

Real-Time Communication Method for mHealth Base on Extended XMPP Protocol

Chenjie Shi, Yu Fang

The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China
Email: tadelesh.shi@live.cn, fangyu@tongji.edu.cn

Received June 2013

ABSTRACT

Considering characteristic of mHealth communication and problems of existing methods, this paper presents a real-time communication method for mHealth based on extended XMPP protocol. The method can maintain the role status efficiently and reduce data latency during the communication process. Meanwhile, it can be extended flexibly to meet increasing communication demands of mHealth services. Furthermore, a system framework is presented to support telemonitoring scene. Finally, system implementation and feasibility tests verify the effectiveness of the method and framework.

Keywords: mHealth; XMPP Protocol; Real-Time Communication

1. Introduction

With the increasing proportion of urban sub-health and aging population, medical resource for diagnosis and treatment of chronic diseases are in high demand. Taking advantage of sensor technology and mobile communication technology, mHealth provides various medical information and services which improve utilization of medical resources efficiently and reduce costs as well. With the development of urban wireless communication infrastructures as well as the popularity of mobile terminals, mHealth has become a new trend of medical informatization [1].

Existing communication methods of mHealth use custom packages to transport data and produce service responses through socket or http request. All these methods are lack of unified application layer protocol and cannot maintain the role status during communication. Moreover, custom binary packages make method extension uneasy. In order to solve these problems, a real-time communication method for mHealth based on extended XMPP protocol is presented. With XMPP protocol, functions of connection maintenance, status tracking and data transfer process of mHealth services are implemented. By extending “iq” and “message” package of XMPP protocol, the demands for different types of data transfers and service requests and responses are met. A system framework using proposed method is presented to support mHealth service of telemonitoring scene and concrete system implementation with tests validates the feasibility of the method and framework.

2. mHealth Communication and Existing Problems

The typical mHealth real-time communication process is shown in **Figure 1**. User wearing physiological signal sensors (ECG, SPO2, NIBP sensors, etc.) holds mobile terminal to collect and transmit signal data and user data. mHealth processing with service results are exhibited in terminals by communicating with mHealth providers via server. Different mHealth providers use clients to provide concrete mHealth services (telemonitoring, online diagnosis, health advisory, etc.) on signal data and service requests received from the server [2,3].

The foundation of mHealth service processes is real-time data transmission of various service requests and responses. The transmission has three main features [4,5]. First, communication roles, users and providers, stand at heterogeneous platforms and network environments. Communication should adapt to different terminal platforms and diverse networks. Second, diversion and expanding of mHealth services need communication pro-

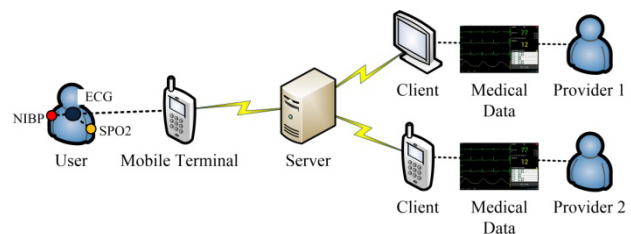


Figure 1. mHealth communication process.

protocol to adapt to changing requirements in data encapsulation. Third, some mHealth services such as telemonitoring require low-latency transmission.

The existing real-time communication methods for mHealth use socket or http to maintain data link as well as perform data transmission using custom binary data packages [6,7]. These methods have two aspects of problems. First, due to stateless transmission which most methods are using, they cannot handle service status renewal on breakout chain and deal with latency of service requests and responses. Second, lacking of unified application layer protocol and using binary format in data package make methods not suitable for expanding transmission demand of mHealth services.

3. Communication Method

3.1. XMPP Protocol Basis

XMPP protocol is an XML based instant messaging protocol, which uses XML based structured information during entire communication process, having good semantic integrity and flexible scalability. SASL and TLS built into the core XMPP specifications guarantees the security of the data transmission [8,9].

XMPP protocol use TCP persistent connection to build XML stream flow which carries “message”, “iq” and “presence” packages to composite instant communication process. This paper uses and extends these basic mechanisms to implement connection maintenance, status tracking and data transfer process of mHealth service. Based on XEP-0045 standard of XMPP [10], proposed method constructs the session room to organize the transmission process and trace the status between user and providers. Data format for mHealth service can be scalable defined by extending “message” and “iq” packages.

3.2. Establishment and Destruction of Communication

mHealth services typically involve several roles as user and multiple service providers. Using session room mechanism of XMPP, the method organizes all roles into a session room initiated by user. With session room, server can track the status of all roles, forward data packages during service process and isolate user’s medical data from external. The communication process is shown in **Figure 2**.

1) User connects to XMPP server and finishes identity authentication. Session room creation request will be sent after connection established.

2) Server receives session room creation request and create the room with identity “username@service/name”. Then server query providers bounded to user and send invitation request to them.

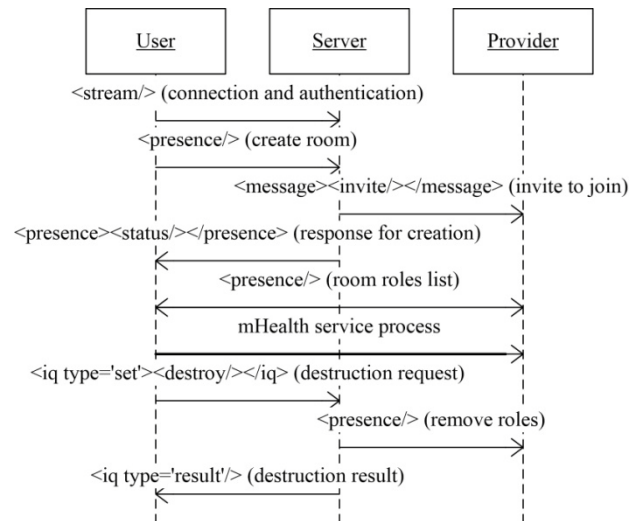


Figure 2. Establishment and destruction of communication.

3) Providers receive invitation request and join the session room. User and providers confirm all the roles in session room by exchanging “presence” packages and meanwhile start the service process.

4) When user decides to terminate process, “iq” package will be sent to server for connection destruction.

5) Server receives destruction request and remove all roles in session room. Then server destroys the session room and notifies user by sending “iq” package.

3.3. Information Data Communication

Information data of mHealth services refers to non-interactive data passing between user and providers. This type of data is mainly designed for information delivery during the service process, including user info data, physiological signal data, diagnostic data, advisory information, etc.

The data is packaged as extended “message” packages according to the data type and sent to server. Server will forward it to corresponding roles of session room. User or providers receive “message” packages and restore the information to finish the service process.

With the scalability of XMPP protocol, various data formats in the mHealth services can be defined. Physiological signal data for instance involves waveform data like ECG and numerical data like NIBP. A child element “monitordata” is added to “message” package to define the type of data. Specific data can be encapsulated in XML semantic way or in-band binary way. The XML way distinguishes types of sensors by “ecg” “nibp” element and record data by body of elements. The In-band way encapsulates sensor’s byte array in Base64 encoding packages and attaches them to “monitordata” element. An instance of ECG monitoring data packaged in XML way is showed in **Figure 3**. The data includes 3 path

```

<message>
<monitordata type="sensordata">
<ecg1>135,137,139,141,143</ecg1>
<ecg2>135,137,139,141,143</ecg2>
<ecg3>135,137,139,141,143</ecg3>
<resp>185,187,189,195,198</resp>
<hr>68</hr>
<rr>15</rr>
<timestamp>1365324293</timestamp>
</monitordata>
</message>
    
```

Figure 3. ECG monitoring data package.

ECG wave, 1 path RESP wave, HR and RR value. An instance of 6-parameter monitoring data packaged in in-band way is shown in Figure 4. The data includes ECG, RESP, SPO2, PR, NIBP and TEMP signals.

3.4. Request Data Communication

Request data in mHealth services refers to querying and setting data between user and providers. This data is mainly for requests and responses during the service process. Using “iq” request-response mechanism of XMPP, the method encapsulates this type of data by adding a new class of “iq” package.

Taking sensor setting for instance, providers require to set user-sensor parameters (ECG lead, RESP gain, etc.) to meet different diagnosis requirements. The communication process is shown in Figure 5.

- 1) Provider sends “iq” package with querysensor/ set-sensor element to query/set user’s particular sensor’s parameter.
- 2) Server receives and recognizes the request by “iq” package’s querysensor/setsensor element and only forward it to user.
- 3) User receives “iq” package and query/set particular sensor, then sends back the parameter value or setting result to the provider through “iq” result package.

4. System Framework

With proposed extended XMPP protocol method, a mHealth system framework is designed to support mHealth service of telemonitoring scene, in which user wearing physiological signal sensor use mobile terminal to transmit real-time signal data while remote doctor uses client system to monitor user condition and give medical suggestions [11]. The framework is divided into two parts: mobile terminal and client.

4.1. Mobile Terminal

Mobile terminal part use Smack library as basis of XMPP communication. The upper level is divided into four layers (Figure 6): protocol extension layer which

```

<message>
<monitordata type="sensordata">
Y2BeXmBka...psbnBxcA=////////////////////////////////////
////////////////////////////////////w==
</monitordata>
<properties><property>
<name>timestamp</name>
<value type="long">1371306851897</value>
</property></properties>
</message>
    
```

Figure 4. 6-parameter monitoring data package.

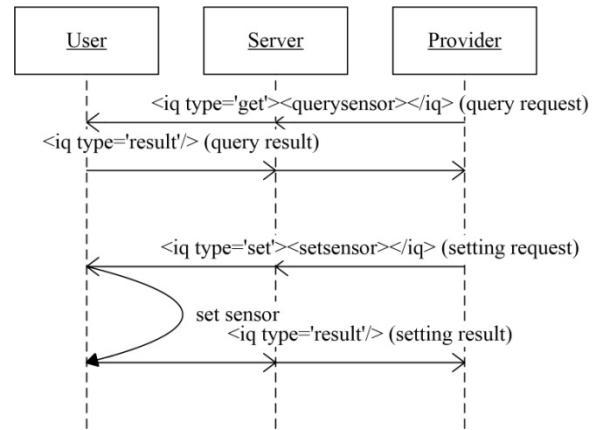


Figure 5. Sensor query/setting communication.

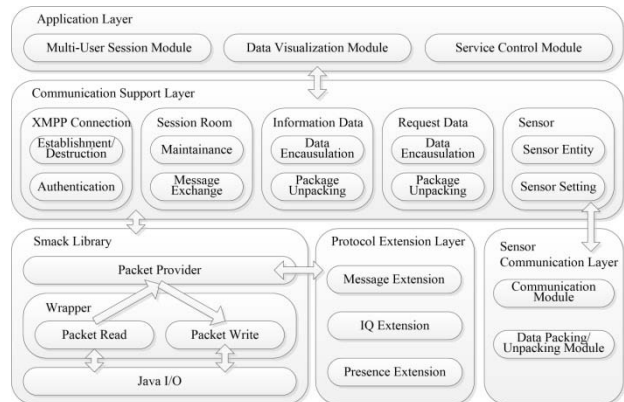


Figure 6. Mobile terminal framework.

handles XML streams of extended XMPP protocol, communication support layer which coordinates underlying communication processes of various applications, sensor communication layer which interacts with physiological signal sensor and application layer which organizes service process of telemonitoring.

Protocol extension layer use packet provider mechanism supported by Smack library to extend XMPP protocol. XMPP packets’ processing of Smack library is divided into three major parts. First, data packages are retrieved or sent from/to socket buffer by Java I/O. Second, wrapper module implements package read/write processes

by observer pattern. Third, provider module uses programmed package extension to unpack “iq”, “message” and “presence” packages. Protocol extension layer designed by this paper programs the package extension mentioned in the communication method via provider module which can handle the information data from “message” packages and request data from “iq” packages.

Communication support layer provides upper application layer with required XMPP communication interfaces. This paper uses singleton pattern to encapsulate five objects for interfaces of different communication processes. XMPP connection object provides interfaces for connection establishment, authentication and connection destruction between user and server. Session room object provides interfaces of creation, setting and destruction of session room, and processes room message exchange. Sensor setting object encapsulates parameters of all sorts of sensors and provides interfaces for querying and setting user’s sensors. Information data object and request data object encapsulate entities with data properties and their operations, and provides interfaces for packing and unpacking “message” and “iq” packages.

Sensor communication layer supports sensor operations through data transfer between mobile terminal and physiological signal sensors.

Application layer is the direct interaction layer for user to use telemonitoring service while feature interaction process is encapsulated into several APIs. Application layer includes three main modules. User use multi-user session module to launch telemonitoring service. Data visualization module and service control module couple communication support layer and application layer to handle data transfer, process user interface display and organize telemonitoring service process.

4.2. Client

Client part implements mHealth service visualization and operation interface for providers. This paper designs a telemonitoring extension plugin for Spark (Figure 7), an open source IM client based on XMPP protocol which can be easily extended by flexible plugin structure, to implement client part. The plugin extends the functionality of multi-user session room window of Spark through three portions. First, package unpacking/packing module handles extended XMPP packages during telemonitoring process through filtering “message” and “iq” packages sent to session room. Second, interface extension module adds physiological signal data visualization function to session room window via Java Swing graphic API to visualize waveform data and numeric data. Third, telemonitoring service module is responsible for coordinating other modules to organize logic flow of telemonitoring service process, including data transfer adaption, user interface display and user action response.

5. Tests and Analysis

A concrete mHealth service system is implemented in this paper to validate the feasibility of communication methods and system framework as illustrated above. The system environment is as follows, Android 4.1.0 operation system used by mobile terminals, service system based on ASmack 0.8.1.1, the monitor powered by Berry Bluetooth 6-parameter monitoring sensor as physiological signal input, Dell Power Edge T110 servers with Openfire 3.8.1, clients acted by PCs with Spark 2.6.3. All the servers are publicly available, mobile terminal and client get access to the public network through local network. Network configuration is shown in Table 1.

5.1. Functional Test

Mobile terminal system runs as shown in Figure 8. System gets physiological sensor data, encapsulates using XMPP protocol and sends them out to server. Client works as shown in Figure 9, system parses received data and provides appropriate services according to the data types.

Control communicated XML packages are captured by taking advantage of debug mode. Figure 10 shows the XML segments of telemonitoring session room creation process which includes “iq” packages for connection

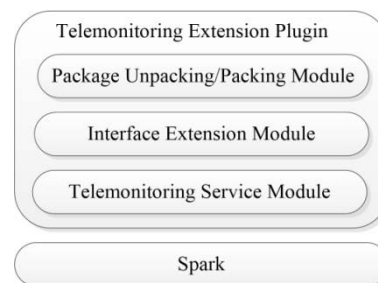


Figure 7. Client framework.

Table 1. Test system network environment.

Device	IP
Mobile Terminal	192.168.0.3
Server	202.120.189.151
PC Client	11.0.17.21



Figure 8. Running status of mobile terminal.

establishment and “presence” packages for status communication. **Figure 11** shows the XML segments when monitoring which contains “message” packages for information data.

5.2. Communication Latency Test

Timestamp is added to the extended protocol in order to test the latency of real-time communication. Latency status of the extended XMPP protocol under different data sizes can be measured by comparing the receiving

time of mobile terminal or client with timestamps. Test results shown in **Table 2** indicate the protocol is able to deal with latencies because of the persistent connection feature of XMPP protocol.

6. Conclusion

This paper analyzes the characteristic of real-time mHealth communication and existing problems, proposing an extended XMPP protocol that can be used in real-time mobile medical communication. This approach implements



Figure 9. Running status of client.

Status	Data
02:22:43 PM SENT	<presence id="e36g9-4" to="patient1@conference.202.120.189.151/patient1@192.168.0.3/Smack"><x xmlns="http://jabber.org/protocol/muc"></x></presence>
02:22:43 PM RCV	presence id="e36g9-4" to="patient1@192.168.0.3/Smack" from="patient1@conference.202.120.189.151/patient1@192.168.0.3/Smack"><x xmlns="http://jabber.org/protocol/muc#user"><item jid="patient1@192.168.0.3/Smack" affiliation="owner" role="moderator"/><status code="201"/></x></presence>
02:22:43 PM SENT	<iq id="e36g9-5" to="patient1@conference.202.120.189.151" type="set"><query xmlns="http://jabber.org/protocol/muc#owner"><x xmlns="jabber:x:data" type="submit"></x></query></iq>
02:22:43 PM RCV	<message type="groupchat" from="patient1@conference.202.120.189.151" to="patient1@192.168.0.3/Smack"><body>This room is locked from entry until configuration is confirmed.</body></message>
02:22:43 PM RCV	<message type="groupchat" from="patient1@conference.202.120.189.151" to="patient1@192.168.0.3/Smack"><body>This room is now unlocked.</body></message>
02:22:43 PM RCV	<iq type="result" id="e36g9-5" from="patient1@conference.202.120.189.151" to="patient1@192.168.0.3/Smack"/>
02:31:16 PM RCV	<presence id="kMWA8-41" to="patient1@192.168.0.3/Smack" from="patient1@conference.202.120.189.151/doctor1"><x xmlns="http://jabber.org/protocol/muc#user"><item jid="doctor1@11.0.17.21/Spark.2.6.3" affiliation="none" role="participant"/></x></presence>

Figure 10. XML segments of session room creation process.

Status	Data
02:34:12 PM SENT	<message id="e36g9-6" to="patient1@conference.202.120.189.151" type="groupchat"><body>data</body><monitordata xmlns="xmpphealth/monitordata" type="sensordata">Y2BeXm...mpsbxBxcXA=//////////////////////////////////////w==</monitordata><properties xmlns="http://www.jivesoftware.com/xmlns/xmpp/properties"><property><name>timestamp</name><value type="long">1371306851897</value></property></properties></message>

Figure 11. XML segments of motoring data.

Table 2. Latency test result.

Direction	Data Size (byte)	Average Latency (ms/100 packages)
Terminal	100	130.29
→	500	131.50
Client	1000	142.41
Client	100	128.54
→	500	136.28
Terminal	1000	138.54
Total Average Latency		134.59

a universal extensible real-time mHealth communication application layer protocol by using and extending the connection maintenance, status tracking and data communication feature of XMPP. Based on that, this paper presents a system framework that supports telemonitoring scene of mHealth service which is validated by implementation and tests. In the future, the method should be improved in efficiency based on quantitative analysis and data compressing.

7. Acknowledgements

This work was financially supported by National Natural Science Foundation of China Youth Foundation (61003222).

REFERENCES

- [1] M. Lin and Z. Z. Qiao, "Thinking of Mobile Medical Demands and Development," *Mobile Communications*, Vol. 23, No. 6, 2010, pp. 31-35.
- [2] D. K. Jang, J. M. Kim, S. Sohn and K. R. Han, "Development of a Mobile e-Health Care System for Rapid Detection of Emergent Situations," *International Conference on New Trends in Information Science and Service Science*, Macao, 24-26 October 2011, pp. 93-96.
- [3] S. Adibi, "Link Technologies and BlackBerry Mobile Health (mHealth) Solutions: A Review," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 16, No. 4, 2012, pp. 586-697.
<http://dx.doi.org/10.1109/TITB.2012.2191295>
- [4] E. C. Kyriacou, C. S. Pattichis and M. S. Pattichis, "An Overview of Recent Health Care Support Systems for eEmergency and mHealth Applications," *Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, Minnesota, 2-6 September 2009, pp. 1246-1249.
- [5] P. Zhou, H. Y. Jiao, F. Wang and C. D. Wang, "Design of Portable Wireless ECG Monitoring Terminals," *Computer Engineering and Design*, Vol. 33, No. 9, 2012, pp. 3407-3410.
- [6] L. H. Zhong, C. L. Li and W. D. Zhou, "Transmission and Processing of Home Telemonitoring Data," *Journal of Biomedical Engineering Research*, Vol. 22, No. 4, 2003, pp. 40-42.
- [7] M. A. Chowdhury, W. Mciver and J. Light, "Data Association in Remote Health Monitoring Systems," *IEEE Communications Magazine*, Vol. 50, No. 6, 2012, pp. 144-149.
<http://dx.doi.org/10.1109/MCOM.2012.6211499>
- [8] IETF, "RFC 6120 Extensible Messaging and Presence Protocol (XMPP): Core," Standard, 2011.
- [9] IETF, "RFC 6121 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," Standard, 2011.
- [10] XMPP Standards Foundation, "XEP-0045 Multi-User Chat," Standard, 2011.
- [11] W. J. Jia, Y. P. Bai, X. J. Sheng and S. C. Wu, "Development of Physiological Multi-Parameters Telemonitoring System," *Journal Press of China Medical Devices*, Vol. 24, No. 8, 2009, pp. 76-78.