in Internet to present them. Meanwhile, pidgin client can access to the sensor clients in wireless sensor networks to get the instant message of sensor clients by sending query information to sensor clients in wireless sensor networks.

XMPP server runs in the Linux environment and the process of authentication between sensor client and its XMPP server is shown in The Fig .5.

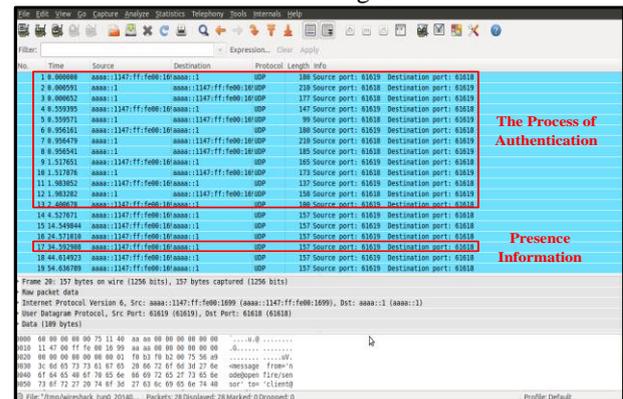


Figure 5. The process of authentication

To capture the packets, the capturing packet software needs to be used in Linux environment. It can clearly display the process of authentication between sensor client and XMPP server. This process uses UDP transport mechanism which can reduce power consumption and the amount of code greatly. It conduces to apply XMPP in the lightweight protocol stack. However, the XMPP server needs secondary development to support UDP and TCP transport mechanism at the same time. Besides, XMPP server also needs to support conversion between UDP datagram and TCP package.

The sensor client reports its information every ten seconds, the reporting information is shown in Fig .6.

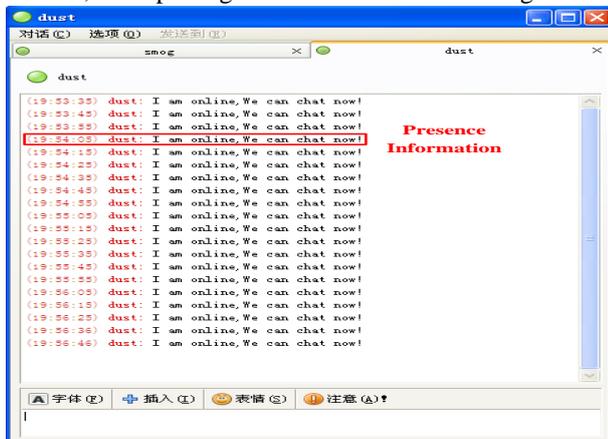


Figure 6. The presence information

To guarantee the intuition of the process, The XMPP sets sensor clients to report themselves every ten seconds. Sensor client does not report itself any more when it is not online. So it is clearly to see which sensor client is not online any more.

In the process of reporting information, Internet client can send query information to sensor client in wireless sensor networks and the sensor client can return corresponding response to pidgin client as well. But the pidgin client in Internet must send the correct query information to the sensor client in WSN. Otherwise, the processing module of sensor client will do nothing after parsing the wrong query information because the special string has been stipulated already in the module of processing data before the authentication. The Fig .7 shows the process of Information/Query.

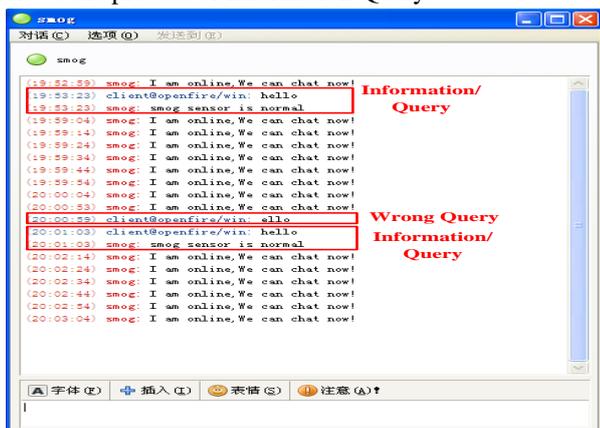


Figure 7. The process of Information/Query

This test system realized interaction of data between XMPP client in wireless sensor networks and the XMPP client in Internet. The instantaneity of communication between sensor client in wireless sensor networks and the XMPP client in Internet was clearly verified as well.

V. CONCLUSION

In this paper, a lightweight XMPP client software based on UDP was developed. The software is designed for resource-limited sensor node and tested in the IPv6 networks. The test results demonstrate the feasibility of the usage of XMPP in wireless sensor networks.

ACKNOWLEDGMENT

This work was supported by the Fundamental and Advanced Research Program of Chongqing (Grant No. cstc2013jcyjA40008), the National Natural Science Foundation of China (Grant No. 61301125), the Youth Top-notch Talent Support Program of Chongqing, and the National High Technology Research and Development Program (863 Program) (Grant No. 2012AA040602).

REFERENCES

- [1] IETF, RFC 6120, Extensible Messaging and Presence Protocol (XMPP): Core (2011)[S], Saint-Andre P, 2011.
- [2] IETF, RFC6121, Extensible messaging and presence protocol (xmpp): Instant messaging and presence[S], Saint-Andre P, 2011.
- [3] Saint-Andre P, Smith K, Tron çon R, “XMPP: the definitive guide,” O’Reilly Media, Inc, 2009.
- [4] XMPP Standards Foundation, XEP-0176, Jingle ICE-UDP Transport Method[S], Beda J, Ludwig S, Saint-Andre P, et al, 2009.
- [5] XMPP Standards Foundation, XEP-0177, Jingle Raw UDP Transport Method[S], Beda J, Saint-Andre P, Ludwig S, et al, 2009.
- [6] XMPP Standards Foundation, XEP-0278, Jingle Relay Nodes[S], Camargo T, 2011.
- [7] Ozturk O, Introduction to XMPP protocol and developing online collaboration applications using open source software and libraries[C]//Collaborative Technologies and Systems (CTS), 2010 International Symposium on. IEEE, 2010, pp.21-25.
- [8] Saint-Andre P, Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP):Media Sessions[J], 2013.
- [9] Bernstein D, Vij D, Intercloud directory and exchange protocol detail using XMPP and RDF[C]//Services (SERVICES-1), 2010 6th World Congress on. IEEE, 2010, pp.431-438.
- [10] Hornsby A, Belimpasakis P, Defee I. XMPP-based wireless sensor networks and its integration into the extended home environment[C]//Consumer Electronics, 2009. ISCE’09. IEEE 13th International Symposium on. IEEE, 2009, pp.794-797.
- [11] Ted Roden, “Building the Realtime User Experience,” O’Reilly Media, July 17, 2010.
- [12] Zhou W Q, Wang L Q, Zhou T, et al, Research and Application on Enterprise Instant Messaging System Based on XMPP[J], 2010, pp.106-111.
- [13] Weis G, Lewis A, Using XMPP for ad-hoc grid computing-an application example using parallel ant colony optimisation[C]//Parallel & Distributed Processing, IEEE International Symposium on. IEEE, 2009, pp.1-4.
- [14] Szymon Lukaszuk, Method and system of transferring electronic messages using an instant messaging protocol, US, 20140089441 A1[P], 2014-03-27.
- [15] Gomes D, Gon çalves J M, Santos R O, et al. XMPP based context management architecture[C]//GLOBECOM Workshops (GC Wkshps), 2010 IEEE. IEEE, 2010, pp.1372-1377.