SIP based HD Video Conferencing on OMAP4

This document is a case study of SIP based high definition video conferencing on Android Ice cream sandwich running on OMAP4460 based Blaze Tab2 and Blaze mobile development platforms. This showcases the performance, quality, latency and interop capability of our video conferencing application on OMAP4.

Mahesha M

25-January-2013

© 2013 PathPartner Technology Consulting Pvt Ltd.,

www.pathpartnertech.com
Introduction

Our SIP based video conferencing solution is based on open standard SIP protocol for signaling, RTCP for bandwidth control, SRTP for secure transfer of video, video codec’s such as high performance H264, legacy H263, voice codecs such as G711, G722, AMR-NB, AMR-WB, Telephony capability such as DTMF. This solution supports video resolution upto 1080p@30fps and can operate at bit rates from 128kbps upto 8Mbps in over wired and wireless environment suchg WiFi. This SIP based video conferencing solution seamlessly interoperates with the hardware and software endpoints from Polycom, Lifesize and Tandberg.

System Architecture

This section describes the overall system architecture in brief. The solution mainly consists of software blocks namely GUI, SIP call manager, media manager and multimedia system. GUI is developed using Android Java which fits range of screen sizes from 7 inch to 10 inch.

The media manager block is the only interaction interface between the Call manager and the Multimedia system. The media manager block act as an abstraction layer between the overall system and the specific multimedia framework implementations like gstreamer, stagefright and our proprietary framework etc.

The media manager block is mainly be implemented with the help of threads, manager thread, player thread and recorder thread. Message queues are used for the data handling. Exposed API's accept requests from the Call Manager and push the requests into the Send Message queue. manager thread would also read the Callback request messages posted by the Media Block API’s into the Reply Message queue and make necessary callback to the Call Manager. The player and recorder threads read commands from the Send Message queue and pass on the intended requests to the Media block. The Media block APIs also push in callback requests into the Reply Message queue. The Media block act as an abstraction layer for the multimedia framework and hence consist of abstraction based APIs. Call manager talks to GUI, media and SIP.

Audio and video mixers are designed for better AV-sync for conferencing call. RTCP is designed to send the feedback about packet loss and adjust the bandwidth according to the network conditions. A simple block diagram is shown below to explain the interaction between different modules. The JNI layer is developed for the mediaphone to communicate to the media layer. The framework wrapper is developed to make the mediaphone independent of the multimedia framework. stagefright or gstreamer framework can be used as framework.
High Level Diagram of SIP-VC blocks Interaction

Sequence Diagrams

All function calls to the Media are asynchronous. The application posts message to the Media for any media service using the below mentioned APIs. The return status shall be intimated back to the application using the callback function. The sequence diagram of call, hold and conference is drawn below.

Basic Call Sequence Diagram

In basic call sequence diagram a call is made between two parties. Before making a call both parties need to register for the common SIP server. With the registered SIP id one can make a call to other party. The init_media() function initializes the media with necessary meta data to make a call. Then the session will be created and handle is given to application. Call manager talks to the underlying media manager functions, which starts the player and recorder threads.
Hold Sequence Diagram

Hold sequence diagram demonstrates how the interactions flows when a called party is put to hold

<table>
<thead>
<tr>
<th>Application</th>
<th>media manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP REGISTRATION</td>
<td></td>
</tr>
<tr>
<td>init_media()</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status init_media, null)</td>
<td></td>
</tr>
<tr>
<td>Enable_session(call_index)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(create_media_session)</td>
<td></td>
</tr>
<tr>
<td>Set_media_params(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(set_media, null)</td>
<td></td>
</tr>
<tr>
<td>start_in_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status start_in, null)</td>
<td></td>
</tr>
<tr>
<td>Start_out_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status start_out, null)</td>
<td></td>
</tr>
<tr>
<td>Stop_in_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>CALL DROP</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status stop_in, null)</td>
<td></td>
</tr>
<tr>
<td>Stop_out_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status stop_out, null)</td>
<td></td>
</tr>
<tr>
<td>End_session(call_index)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status end_session, null)</td>
<td></td>
</tr>
<tr>
<td>Close_media()</td>
<td></td>
</tr>
<tr>
<td>SIP UNREGISTRATION</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status media_close, null)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>media manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP REGISTRATION</td>
<td></td>
</tr>
<tr>
<td>init_media()</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status init_media, null)</td>
<td></td>
</tr>
<tr>
<td>Enable_session(call_index)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(create_media_session)</td>
<td></td>
</tr>
<tr>
<td>Set_media_params(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(set_media, null)</td>
<td></td>
</tr>
<tr>
<td>start_in_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status start_in, null)</td>
<td></td>
</tr>
<tr>
<td>Start_out_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status start_out, null)</td>
<td></td>
</tr>
<tr>
<td>Stop_in_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status stop_in, null)</td>
<td></td>
</tr>
<tr>
<td>HOLD</td>
<td></td>
</tr>
<tr>
<td>CREATE new SESSION</td>
<td></td>
</tr>
<tr>
<td>Create_session(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Stop_out_rtp(session_handle)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status stop_out, null)</td>
<td></td>
</tr>
<tr>
<td>Media_callback(status create_session, session_handle)</td>
<td></td>
</tr>
<tr>
<td>UNHOLD</td>
<td></td>
</tr>
<tr>
<td>Set_media_params(session_handle)</td>
<td></td>
</tr>
</tbody>
</table>
Conference Sequence Diagram

The conference call sequence given below shows how the sequence unfolds when more than two parties connects to the conference call.

```
Application                                                                                                     media manager

SIP REGISTRATION                                                                                                     
init_media()                                                                                                      
Media_callback(status init_media, null)                                                                             
Enable_session(call_index)                                                                                         
Media_callback(create_media_session)                                                                               
Set_media_params(session_handle)                                                                                   
Media_callback(set_media, null)                                                                                   
start_in_rtp(session_handle)                                                                                                
Media_callback(status start_in, null)                                                                             
Start_out_rtp(session_handle)                                                                                      
Media_callback(status start_out, null)                                                                             
Stop_in_rtp(session_handle)                                                                                         
Media_callback(status stop_in, null)                                                                               
HOLD                                                                                                             
CREATE NEW SESSION                                                                                                  
Create_session(session_handle)                                                                                    
Stop_out_rtp(session_handle)                                                                                      
Media_callback(status stop_out, null)                                                                             
Media_callback(status create_session, session_handle)                                                            
Set_media_params(session_handle)                                                                                   
Media_callback(set_media, null)                                                                                   
start_in_rtp(session_handle)                                                                                                
Media_callback(status start_in, null)                                                                             
Start_out_rtp(session_handle)                                                                                      
Media_callback(status start_out, null)                                                                             
MAKE CONFERENCE                                                                                                    
Make_conf(session_handle)                                                                                        
Media_callback(conf_call_handle, null)                                                                             
Remove_conf_call(session_handle)                                                                                    
Close_media()                                                                                                         
SIP UNREGISTRATION                                                                                                  
Media_callback(status media_close, null)                                                                            
```
Specifications

The following parameters details about the richness of the mediaphone features.

Voice and Video Call

Caller ID

Call Controls
- Dial
- Call Accept
- Speed Dial
- Re-dial
- Call Hold
- Call Cancel
- Call Resume
- Call Reject
- Call Terminate
- Call Transfer/Forward
- Join Call(3 – Way)**
- Quit Call (3- way)

Video Controls
- Block Camera
- Preview local
- Picture-In-Picture
- Full screen

Voice Controls
- Audio/ Speech mixing
- Mute/Resume

Tones
- Ring Tone
- Busy Tone
- Un-available Tone
- DND (Do not Disturb) Tone
- Call progress Tone

Record Session
- Synchronized two-way session(Video & Voice, Only voice)
- Session Playback

Answering Machine
- Record welcome message
- Record visual voice mail
- Retrieve visual voice mail

Call Quality
- Camera Noise Filter
- Camera Auto Exposure /Auto Focus
- Acoustic Echo canceller
- Voice Activity Detection (VAD)
• Comfort Noise Generator
• Background noise reduction
• Packet loss concealment
• Adaptive bandwidth adaptation

Video Quality
• H.264 BP Video Codec
• 1-2 Mbps one-way per user
• 1080p, 30fps

Voice Quality
• G.711
• G.729
• Wideband G.722

Session
• AV File format - MP4

Quality of service
• Latency: 150-250ms

Call Signaling
• SIP (based on RFCs 3261, 3263, 3264, 3265, 3515, 3911, 2327, 2327, 3428, 2976)
• SDP(2327)
• RTSP*

Call Data Transport
• RTP/RTCP
• DTMF
• Audio only call

Messaging
• Text Instant Messaging

User account
• Multiple SIP account

Servers
• Proxy
• Voice Mail & IVR

Networking
• TCP/IP, DHCP, DNS, NAT Traversel

User Interface
• Home Screen
• Phone Book / Dialer
• Call History
• Message Indicator
• Date and Time / Calendar
• Settings and Configurations
• Call Controls
• Visual Voice mail retrieval
• Text message Retrieval
• Alerts – Incoming Call, Call Join/Quit
• Volume Control

Software Platform
Video conferencing on OMAP4

- Operating System
- Telephony Framework
- Multimedia Framework
- Application Framework

Hardware Platform
- OMAP44xx

Conclusion

SIP based HD video conferencing solution is a enterprise class video conferencing on Android Ice cream sandwich running on OMAP4460 based Blaze Tab2 and Blaze mobile development platforms. It seamlessly interops with third party hardware and software endpoints. Video resolution is supported upto 1080p@30fps in OMAP4 processors. The architecture of SIP-VC application explores the multi-core capability of the OMAP processors very efficiently thus able to achieve high video resolution, quality with low latency.

References

1. OMAP4 Technical reference manual