Improving Quality in Voice Over Internet Protocol (VOIP) on Mobile Devices in Pervasive Environment

Mr. S.Thiruppathi
Assistant Professor,
Department of Software Engineering,
Periyar Maniammai University, Vallam, Thanjavur,
E-mail: thistsp83@gmail.com

Abstract - Today the customer care in telecommunication using the Voice over Internet Protocol (VOIP) is a way to carry out telephone conservation over a data network for free. VoIP products promise converged telecommunications data services that are cheaper, more versatile and provide good voice quality as compared to traditional offerings. Although VoIP is widely used, VoIP on mobile devices is still in infancy. Currently there are a number of VoIP solutions for mobile phones. However, VoIP solutions developed using Java 2 platform Micro Edition (J2ME) are not available .Java based solutions are widely compatible with many devices. In this paper, strong focus has been granted to devices compatibility through the use of the widely supported J2ME framework for the people using environment. The implementation details of VoIP client using J2ME are illustrated.

Keywords: *VoIP, RTP, SIP, J2ME, GPRS, Asterisk, PBX, UDP, Pervasive Computing*

I. INTRODUCTION

To date, the public Switched Telephone Network (PSTN) has been used to conduct telephone calls over a wired network. With the development of computing technology, voice over Internet Protocol (VoIP) has been established as an alternative to traditional telephony networks. VoIP allows telephone conservations to take place over a data packet-switched network like the Internet.

VoIP products promise converged communications and services that are cheaper, more versatile and provide improved voice quality as compared to traditional offerings. Although VoIP is widely used, VoIP on mobile devices is still in its infancy.

The second section of this paper details the VoIP on mobile devices solution. Section three provides a proposed solution and architectural overview**.** Section four details the system implementation. Section five demonstrates and evaluates the implementation outcomes. The last section concludes this paper and recommends some future work.

II. BACKGROUND AND RELATED WORK

VoIP is the digitalizing of voice using an analog to digital converter (ADC), sending this data through a data network and the reassembling of this data to form the original analog format using a digital to analog converter (DAC). VoIP is made of two parts, signaling and data transport. The VoIP signaling function can be performed using protocols such as SP[1], H.323 [3] and MGCP [4]. Data transport can be performed by The Real time Transport Protocol (RTP) [2]. This protocol is used to deliver voice data during conversation.

A. Session Initiation Protocol (SIP)

The Session Initiation Protocol[l](SIP) is an application-layer control signaling protocol. SIP is used to create, modify, and terminate multimedia sessions or conferences such as Internet Telephone calls. The SIP message format is similar to the Hyper Text Transfer Protocol (HTTP) message format[l].

Two main components in SIP are user agent (UA) and Servers.

- UAs: are regarded as a client that can send the request and response together. This includes a user agent client and a user agent server.
- Servers: are used to receive requests from clients for servicing and sending responses back to the client.

B. Real-time Transport Protocol (RTP)

- RTP: is used to carry voice data
- RTCP: is used to monitor the quality of services and information about the participants who are in a sessions

C. Voice Codecs

In VoIP, there are many different audio codecs. The bandwidth required during a VoIP conversation naturally depends on the codec. Table I describes the audio codec supported in VoIP. The G.711 codec is used on most telephony systems all over the world. The G.729 codec provides the best voice quality. However, due to the native support of G.711 by mobile devices, it is more suitable to use G.7 11.

Table 1: Audio Codecs Supported in VOIP [5]

Codec	Sampling Rate	Bandwidth	Payload size
G.711	8KHz	64Kbps	20ms
G.722	16KHz	48,48,64Kbps	30ms
G.723.1	8KHz	5.3,6.3Kbps	30ms
G.726	8KHz	24,42,40Kbps	20ms
G.728	Unknown	16Kbps	
G.729	8KHz	8Kbps	20ms
GSM	8KHz	13Kbps	
	Unknown unknown	13.33Kbps 15Kbps	30ms 20ms

Table 2: Comparison of Mobile Platform Development

	Symbian	J2ME	BREW
Foundation	C++	Java	C++
Learning Curve	Difficult	Average	Difficult
Emulator	Free	Free	Limited
Debuggers Available	Good on Latest version	Excellent	Need Payment
Development Tool cost	Varies(free Tools Available)	Free	Expensive
Cross-platform Deployment	Compile Per target	Average	CDMA Handsets only
Developer community and support	Extensive	Extensive	Limited
Market penetration	Extensive	Extensive	In few countries
SIP/RTP support	Yes/Yes but complicated	Yes/No	Unknown

D. Mobile Application Development Environment

There are a number of integrated development environments, languages, frameworks and libraries that can be used to develop the solution proposed in this paper. Table II details the differences between the most popular mobile development environments. J2ME is chosen as the development tool in this paper. Although RTP is not supported on J2ME, implementation of RTP on J2ME framework is one of key features of this paper.

E. Existing Related Work

Although there are no VoIP solutions developed for J2ME enabled mobile devices, some solutions exist:

1) Burdet el. al. has developed a VoIP solution for mobile device using the Symbian operating system through Bluetooth [6]. SIP, SDP and RTP features operate on a Symbian emulator. The work of Symbian on mobile phone shows that VoIP can be performed over a Symbian Mobile Phone. The proposal illustrated in this work is different to the ideas illustrated in this paper.

2) Vikram Goyal experimented with streaming content data using JAVA ME [7]. An existing audio file on Darwin server is requested by a client through RTP protocol. A custom data source class, source stream class and RTSP class are created to handle the incoming RTP packets. However, this work does not follow RTP standard. It also does not mention how to deliver and play voice data.

III. Proposed Architectural Framework

In this proposed method this can be used in the any type of generation. Now this could be implemented as follows.

A. My Proposed SIP Framework

SIP performs signaling functions in a session. According to the SIP specification[l], the sequence for making a SIP call is illustrated in Figure 1. In this paper, SIP functions are implemented based on the sequence in Figure 1.

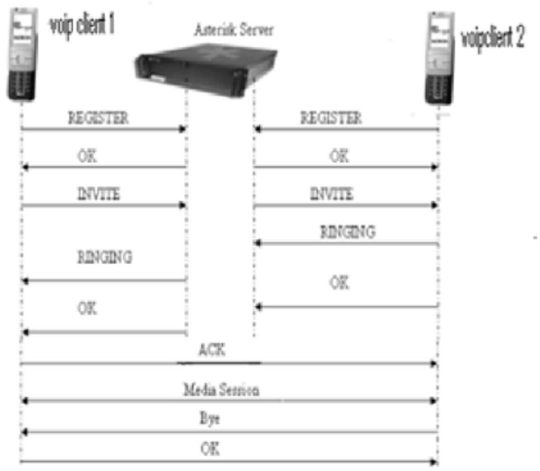

Figure1: VOIP Client SIP Message Sequence

As signaling functions are provided, SDP is used to negotiate media session description between clients. The media session description includes encoding format, payload type and sampling rate of voice packets. Also, IP address and port number for receiving voice packets is included.

It can run on devices with limited memory. There are six SIP request methods are explained as following in SIP specification[l]: REGISTER, INVITE, ACK, CANCEL, BYE, OPTIONS. Two main requests are introduced in the following:

1.) **Registration:** registration is the first essential part for starting a session. A SIP[l] registration message is generated and sent to Asterisk server. According to SIP specification, the transactions between client and asterisk server.

2.) **Invitation:** The invitation is the second step that a VoIP call needs to do after registration. According to SIP API, SIP Invitation Message is generated.

Figure1 shows VOIP client sends registration request to asterisk sever. The Registration request consists of IP address, contact number and other details. Asterisk server gets the request and responds to VOIP client.

VOIP client 1 invite to VOIP client 2, client 2 call try to client 1 that status code is maintained 100 and leads to status code maintained 180.client 1 attend the call means that status code maintained 200 ok. Client 1 acknowledge to client 2, after call should be finished it say bye.

IV. IMPLEMENTATION USING ASTERIX SERVER

Asterisk is an open source Private Branch Exchange (PBX) that gives you real-time connectivity for both PSTN and VOIP networks. It also functions as a SIP proxy.

V. EXPERIMENTAL RESULT

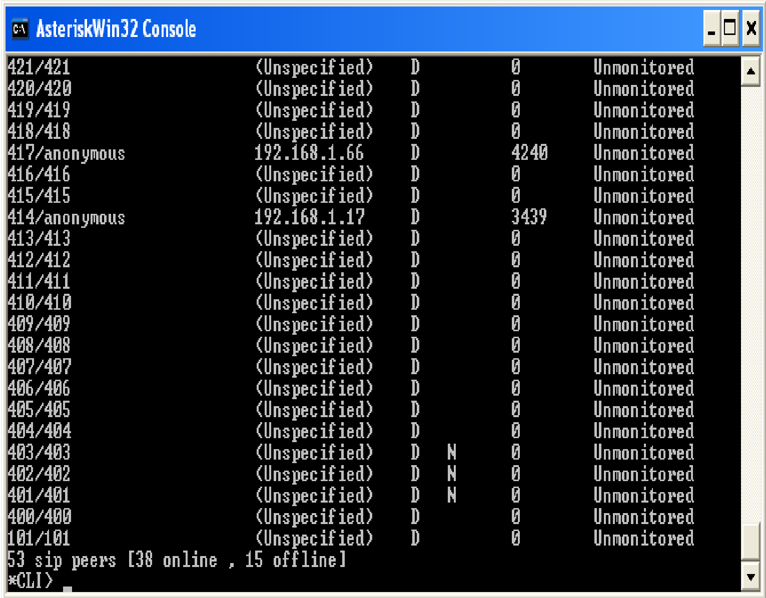

Figure2:Voice Message Communication Module

Fig2. Shows Mobile clients validated, call initiated through sip through mobile server. Transmitter port, transmitters' local, receiver port, receiver local is for full duplex communication. Both are communicating using port.

VI. CONCLUSION

The above result as shown in fig 2.the implemented client was demonstrated. VOIP client can be developed using J2ME followed the standard and deploy on a mobile phone with the necessary features. The features of the implemented client are suitable for mobile devices.
Although the implemented client is compatible with the VOIP standard, the client is not implemented completely. There feature work and implementing RTCP in RTP and implementing duplex conversation.

REFERENCES

1. Guo Fang Mao, Alex Tale ski, Elizabeth chang, VOIP on mobile devices, April 2007.
2. Rosenberg, H. Schulzrinne, G. Camarillo, A.Jonshon, R. Spark, M. Handley, E. Schooler. SIP: session Initiation Protocol. The Internet Engineering Task Force, June 2002.
3. H.Schulzrine, S. Casner, R. Frederick, V. Jacobson. RTP: A Transport Protocol for, July Real-Time Application. Network Working Group 2003.
4. W.Muller, H.Glasmann, J.Kellerer.service Development and Deployment in H.23 and SIP. IEEE, 2001.
5. F.Andreasen, B. Foster. Media Gateway Control Version 1.0, pages 5-33. Network Working Group, January 2003.
6. Oz VoIP. VOIP Codecs. Viewed 18 April 2006 [ONLINE] http:// compare. Ozvoip. com/codecs.php.
7. L.Burdet, P.Stuedi, A. Frei and G. Alonso. Symphone: Design and implementation of a VOIP peer for Symbian Mobile Phones using Bluetooth and SIP. The ACM Digital library 2006.
8. V.Goyal. Experiments in Streaming Content in Java ME. Sun Microsystems, viewed 06 September 2006 [ONLINE] http://today.java.net/pub/a/today/2006/08/22/ Experiments-in-streaming-java-me.html?Page=1, 2006.
9. M.Handley, Jacobson. SDP: Session Description Protocol. Network Group, 1998. Inc Sun Microsystem. Porting Guide Java 2 platform, Micro Edition. Technical Report: Mobile Media API, June 2002.