

Investigation on the data transfer protocol for ONEChat

Author: Heming Cui

Advisor: Professor Henning and Mr. Suman

1 Abstract

Since mDNS is not suitable for data transfer, I am looking for a real data transfer protocol to support both multicast and real-time text transfer for ONEChat. I have investigated four protocols, RTP, SRTP, Jabber XMPP, and Real-time Text. I have run all of them and find that the Real-Time Text mostly matches our requirement for two reasons. First, the Real-Time Text is built on top of RTP, which supports both multicast and real-time text transfer, so it matches the requirement of our group chatting application. Second, the RTP library used in Real-Time Text is implemented in Sun's JMF, which is a professional and reliable framework, while the other RTP library I am investigating has some bugs. We need a reliable RTP library to develop some audio/video sharing features as well as text transfer in my research project at Spring 09.

2 Introduction

In this section, I will give a brief introduction to the four data transfer protocols I have investigated. Since different libraries for the same protocol is similar, we just introduce one library for one protocol. I have investigated several libraries for each protocol, and the libraries we are introducing below are the ones mostly match our requirements.

RTP (Real-time Transfer Protocol, published as RFC 3550) is the Internet-standard protocol for the transport of real-time data, including audio and video. It can be used for media-on-demand as well as interactive services such as Internet telephony. RTP consists of a data and a control part. The latter is called RTCP. RTP sessions are built on top of UDP. RTP was originally designed as a multicast protocol, but has since been applied in many unicast applications.

I have investigated the jlibrtp library [1] (before I investigated the Real-Time Text, I did not know that JMF has implemented RTP, so I investigated jlibrtp), which is an RTP implementation developed by Arne Kepp with Java. The jlibrtp library is one of our IRT student projects. I have written some simple code in Java to test multicast text delivery with this library. It works after I fix three bugs in the library. These bugs are all involved with multicast. I have sent an email to Arne Kepp to make sure.

SRTP (Secure Real-time Transfer Protocol, published as RFC 3711) defines a profile of RTP, intended to provide encryption, message authentication and integrity, and replay protection to the RTP data in both unicast and multicast applications. As far as I know, there is no open source SRTP library written in Java. There is an open source SRTP library named libsrtp [2] written in C, which is lead by Cisco, but it only works in Posix-style operating systems. I have run this library and make sure it works after I fix one bug (comparing an unsigned integer with -1). If we decide to use this library, we may need to build a bridge (I would like to use sockets) to connect our Java application and this library within the same machine.

Jabber XMPP (eXtensible Messaging and Presence Protocol) is an open, XML-inspired protocol originally aimed at near-real-time, extensible instant messaging (IM) and presence information (buddy lists), but now expanded into the broader realm of message oriented middleware. It remains the core protocol of the Jabber Instant Messaging and Presence technology. Built to be extensible, the protocol has accumulated features over time such as Voice over IP and file transfer signaling. Jabber streams are built on top of TCP. It does not support multicast.

I have investigated a Jabber XMPP library called Smack [3]. Although Smack contains fluent APIs aiming to support group chatting, messages are transferred via unicast communication. Jabber needs a server to forward messages. For example, in a chat group, a client sends a message to the server, and the server forwards the message to every other group member one by one. A Jabber XMPP server is very complicated, so we could only use an open source Jabber server rather than writing a server by ourselves. For each application written with the Smack library, there is a corresponding open source server called Openfire [3]. I have written some Java code to make sure the Smack library works. Since Jabber XMPP does not support multicast, it does not match our requirement.

Real-time text is conversational text that is sent and received on a character-by-character basis. The characters are sent immediately once typed and also displayed immediately to the receiving person or people. This allows text to be used in the same conversational mode as voice. Real-time Text is defined in ITU-T Multimedia Recommendation F.700 2.1.2.1 and built on top of RTP. I have investigated a Real-time Text library called T140 [4]. This library uses Sun's JMF (Java Multimedia Framework) as the RTP library. I have written some Java code to know how the T140 library works.

I find the Real-Time Text mostly matches our requirements for these reasons. First, the Real-Time Text is built on top of RTP, and it supports multicast in order to save bandwidth. Second, the RTP library used by the Real-Time Text is a new feature in Sun's JMF 2.1.1, which is a professional and reliable framework, while the other RTP library named jlibrtp has some bugs. We need a reliable RTP library to develop some audio/video sharing features as well as text transfer in my research project at Spring 09. Below is the comparison result of the four protocols.

3 Comparisons (in tables)

The comparison result is given in the table below. The red color means bad.

Investigated Protocol	Investigated library	Language	Platform	Multicast support	Easy to use	Security	Other Concerns
RTP	jlibrtp	Java	Cross-platform	Yes	Yes	No	Three bugs were found.
SRTP	Only one library named libsrtp written in C is found	C	Only works in a Posix-style OS	Yes	No, we need a socket between the C code and Java.	Yes	One bug was found.
Jabber XMPP	Smack/Openfire	Java	Cross-platform	No	XML, easy to use	Yes	
Real-time Text	T140	Java	Cross-platform	Yes	Yes	Not sure. I need more time to check the code.	

4 Bugs found in the libraries

I found four bugs in the libraries while investigating, three for jlibrtp, and one for libsrtp. The jlibrtp library is one of our IRT student projects, and I have sent an email to the author Arne

Kepp to make sure, but no reply yet. For libsrtplib, I have also sent an email to the author (a Cisco employee), also no reply yet.

5 Conclusion

The Real-Time Text mostly matches our requirement. I would like to use it as our data transfer protocol. Since the RTP library in JMF does not provide any security guarantee, we are going to add encryption and decryption mechanisms by ourselves.

6 Links to the libraries

- [1] For RTP, I investigated jlibrtp: <http://jlibrtp.org/>
- [2] For SRTP, I investigated libsrtplib: <http://sourceforge.net/projects/srtplib/>
- [3] For Jabber, I investigated Smack: <http://www.igniterealtime.org/projects/smack/index.jsp>
- [4] For Real-time Text, I investigated T140: <http://sourceforge.net/projects/rtp-text-t140/>