

Video Files Recording and Playback with VoiceXML

MIROSLAV VOZNAK, LUKAS KAPICAK, JAROSLAV ZDRALEK, PAVEL NEVLUD, JAN ROZHON

VSB-Technical University of Ostrava
Department of Telecommunications
17. listopadu 15, 708 33 Ostrava
CZECH REPUBLIC

miroslav.voznak@vsb.cz, lukas.kapicak@vsb.cz, jaroslav.zdralek@vsb.cz, pavel.nevlud@vsb.cz,
jan.rozhon@vsb.cz

Abstract: - The paper deals with SW PBX Asterisk and with features supporting video calls but our primary object is not switching video calls but their automatic recording and replaying. Asterisk is open source telephone exchange which brings capability for audio and video calls but does not allow recording and playback of video files. We have implemented these capabilities and our solution is designed for customers using mobile smartphones. End-users can record a video directly on server. The video files are saved in a directory and a list of these files is generated. Names on the list are read to users by means of the speech synthesis, and subsequently the DTMF key input is used to make the choice. We applied VoiceXML for this purpose and the Interactive voice response as a tool enabling browsing in the directory and playback the selected video. Nowadays, we record video on memory card inside mobile phone but our presented solution offers an option to store the video on server in real-time, for end-user is simple to make a call and video is being automatically recorded.

Key-Words: - Video calls, Recording, Playback, Smartphones, VoiceXML, Asterisk.

1 Introduction

Asterisk is software that can be installed on any personal computer, turning it into a communication server. Main functions of Asterisk include establishing audio and video calls, serving as a voicemail VoIP gateway etc. [1], [5].

Asterisk has native support for transmitting video calls, and it is possible to establish a video transmission between two or more participants. Asterisk also has native support for many video codecs, including H.263, H.263p and H.264. On the other hand, recording videos and their playback is not supported by Asterisk.

In our configuration, Asterisk functions as a multimedia video server. Our modified Asterisk enables saving video calls into a file and subsequently replaying these files to other users through their terminals which also support video calls. Desktop computers and/or mobile devices with appropriate software can serve as user terminals.

We will connect IP cameras to end-user terminals and play streamed video from these cameras on them. Users can watch the video captured by means of their user devices such as smartphones or softphones installed on their personal computers.

Our solution is also suitable for users in offices who use personal computers for video calls but it is primarily targeted at customers using smartphones. Mobile devices support video calls through the video

service available in a 3G network but the quality of such video calls does not correspond with mobile device's hardware in mobile devices [4].

Mobile devices with an open operating system can be upgraded with third party software, including VoIP clients. VoIP clients enable a direct connection to Asterisk without any gateways. It is possible to use Wi-Fi connection to access Asterisk but Wi-Fi does not cover large areas. We tested UMTS networks to transmit audio and video files. These networks enable using VOIP or video calls.

Asterisk that enables recording and playback of video files has PULL and PUSH functions. The PULL function requires some end-user interaction, such as choosing a particular video. In this situation, the DTMF key input has to be applied to select a specific file. We implemented VoiceXML support for easy navigation through saved video files [12].

The PUSH function, on the other hand, allows other users to choose a specific video file and subsequently routes these video files to a particular end-user or a group of the end-users. This function does not require any interaction by end users

2 Video Playback and Recording

Another software necessary for streaming video content is MPEG4IP. MPEG4IP provides a standards-based

system for encoding, streaming and playing encoded audio, video and text [7].

FFmpeg and MPEG4IP extensions enable operating systems to support multimedia content. In order to connect Asterisk with FFmpeg and MPEG4IP, an `app_mp4.c` extension has to be installed into Asterisk. This extension consists of two components. First one is `mp4play` that enables playing video files directly on end-users devices. The second is called `mp4save` and enables saving video files on a host's disk or a different storage media.

The `app_rtsp.c` extension enables streaming multimedia contents. This extension allows us to connect streaming contents from IP cameras or VLC player [8] with end-user devices. During our tests, we did not take advantage of the video call service offered by 3G networks [4]. This service has its limitations as regards video quality and resolution of transmitted video files. We used the data connection offered by UMTS networks. Data bandwidth and latency enables us recording and playing video files with a higher resolution compared to the video call service implemented in UMTS networks.

With VoiceXML support implemented into Asterisk, we used synthesised voice to read names of saved files. Our solution uses VoiceXML in version 2.0 and VoiceGlue [12], [13].

The configuration for video playback and recording is similar. We installed the Asterisk communication server on a personal computer equipped with the Ubuntu operating system. End-user devices were connected to this server via a local computer network. Smartphones were connected to the server through Wi-Fi or UMTS network.

2.1 Video Recording

To record videos, we used smartphone with the Android operating system and the Ekiga softphone installed on a personal computer. Ekiga is a multiplatform application that can be installed on personal computers running on Linux and Windows operating systems [9], [10]. Ekiga supports H.261, H263, H263p, H.264, theora and MP4V-ES video codecs. We chose the IMSDroid application for the Android operating system [11]. This application supports the same video codecs as the Ekiga softphone. To record video files, we can usually use front or rear facing cameras in mobile devices. Certain new HTC smartphones, however, are not equipped with a front-facing camera.

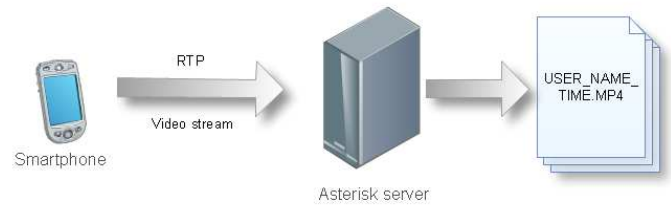


Fig. 1. Video recording

Fig. 1 depicts our testing configuration for video recording. The Asterisk server was the destination. The code below illustrates the `extension.conf` set up for an automatic recording of all video files.

```

exten => 784,1,Answer()
exten =>
784,2,mp4save(/tmp/${CALLER(num)}${EPOCH},,D%d:M%m-%H:%M:%S_%$.mp4)
exten => 784,3,Hangup()
  
```

Recorded video files are tagged with caller's ID and time when the video call was established. The name of the video file will be subsequently adjusted for voice synthesis.

If the user chooses the "send video" function, an RTP relation between the end-user device and Asterisk is automatically established and the video is saved to a pre-defined storage in `extension.conf`. `Extension.conf` can be configured to playback video files which the user saves into the pre-defined storage space on the server.

After recording, users can watch their recorded video file and subsequently decide whether the file should be saved or deleted from the database. Navigation between these choices is extended by the DTMF key input with synthesized voice navigation.

2.2 Video playback

Video playback can be initiated from both sides. Fig. 2 shows its initialization by the user. The end-user chooses a particular video file through the DTMF key input. This technique is limited as regards the number of possible choices. Where the database is too extensive, the user can not choose the video file that s/he needs. S/he can only choose from restricted options, such as for instance the last 10 video files.

Names of video files are saved into a database and users choose for example the last ten video recordings viewed. We used names of video files from specific users or last records of all users. Alternatively, a coordinator could draft a list of most interesting video files. In this case, the video file name - `video.mp4` is transformed into a specific name of video file.

With VoiceXML support installed, names of video files are being read. Users hear the names of saved video

files in a particular list. Using this technology, users do not have to watch the screen of their mobile device and can choose a specific video file using the DTMF key input.

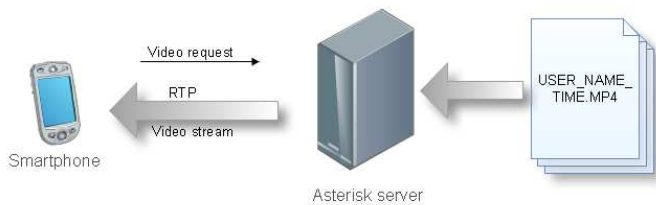


Fig. 2. PULL video playback

We show the extension.conf configuration file for video playback. This configuration file contains information about the number which the user will dial for the video playback from the server side. The name of the video file depends on the content of the saved video. The name can be fixed or generated dynamically.

```
exten => 782,1,Answer()
exten => 782,2,mp4play(/tmp/video.mp4)
exten => 782,3,Hangup()
```

Fig. 3 shows the concept that we are working on at the moment. Selecting video files will be initiated through a website, and the requested video file will be chosen by a coordinator and PUSHed to the end-user device. All the user has to do is to answer the incoming call. With the DTMF key input, users can replay, pause or continue watching an incoming video.

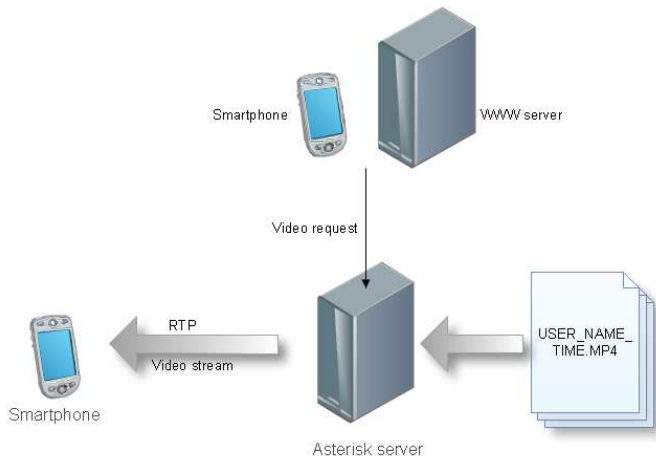


Fig. 3. PUSH video playback

End-users using softphones and smartphones can sort out videos from the database through their web browser and PUSH the video files to their end-user device. They can also route a video into their device from someone else's device.

The recorded video is saved in the storage space in its native format with standard codecs. Mobile devices can

play these video files and users can download them to their local storage media and subsequently play them.

Currently, we are working on the parameters enabling video transcoding such as resolution and/or frame rate in a real-time video transmission.

2.3 Video Streaming

To be able to stream video, we need a source of streaming content. Streaming content's sources include IP cameras or media players that support streaming. Such media players include for example VLC media player [8]. IP cameras and media players should support streaming in codecs that are compatible with Asterisk and VoIP clients in end-user devices. Fig. 4 shows the video streaming configuration. Where the end user wants to choose a specific camera, s/he can use the DTMF key input. IP cameras' names are read by voice synthesis software. If the camera is located in a specific place, the camera's address is generated into the synthesis speech output. Currently, we are also working on establishing a connection between information provided by the GSM network and the list of cameras for the DTMF key input. If we want to make the list of cameras shorter – i.e. suitable for the DTMF key input – we need to choose the closest cameras. Using the information provided by the GSM network, we can determine the approximate location of a mobile user in terrain. Once we have this information about location, we can form a list of the closest IP cameras.

IP cameras' names and/or destinations will be automatically read to the end user and it will be easy to choose the right one. The choice will be facilitated through the DTMF key input.

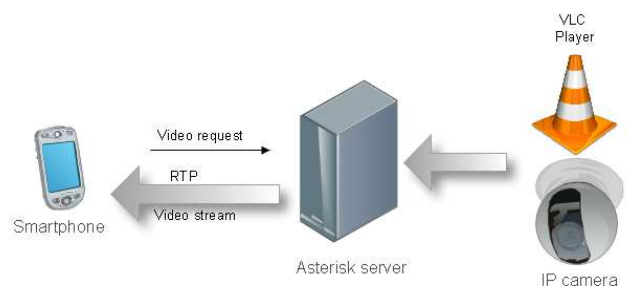


Fig. 4 PULL initialized a streaming to host

We show Asterisk playing the streaming content to more end-users, see below.

```
exten => 783,1,Answer()
exten =>
783,2,Rtsp(rtsp://192.168.1.41/live.sdp)
exten => 783,3,Hangup()
```

End-users select a specific IP camera or media player through the dialled number. Thanks to web pages, it is possible to initialize the PUSH technology as well as to send videos from a specific IP camera or multimedia player to end-users devices.

In Fig. 5 you can see that Asterisk perform streaming content to more end-users. End-users select specific IP camera or media player through dialed number. Thanks to web pages it is possible to initialize PUSH technology and send video from specific IP camera or multimedia player to end-users devices.

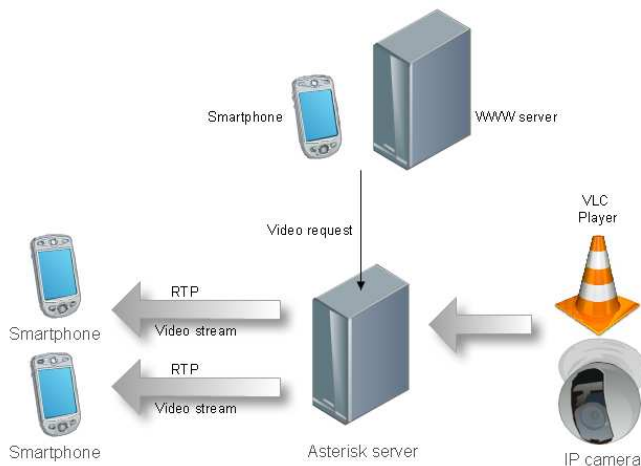


Fig. 5. PUSH initialized streaming to multiple hosts

4 VoiceXML Implementation

For VoiceXML implementation in Asterisk, we used an open-source project called VoiceGlue [13]. VoiceGlue provides a VXML interpreter for Asterisk using the OpenVXI interpreter. VoiceGlue enables speech synthesis for any kind of texts. In our solution, we used VoiceGlue software to read names of video files saved in a server's storage space.

The list of saved video files is generated automatically. New video files are subsequently saved according to the date on which they were created. All records should have a specific name format. The format also contains author's name and date of creation.

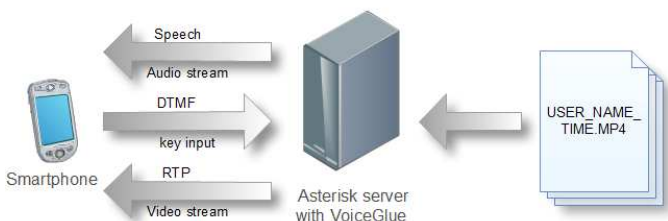


Fig. 6. Architecture with VoiceXML support

VoiceGlue software was installed on the same Ubuntu system on which the video support was installed. Fig. 6

depicts our solution's architecture and the following code shows an example of VoiceXML structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.0"
xmlns="http://www.w3.org/2001/vxml">
  <menu id="top" dtmf="true">
    <property name="inputmodes" value="dtmf"/>
    <prompt>
    <enumerate>
      For <value expr="_prompt"/>, press <value
      expr="_dtmf"/>
    </enumerate>
    </prompt>
    <choice next="#video1.mp4">
      Video1
    </choice>
  </menu>
  <form id="video1">
    <block>
      <prompt>
      Video 1 from user Bob established in 10:31 AM
      </prompt>
      <goto next="#top" />
    </block>
  </form>
</vxml>
```

5 Results

Mobile device establishes audio connection with Asterisk via SIP protocol. Users listen possible choices like as send video files, play video files, watch streaming content from IP camera. Through DTMF key input they will choose appropriate choice defined in extensions.conf.

New 3G mobile networks like HSDPA or HSPA+ support high speed transmission rates and if mobile devices have sufficient hardware for playing and recording high resolution video we can use them for video recording and playback via Asterisk.

6 Conclusion

Asterisk is communication server with support for switching voice and video calls through number of audio and video codecs. Unfortunately, Asterisk has no capabilities for recording and playback video files. Our solution is primary designed for customers using smartphones. These customers will apply their end-user devices for recording video and immediately save this video files to server. In end-user devices is not necessary a large storage memory because video is not saving into their storage but it is saving directly on the server side. Other users will have this video immediately available.

Video files are saved on server side and it is generated list of these files. List will be managed automatically and/or some coordinator could manage sequence of video files in list. Names in this list are read to users with speech synthesis and then is used DTMF key input for choosing the right choice.

Asterisk with program mp4play plays video files saved on server storage directly in the end user equipment. From Asterisk log we even know if user received this video and then we have backwards control if user received video call from server.

Now we are working on connection between Asterisk and web server. With this connection it will be easy to choose video from database and PUSH this video directly to the end user device.

The paper focuses mainly on video transfer to or from the mobile end user devices over a wireless network. The video quality is important for end users, so the next challenge will be to assess the quality of transmitted video.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

References:

- [1] Jim Van Meggelen, Jared Smith, Leif Madsen, *Asterisk: The Future of Telephony*, O'Reilly Media, pp. 15 - 48, September 2005. ISBN 978-0-596-00962-5.
- [2] Timo Halonen, Javier Romero, Juan Melero, *GSM, GPRS and EDGE Performance Evolution Towards 3G/UMTS*, 2.edition, West Sussex, England: John Wiley & Sons, Ltd, 2003, pp. 11, 14, 49. ISBN 0-470-86694-2.
- [3] Jörg Eberspächer, Hans-jörg Vögel, Christian Bettstetter, Christian Hartmann, *GSM-Architecture, Protocols and Service*, 3.edition, West Sussex, England: John Wiley & Sons, Ltd, 2002, pp. 58, 108, 139, 212, 234, 264, 266. ISBN 978-0-470-03070-7.
- [4] Holma, Toskala, *HSDPA/HSUPA for UMTS*, 1. edition, West Sussex, England: John Wiley & Sons, Ltd, 2006, p. 31 and 61. ISBN 0-470-01884-4.
- [5] M. Voznak, *Voice over IP*. VSB-Technical University of Ostrava: College book, 1st. ed., Ostrava, 2008.
- [6] F. Caron, S. Coulombe, Tao Wu, A Transcoding Server for the Home Domain, In: *Portable Information Devices*, Ecole de Technol. Supérieure, Montreal, May 25-29, 2007, pp.1 - 5.
- [7] S. Zoi, D. Loukatos, L. Salakis, P. Stathopoulos, N. Mitrou, Extending an Open MPEG-4 Video Streaming Platform to Exploit a Differentiated Services Network, In: *High-Speed Networks and Multimedia Communications 6th IEEE International Conference, HSNMC 2003*, July 2003, pp. 385-405.
- [8] H. Nguyen, P. Duhamel, M. Brouet, D. Rouffet, Optimal VLC sequence decoding exploiting additional video stream properties, In: *Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP '04). IEEE International Conference*, May 17-21, 2004, pp. iv-621 – iv-624 vol.4.
- [9] K. Paul, T.K. Kundu, Android on Mobile Devices: An Energy Perspective, In: *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on Digital Object Identifier*, June 29 – July 1, 2010, pp. 2421 – 2426.
- [10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R.Sparks, M. Handley, E. Schooler, *RFC 3261 - SIP: Session Initiation Protocol*. Technical report, IETF, 2002.
- [11] Gonzalo Camarillo, Miguel-Angel Garcia-Martin, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, England: John Wiley & Sons, Ltd, 2004, pp. 205 - 261. ISBN 978-0470871560.
- [12] Hyeong-Joon Kwon, Kwang-Seok Hong, A Design of User-Initiative Voice Web Using RSS and VoiceXML, In: *Software Engineering Research, Management & Applications, 2007. SERA 2007. 5th ACIS International Conference*, Aug. 20 – 22, 2007, pp. 281- 288.
- [13] D6.3 Data Formats and Protocols for Information Handling in INDECT Portal; Indect consortium, deliverable Dec. 29, 2010.