Multimed Tools Appl (2013) 62:733–759
DOI 10.1007/s11042-011-0875-1

Socio-technical aspects of remote media control for a NG9-1-1 system

Vikram Chandrasekaran · Ram Dantu · Kalyan Pathapati Subbu

Published online: 13 September 2011

Abstract 9-1-1 emergency calls mostly involve distress situations that cause people to panic while trying to answer questions or follow instructions given by a dispatcher. To obtain precious information with the least user intervention and reduced stress on the user, there is a need for the dispatcher to have a better control and understanding of the condition or situation at the other end. The increasing growth of smartphones embedded with camera, speaker phone, GPS, microphone and various other sensors, extends their usage from merely making calls to life saving gadgets during critical situations. By integrating these sensor rich smartphones and the rapidly growing VoIP technology, we propose a VoIP based Next Generation 9-1-1 (NG9-1-1) system for remote media control. Specifically, we use Session Initiation Protocol (SIP) in the implementation of the system using a mobile and a PC client. The proposed system on the mobile client accounted for less than 25% of CPU utilization even with video transmission. The average network utilization was about 10 and 72 kbps for audio and video, respectively. With these encouraging results, we believe the proposed remote media control system will facilitate information acquisition and decision making in emergency situations.

Keywords Media control · Remote · SIP · VoIP · Voice quality · Image transmission · PSAP · NG9-1-1

V. Chandrasekaran · R. Dantu (✉) · K. P. Subbu
University of North Texas, Denton, TX, USA
e-mail: rdantu@unt.edu

V. Chandrasekaran
e-mail: vc0063@unt.edu

K. P. Subbu
e-mail: kp0124@unt.edu

1 Introduction

Effective information exchange is the key for prompt and competent response from a dispatcher during a 9-1-1 situation. In most cases people panic and feel handicapped to perform simple instructions or handle devices. Meanwhile, dispatchers constantly face a "60-s dilemma"; i.e., when they need to make complicated but important decisions, whether to dispatch emergency services and if so, what services to dispatch, since in today's voice oriented 9-1-1 calls, the dispatcher first collects critical information (e.g., location, call-back number) from the caller and assesses the situation.

The advent of new technologies in telecommunication has led to revolutionary changes in the design and functionalities of mobile handheld devices such as PDAs, smartphones, mobile phones, and tablet PCs. Smartphones for instance, can be used to measure and transmit vital signs [38] of a person such as heart rate, respiratory rate and blood pressure during critical situations. Also with increasing demands for IP technology due to its low cost and greater benefits over traditional cellular networks/PSTNs, Voice over IP (VoIP) has been under rapid development over the past few years. With IP technology, any kind of service provided for a network terminal can be offered in a mobile phone.

In this work, we attempt to move one step towards providing a better service than the traditional cellular network by introducing *remote media control (RMC) during emergency situations*. Similar to Next Generation multimedia systems [28], this paper demonstrates a new IP based communication system model for Next Generation 9-1-1 [15, 16] (NG9-1-1) calls using smartphones, where the dispatcher can obtain control over the multimedia elements in the phone from a remote location. The system consists of a mobile and PC client communicating with each other over SIP. The mobile client is a smartphone embedded with sensors such as microphone, speakerphone, camera etc whereas the PC client is a VoIP phone. Both the mobile and PC clients connect to the same VoIP server for initialization. Owing to the communication requirements, the system has been modified to work under simplex or duplex mode for some of the features in the system. We will discuss about these features in detail in the later sections.

The paper is organized as follows: Section 2 explains the motivation for the proposed system. Section 3 describes the testbed infrastructure, the mobile, PC clients, and the message exchange between them. Section 4 discusses a SIP-Assisted FTP image transfer followed by an analysis of the impact of media features such as image resolution, file size, brightness, contrast, sharpness and camera exposure on image quality. In Section 5, the performance of the system is evaluated by analyzing the network and CPU utilization of the mobile client. This section also discusses call quality and voice quality analysis of the system. Section 6 provides a brief discussion about the security issues in our VoIP based remote media control system. Section 7 presents some existing work in this domain and finally Section 8 concludes the paper throwing some light on the future work.

2 Need for media control

The objective of a remote media control system is to facilitate the call taker at a public safety answering point (PSAP), to connect with the caller's device and execute operations to control media elements like the camera, speaker phone, and microphone. Figure 1 shows some of the media and sensor elements that can be controlled in the phone over the network.

As seen from Fig. 1, the smartphone embedded with different sensors as shown, can be used to communicate with an IP enabled video phone at the call taker's end. The call taker obtains every possible information regarding the scene or can remotely connect with the phone and control the media elements and other sensors.

Motivation A mobile phone is capable of transferring audio data across a network; whereas a video phone can be used to visualize the actual situation. Due to bandwidth limitations in the traditional cellular network, transmitting video might cause huge lags, delay and dropped calls. However, in the IP network, bandwidth is no longer a hurdle even with the ability to stream high definition (HD) videos on mobile IP networks. Despite the availability of huge bandwidth for the network and dedicated routes, all 9-1-1 emergency situations are dealt only through audio calls. This clearly indicates the under utilization of present technology.

Also Emergency Dispatch services (9-1-1) are an important part of modern society. New telecommunication technologies using Internet have made it possible to use multimedia services for 9-1-1. The new services can use images, video and text transmissions in addition to the traditional voice transmission. This new technology is also turning mobile phones with several embedded sensors into more than just personal communicating devices. These sensors can be used for some initial health diagnosis. For example, the embedded accelerometer, can be used to measure the breathing rate of a person with about 90% accuracy [38], or it can be used to guide a person to give proper CPR in case of emergency. The authors have used different sensors in a mobile phone for measuring heart rate, blood pressure, respiratory rate and oxygen saturation with accuracies of about 90% [38]. The operators handling the 9-1-1 calls use standard protocols, called Dispatch Protocols, to answer the calls for help. The protocols guide the operator about what questions to ask and what actions to take during a given emergency. The existing dispatch protocols assume a voice only call for 9-1-1, but with the deployment of NG9-1-1, the protocols can be modified to make these services more effective.

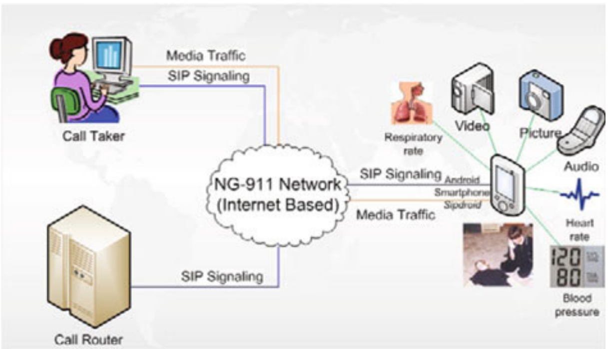


Fig. 1 System design of remote media control where the call taker can remotely control the camera, microphone, speaker phone and also obtain the victim's heart rate, respiratory rate, blood pressure with the aid of the user or a by-passer

The remote media control system thereby integrates VoIP and NG9-1-1 into a single protocol level in a low powered processing unit such as a smartphone. We hereby highlight the novelty:

– Developing and integrating a remote media control system for NG9-1-1.
– Analyzing and evaluating end-end performance of RMC on upcoming Android based devices.
– Revamping emergency dispatch protocols using the proposed remote media control system.

3 Testbed infrastructure

In this section, we first describe the proposed system consisting of a mobile (caller) and PC client (receiver). We then describe the message exchange and control message transfer process between the clients.

3.1 Mobile client

The mobile client runs on Android based HTC Hero, with a 528 MHz Qualcomm MSM7201A processor and 512 MB ROM. The Hero has a built-in GPS, WiFi and a 5-MP camera. Android operating system from Google provides easy to use application programming interfaces (API) for control of the various multimedia components in the phone. The client application was also tested on T-Mobile G1 and the Nexus One smartphones. The data communication was carried over directly between terminals and without routing through a proxy server. Figure 2 illustrates the layered architecture of the mobile client.

– The user interface (UI) lies in the application layer of the stack and it forms the foreground of the application providing the options and interface to make calls for the client. This layer gives controls to the user for operation.
– Due to the unavailability of SIP or RTP stack in the Android platform, the MjSIP stack has been used for the session and transport layers. MjSIP is a layered

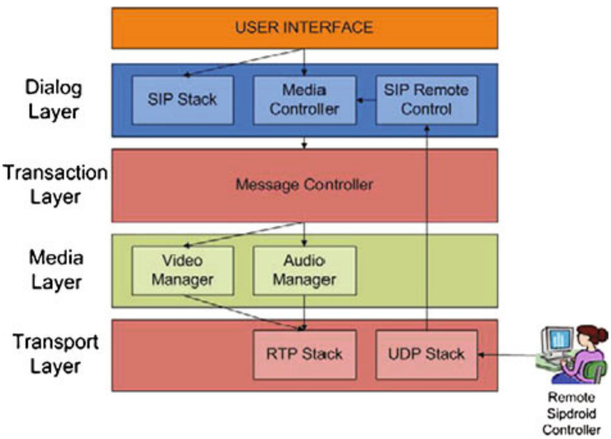


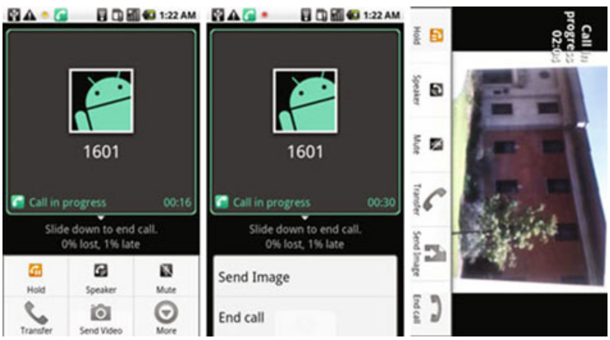
Fig. 2 Architecture of mobile client, where the remote control module is directly linked with the media controller to perform I/O operations with hardware

architecture which complies with the RFC 3261 [33]. It has three core layers: Transport, Transaction and Dialog.
- For the purpose of media control, a new media layer has been introduced to access the API level media components of the phone through its SDK.
- The dialog layer helps in managing the dialog objects and maps the transaction to the corresponding dialog objects.
- The transaction layer creates and manages transactions by mapping the messages to their transaction objects. Each transaction will maintain its transaction states, send and receive requests through the lower layers.
- The media control layer is introduced in the dialog layer, where control messages are received and parsed to give the control codes. The commands to be executed directly to the hardware during a communication are dispatched by this layer.

Figure 3 shows the user interface of the mobile client.

SipUA (SIP User Agent) is the application developed for IP based telephony service for the mobile client. It is a generic term used in the VoIP environment for clients. The user interface consists of a dialer and a window for changing different settings like the username, password, server address and the port to which it needs to connect. When the information is entered by the user, the user agent registers to the registrar server, which is notified by a small light in the notification area of the mobile phone. When a number is dialed, the telephony manager is called to pop up the default dialer interface for calling.

For creating a SIP/VoIP application for the Android platform, Google code has hosted an open source project—*Sipdroid* [19]. Since the Android platform does not have API's to support VoIP applications, we used an open source VoIP application as a base to build on the video, image transfer and media control services. The mobile client runs on WiFi and connects to a generic SIP server, (e.g., a SIP server in Fig. 4 and proxy server and a soft switch in Fig. 21). The video is transmitted as dynamic payload from the mobile to the PC client. SIP messages are passed through the MjSIP SIP/RTP stack which provides the methods for VoIP calls. The audio from the microphone and the packets received are ported by the Sipdroid to the ear piece through the media management. When the Back key is pressed, a disconnect message is sent and the call information is stored in the phone's call log.



(a) (b) (c)

Fig. 3 **a** Call options on phone. **b** Image transfer option. **c** Video transmission

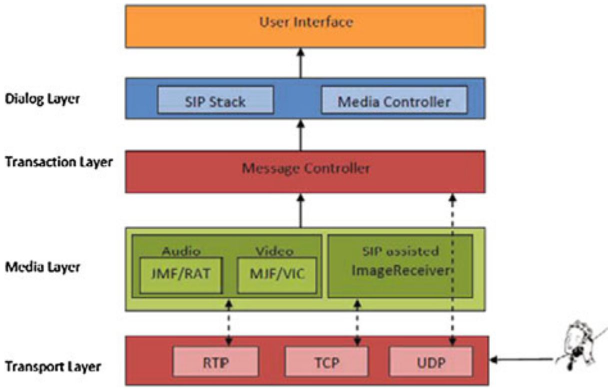


Fig. 4 The architecture consists of transport, media, transaction and dialog layers. Transport layer takes care of communication, media layer access the media components, transaction layer handles the messages and the dialog layer helps in managing the dialog objects

The information for the call log is formatted by the SipUA, by considering the user name of the number called/received as the display name and the entire SIP URL as the user number, along with the duration and direction of calling. On event of incoming calls, the calls are received by SIPUA which starts the ringer manager. When the user presses the connect button, the call is accepted and the RTP traffic starts to flow through.

3.2 PC client

The PC client was developed over pure Java and it extends the support from an open source SIP stack called *MjSIP* [27]. Figure 4 illustrates the PC client architecture with different layers.

MjSIP is a Java SIP/RTP stack which runs on any operating system. It makes use of the Java direct audio and video conferencing tool (VIC) [22] for receiving video from the mobile client. The PC client runs on both Linux and Windows operating system with JVM 1.5 or higher. The current setup uses a Core 2 duo 2.53 GHz machine running Ubuntu 9.10 with open-jdk 1.6. There is also support for (1) Java Media Framework (JMF) for adding audio, video and other time-based media into applications and applets built on Java technology and (2) Robust Audio Tool (RAT) for audio transmission and reception. Video reception was taken care by VIC, but since it did not support the reception of dynamic payload types, a JMF was used which partially supports H.263+ with minor tweaks to the codec information. Most of the connections were unidirectional for bandwidth conservation. Both the mobile and PC client use the same SIP/RTP stack for communication.

Figure 5 shows the user interface of the PC client.

The technical features of the mobile and PC client are compared in Table 1.

3.3 Message exchange between mobile and PC client

Figure 6 shows the message exchange architecture of the remote media control system, where the caller and callee establish a session using SIP to send control messages over the network to access the device.

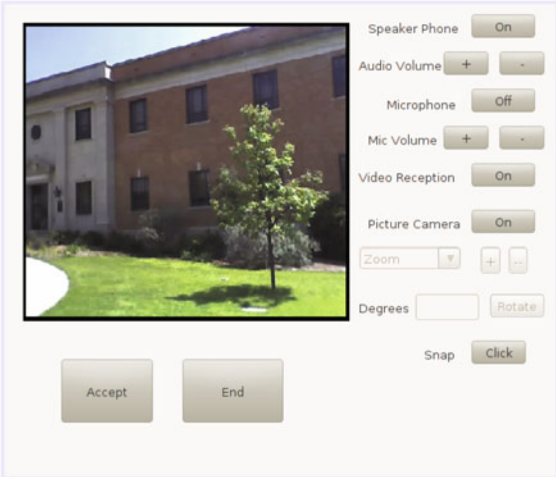


Fig. 5 Prototype of PC Client UI to be used at the dispatcher end

The initialization of SIP stack is accompanied by the system configuration, initialization of logging activities and other resources like, the SIP providers, listening agents. The initialization of the SIP session involves sending an invite message by the caller and negotiation of both the parties upon the media codecs to be used for the session. Once the negotiation is complete on accepting the call, the session is established and the data transfer starts between the ends. Since the default SIP stack from Sipdroid did not provide any extension for sending messages, we incorporated a SIP extension for instant messaging (RFC3428) [7] into the system.

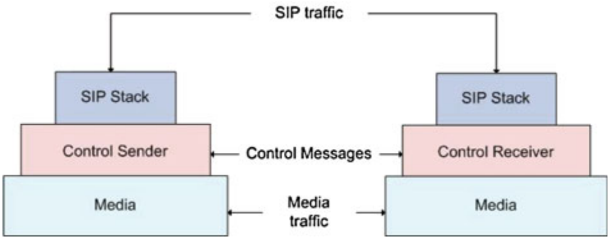
Control messages are sent from the PC client to perform an action in the mobile client. The control sender module interacts with the user interface and forms the control code for transmission. As soon as the control messages reach the mobile client, they are parsed and sent to the appropriate media sections for execution. An acknowledgment is sent back to the PC client on successful delivery of the message. A sample control message is given below.

```
MESSAGE im:user2@domain.com SIP/2.0
Via: SIP/2.0/UDP user1pc.domain.com
From: im:user1@domain.com
```

Table 1 Feature comparison of the mobile and PC clients

Media feature	Mobile client	PC client
Audio	Full duplex	Full duplex
Audio codec	G.711	G.711
Audio encoding	PCM A-law 8 Khz mono	PCM U-law 8 Khz mono
Audio bit rate	64 kbps	64 kbps
Video	Send only	Receive only
Video codec	H.263+	N/A
Video encoding	3GPP. 90 Khz (Android OS)	N/A
Video bit rate	Variable	N/A
Image	Send only	Receive only
Image encoding	JPEG	N/A

Fig. 6 Message exchange at each layer between mobile and PC client




```
To: im:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 <MESSAGE>
Content-Type: application/control
Content-Length: 18
```

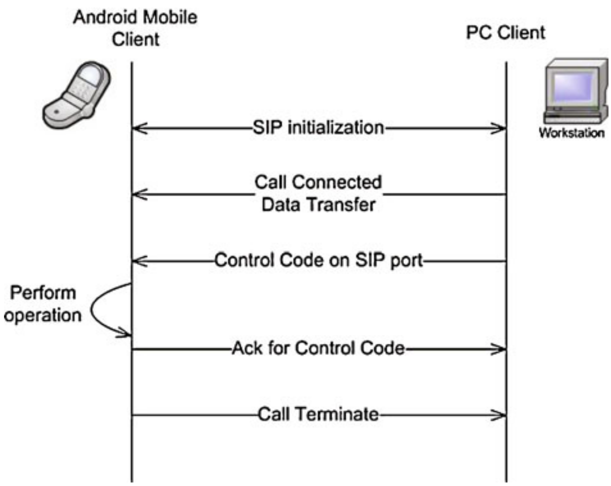
The above described control message transfer process is depicted in Fig. 7.

3.4 Need for image transfer

Bandwidth requirements have been a bottleneck for video over IP over a long time. However, with the availability of high speed data services, video transmission is no longer a huge problem, but there is still hesitation from users in using the service because of its disadvantages such as lack of privacy, bandwidth overhead in case of multiple video communications etc. Hence in this system, we opted to transmit images rather than video during SIP communication.

The proposed architecture is capable of moving back and forth between video and image, or using only audio depending on context. The system can have a module to enable or disable RMC during 9-1-1 or rights can be given to the user to disconnect RMC if they feel their private space is invaded. This is the primary reason for

Fig. 7 Flow of events between the two clients: Firstly the session is initialized during the SIP initialization followed by data transfer when the session is established. Upon session establishment, the media control module will be active to send control messages from the PC client. On receipt and successful execution of the control message, an acknowledgment is sent back from mobile client. Note that there is an intermediate device called a SIP proxy server, not shown here



choosing image transfer. Also when the comfortability factor is considered, sending images of a situation is preferred over streaming videos.

Moreover, the video resolution is very low as compared to a higher image resolution. For instance, during a crime situation, when an onlooker calls 9-1-1 to inform, he can either relay a low resolution video or a high resolution image which can be much more helpful for analyzing and extracting facial information.

In the next section, we present a detailed analysis of image transfer performed in this work.

4 Image transfer

Image transmission has its own advantages like resource conservation and user control over the scenes transmitted. To determine the optimal factors for image transmission, a series of experiments was conducted to evaluate factors like resolution, zoom level, and brightness. With the study of these parameters, *the conditions for optimal transmission of images to the call taker in an emergency calling situation can be determined.*

As described earlier in Section 3, the clients at both ends run on Java. One of the enhancements for the client is the ability to send pictures to the other end for conserving bandwidth. The mechanism for sending the images/files during a SIP call is described in the RFC4976 [20], where a file is sent by using a Message Session Relay Protocol (MSRP) relay server to initiate sessions for file transfer and help in the NAT traversal for file communication. MSRP protocol is a protocol for transmitting a series of related instant messages in the context of a session. We opted for FTP file transfer since a reliable communication is required between the end clients without much hassle. Using FTP, the data is pushed by the client to a centralized FTP server and read at the other end. This mechanism is simple to implement and at the same time, more robust to having a guaranteed file transfer.

There are some restrictions in the directional usage of image transfer of both the mobile and PC client. From Table 1 it can be inferred that both devices work only in half duplex mode. There are many options/parameters which can be controlled in the camera hardware in the image mode, unlike the video recorder discussed earlier. In the next section, we highlight the most important reasons for using image based transmission in this work.

4.1 Need for images

High resolution Currently video communication from a mobile phone supports only a low resolution, lacking details of objects. Hence, images can be transmitted when there is a need for high resolution feeds.

Low network and CPU The image transmission module requires very less CPU power and does not require continuous block of CPU resource for a single process as in the case of video transmission. Similarly, the bandwidth consumed by images is far less compared to video transmission. Hence with image transmissions, there will not be any loss of important conversations during limited network availability.

SIP-assisted FTP The image transmission module requires a reliable service to transport huge chunks of data. SIP merely being a signaling protocol, coordinates with the FTP service in the image transmission which is discussed in the next section. The following paragraph describes why direct FTP transfer was not selected in our system.

A PC Client with a FTP server hosted on it can be implemented. But the system is implemented over a network connecting multiple PC and mobile clients at the same time and not a single PC. The image in the FTP will serve as a record for the conversation and future reference pertaining to that call alone. This way, a log of information can be maintained at a centralized location. Implementing FTPs in individual machines are expensive and costing IP addresses in a country wide scale. Direct transfer to an end system was the primary design for this system using RFC 4976 but soon we discarded because of the need for central record maintenance, transmission affecting the network performance(if the caller is sending a 5 MB image with a 8.1 MP camera, the voice communication will be disrupted due to a direct connection with the system).

Having an offsite FTP reduced the network load with direct machine and also, having an FTP transmission on a different thread gave the flexibility to manage large image sizes to optimize if needed. Also, the images need a reliable delivery as that of SIP/control messages, so it needs to be a TCP connection for proper functioning. Direct transfer with SIP standards, cannot guarantee delivery. Having a central image database will have the ability for a readily available resource if additional help is required by the dispatcher to come to a decision, like a conference view.

4.2 SIP-assisted FTP image transfer

Figure 8 shows the basic architecture of the system and the components involved in image transfer. The file is transferred to the other end with the help of FTP via port 21 and the SIP signaling/initialization takes place through the registrar/SIP proxy. Once the SIP connections are established, the clients can communicate with each other directly. The control messages are transferred from the PC Client to the mobile client through the instant messaging protocol.

Figure 9 represents a small module stripped from the Sipdroid application for sending images to the server from the mobile phone. It basically contains three major components: File Namer, FTP Sender and the camera module. Whenever the user wants to send the image to the call taker, they can start sending by clicking the menu option of the application, otherwise, the call taker can remotely turn the camera ON and send them. The File Namer module generates a random filename for the image file that is to be created and conveyed to the destination/receiving side. The random file names provide a certain level of privacy from attackers and a mechanism of authentication since the file name will be visible only to the people involved in the communication.

The FTP sender module creates a client side socket to start a file transfer to the server. The FTP sender uses standard authentication mechanisms to open the FTP socket. The camera module includes a picture capturing operation along with features for controlling the image factors like auto focus, brightness, contrast, and sharpness. Most of these features are embedded in the camera module.

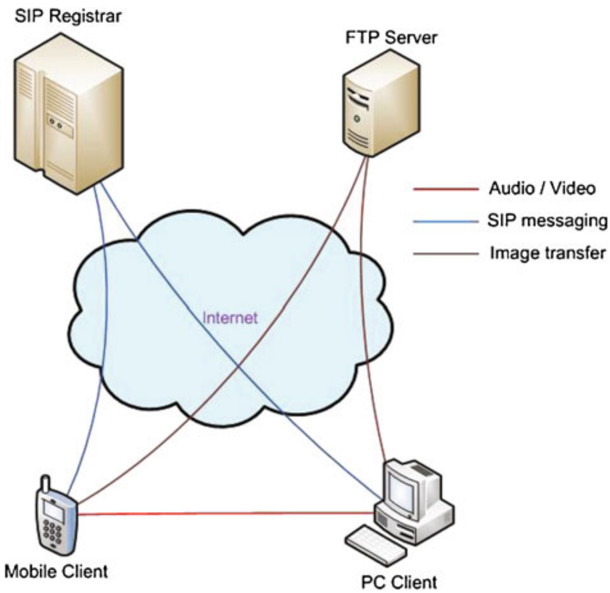


Fig. 8 Architecture of image transfer mechanism: FTP server stores the images sent by mobile client and the PC client reads them with the file name sent through the SIP messages

4.3 Event flow for SIP-assisted FTP transfer

The image module uses the idea of MSRP relay servers but without handling all the negotiation to accept/reject the transfer. First the image transfer mode starts with the establishment of voice connection between the two SIP clients. Once the call is established, the image transfer module is enabled and when the module is started, the preview is displayed on the user screen. Images can be captured either by the user or remotely by the click of a button. Figure 10 depicts the sequence of events for a complete image transfer. When an image is captured, the filename is generated for the image and saved in the mobile phone. The FTP uploader establishes a connection with FTP server after the authentication mechanism. Once the authentication is completed, the image is transmitted to the FTP server and deleted at the mobile

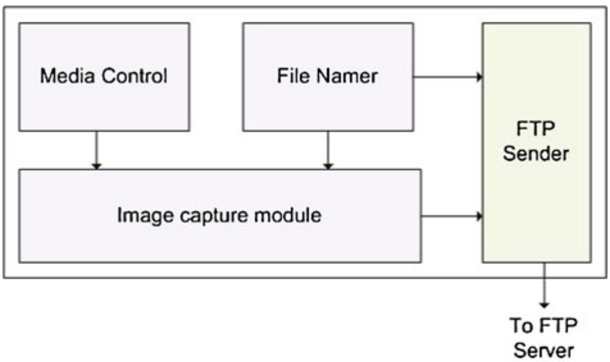


Fig. 9 Image transmission module: Media control enables the image capture module. When the image is taken, the File namer generates a random sequence of name for the file and uploads to the ftp server

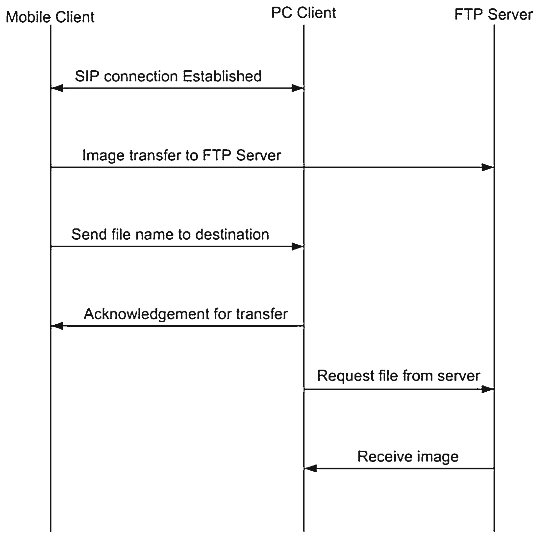


Fig. 10 Flow of event for image transfer

phone. As soon as the mobile client receives the notification for file completed, the name of the file is transmitted across the SIP clients using the instant messaging protocol. The PC client which knows the filename can directly request the FTP server and download the image file after establishing an authenticated session. Thus the image is completely transmitted to the other end. Table 2 lists some of the features that could be controlled remotely.

4.4 Impact of media features on image quality

Image quality is an important feature when there is a need for image analysis and in particular while sending images from a smartphone. The mobile device has limited CPU and RAM. Hence it is required to select the right resolution, bandwidth for accurate interpretation of the scene.

The size of the image is directly related to the quality of an image. For an image of the same resolution and fixed scene, the quality of the image will increase with increase in file size for JPEG images [21], since increase in quality means increase in the amount of spectral information present in a pixel. When the number of pixels per square inch increases, the data volume increases resulting in increase in the size of image.

Table 2 Range of values that can be set by from remote by media control

Feature	Range
Zoom	1–40
Brightness	0–10
Contrast	0–10
Sharpness	0–30
Saturation	0–10
Rotation	0–360

Surendar and Ellis [9] analyzed the trade off between image size and the quality of the image. In particular they use jpeg transcoding for reducing the image size. They have defined the notion of a quality aware transcoding that explicitly trades off image quality for reductions in image size. They characterized a sample of typical images available on the Internet with respect to their original size and quality.

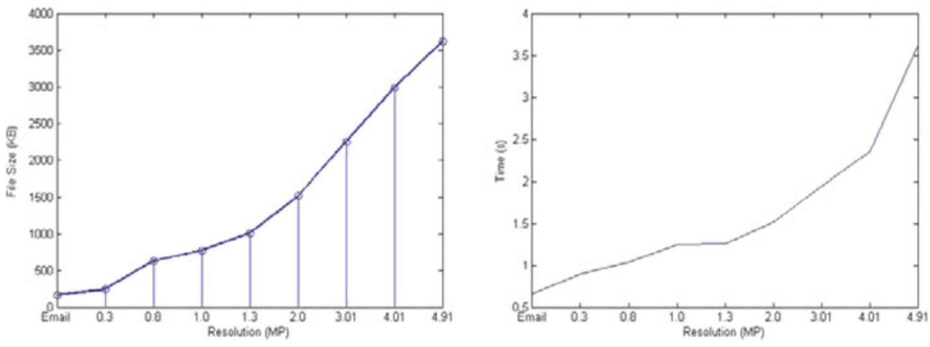
In the following sections analyze the impact of size on the quality of the image.

4.4.1 Perceived quality to image resolution

In this experiment, the relationship between file size and image resolution was studied. The test was conducted by attaching the mobile phone to a fixed stand and varying the resolution given in number of mega pixels of the camera (0.3 MP to a maximum of 5 MP). From Fig. 11a, it can be observed that *file size and the resolution are linearly related*. A continuation of the first experiment was to identify the time taken for the image to be uploaded in the FTP server. This mainly depends on the available bandwidth and the link speed for the system. For our experiment, a 54 Mbps link speed was used for the mobile phone over Wi-Fi, which resulted in a near zero latency for the transmission. From Fig. 11b, it can be observed that the transmission time increases with increase in the image resolution. This means, *the file size increases when the transmission time of the system increases*. It can also be observed from the graph that the transmission time increases gradually until a certain size, then followed by rapid increase. This is because of the buffer size used in the transmission.

4.4.2 Perceived quality to brightness, contrast and sharpness

Some experiments were conducted to identify the factors that could possibly affect the quality of the image upon control. In all the experiments, the mobile phone was fixed and stationary. There were no changes to the scene except, the variable factor that was considered for testing. All images were taken at the email size of 512 × 384 for the best performance in terms of image transmission. Now the increase in zoom of the camera with image quality is considered. From Fig. 12a, it can be seen that the *file size of the image decreases as the zoom factor increases*. This infers that the quality of the image decreases on zooming.



(a) (b)

Fig. 11 **a** File size vs. image resolution. **b** Transmission time vs. image resolution

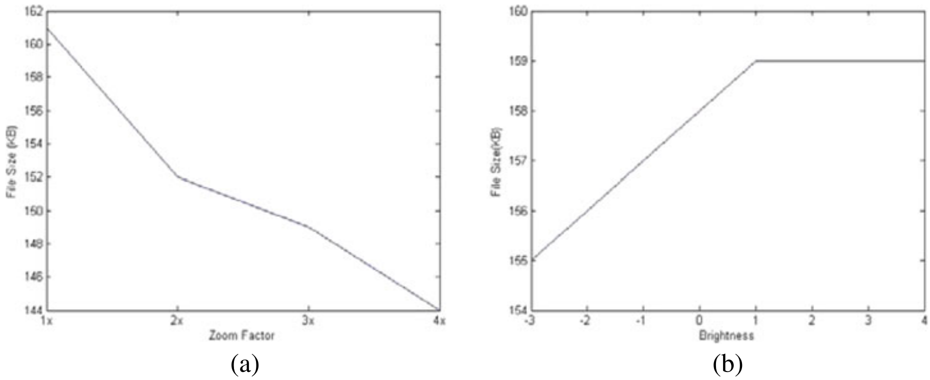


Fig. 12 **a** File size vs. zoom factor. **b** File size vs. brightness

The brightness parameter of the camera was varied, keeping all the other factors constant. From Fig. 12b, it can be observed that the image quality is degraded in low intensities and increases gradually with increase in the image brightness. It can also be seen that, the quality remains constant after a certain level of brightness. This shows that the *camera works better in medium and bright lighting conditions than the low ones*.

Another factor, contrast of an image is the difference in visual properties that makes an object distinguishable from other objects and the background. It brings about many changes to the perception of the image by an individual. For observing the quality of the image, its contrast was varied between −3 and 4. From Fig. 13a, *it can be concluded that the image quality degraded when the contrast was high or low*. However there existed only one optimal contrast value for maximum quality.

Sharpness of an image contributes to the demarcation of an object from that to the other objects in the image. Blurriness increases when the image is not sharp and tends to overlap pixels on the boundaries of other objects. Based on the observation from Fig. 13b, it can be seen that *the quality of the image increased as the sharpness*

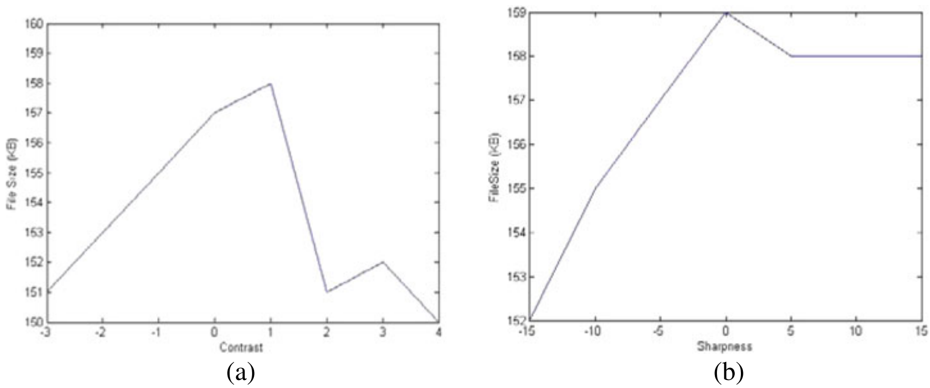


Fig. 13 **a** File size vs. contrast. **b** File size vs. sharpness

Table 3 File size under different lighting conditions

Light condition	File size (KB)
Dark (no light source)	29.2
Indoor natural light	67
Indoor low light	72.4
Indoor bright light	118
Indoor on light source	132
Outdoor low light	169
Outdoor medium light	197
Outdoor bright light	202

increased and remained constant above a certain line. This is because, the sharpness cannot impact an image after a certain limit and the increase in the sharpness beyond a threshold will result in the increase of every pixel detail of the image, adding up noise as a whole.

4.4.3 Perceived quality to camera exposure

9-1-1 situations can happen in places such as office, home, on the road, parks, movie theater, night clubs, etc. Every place has different lighting situation, depending on the time of the day, season, electrical equipments and so on. The camera's exposure to light plays a major part in the capture quality which in turn helps in better analysis of video or images at the dispatcher end. Hence in this section, we present perceived quality to camera exposure based on some of the generic everyday lighting conditions under natural and artificial light sources.

The quality of the picture depends on the type of light source and the amount of light exposure it receives. A simple experiment was conducted to identify the best light source for the mobile phone (HTC Hero).

Based on the results tabulated in Table 3, it can be seen that the file size/quality is more in outdoor lighting conditions than indoors. Also, there is a better performance of the camera in the presence of brighter sources. The tests were conducted in different places like complete darkness, indoors with a single incandescent light source and a natural light source from the windows and outdoor conditions. Hence it can be concluded that *the optimal quality can be obtained in outdoor environments under adequate lighting conditions*.

5 Performance evaluation

In this section, we analyze the bandwidth usage of the system and the CPU utilization of the phone at various instances.

5.1 Testbed

The test setup comprised of the mobile and the PC client registered under the same registrar server which is one that accepts register requests and places the information in those requests into the location service for the domain it handles. The mobile client was running the utilization monitoring services in the background, constantly reading the file and computing the required result from the statistics file. The PC client was replaced with IP hard phones which are VoIP enabled phones with a traditional keypad and slot for connecting network cable and capable of supporting dynamic payload data to test the performance under different hardware equipments. Figure 14 shows the test bed setup.

For evaluating the system, separate routines were developed for monitoring the network and CPU usage of the mobile phone. The routines run as services in the background, continuously recording data. Although the Android tool kit provides a benchmark tool to measure the overall performance of the system, we created system services to monitor every individual process in the phone for better clarity of the results. Figure 15 shows the basic architecture of the utilization module.

5.2 Network utilization of mobile client

Once the application was installed, data from the files was recorded by the monitoring service and stored in the user given location on the SD card. The CPU utilization monitor was designed to collect only the CPU usage of the Sipdroid and all its child processes. As soon as the mobile client registered to the server creating a process, its utilization values were recorded to the file. In this experiment, the total network rate at any given instance of time was computed using (1).

$$Bitrate = \frac{BitsOut_t - BitsOut_{(t-sample\ time)}}{sampletime} \tag{1}$$

where t is any time instant and sample time is the interval between two data recordings. Figure 16 shows the bit rate when the application is idle i.e., no media transfer across terminals. The small amount of bits sent is due to the keep alive mechanism in the client, to probe the server at constant intervals of time. It can be observed that, there is almost negligible amount of data transfer when the client is *idle* and the small amount of bit rate is due to the transfer of around 225 bytes of register message.

The experiment was extended to compute the bit rate of the audio. Since a static PCM audio was used, data transfer occurred in the absence of a voice signal. On the onset of the 30th second, the speaker was turned on to detect any changes in bit rate.

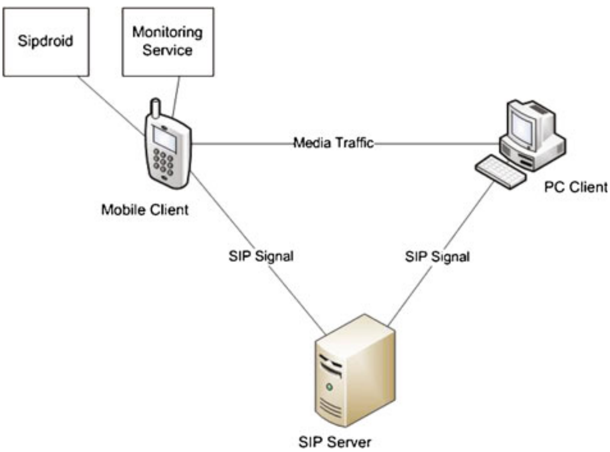


Fig. 14 Test setup consists the PC and mobile SIP client registered through the same SIP server under a local wireless network. The mobile phone runs the monitoring service in parallel to the client application

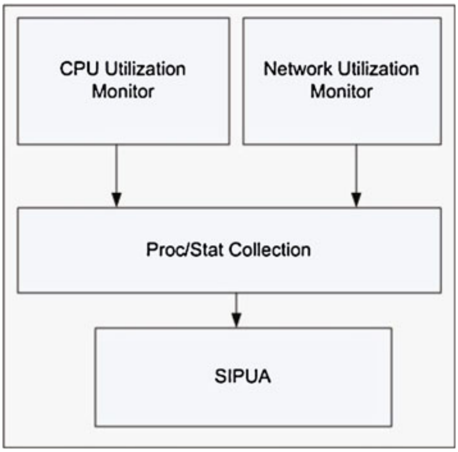


Fig. 15 Architecture of utilization monitor: Android system collects statistics about processes. Utilization monitor parses the collected information to retrieve data for desired process

The results showed that there was no influence over the bit rate with the change in the microphone gain. From Fig. 17, it can be concluded that there is no change in the bit rate unless the codec is changed in the system.

Next, separate experiments were performed by transmitting static and moving video. In the first case, the phone was placed in a stationary position and made to transmit a static video whereas in the second case, the camera was moving along with its bearer and the speaker phone turned on for audio. In both cases, a variable bit rate codec was used with the packet size varying between 240 and 1,458 bits depending on the current frame size. Figure 18 depicts the results.

Comparing Fig. 18b with Fig. 18a, it can be seen that the frequency of change in bit rate is much lower in the static picture. All the measurements were taken at a probing interval of 2 s, i.e., the service probes the file every 2 s for collecting utilization values. The results can be continuous if the sampling interval is small.

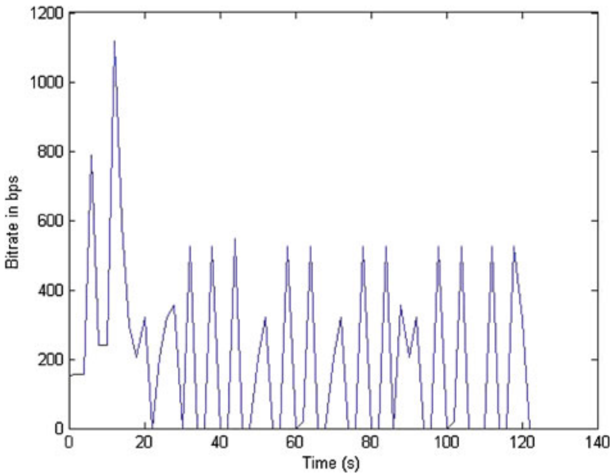
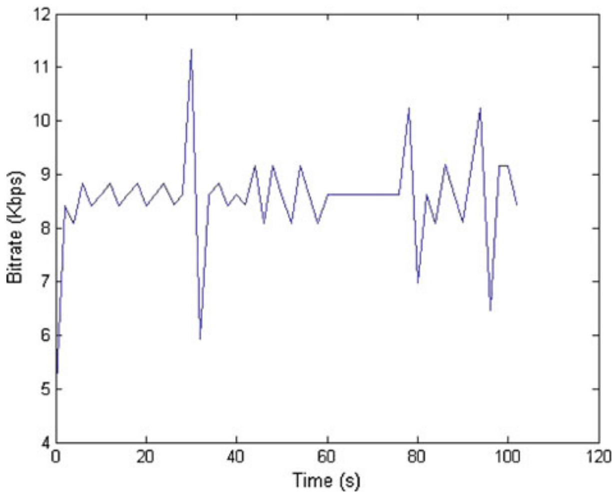


Fig. 16 Network utilization with client is idle

Fig. 17 Network utilization over a voice call from client



5.3 CPU utilization of mobile client

This experiment attempts to find the amount of CPU used by the *SIPUA* process, which is the SIP user agent for the mobile. First the CPU utilization was performed with the phone in idle state. From Fig. 19a it can be observed that the spikes in the CPU utilization ranged between 0 to 25%. This is because the SIPUA initiates a keep alive mechanism in regular intervals of time to describe its presence to the server. This timely probing of the server takes up some of the CPU resources given by the spikes in data.

The most power consuming peripherals of the mobile phone are the display, camera and the speaker phone. An experiment was conducted to identify the load over the CPU when the camera is turned ON. It was possible to obtain a resolution of a sample for every 8 s, even though the monitoring was set to a sampling rate of

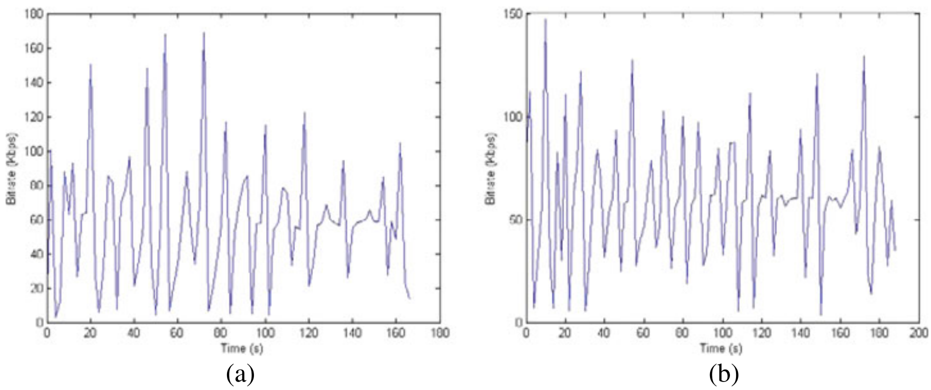


Fig. 18 **a** Network utilization for a video call on static position shows low frequency of bit rate changes over time. **b** Network utilization over a video call under constant motion of the scene shows high frequency in bitrate changes over time

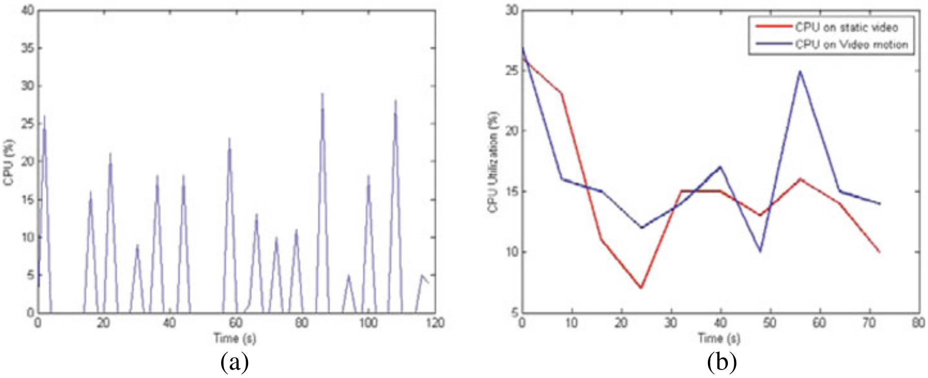


Fig. 19 CPU utilization when application is **a** idle and **b** camera turned ON

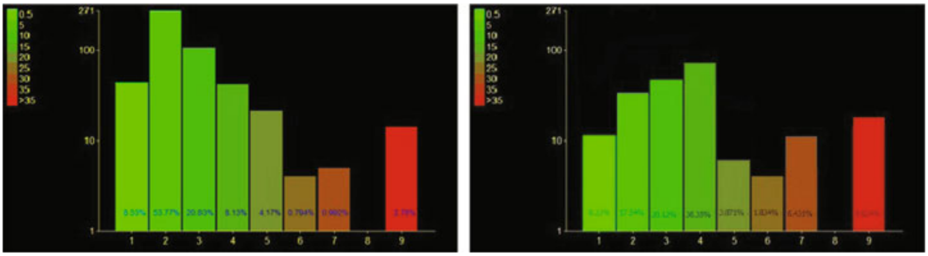
2 s. From Fig. 19b it can be concluded that although the video consumes a maximum 26 % of CPU while transmitting video at CIF (352 × 288) resolution, it only takes up *an average of 12–18% of CPU* all the time. It can also be seen that there is only a minimal impact on the CPU usage by video taken while walking and staying still.

5.4 Call quality analysis

For this experiment, a commercial tool 'Hammer Call Analyzer' [13] was used to monitor the audio quality during a call. The hammer call analyzer is an application-aware analysis tool that uses a passive approach to monitor and capture network traffic. The analyzer has many features such as media analysis, VoIP decoder, multi stage call flow display, etc. When codecs are used to compress audio for bandwidth requirement, the audio quality is described by the Mean Opinion Score (MOS) which is a parameter that is used to indicate the perceived quality of the media after compression. The scale of MOS values range from 1 (bad) to 5 (best). The tool predicts the human rating behavior and assigns the value for MOS. The estimation of MOS depends on a prime component called the R-Factor. The tool follows ITU E-model [3] in calculating R-Factor which considers end-to-end delay, echoes, side-tones, loudness and other factors along with speech quality.

5.4.1 Stream Quality Signature (SQS)

This is a unique display that shows the frequency and distribution of inter-packet arrival variation during a call. The distribution is shown over nine time bins with static ranges. The bins are color-coded and arranged on the x-axis in ascending order. The number of packets that fit each bin is shown on the Y-axis using a logarithmic scale. For each bin, the display shows the percentage of total packets contained in the bin. The percentage shown inside the bin is the number of packets with an SQS score in the range depicted by the legend of Fig. 20a. With greater number of packets in the first three bins, the quality of the audio is good. Figure 20a was obtained for the audio played in a loop by a music player without video transmission with normal microphone gain. The same experiment was repeated with the video transmission to the other end and the microphone gain at maximum. The results showed the number



(a) (b)

Fig. 20 **a** SQS metric without video transmission: most packets fall into 1–3 bins which states high audio quality. **b** SQS metric with video transmission: the audio packet inter-arrival time increases and falls in 4–9 bins, which depicts poor quality of audio than **a**

of packets in the first three bins with video transmission was very low as in Fig. 20b when compared with Fig. 20a. Hence it can be concluded that *the audio quality gets worse because of video transmission and interferes with the packet transmission in the mobile phone.*

Table 4 shows some of the useful metrics for calculating the R-Factor. The audio quality is determined based on the R-Factor where the value 100 being the best, 70 being minimal quality for telephony conversation and 0 being the worst. In the case of an audio stream with normal microphone gain i.e., with the speaker phones OFF, value of R-factor obtained was 81, whereas the R-Factor was 62 for the audio stream with the microphone gain at maximum because, with the speaker phone ON, the phone could be away from the person and not close to ear. The proximity is varied and thus the microphone gain has to be counteracted to listen to the person's voice. Based on the metrics, the MOS was calculated to be 3.875 for microphone gain at normal and 3.04 for the gain at maximum. A MOS score of 5 indicates the best audio quality. Thus it can be concluded that *the audio quality was good with normal gain and speaker phone turned off.*

5.4.2 Voice quality under different microphone modes

The mobile phone audio operates in three different modes namely routing via the ear piece, the speaker phone and through the hands free kit. The microphone is built in a way to accustom to these three modes for maximum performance. In the ear

Table 4 Comparison of microphone quality during voice transmission with normal and maximum gain

Metric	Speaker phone OFF	Speaker phone ON
Payload type	PCMA	PCMA
Received packets	506	612
Lost packets	0	1
Out of sequence packets	0	0
Duped packets	0	0
Jitter	60	94
Maximum jitter	177	242
R-Factor	81	62
Mean opinion score	3.875	3.045

Table 5 Audio quality under different audio modes

Mode	Jitter (s)	R-factor	MOS
Ear piece	0.063032	76	3.69
Hands free	0.059344	85	4.00
Speaker phone	0.058206	60	2.96

piece mode, the microphone gain will be low to capture the audio, spoken close to the mouth piece. When the audio is routed to the speaker phone, the gain of the microphone is increased to receive audio from signals of a certain distance. The mobile phone enters the hand free mode as soon as the kit is plugged in. The hands free kit contains a dedicated microphone and does not use the built-in microphone.

A small experiment was conducted to find out the best mode for good audio quality. A tool called 'OmniPeek' [40] from Wildcard was used to analyze the call quality by capturing the packets on the fly. The application calculates the MOS, R-factor, delay and jitter for the audio and presents a qualitative value. For experimental data in the three modes, the results are tabulated in Table 5. Based on the results, it can be observed that *the hands free mode gives the best quality*, due to its elimination of most of the background noise and the echo. Also, speaking in the ear piece had considerably lower quality when compared with the hands free kit. The quality of audio was the lowest in the speaker phone mode because most of the background noise was captured along with the speakers' voice. The microphone in the Android was too sensitive to capture even noises from 6 to 10 ft in a quite environment. In addition to this, the speaker phone gave a feedback to the microphone, making it to echo for the receiver. On a general case, the jitter in the overall system was high, which degraded the performance of the system. Reducing the jitter and processing the noise would yield a good reception in audio quality.

6 Security in RMC

Security issues in VoIP are unique and, in most cases, quite complex. As stated in [11], VoIP protocols were not designed with security as a primary concern like for instance, SIP is unaware of media misuse. It is an application-layer control protocol that establishes, modifies, or terminates user sessions. It uses two main security mechanisms: end-to-end and hop-by-hop [11]. For end-to-end protection, hypertext transfer protocol (HTTP) digest provides authentication. SIP body encryption utilizes the secure multipurpose Internet mail extension. In a hop-by-hop mechanism, encryption is supported, and the SDP conveys the SSL keys for media encryption [11]. A high level VoIP security architecture proposed by [11] is depicted below.

A VoIP based network consists of end-user equipments, network components and gateway PSTN networks. Relating Fig. 21 to Fig. 1, we can see that the end-user equipment provides an interface for users to communicate with other end users. Equipments such as hard phones, smartphones etc fall into this category. The security of such end-user components depends on how they are installed. Rarely, however, does the equipment have security features built in, making them vulnerable to exploitable flaws. Since VoIP normally uses the existing IP network, it inherits its vulnerabilities. Each network component including routers, switches, and firewalls

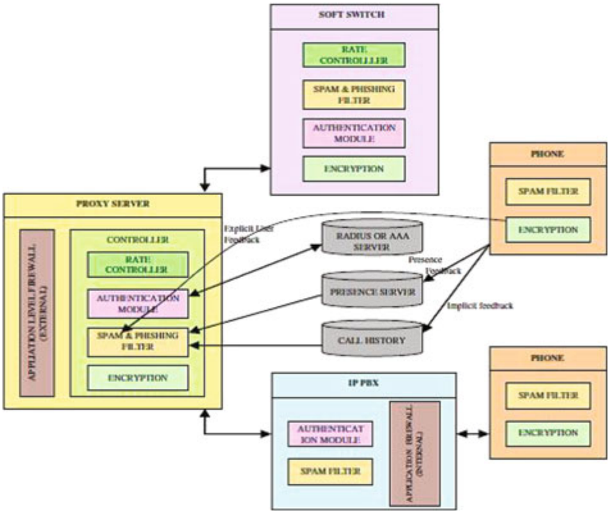


Fig. 21 High level VoIP security architecture

has its own security concerns. However, intermediate devices such as proxy servers and gateways are not shown in Fig. 1, but it is essential to handle security issues in these devices for an end-to-end security.

Depending on the type of attack such as Denial of Service (DoS), Spam and Phishing, Eavesdropping etc., mechanisms such as

- Application-level firewall (external), Rate controller, Authentication module can be performed at the proxy server
- Authentication module, Application-level, firewall (internal) at the IP PBX
- Authentication module, Application-level, firewall (internal) at the end-user equipment.

This high-level security architecture can be used to address the security concerns to make the VoIP infrastructure more secure and robust.

7 Related work

Traditional emergency 9-1-1 (E 9-1-1) systems, since their very inception in the 1970s, have been of immense help to communities. However, these systems are based on outdated technologies [29, 30] and purely voice communication methods. With the recent advent of multimedia communication systems and services based on Internet protocol (IP) [32], it is now possible to revitalize [6, 37, 39] the national emergency response infrastructure with multimedia information. The new technologies (e.g., end-to-end voice and video over IP networking and third-generation (3G) cellular networks, WiFi/WiMAX technology) facilitate drastic improvements in 9-1-1 services and response times with the use of text messaging, transmission of video clips captured by the cell phones, and automatic crash notifications [25].

Communicating visually is an important factor in the proposed remote media control. The call taker has to be very efficient in communicating what information he/she requires to analyze the situation. We highlight some work related to this.

Kraut et al. [24] discuss the importance of visual communication in collaborative tasks. They perform experiments such as collaborative bicycle repair task and analyze the performance of the participants. Their work provides support for remote collaborative tasks like the user and the call taker jointly working to provide, obtain and act upon a emergency situation.

Networked control systems (NCS) comprise of the system to be controlled, sensors, actuators, and controllers whose operation is coordinated through some form of communication network. The remote media control system is a kind of NCS with the smartphone based sensors, and system for remotely controlling these sensors. Hence some challenges highlighted in existing work can be related to the proposed work. Baillieul et al. [2] survey NCS and discuss fundamental issues involved in designing successful NCSs. They provide emphasis on distributed control systems where sensor data can be processed locally, remote control units can be coordinated etc. Authors in [12] propose a hybrid automaton based model to analyze the queue dynamics of a VoIP network using differential equations and other feedback control models. They conclude that under network congestion, the packet sending rate depends upon packet loss ratio. The work in [14] uses uncertainty threshold principle and a kalman filter model to analyze the congestion control and obtain the rate of dropped packets in a network controlled system. The authors in [1] use game theory concepts and apply mechanism design to defend congested calls in a VoIP network. Their solution solved the network congestion problem by choosing the most cost efficient users.

Henning et al. [35] discuss how Internet telephony can be used to provide emergency call ("9-1-1" or "1-1-2") services, enhancements required to allow prioritized access to communications resources during emergency-induced network congestion and describe an Internet based event notification mechanism to alert communities to emergencies. Song et al. [36] discuss some technical challenges in integrating instant messaging and SMS based system with NG 9-1-1 system and provide a working prototype. Kim et al. [23] describe an enhanced version of a prototype for emergency services in VoIP. They propose Cisco Discovery protocol for determining physical location of a person and also a mechanism for routing emergency calls.

The Call routing capability use case described in the NG 9-1-1 system description [17] provides information required for a 9-1-1 call or other emergency initiated event to be sent through a system. Also RFC 5031 [34] describes a uniform resource name (URN) for emergency and other context dependent services. As per this RFC, there is no national coordination or call center for 9-1-1 in the United States. Hence the RFC allows to define such global, well-known services, while distributing the actual implementation across a large number of service-providing entities. Henning in [18] also describes how the Session Initiation Protocol can be used to set up emergency services for Internet telephony. There are also some availability requirements provided in [17] such as SR-SYSTA-04, which state that the system shall implement redundant infrastructure to support intelligent routing of calls/data in the event of outage conditions. This can also be termed as Continuity of Operations (COOP), which is a system's ability to prevent critical system failures (e.g., via component redundancy) and to seamlessly conduct updates and repairs.

Bergstrand et al. [4] present their findings on how mobile live video capabilities can improve information sharing and situation awareness in emergency response work. They show how live video can be broadcasted from the incident site to a web-application using mobile terminals with 3G capabilities. Ranjan et al. [31] discuss the relationships between physical movements and visual information using a camera in different modes. They discuss camera placement, orientation and control available to a helper during emergencies in remote situations. Some image stabilization mechanisms in camera have been proposed [10, 26]. These mechanisms could be useful in compensating for the blurriness that could occur when the due to shaking of the mobile phone camera.

Since the work proposed in this paper is based on a typical SIP based VoIP network, there would be a need to analyzed the VoIP traffic in a network. We highlight some of the models developed in the literature for this purpose. Work in [41] uses point processes and their superposition, time series autocorrelations and power spectra, long-range dependence, random effects and hierarchical modeling, bootstrapping, and robust estimation to model the VoIP traffic and present the validation of the models for multiplexed processes. Similarly work in [5] uses hyper-exponential distribution for approximation of audio/voice streams duration to perform a statistical analysis of VoIP packet streams. Their results showed that the precise VoIP source model can be based on the five-state Markov process. Authors in [8] model and analyze voice traffic over VoIP networks and conclude that the voice data is log normally distributed. There are several other works related to statistical analysis of the traffic data over VoIP network. The proposed VoIP based NG9-1-1 system can very well utilize the models in the above mentioned works to analyze multimedia data.

Most importantly, there has not been a model developed by integrating all of the individual components discussed in the literature. The remote media control system proposed in this work for NG 9-1-1 integrated different technologies such as VoIP, SIP, and remote network control protocols into a single entity using a smartphone.

8 Conclusions

This paper proposed a new method for remotely controlling the media elements of the mobile phone over a SIP call. The performance evaluation of the test bed showed that a considerable amount of CPU is free all the time for other processes. The CPU utilization for the proposed system reached a maximum of 25% only when using the camera to transmit video. The network utilization of the system maintained an average bit rate of 72 Kbps during a video transmission, whereas the audio transmission utilized a constant 8 Kbps. This proposed system could be integrated with the vital sign diagnostic modules for transmitting realtime physical status of a subject.

By integrating existing pieces into a single protocol level in a low powered processing unit such as a mobile phone, we showed that our system is capable of complying with existing VoIP standards and can also be modified to fit any existing standard. The major changes need to be done only with the device levels and not the infrastructure level for the communication. We believe this remote media control can bring about a revolution in the usage of Internet telephony and serve as a tool in an emergency condition.

Acknowledgement This work is partially supported by the National Science Foundation under grants CNS-0751205 and CNS-0821736.

References

1. Azimi-Sadjadi B (2003) Stability of networked control systems in the presence of packet losses. In: Proceedings of IEEE conference on decision and control
2. Baillieul J, Antsaklis PJ (2007) Control and communication challenges in networked real-time systems. In: Proceedings of the IEEE
3. Bergstra J, Middelburg C (2003) ITU-T recommendation G.107: the E-model, a computational model for use in transmission planning. Tech rep
4. Bergstrand F, Landgren J (2009) Information sharing using live video in emergency response work. In: Proceedings of the 6th international, ISCRAM conference
5. Biernacki A (2006) VoIP source model based on the hyperexponential distribution. In: PWASET
6. Caesar MC, Ghosal D, Katz RH (2002) Resource management for IP telephony networks. In: 10th international workshop on quality of service (IWQoS 2002)
7. Campbell B, Rosenberg J, Youjian L (2002) Session initiation protocol (SIP) extension for instant messaging. RFC 3428, Internet Engineering Task Force. http://rfc-editor.org/rfc/rfc3429.txt
8. Casilari E, Montes H, Sandoval F (2002) Modelling of voice traffic over IP networks
9. Chandra S, Ellis CS (1999) JPEG compression metric as a quality aware image transcoding. In: 2nd symposium on Internet technologies and systems, USENIX
10. Chiu CW, Chao PCP, Wu DY (2007) Optimal design of magnetically actuated optical image stabilizer mechanism for cameras in mobile phones via genetic algorithm. IEEE Trans Magn 43:2582–2584
11. Dantu R, Fahmy S, Schulzrinne H, Cangussul J (2009) Issues and challenges in securing VoIP. In: Computer security
12. De Cicco L, Mascolo S (2010) A mathematical model of the Skype VoIP congestion control algorithm. IEEE Trans Automat Contr 55:790–795
13. Empirix (2009) Hammer call analyzer. http://www.empirix.com/products/hammer_call_analyzer.asp
14. Gupta P, Shmatikov V (2007) Security analysis of voice-over-IP protocols. In: 20th IEEE computer security foundations symposium
15. http://www.nena.org/ng911-project. Accessed 2009
16. http://www.cs.columbia.edu/irt/project/ng911/. Accessed 2009
17. http://www.its.dot.gov/ng911/pubs/NG911_FINAL_Requirements_v2.htm. Accessed 2009
18. http://tools.ietf.org/html/draft-schulzrinne-sipping-sos-04. Accessed 2009
19. I-P-tel GmbH (2009) Sipdroid-free SIP/VoIP client for android. http://code.google.com/p/sipdroid
20. Jennings C, Mahy R, Roach A (2007) Relay extensions for the message session relay protocol (MSRP). RFC 4976, Internet Engineering Task Force. http://tools.ietf.org/html/rfc4976
21. JPG Quality Comparison Chart (2007) http://en.wikipedia.org/wiki/Wikipedia:Featured_picture_candidates/Image:Quality_comaprison_jpg_vs_saveforweb.jpg
22. Kevin S, Neil R, Yves L (1996) MBONE: multicasting tomorrow's Internet. Wiley, New York
23. Kim JY, Song W, Schulzrinne H (2006) An enhanced VoIP emergency services prototype. In: Information systems for crisis response and management
24. Kraut RE, Fussell SR, Siegel J (2003) Visual information as a conversational resource in collaborative physical tasks. In: Human–computer interaction
25. Langle L, Dantu R (2002) Are you a safe driver. In: IEEE international conference on embedded and pervasive systems
26. Lin YJ, Shen CT, Lin CC, Yen HC (2009) A fast image fusion algorithm for image stabilization on hand-held consumer electronics. In: ICCE '09, digest of technical papers international conference on consumer electronics
27. Luca V (2005) MjSIP-mini tutorial. http://moya.iyard.org/pub/Technology/JavaSIP/mjsip_mini_tutorial_01.pdf
28. Nahrstedt K, Yang Z, Wu W, Arefin A, Rivas R (2011), Next generation session management for 3D teleimmersive interactive environments. Multimedia Tools Appl (Springer) 51:593–623
29. NSF Workshop Report (2002) Network research testbeds. Tech. rep
30. NSF Workshop Report (2005) Overcoming barriers to disruptive innovation in networking. Tech. rep
31. Ranjan A, Birnholtz JP, Balakrishnan R (2006) An exploratory analysis of partner action and camera control in a video-mediated collaborative task. In: Proceedings of the ACM conference of computer-supported cooperative work
32. Reynolds B, Ghosal D (2002) STEM: secure telephony enabled middlebox. IEEE Commun Mag (Special issue on security in telecommunication networks)

33. Rosenberg J, Henning S, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E (2002) SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force. http://www.rfc-editor.org/rfc/rfc3261.txt
34. Schulzrinne H (2008) A uniform resource name (URN) for emergency and other well-known services. In: RFC 5031
35. Schulzrinne H, Arabshian K (2002) Providing emergency services in Internet telephony. In: IEEE Internet computing
36. Song W, Kim JY, Schulzrinne H, Boni P, Armstrong M (2009) Using IM and SMS for emergency text communications. In: IPTComm'09
37. The Emulab Testbed (2009) http://www.emulab.net/
38. Vikram C (2010) Measuring vital signs using smartphones. Dissertation, University of North Texas
39. White B, Lepreau J, Stoller L, Ricci R, Guruprasad S, Newbold M, Hibler M, Barb C, Joglekar A (2002) An integrated experimental environment for distributed systems and networks. In: Proceedings of the 5th symposium on operating systems design and implementation
40. Wild Packets (2009) OmniPeak: network analyzer. http://www.wildpackets.com/products/network_analysis_and_monitoring/omnipeek_network_analyzer/voip_monitoring
41. Xi B, Chen H, Cleveland WS, Telkamp T (2010) Statistical analysis and modeling of Internet VoIP traffic for network engineering. Electron J Stat 4:58–116

Vikram Chandrasekaran is pursuing his master's degree in computer science and engineering at the University of North Texas. His research interests include Wireless Networks, Computer Networks and Information Security, VoIP, Cryptography, Social Network analysis and Multimedia.

Ram Dantu is an associate professor in computer science and engineering at the University of North Texas. He is currently a visiting professor at the Massachusetts Institute of Technology. His primary research areas are Secure Protocols, Wireless Security, Sensor Networks, Forensics, Mobile Computing, and Multimedia.

Kalyan Pathapati Subbu is a PhD candidate in computer science and engineering at the University of North Texas. His primary research areas are Mobile Computing, Wireless Networks and Multimedia.