Investigation, Development and Test of Video over IP Applications for IMS/SIP Clients

R.Bonhomme, L.Sun and I.Mkwawa

Centre for Security, Communications and Network Research,
Plymouth University, Plymouth, UK
e-mail: info@cscan.org

Abstract

The main purpose of this paper is to investigate and develop Video over IP solutions regarding the current technologies. The objectives are to understand how a mobile device or client can communicate with multimedia servers and others devices and to define the differences with traditional telephony server on the one hand. To investigate and to understand how video calls and real time Quality of Service can be implemented on these clients using these servers on the other hand. A VoIP testbed was set up which includes open IMS core, Asterisk server, Android G1 phone and X-lite VoIP client to facilitate this task. Video calls between G1 phone and X-lite had been enabled using Asterisk server. These calls had been monitored: calculation of delay, jitter and packet loss and a Quality of Experience metric had been implemented to provide information on the call quality.

Keywords

Video over IP, SIP, Quality of Service

1 Introduction

"Users are expecting new multimedia applications that make use of the full capabilities of their devices." This statement from a research on a mobile multimedia framework (Albaladejo et al, 2008) is a way of introducing this thesis. Today, there are a lot of technologies around us such as 3G, UMTS or IP. Information is everywhere and users are in the middle (Adamek et al, 2002 and Long, 2001). New services appeared these last few years: voice and video call, conferences, video messaging and operators have to improve and modernize their systems and their infrastructures. All these services have a direct cost and users pay it. However open sources solutions can be found such as Open IMS for 3G technology or Asterisk for WI-FI: Internet Protocols allow users to break free from their operators, to communicate between themselves.

2 IMS and Asterisk server

2.1 IMS

Like defined in 3GPP TS 23.228 (3GPP, 2009b), IP Multimedia Subsystem also called as IMS architecture has the capability to provide multimedia services such as voice, video and data for wireless users. The first contact between user equipment and an IMS network is P-CSCF or Proxy Call Session Control Function. P-CSCF is like a proxy, it forwards information from one point to another: Its mission is to ensure secured communications between end users and the IMS network, to forward requests and responses between end user and others IMS services (I-CSCF for a registration or S-CSCF for a service) and to provide an access to emergency services. Quality of service and SIP messages compression or decompression could be handling at this point (3GPP, 2009b). The second element is the I-CSCF or Interrogation Call Session Control Function. It assigns an S-CSCF during user registration based on information provided by the HSS such as user services or operator preferences (3GPP, 2009b) or it forwards requests for another network. The third element is S-CSCF or Serving Call Session Control Function. It handles all the session services (registration and control) and user services (bill information), accesses HSS to obtain user information. The fourth element is the HSS or Home Subscriber Server. It contains information on the end user such as identification, number, address, access information and location (3GPP, 2009b).The last element is AS or Application Server, it provides services for users on media sessions. As defined in the 3GPP 23.228 (3GPP, 2009a), users can be able to contact others from another network such as PSTIN or GSM.

2.2 Asterisk

Another solution can be found to allow WI-FI devices to communicate between themselves or to different networks: Asterisk. It is an open source project written in C which implements an IP Private Branch Exchange (PBX).This server is a combination of a core and different Application Programming Interface (API) as shown on the figure 4 (Asterisk, 2010). The core is a PBX system: it connects different users together. The four principals are channel, application, codec translator and file format. Channel is used to provide a support for different technologies such as SIP, Inter Asterisk eXchange or Integrated Services Digital Networks. Conferencing, voicemail or paging are added by the application API. Asterisk can also encode or decode different formats with the codec translator API.

3 SIP /SDP protocols

The main purpose of an IMS or Asterisk is to allow users with different clients to communicate together. As soon as a user has connection to a server, though 3G and WI-FI network for IMS or WI-FI network only for Asterisk, he can access services. However it is not “plug and play”, there are several procedures describing exchanges between users and servers for discovery, registration, authentication, calls or servers’ services. Portions of these interactions use SIP and SDP protocols. As defined into different rfc, they are protocols on application layer for “creating, modifying and terminating sessions with one or more participants” and “describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation”. According to rfc 3261 (Rosenberg et al, 2002) SIP protocol has got several features: register and subscribe methods for client declaration and registration, invite method for session creation and bye method for session termination. There are also different responses characterised by a status-code of three digits. These digits are defined from 100 to 699 with six different parts. For example from 100 to 199, it is a provisional state: request has been received and its treatment is in progress. From 200 to 299, it is a success: request has been received and its treatment is ok. From 400, server has problems to understand client requests or from 500, server failed with a valid request. They are important keys in an IP communication, every call starts with a common diagram. As shown on the figure 1, SIP protocol is based on an exchange of requests and responses between servers, proxies and clients

```
 Alice's  . . . . . . . . . . . . . . . . . . . .  Bob's
softphone                                        SIP Phone
   |                |                |                |
   |    INVITE F1   |                |                |
   |--------------->|    INVITE F2   |                |
   |  100 Trying F3 |--------------->|    INVITE F4   |
   |<---------------|  100 Trying F5 |--------------->|
   |                |<-------------- | 180 Ringing F6 |
   |                | 180 Ringing F7 |<---------------|
   | 180 Ringing F8 |<---------------|     200 OK F9  |
   |<---------------|    200 OK F10  |<---------------|
   |    200 OK F11  |<---------------|                |
   |<---------------|                |                |
   |                       ACK F12                    |
   |------------------------------------------------->|
   |                   Media Session                  |
   |<================================================>|
   |                       BYE F13                    |
   |<-------------------------------------------------|
   |                     200 OK F14                   |
   |------------------------------------------------->|
   |                                                  |
```

Figure 1: SIP session example with two proxies (Rosenberg et al, 2002)

4 Android and Sipdroid

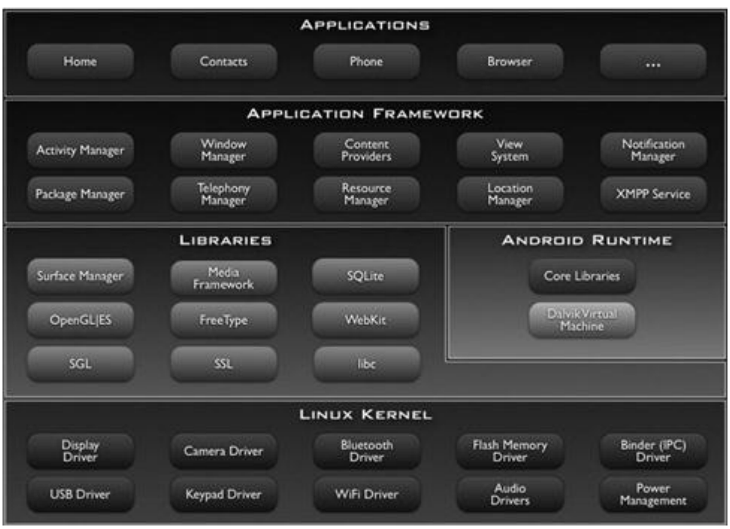

Figure 2: Android architecture (Burnette, 2008)

As shown on figure 2, Android is a mobile operation system based on a Linux kernel and code in java. Different tools are available to allow development and debugging on Android system: Android DDMS and Android development tools (ADT). According to Android developers website (Android developers, 2010a), Android Dalvik Debug Monitor Server (DDMS) provide a debugging tool which can be used on an emulator or a device connected. It can provide data on all applications running. ADT package regroups all Android tools and make them accessible from Eclipse interface or command lines. Sipdroid is a java based SIP client compatible with PBX and IMS architecture. These sources will be used to allow Video communications.

5 Testbed

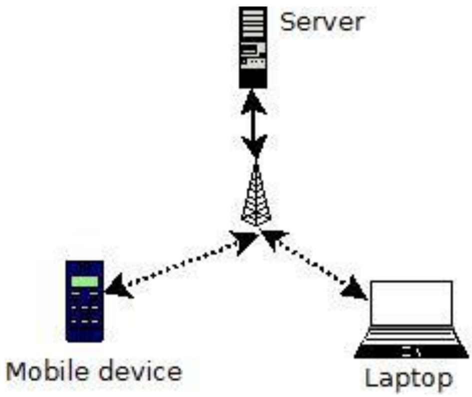

Figure 3: Testbed

As show on Figure 3, the testbed is composed with a mobile device (G1 phone), an Asterisk sever, a wireless connection and a computer with soft phones applications such as Ekiga or X-lite. Monitoring tools have been installed on the G1 phone and the computer to provide traces during the communications. A USB connection between the G1 phone and the laptop is used to transfer applications from Android SKD to the phone and to access tools on the phone.

6 Video applications

The final aim of this project is to investigate and add video calls capabilities to a SIP client (IMS or SIP based). So the first idea was to check if video from G1 phone can be sent to another device or a laptop. In order to do that a one way application is developed: when user has to enter an IP address into an interface and validates his action, a stream will be created between the device and the target. The application has to capture video frames from the camera, to encode them and to send them into the network. According to OpenCORE documentation (PacketVideo, 2009), video can be only encoded in H.263 version 1998 or 2000 known as H.263-1998 and H.263-2000. There are different parameters available such as frame size (QCIF 176 × 144 or CIF 352 × 288), frame rate (15 fps or 30 fps).The real objective of this project is to investigate and develop a video application for an IMS client. The first application was a mean to understand and test how android system reacts to a video stream: use of mediarecorder to create videos packets and creation of rtp packets. The next step is a video call. The Sipdroid project is under a GNU GPL, the sources can be modified but copyrights have to stay and it is not possible to do a commercial version without an authorization of the copyrights holders. On the Sipdroid documentation, it seems that there is not support for Asterisk server for the video calls and users have some problems to register with an Android device (Sipdroid, 2010). In considering the figure 13, we use Asterisk as server for the communication setup. In order to have the device registered to Asterisk server, two files need to be modified: SIP configuration file and Asterisk dial plane file. User profiles have to be added into the first file, they will allow SIP clients to connect and register to the server. To parameters have to be added: Qualify=no (disable round trip time timers) and canreinvite=no (use to resend an INVITE message to the clients before the beginning of the call, so all exchanges go through Asterisk) or the application crashes when there is a call. The last step is to configure the Dial plane to allow the clients to call each other. Different actions can be defined in the plane as such wait, answer, voicemail, dial or hangup. For the testbed, one client call another, so when someone calls extension 100, Asterisk takes the call, dials a SIP id and hangs up when the call is over.

To test the calls, two soft phones were installed on the laptop under Window 7, Xlite and Ekiga. They have to be configured to connect Asterisk server, in each case a SIP account is created with the usernames, passwords defined in the sip.conf file and the IP address of the server. Sipdroid code has to be modified to work properly with an Asterisk server. Android platform is not compatible for the moment with RTP stream, so video calls work in two directions but only one client can see the other camera.

7 Quality of Service and Quality of Experience

With the video application working: packets are sent and are received on the Android device, a monitoring on the quality of the video call can be implemented. It is one of the objectives of this project to provide to users information on their calls and if it is possible to adapt the call quality regarding the network conditions. Two sorts of information will be displayed: packet loss, jitter and delay for an indication of the Quality of Service and a Mean Score of Opinion for the Quality of Experience. As defined in the first part on this thesis, the calculation of the delay, of the jitter and the packet loss are direct measurements opposite to MOS score that is defined by a model. To calculate this score, different models are available such as PEVQ or V-MOS. However in this project, there is only a limited access to the media and only a reference-free algorithm can be used. One algorithm called “regression-based video quality prediction” and defined by Khan et al (2009), uses parameters from the application level to score a MOS for the video and to provide a QoE score that can be used to implement adaptation. Parameters used by this algorithm are frame rate, send bitrate and the packet loss rate. The frame rate and the send bit rate are linked to the softphone encoder. A method is created and it calculates QoS parameters and QoE score. Different variables are defined in static and there are available in all the Sipdroid application: lossvideo, latevideo, data, jitter and Mos. Others variables are used locally for the calculations. When the measure method is called, if the receiving of a RTP packet is possible the parameters are calculated. For the QoS parameters, the first calculated is packet loss rate: the current sequence number is compared to the previous one and if there is a difference: packets are lost. The delay is the time difference between the packet arrival and the previous one. For the jitter, the formula used is defined by Schulzrinne, et al (2003) in the rfc 3550. D is the difference between a packet i and the previous j, R is the time when the packet arrives and S is the RTP timestamp. J is an estimation of the jitter as called interarrival jitter.

$$D(i,j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Ri - Si)$$

$$J = J + \frac{(|D(i-1,i)| - J)}{16}$$

To calculate a QoE score, model defined by Khan et al (2009) is used. In this model, constants characterize different environments: Slight Movement, Gentle Walking and Rapid Movement. As the device is mobile and the user can walk when he uses it, the second category was chosen. So the MOS is equal to:

$$MOS = \frac{a_1 + a_2 \times (frame\ rate) + a_3 ln(send\ bite\ rate)}{1 + a_4 \times Packet\ error\ rate + a_5 (Packet\ error\ rate)^2}$$

with a_1=3.4757, a_2=0.0022, a_3=0.0407, a_4=2.4984, a_5=-3.7433

The range of this value is from 0 to 5, from bad to good.

8 Results

At the beginning, packet loss, frame rate and bitrate were calculated by the monitoring class. However Sipdroid seems to crash and/or the camera blocks itself during the tests. To limit these problems, only one parameter varies: packet loss rate and the others are set up. For testing, frame rate is 27 frames per seconds and bitrate is 140Kbytes/seconds. The tc command available under Linux platform is used on the Asterisk machine to add on the network some perturbation such as packet loss or delay. To limit the variation of the packet loss during the call, one minute is used as a reference for all calls.

		Test number				
		1	2	3	4	5
Packet loss variation	0%	MOS:4	MOS:4	MOS:4	MOS:4	MOS:4
	2%	MOS:3.8	MOS:3.7	MOS:3.7	MOS:3.8	MOS:3.7
	5%	MOS:3.6	MOS:3.4	MOS:3.5	MOS:3.4	MOS:3.4
	10%	MOS:3.3	MOS:3.2	MOS:3.3	MOS:3.2	MOS:3.2
	15%	MOS: 3.0	MOS: 2.8	MOS: 3.1	MOS: 3.1	Crash

Figure 4: MOS evolution with packet loss

In the figure 4, there is a summary of the MOS score with the evolution of packet loss during different tests. A visual evolution can be monitored in Figures 5 that was captured from the laptop with different packet loss rate. There is no visual difference between the two captures. When the packet loss is set to 5%, some errors appear in during the calls however overall quality is still good. With 10% and 15%, errors are increasingly presented and Xlite or Sipdroid can crash during the call.

Figure 5 – Video call with 10% and 15% packet loss

With the variation of the QoS parameters during the tests without any modification, the MOS is measured with a delay variation. As with the packet loss test, delay is added on the network interface.

		Test number				
		1	2	3	4	5
Delay variation	0 ms	MOS:4	MOS:4	MOS:4	MOS:4	MOS:4
	10 ms	MOS :4	MOS: 3.9	MOS:4	MOS:4	MOS:3.9
	20 ms	MOS: 3.9	MOS: 3.8	MOS:3.9	MOS:4	MOS:3.9

Figure 6: MOS with delay evolution

According to the formula for the MOS calculation, delay variation should not affect MOS values. However with a packet loss variation during the call, MOS is affected as shown on Figure 6.

9 Conclusions and Future work

This paper has built an Asterisk testbed with a mobile device to implement and test video quality measurements with a no reference algorithm. The next step in this project is to modify the testbed and use these applications with an IMS architecture. As defined at the beginning, IMS server is based on SIP protocol like Asterisk server but using different servers the communications are more complicated. Two scenarios can be made, implementation of Asterisk server as an Application server or implementation of an IMS client.

10 References

3GPP (2009a) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 9). Available at http://www.3gpp.org/ftp//Specs/archive/23_series/23.228/ [Accessed on 1/01]

3GPP (2009b) 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Network architecture(Release 9) . Available at http://www.3gpp.org/ftp/Specs/archive/23_series/23.002/ [Accessed on 2/01]

Albaladejo, A. M., Murarasu A. And Magedanz, T (2008) 'Design of a Coherent Mobile Multimedia Framework for Convergent Services'. New York: ACM

Adamek, J.G., Henrikson, E.H., Lassers H.A, Lee, A.Y. and Martin, R.B. (2002) 'Services and Technical Considerations for Wireless IP Multimedia Subsystem' Bell Labs Technical Journal, 7 (2), pp. 91-104. Available at http://www.informatik.uni-trier.de/~ley/db/journals/bell/bell7.html [Accessed on 01/02]

Android developers (2010a). Dev Guide. Available at http://developer.android.com/index.html [Accessed 1/08]

Asterisk (2010) The open source telephony project. Available at http://www.asterisk.org/ [Accessed on 29/03]

Burnette, E. (2008) How Android works: The big picture. Available at http://www.zdnet.com/blog/burnette/how-android-works-the-big-picture/515 [Accessed on 10/06]

Khan, A., Sun, L. and Ifeachor, E (2009) 'Content Classification Based on Objective Video Quality Evaluation for MPEG4 Video Streaming over Wireless Networks', World Congress on Engineering, track on International Conference on Wireless Networks, London,1-3 July. pp889-894.

Long, C. (ed.)(2001) Wireless and Mobile Network Architectures. New York: John Wiley & Sons.

PacketVideo (2009) OpenCORE Multimedia Framework Capabilities. Available at www.opencore.net/files/opencore_framework_capabilities.pdf [Accessed on 03/02]

Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V (2003) 'RTP: A Transport Protocol for Real-Time Applications', IETF, available online at http://www.ietf.org/rfc/rfc3550.txt [Accessed on 10/06]

Sipdroid (2010) Available online at http://code.google.com/p/sipdroid/ [Accessed on 23/03]

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E. (2002) ' SIP: Session Initiation Protocol', IETF , Available online at http://www.ietf.org/rfc/rfc3261.txt [Accessed on 29/01]