

Multimedia services in Asterisk based on VoiceXML

M. Voznak, L. Kapicak, J. Zdralek, P. Nevlud and J. Plucar

Abstract—The paper deals with multimedia services that cooperate with the Asterisk, as the communication server. Asterisk supports many services and we focused on the services that are not supported by native Asterisk. These services include recording and playback of video files, SOAP protocol support, VoiceXML and user location information. We put together all these services and created a comprehensive system. Its primary roles are performed by Asterisk. Our system enables users to save recorded video files directly to the server. Subsequently, they can replay their own videos or someone else's files. Video files or IP cameras are easily selected by dialling a specific number. A video transmission similar to a video call is then automatically established. In our solution, the management provided by a web services server that uses SOAP protocol. User applications can use this SOAP interface to connect to Asterisk. The project is aimed in particular at users equipped with mobile end-user devices; certain services have been explicitly designed for these users. Transmission in mobile networks is limited in terms of transmission rates and latency. We defined limits for transmitting video streams in mobile networks.

Keywords—Video recording, Video playback, UMTS, GSM, Smartphones, VoiceXML, Asterisk, web services.

I. INTRODUCTION

THE well-known open-source software solution Asterisk can operate as a voicemail, VoIP gateway, call center, protocol 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 and 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 and 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]. For choosing video files exist other way. Mobile devices are equipped by web browser and through the web portal, users can establish video call to their devices.

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.

II. VIDEO IN MOBILE NETWORKS

UMTS is a mobile network optimized for video calls. It features a channel designed for transmission video calls [3]. Support for video calls has to be implemented into the mobile end-user device. It also has to be equipped with a front camera. UMTS network's video call channel capacity is limited to 64 kb/s [2]. The quality of video transmission is rather poor.

Mobile networks also support data transmission. We tried to transmit video files through data connection in the 3G network. The 3G mobile network supports data rates as shown in Figure 1 and 2. Transmission rates depend on hardware implemented by the mobile provider as well as on the configuration what he had chosen. You can find more information about transmission rates in literature [2].

Manuscript received March 31, 2011. 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.

M. Voznak, L. Kapicak, J. Zdralek, P. Nevlud and J. Plucar are with Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava Poruba, Czech Republic (phone: +420 596 991 699; e-mails: miroslav.voznak@vsb.cz, lukas.kapicak@vsb.cz, jaroslav.zdralek@vsb.cz, pavel.nevlud@vsb.cz and plu042@vsb.cz).

HS-DSCH class	Transmission speed [Mbps]
1	1.22
2	1.22
3	1.82
4	1.82
5	3.65
6	3.65
7	7.21
8	7.21
9	10.13
10	13.98
11	0.91
12	1.82

Fig. 1 Transmission speed of HSDPA.

E-DCH class	Transmission speed [Mbps]
1	0.71
2	1.45
2	1.4
3	1.45
4	2
4	2.89
5	2
6	2
6	5.74

Fig. 2 Transmission speed of HSUPA.

Coding Scheme	Transmission speed [kbps]
MCS-1	8.8
MCS-2	11.2
MCS-3	14.8
MCS-4	17.6
MCS-5	22.4
MCS-6	29.6
MCS-7	44.8
MCS-8	54.4
MCS-9	59.2

Fig. 3 Transmission speed of EDGE in one channel.

UMTS coverage is not as extensive as that of the GSM network. Unfortunately, GSM network's capacity and data rates is not comparable with that of the UMTS network. GPRS data connection does not enable video transmission. The EDGE technology offers higher bitrates and also enables

video transmission. Figure 3 shows EDGE technology's bitrates in a single channel. Higher bitrates can be achieved by combining multiple channels. Mobile networks in the Czech Republic support the MCS-9 coding scheme and offer a combination of four channels for downlink and one channel for uplink. Alternatively, the combination is three channels for downlink and two channels for uplink is also available. This combination has advantage for uplink.

III. COMPARISON OF TRANSMISSIONS IN MOBILE NETWORKS

In order to optimize the quality of transmitted video files, we tested the real transmission speed of mobile networks. Using an FTP server, we tested both uplink and downlink. We also tested latency.

We discovered that the minimum technology enabling video transmission is the EDGE technology. GPRS data transmission has large latencies and its transmission rate is very poor, therefore it cannot be used for video transmissions. Due to specifications [2], the maximal theoretical transmission speed with multiple channels is only 85.6 kb/s for downlink and 42.8 kb/s for uplink.

Figure 4 shows transmission speed in various mobile networks. Given the specifications, we only tested networks that support video transmission in real time.

Technology	Latency [ms]	Downlink [kb/s]	Uplink [kb/s]
EGPRS	189	128.83	102.97
UMTS R99 FDD	150	124.74	112.67
UMTS R99 TDD	130	217.58	153.24
HSDPA R5	91	1989.56	321.31

Fig. 4 Tested mobile network specifications.

Characteristics of mobile networks are very variable. They depend on many factors (load, network quality, number of connected users, network settings etc.). We measured network specifications in areas with good coverage and good network signal.

Thanks to parameters measured, it is possible to choose video codec, optimize transmitted video's resolution, bitrate and number of frames per second.

We tested video codecs supported by Asterisk. We used the resolution supported by IMSdroid, i.e. 176x144 pixels. Figure 5 shows information about codec and bitrate of a video file. We carried out the tests in a Wi-Fi network which has enough transmission capacity for real time video transmission. We used the Wi-Fi network for testing because we need to know which video settings will be available in a particular mobile network.

Codec	Bitrate [kbp]	File [kB]
ITU H.263 video (3GPP)	164	397
ITU H.263 video (3GPP)	154	380
H.264/MPEG-4 AVC	225	543
H.264/MPEG-4 AVC	538	1246
H.264/MPEG-4 AVC	1432	3291

Fig. 5 Codecs and bitrate of saved video files.

The architecture for testing real time video transmission in a mobile network is shown in Figure 6. We selected a smartphone with the Android operating system for testing. Mobile devices enable switching between UMTS and GSM networks. The Asterisk server is the final component.

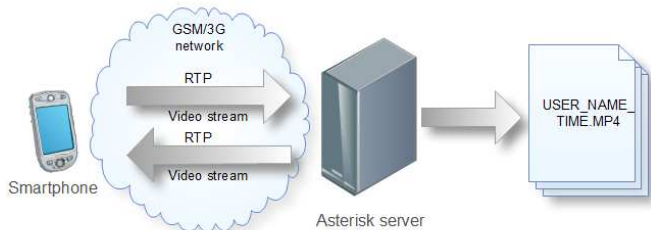


Fig. 6 Transmission through mobile network.

Video files were saved directly on the server side. The quality of these video files depends on bitrate and codec. We played video on the same device with the same settings for video transmission. The best results were achieved using the H264 video codec. The difference in video quality between 538 kb/s and 1432 kb/s bitrate is not significant. The H.263 video codec yielded the poorest results. Figure 7 shows the differences in video quality achieved. Pictures in Figure 7 are screenshots from another device on which we played the files recorded on Asterisk.

The best results were achieved using the H264 video codec. When comparing them to H.263 video codec results, the differences were mainly in sharpness and in colours. The bitrate of video files is very important for video quality. When using the H.264 video codec, the video stream at bitrate 225kb/s could theoretically be transmitted through the EDGE network. During testing, we encountered certain issues with dropped frames. Also, the SIP client sometimes froze. The testing in the 3G network yielded better results. This time, we encountered no problem with dropped frames and the application did not freeze. We were able to use the H.264 video codec with 1432kb/s in the HSDPA network. In further stages, we will be testing a higher streamed video resolution. A higher resolution is only available in the UMTS network.

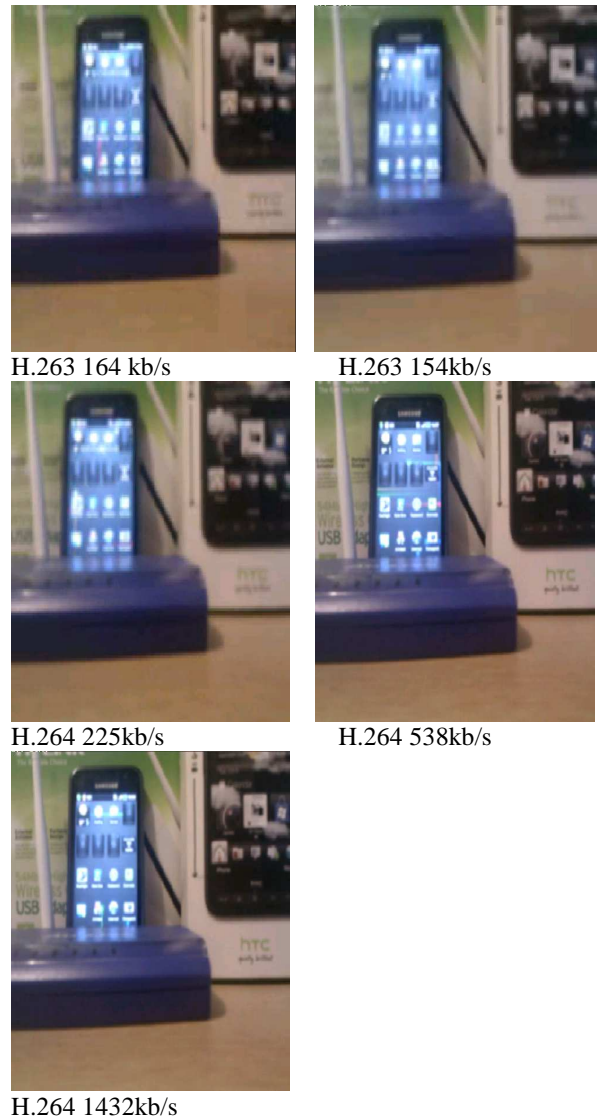


Fig. 7 Differences in video quality.

IV. 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. FFmpeg is very important for transcoding video files into format supported by mobile device. 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. We were able to use 2G mobile network with some limitations in quality and bitrate of recorded video files.

A. 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, H.263, H.263p, 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.

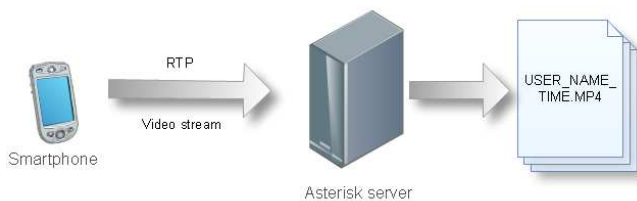


Fig. 8 Video recording.

Figure 8 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.

B. Video playback

Video playback can be initiated from both sides. Figure 9 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.

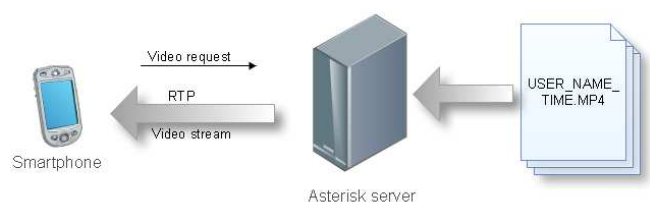


Fig. 9 PULL video playback.

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()
```

Figure 10 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.

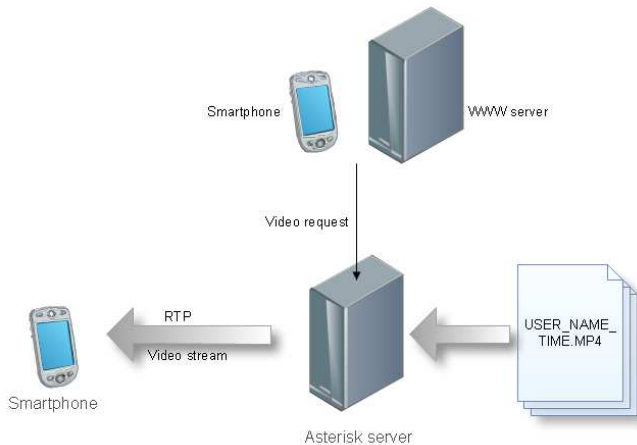


Fig. 10 PUSH video playback

C. Web access to the Asterisk

The web services server will be cooperating with the Asterisk. The web services server will be communicating with other web services servers or with web services clients through the SOAP (link) protocol [14]. Asterisk does not have the SOAP interface so we had to develop some techniques to enable Asterisk to cooperate with the web services servers and web services clients.

Figure 11 illustrates the architecture that features a web services server. Through the WSDL file, it enables cooperation among different web services clients. Clients may include other web servers, end user devices, desktop computers etc. This technology allows other end-user devices to connect to Asterisk through the SOAP protocol. Subsequently, commands can be used to establish calls and to obtain information about the calls or connected users. We can implement any other CLI commands [5].

D. Web access Architecture

Using AMI (Asterisk Manager API) [5], we have created an application that can connect to the Asterisk server and issue various commands or read events over SSH. The application uses commands over the CLI asterisk interface [5]. We implemented these commands:

```
dial [extension[@context]]
sip show peers
originate local/1001@default extension 1002@default
```

The most important command is the command to establish a connection. We can establish an audio connection between two users or between the Asterisk server and one or multiple

users. We can also establish a video connection and subsequently play a video file saved on the server.

Generally, our application sends a request over SSH to the Asterisk server, which processes the request and generates a proper response. Response messages are then sent back to the client application where they are displayed to the user.

This application is hosted on a web server, either on the same server as Asterisk or on any other remote server. We need to host this application on a server in order to create a public interface for this application. We have achieved this by using the web services.

Web services are used to communicate between two electronic devices over a network. This communication is handled by SOAP (simple object access protocol) messages. The web service interface is described using WSDL [15] (web service description language). Both WSDL and SOAP are described in XML. We made this WSDL file public, so developers may use it to create new applications.

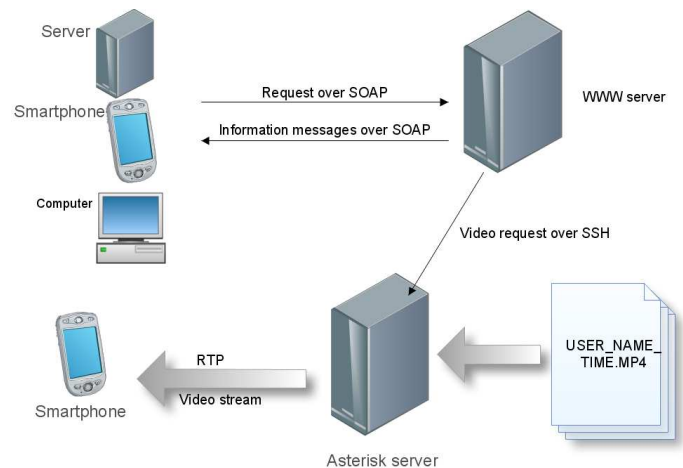


Fig. 11 SOAP and its integration into scheme.

An important part of our project is a web page that acts as a web service client. It is used to visualize requests and responses. It also displays a list of users in the Asterisk database, so anyone who has been authenticated and authorized to use the web page interface could manage these users. As for now, the main functionality is to establish the call between two selected users.

A sample from the web service server code is below. This code was created using java programming language and NetBeans as IDE. We can see that the method takes two arguments (each argument represents a specific user in the Asterisk database) which are created by WebParameters. The method itself creates an instance of the connection class which manages the connection between two users. Once the call is established, the programme sends an info message.

```
@WebMethod(operationName = "establishConection")
public boolean establishConection(@WebParam(
    name = "caller") User caller, @WebParam(name = "called")
    User called)
{
```



```

try
{
    Connection con = new Connection(caller, called);

    if (con.isActive()) {
        System.out.println("Connection between ["
            + caller.getUserName() + "] and ["
            + called.getUserName() + "] has been
established.");
    }

    return con.isActive();
} catch (Exception exc) {
    //handle exception
}
return false;
}

```

Output from this method is shown below. We can see successfully established connection between two users.

INFO: Connection between [John Black 0] and [John Black 2] has been established.

This message was generated from web services server and was send to the client, which is able to recognize successfully established connection from this message.

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.

E. 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. 14 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.

F. Information about User Location

An user location could be determined in different ways. User terminals are equipped with a GPS module and they provide the server with a very accurate user position. This solution has its disadvantages. They include the time required to establish user location and determining the location inside of a building. We tested the time required to establish the user location by means of GPS. It ranges from one minute to five minutes. A-GPS technology is capable to establish the position faster but it still takes about one minute. The time to establish the satellite position depends on many factors such as weather condition, location, GPS platform etc.

It is very difficult or sometimes impossible to receive GPS signals from satellites inside buildings and in cities with high buildings.

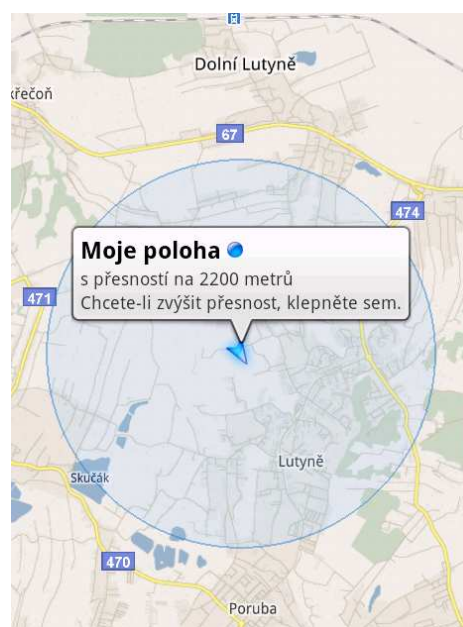


Fig. 12 Google location in Google maps.

The accurate user position is not important for us. However, we need to know the position as soon as possible in order to be able to create a list of closest IP cameras. For this purpose, we use GSM or UMTS stations [2]. Every station has its own unique code and channel on which it is transmitted. If we want to use this approach we have to know location of all stations and their respective coverage. This information is handled by mobile operators and by Google. Such information is classified and can only be obtained by signing a contract with an operator. Google [16] has many applications such as Google maps, Google earth etc. that use location techniques. Figure 12 shows locations in Google maps.

The accurate or approximate user position is determined through GSM/UMTS network, Wi-Fi or by means of GPS.

Location requests are sent to a specific location server administered by Google. The address of the location server is:

<http://www.google.com/loc/json>

Messages sent to the Google location server need to have a specific format and have to be sent as HTTP Post messages. The code below shows the structure of HTTP POST message with information about the GSM network:

```
{
  "version": "1.1.0",
  "cell_towers": [
    {
      "cell_id": "7132",
      "location_area_code": 1721,
      "mobile_country_code": 230,
      "mobile_network_code": 2,
      "age": 0,
      "signal_strength": -61,
      "timing_advance": 9
    }
  ]
}
```

Message contents this information:

cell_id – unique identifier of the cell. (CID for GSM, BID for CDMA)
 location_area_code – location Area Code (LAC for GSM, NID for CDMA)
 mobile_country_code – Mobile Country Code (MCC for GSM and CDMA)
 mobile_network_code – Mobile Network Code (MNC for GSM, SID for CDMA)
 age – the number of milliseconds since this cell was primary. If age is 0, the cell_id represents a current measurement
 signal_strength – Radio signal strength measured in dBm
 timing_advance – Radio signal strength measured in dBm

Network information provided by the mobile end-user device is immediately sent to the Google location server and then Google responds with the following message:

```
{"location":{"latitude":49.877187,"longitude":18.415588,"accuracy":2272.0},"access_token":"2:ZpschXrzf1WRNvwQ:0RQASmQxllmifvCQ"}
```

The accuracy of this location technique is very poor but the method is sufficient for our purposes. We need information about the user location very fast. Once we know the location, we can offer a list of the closest IP cameras.

Architecture for using Google location server is depicted in figure 13:

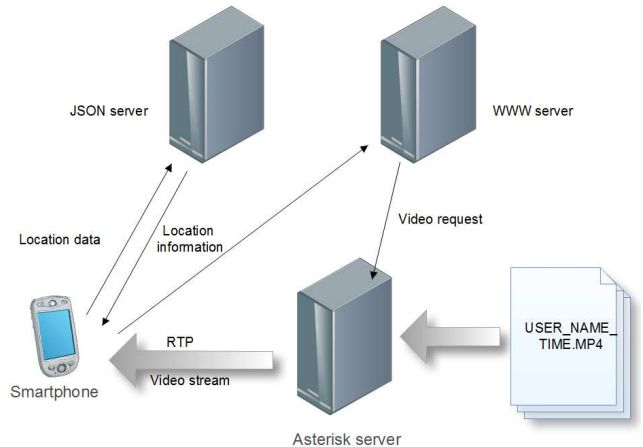


Fig. 13 Architecture of cooperating with Google location.

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

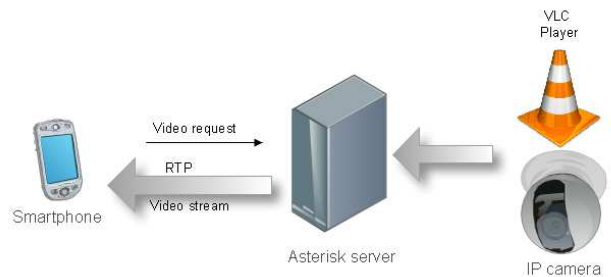


Fig. 14 PULL initialized a streaming to host.

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 dialed 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 Figure 15 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.

The SOAP protocol is used to establish a connection between the video source and the end user.. The coordinator in an office can see the users online – users connected to Asterisk. He can also see the saved video files. The

coordinator can choose a specific user and a specific file on the website. Subsequently, Asterisk establishes a connection with this user. Location could be one of the parameters to choose the destination user.

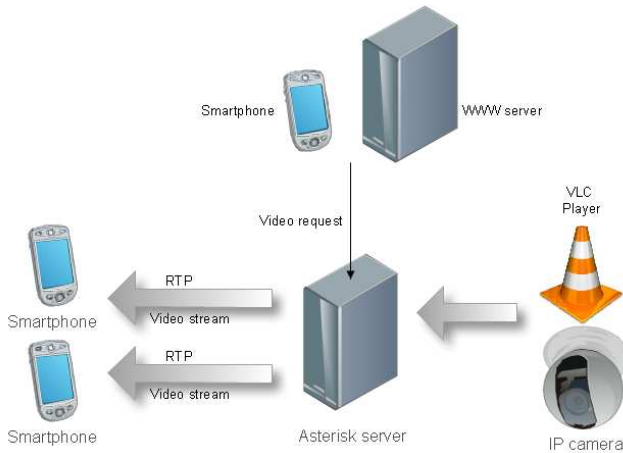


Fig. 15 PUSH initialized streaming to multiple hosts.

V. 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.

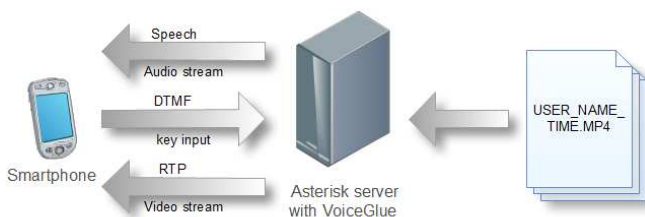


Fig. 16 Architecture with VoiceXML support.

VoiceGlue software was installed on the same Ubuntu system on which the video support was installed. Figure 16 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>
```

VI. RESULTS

Mobile device establishes audio connection with Asterisk via SIP protocol. Users listen possible choices as 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.

Another way for deploying video contents is web management. Authorized users can choose particular users and establish video streams to their end-user devices. It is possible to stream a video from a file or the video comes from an IP camera to end-user devices. The selection of a particular user depends on many factors, their location being one of them. We use the Google location server to obtain information about the user location.

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.

VII. 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.

ACKNOWLEDGMENT

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] J. Meggelen, J. Smith and L. Madsen, *Asterisk: The Future of Telephony*, O'Reilly Media, pp. 15 - 48, September 2005. ISBN 978-0-596-00962-5.
- [2] T. Halonen, J. Romero and J. 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.
- [1] M. Voznak, F. Rezac and M. Halas, "Speech Quality Evaluation in IPsec Environment, " *WSEAS 12th International Conference on Networking, VLSI and Signal Processing (ICNVS '10)*, FEB 20-22, 2010, Cambridge, England, pp. 49-53.
- [3] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS*, 1. edition, West Sussex, England: John Wiley & Sons, Ltd, 2006, p. 31 and 61. ISBN 0-470-01884-4.
- [4] M. Voznak, Advanced implementation of IP telephony at Czech universities, *WSEAS Transactions on Communications*, Volume 9, Issue 10, October 2010, Pages 679-693.
- [5] F. Caron, S. Coulombe and T. Wu, A Transcoding Server for the Home Domain, In: *Portable Information Devices*, Ecole de Technol. Superieure, Montreal, May 25-29, 2007, pp.1 - 5.
- [6] S. Zoi, D. Loukatos, L. Salakis, P. Stathopoulos and 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.
- [7] H. Nguyen, P. Duhamel, M. Brouet and 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.
- [8] K. Paul and 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.
- [2] M. Voznak and J. Rozhon, Methodology for SIP infrastructure performance testing, *WSEAS Transactions on Computers*, Volume 9, Issue 9, September 2010, Pages 1012-1021.
- [9] G. Camarillo and M. 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.
- [10] H. Kwon and K. 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.
- [11] D6.3 Data Formats and Protocols for Information Handling in INDECT Portal; Indect consortium, deliverable Dec. 29, 2010.
- [12] M. Kalin, *Java Web Services: Up and Running*, O'Reilly Media, 1 edition, 2009, pp. 19, 75, 159. ISBN 978-0596521127.
- [13] S. Weerawarana, F. Curbera, F. Leymann, T. Storey and D. F. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*, Prentice Hall, 2005, pp. 47, 96, 112, 303. ISBN 978-0596521127.
- [14] Google geolocation API, Google. Retrieved 2011-03-8. <http://code.google.com/p/gears/wiki/GeolocationAPI>



Miroslav Voznak holds position as an associate professor with Department of Telecommunications, VSB-Technical University of Ostrava, Czech Republic. He received his M.S. and Ph.D. degrees in telecommunications, dissertation thesis "Voice traffic optimization with regard to speech quality in network with VoIP technology" from the Technical University of Ostrava, in 1995 and 2002, respectively. Topics of his research interests are the next generation network, IP telephony, speech quality and network security.



Lukas Kapicak received his M.S. degree in telecommunications from VSB-Technical University of Ostrava, Czech Republic in 2007. Since 2007 has been studying Ph.D. degree at the same university. His research is focused on wireless transmission and data flow analysis, simulation and optimization.



Jaroslav Zdralek holds position as an associate professor with Department of Telecommunications, VSB-Technical University of Ostrava, Czech Republic. He received his M.S. degree in Computer Science from Slovak Technical University of Bratislava, Slovakia in 1977. He received his Ph.D. degree from VSB-Technical University of Ostrava in 2002, dissertation thesis „Diagnostic system without dismantling of locomotive controller“. His research is focused on fault tolerant system, communication technologies and coding.



Pavel Nevlud received his M.S. degree in telecommunication engineering from VSB-Technical University of Ostrava, Czech Republic in 1995. Since this year he has been holding position as an assistant professor at the Department of Telecommunications, VSB-Technical University of Ostrava. The topics of his research interests are communication technologies, networking and security.

Jan Plucar is a student with Department of Computer Science, VSB-Technical University of Ostrava, Czech Republic. He received bachelor's degree in 2009 and he continues in the master study programme at Faculty of Electrical Engineering and Computer Science in Ostrava.