8

A Framework for VoIP Testability and Functionality Extension with Interactive Content Delivery

Janez Stergar, Janez Klanjšek and Sibila Vadlja
University of Maribor, Faculty of Electrical Engineering and Computer Science
Slovenia

1. Introduction

The telephony as we know it has changed when VoIP emerged in 2004. In the third quarter of 2000 the second generation of IP enabled phones came to the market with full QoS capability. Performance was quadrupled and the basic voice functionality was extended with the addition of a large screen with HTTP/XML driven capability. That was a major market breakthrough extending VoIP phones with the possibilities of interactive content delivery.

Classic telephony systems are being rapidly replaced by IP Telephony (IPT) in corporate and home environments. Especially IPT has gained wide acceptance in the industry offering new ways to exchange information with rich media communication capabilities. Therefore stationary telephony is migrating into the internet. Voice over packet switching networks can significantly reduce the per-minute cost, resulting in reduced long distance costs. Therefore many dial-around-calling schemes already relay on VoIP backbones to transfer voice. There is even more potential extending VoIP with interactive content using tailored applications for the end user.

Successful deployment of any new technology solution requires thorough understanding of the function of various components involved and the interaction among them. The architects and engineers who are tasked with implementing the IPT solution must ensure that the proposed architecture meets all the requirements and is also scalable in the future (Kaza & Asadullah, 2005).

Therefore an IP Telephony framework will be introduced with the goal to demonstrate typical limitations and an IP Telephony delivery platform for interactive content applications. The VoIP framework can deliver IP Telephony services and internet access at the same time based on one core device. We implemented VoIP technology with content delivery support using a common unified communication device in combination with typical networking devices such as a router and switch. The platform is capable of carrying voice data and multimedia traffic with QoS management. The system consists of diverse IP phones with dissimilar capabilities and a PoE entry access switch (delivering power for the

IP phones). Also a router is used as a gateway with IP telephony services operation system to simulate the WAN network environment. The framework is designed for IP phone applications development and testing. Therefore a discussion will be included where critical implementation parameters in a real-time environment are evaluated and tested (jitter, frame dropping, priority queuing, etc.).

The extra value of IP telephony is the support for applications extending the voice capability of IP phones with interactive content delivery. These are the so called IP phone services. These enable representation of multimedia content from a server locally or anywhere in the Internet cloud. Applications are delivered from a server and rendered on IP phones using HTTP protocol. The deployment of a typical application regarding the XML capabilities will be discussed and the workflow of an application realization presented. Also limits for image streaming will be evaluated.

2. IP Telephony components

When deploying the IP Telephony solutions of almost any prospering IPT solutions provider on the market the key areas to emphasize are (Kaza & Asadullah, 2005):

- Network Infrastructure,
- Call Processing,
- Call Manager Directory Services,
- IP Telephony endpoints,
- Call Admission Control,
- Legacy Fax Messages,
- Media Resources and
- Applications.

2.1 Network infrastructure

Network infrastructure plays a key role in building multiservice networks e.g. Cisco AVVID. Integration of data and voice traffic puts strong requirements on packet loss, delay and jitter (variable delay of VoIP packets). LAN/WAN components with QoS mechanisms support are indispensable when designing IPT networks as well faster convergence in case of network failures to avoid destructive delay, jitter and frame dropping.

Voice traffic in addition to the existing data traffic increases the bandwidth a critical issue because of the high-speed LAN switching technologies availability. However, when transporting the voice traffic across the WANs, one has to ensure that adequate bandwidth is available to support the additional bandwidth required to transport voice calls. If the WAN links do not have adequate bandwidth their bandwidth has to be increased to support the additional voice traffic. After the bandwidth is assured, QoS mechanisms have to be properly configured on operating systems of LAN/WAN networking devices to prioritize voice traffic with adequately bandwidth allocation.

Network infrastructure plays a key role in building multiservice networks e.g. Cisco AVVID (Architecture for Voice, Video and Integrated Data) is the foundation of converged enterprise communication networks.

2.2 Call processing

The core of an IP Telephony solution is usually a software call manager. For Cisco devices this software component is called Cisco CallManager. The specially designed IOS software with embedded CallManager handles all the call-processing requests received from various clients in the IP Telephony network. For the Cisco IP Telephony AVVID solution CallManager software runs on a compatible router or the Microsoft Windows Server operating systems. Call manager is installed on the Cisco Media Convergence Server (MCS) for medium to large-scale networks but can be also operated from a router (CallManager Express) or a specific device for smaller unified networks e.g. the Cisco UC 500. The selection of the hardware platforms depends on the size of the network in which IP Telephony is going to be deployed, including its high-availability and performance requirements (typically 300-7500 devices per dedicated server for medium to large-scale systems). For large-scale systems a clusterization of servers is inevitable. Call Manager servers are grouped to form clusters to support more devices (IP phones, gateways, etc.). For the current Cisco CallManager version, a Call Manager cluster can have up to eight Call Manager servers running the call manager service.

2.3 Call Manager directory services

Call Manager stores system and device configurations in a Microsoft SQL database. The application scripts and the subsequent information are stored in a Lightweight Directory Access Protocol (LDAP) compliant directory the so-called DC Directory (DCD):

- User authentication and authorization
- Extension Mobility profiles
- Personal Assistant profiles
- Internationalization information
- Personal Address Book (PAB)
- Spoken name
- Fast dial
- Call Forward All information

The DCD process replicates the information among the members of the cluster. This process is similar to Microsoft SQL replication.

2.4 IP Telephony endpoints

In an IPT network, endpoints are the devices that accept or initiate a VoIP session. Typical endpoints that are used are: IP Phones, Soft Phones e.g. Cisco IP Communicator, Wireless IP Phones, Voice gateways which connect the IPT network to the PSTN or a PBX, Survivable Remote Site Telephony (SRST) which provides the fallback support for the IP phones that are connected behind a router running a suitable operating system software version that supports SRST and in a specific Cisco environment the so called Call Manager Express - CME which delivers key system functionality for small and midsize branch offices using Cisco IP stationary, wireless and software phones.

2.5 Call admission control

In VoIP networks, Call Admission Control (CAC) does the bandwidth management. CAC ensures that enough bandwidth is available before granting permission to a gateway for placing the call across the IP WAN. When deploying IPT solutions with multiple locations, there are two choices for implementing the CAC: Call Manager locations-based CAC and Gatekeeper CAC. Call Manager locations-based CAC is one mechanism to limit the calls sent across an IP WAN in a single Call Manager cluster deployment, whereas the Gatekeeper CAC provides call admission and call routing between the Call Manager clusters in distributed call processing deployments.

2.6 Legacy FAX messages

There is still a large portion of long-distance minutes based on legacy fax traffic. One of the most important functionalities in the transition to converged networks is therefore support for fax communications. As network implementations increasingly provide for e-mail attachments and web-downloadable documents, fax communication nonetheless is still a significant method of immediate document delivery worldwide. Three methods to transmit legacy fax traffic across the IP network are common: the Pass-Through mode, where the gateways do not distinguish a fax call from a voice call, the Cisco proprietary Fax Relay mode, where the gateways terminate the T.30 fax signalling and the plain old T.38 Fax Relay mode.

2.7 Media resources

The function of media resource devices is to mix the multiple streams into a single output stream, converting the data stream from one compression type to another, and so forth. The media resources can be hardware or software. The limitation of software media resources is that they can't combine the streams that use different compression techniques. Hardware media resources have the same features as software media resources with an additional advantage of mixing the streams that use different compression types. Characteristic media resources are conferencing, transcoding, and MoH (Music on Hold) which provides music or announcements when the users are put on hold.

2.8 Applications

There is a wide range of applications that can be deployed in an IPT network. These applications are optional, and their deployment adds more features and capabilities to the overall IPT network. Design and deployment of the applications, such as Customer Response Solution and IP Phone services is a very important topic for new converged IPT networks. Cisco offers many proprietary services e.g. IVR, IPCC Express, Cisco Unity, Cisco Emergency Responder, Cisco Conference Connection and so on. In the following we will try to emphasize the non-proprietary solutions and therefore focus on IP Phone Services.

3. IP Telephony deployment architectures

By using the Call Manager software it is possible to bypass the plain old PBX and replace it with IP Telephony over a next generation converged network. The Call Manager application software provides call-control functionality and, when used in conjunction with IP hardware/software phones, can provide PBX functionality in a distributed and scalable manner. The deployment solution models of Cisco IPT can be categorized into one of the following architectures (Kaza & Asadullah, 2005):

- Single-site deployment
- Centralized call processing with remote branches
- Distributed call-processing deployment
- Clustering over the IP WAN

Selection of the deployment model depends on implementation requirements, such as the size of the network, features, and availability of the WAN bandwidth.

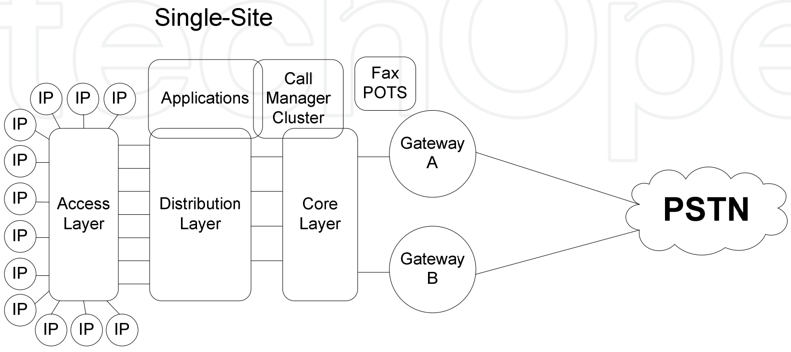


Fig. 1. Single - Site IPT Model

3.1 Single-site model

In this deployment model, Call Manager applications such as voice mail, IP-IVR, AutoAttendant (AA), Transcoding, and conferencing resources are located at the same physical location (Fig. 1).

All the IP phones are located within this single site. The PSTN is used to route the off-net calls.

3.2 Centralized call processing model

In this deployment model all the call processing is done at the central site. This is suitable for implementations in which the majority of the workforce is concentrated at a single site and small numbers of employees work at the remote branches (Fig. 2).

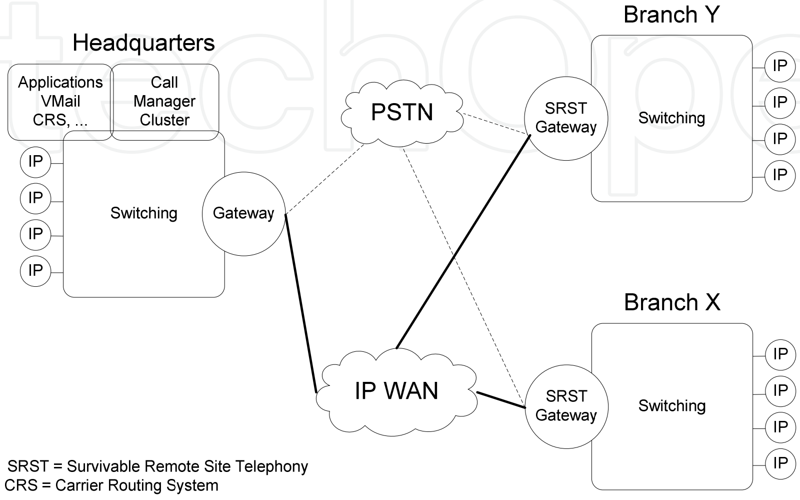


Fig. 2. Centralized Call-Processing Model

At each remote branch, SRST (Survivable Remote Site Telephony) routers ensure that call processing is preserved in case of WAN link failure. The voice traffic travels via the IP WAN and falls back to the PSTN if not enough bandwidth is available across the WAN link, by using the Automated Alternate Routing (AAR) feature available in the Call Manager software application.

This deployment model is cost effective and provides many benefits, such as a unified dial plan, less administrative overhead, and potential savings on communications costs as the remote branches calls use the IP WAN as first choice. The only limitation is that the remote sites will have limited features available in the case of a WAN failure.

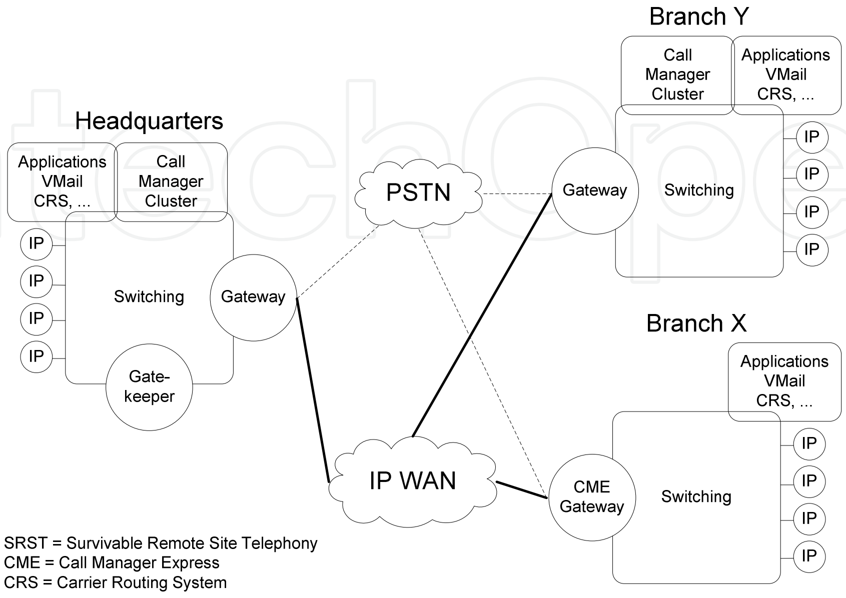
3.3 Distributed call processing model

In the distributed call-processing deployment architecture, Call Manager software and applications are located at each site. Device weights and dial plan weight calculations determine the number of IP phones supported at each site.

In the figure (Fig. 3) a distributed call-processing model is depicted in which headquarters and branch Y IP phones are served by separate Call Manager clusters and branch X is served by the Cisco CallManager Express (CME) feature that is enabled on the router. CME solution is suitable for a small branch.

3.4 Large scale architecture – Clustering over the IP WAN

The Cisco IPT solution allows organizations to build disaster recovery sites by separating the single Call Manager cluster across the WAN. Call Manager servers in a cluster update the configuration information via the Microsoft SQL replication process. To ensure successful SQL replication and propagation of other critical information in real time, the round-trip time (RTT) between any Call Manager servers in the cluster should not exceed 40 ms. Many other requirements have to be satisfied before selecting this deployment model.



SRST = Survivable Remote Site Telephony
CME = Call Manager Express
CRS = Carrier Routing System

Fig. 3. Distributed Call Processing Deployment

When using the clustering over the IP WAN deployment model, voice gateways, media resources, and voice mail have to be deployed locally at each site. Essential services such as DHCP, DNS, and TFTP that are critical for the functioning of IP phones and other IPT endpoints also require local implementation. This configuration avoids dependency on a single site for crucial resources. Clustering over the WAN can support two types of deployments:

- Local failover deployment model where each site contains a primary Call Manager subscriber and at least one backup subscriber. All the servers are part of the same Call Manager cluster.
- Remote failover deployment model where each site contains at least one primary Call Manager subscriber and might or might not have a backup subscriber. Branch X and branch Y as depicted on the previous models have only primary subscribers, and the backup subscriber is not located in each site.

4. The framework for VoIP testability and functionality extension

In this chapter we will introduce the educationally tailored IP Telephony platform based on the Centralized Call Processing Model (Fig. 4). The IPT platform is intended as a test framework for QoS parameters evaluation, configuration, critical examination of the VoIP network parameters as well as delivery and multimedia media applications platform for the Cisco IP Phones.

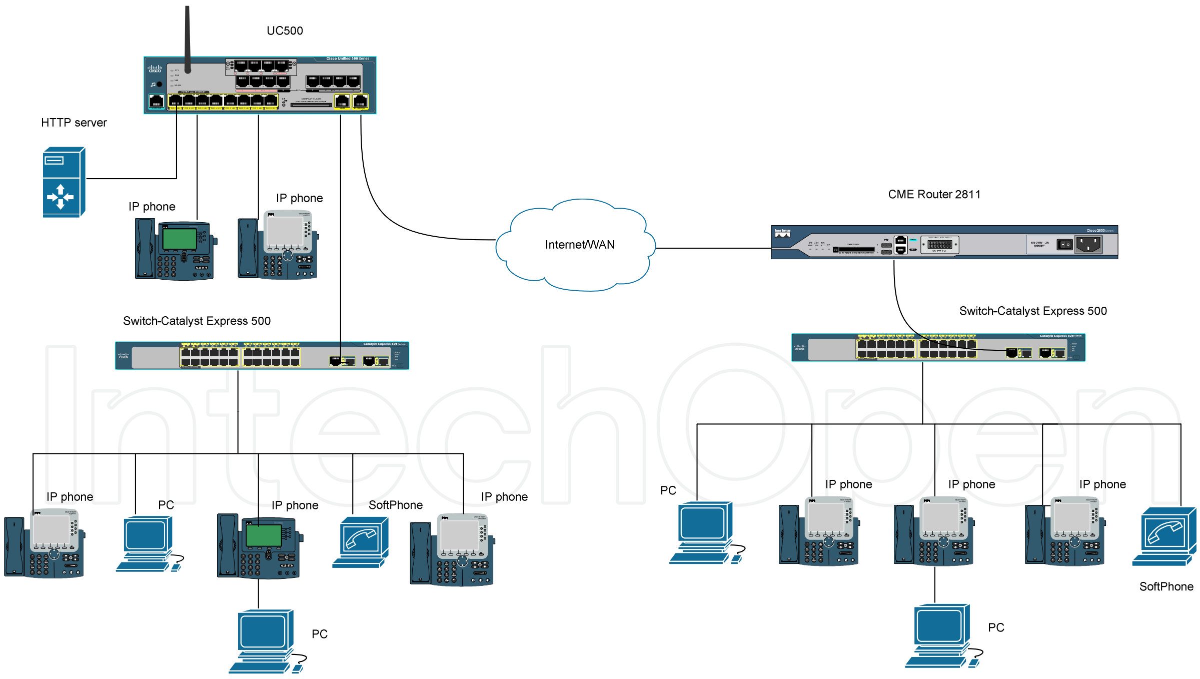


Fig. 4. The VoIP test framework

The introduced VoIP framework represents a development environment for IP phone applications using XML to add the extra value with additional interactive content on phones with touch screen display option. The architecture model is applicable in small and mid-sized business environments. It supports up to 100 IP phones and includes all the functionalities that small businesses need; voicemail, auto-attendant, e-mail integration, and call attainability functions. Voice mail provides a voice messaging system for cases when called persons are not available, auto-attendant provides voice narrated guide system and greeting if required. Call attainability functions support the user call seeking service (Cisco Systems, 2011).

The core of the system is a unified communication (UC) appliance and a VoIP services enabled router with a customized operating system (VoIP software extensions with CME). The UC appliance and router support the convergence of voice communication, data communication, mobile phone support and video (multimedia) support. Additionally the UC appliance offers a wireless module supporting connection of wireless IP phones. It runs special software for VoIP control, the so called CallManager. The call manager handles processes and routes the incoming calls comparable to a PBX system. It is actually an IP-PBX system. The UC appliance functions as a router and a switch for all connected devices e.g. IP phones, computers. The Call manager software offers centralized/distributed control of the calls and routes the calls to the intended users (Cisco Systems, 2011). In the presented platform a switch is used to connect and power IP phones in combination with a router. The installed Call manager software works as an IP-PBX and simulates two separate VoIP networks connected through a WAN via the H.323 trunk. The H.323 trunk encapsulates the calls appropriately. Actually our platform is build of two separate local VoIP networks. The ports that are used are all FastEthernet 100Mbps ports.

The IP phones represent the end devices of the presented network platform. These are entry to high-end IP phones intended for home and business environments. Some models have a HiRes color display and support up to eight call lines which can be configured with different numbers and speed dial functions. Supported are traditionally soft keys which are programmable with functionalities change based on the configuration. Templates can be used to apply the same configuration to multiple phones. All the configurations are centralized. The Call manager software is used for central point phone management with complete control over the IP phones in the system. The IP phones load the configuration settings using the Call manager software integrated trivial file transfer protocol (TFTP) server (Cisco Press, 2006).

Similar UC management software as on the router is implemented on the UC appliance. It has a graphical user interface and supports the control and creation of user IP telephony system. Users can be incorporated into the voicemail system and the creation of voice messages mailbox and e-mail notification of messages are also supported for every individual user mailbox. Thus users can access voice messages from anywhere (Au et al., 2005, Cisco Press, 2011c). The system is configured on network devices in console line in text mode with exception of the entry level access switch which is managed through a GUI. The call manager software takes care of network operation in VoIP network while unity express software takes care of the users that are connected to the system.

4.1 VoIP communication and protocols

All main VoIP protocols can be used and analyzed in our system. The data examination access to forwarding connections is implemented with port mappings. The network is flexible so it can support various Session initiation protocol (SIP) IP phones from different vendors as well as soft phones that run on computers. It can offer a SIP trunk or a H.323 trunk for WAN connection to a telephony service provider (Cisco Press, 2007; Hatting, S. et al., 2010; H.323 Implementation, 2011). Students can observe operations in VoIP network of SIP, H323 and other proprietary protocols (e.g. SKINNY protocol) with Wireshark network analyzer (Wireshark, 2011). For voice transfer and direct communication between two users Real time protocol (RTP) protocol is used.

IP phones and attached computers use Virtual local area networks (VLAN) and Dynamic host configuration protocol (DHCP). VLANs are used to provide basic security and Quality of service (QoS). Actually data traffic is separated from voice traffic applying two VLANs, thus voice traffic does not interfere with data traffic. Interference of the two kinds of traffic could cause degradation of voice communication service. With basic security implementation we mean that any computer in the network cannot see the IP phones on OSI L2 and compromise its identity. DHCP is used to automatically configure network parameters such as IP address, network masks, and DNS address for both the IP phones and computers connected to the network. That optimizes time necessary for adding new users and reduces administrators work - there is no need to manually enter device configuration. Every configuration for any device in the network can be done from a single point. That is one of the tasks performed by the Call Manager software which runs on UC appliance (Deel, D. & Nelson, M., 2004; Cisco Resources, 2011).

There is a trunk connection between the UC appliance and the router. A trunk is used to carry multiple calls or data transfers over a single Ethernet link. The educational platform supports SIP and H.323 trunks. We used a H.323 trunk. Using this kind of connection the UC appliance and router act as two dial peers. The right parameters have to be configured in the dial peer list to correctly route the incoming and outgoing calls. The dial peer list presents similar information to the Call Manager software devices as a routing table does to forwarding router. A dial plan is also implemented. It defines mapping of local phone numbers into global phone numbers and translation of local phone numbers into global phone numbers for and outgoing call and vice versa for an incoming call (similar to network address translation). The system can also be designed that the UC appliance has only one global telephone number which is used for registration at the telephony service provider. The UC appliance then routes calls appropriately inside the local IP telephony network (H.323 Implementation).

5. IP Phone services application development

Applications on stationery IP phones are an added value to IP telephony especially in the business environment. A phone actually transforms to a tool displaying business information, multimedia applications or entertainment applications that serve the user needs. IP network enables that functionality. Applications also called IP phone services are interactive services which relay on the IP phones keyboard or touch screen capability.

First we have to introduce how the IP phone actually invokes a service. The phone is drawn into a service by virtue of a URL attached to a button on the phone the so called services buttons. If the phone is equipped with a touch screen, the button functionalities are taken over with hot spot fields on the screen of the IP phone. The button assignment is done in one of the Cisco CallManager Administration screens. The services button by default is assigned a URL that points to a Web page on the CallManager server (GetServicesMenu.asp). It simply presents the user with a menu of services that have been configured for that particular phone. Each of the different menu items is connected to another URL.

It is important to remember that the Services Menu usually seen on the local CallManager system is only one of many possible ways to invoke services. The Cisco IP Phones can have a URL attached to the directories button and the messages button. Most system administrators will confine the use of those buttons to specialized services.

5.1 Basic concepts

Here we present the introduction to building blocks for development of applications on IP phones. The HTTP integrated client on the IP phone enables the capability to deliver content on the display of an IP phone. The content is gathered from a Web server with the HTTP protocol. It has to be emphasized that applications are not part of a call service but actually are data applications separated from VoIP communication. All they have in common with VoIP is the shared IP network infrastructure (Fig. 4). Applications are displayed on the IP phone on user demand. They are loaded from the Web servers on which they are resident. On the Web server applications are developed, executed and changed. The interface for communication between the Web server and the phone is the TCP/IP stack. Applications use TCP for reliable transfer which is invoked by the HTTP and the proprietary SKINNY protocol. The SKINNY protocol relays on top of the TCP stack intended for signalization and control of the IP phone with the IP phone central unit on CME (Call Manager Express). The connection with the central unit (CME, IP-PBX) is necessary because this is how the IP phone acquires its IP address. The logical address is necessary for the connectivity between the IP phone and the server. The SKINNY protocol uses the IP phone firmware for data link layer communication with the central unit. The firmware is also responsible for the initiation of the HTTP request calling the application on the server. Basically the data travels between the IP phone, IP PBX (which serves as a proxy) and the Web server (Deel, D. & Nelson, M., 2004).

Let us examine the data flow more in detail: The firmware on the IP phone initiates a HTTP request to the server (HTTP GET message). The request includes a URL (Uniform Resource Locator) containing the address of the server and a file, script or program ID resident on that server. Data that is requested is embedded in an XML object and sent to the Web server. The Web server responds and also embeds data in XML objects. Then the data is sent back to the IP phone with the HTTP protocol. The IP phone understands and is capable of parsing only supported XML objects which are intended for displaying the dynamic or static content (Deel, D. & Nelson, M., 2004, Cisco Developer Community – Resources, 2011).

5.1.1 Choosing a Web server and the programming language

There is an open choice of the Web server and Web programming languages. IP phone understands only XML and the applications are processed on the server. This is why we need to select a Web server and a compatible programming language. If we decide for a Microsoft IIS then the optimal programming languages to use are C# and ASP. If we decided for a JAVA server, the JAVA language is the most suitable. If we decided for the Apache, then scripting languages e.g. PHP, Perl, Javascript are the advised programming tools (Deel, D. & Nelson, M., 2004).

For the introduced test-bed we decided to use the Apache Web server with the PHP scripting language. PHP enables a programmer to implement much work with small amount of code making the writing of applications faster. In addition with implementation of MySQL database, one has the capability of saving data from IP phones into a database or displaying data from the database on the IP phones (Deel, D. & Nelson, M., 2004, Gilmore, W. J., 2010). That enables far broader spectra of applications that can be designed. Apache is the most popular Web server and it's also open source. It represents approximately 60% of all the active Web servers implemented on the internet (Apache Usage Statistics, 2011).

5.1.2 Web server usage options

Application can also communicate with distributed Web services on the internet from the local Web server. A Web server can request content from the server on the internet. That server responds with the requested data. The Web server then processes the received information and sends it in XML form to the IP phones. Examples of such applications are stock market information, weather information and even Google maps (Cisco Developer Community – Resources, 2011).

The Web server is the core unit of applications development but also involves many distributed processes. These processes can be database queries, other server queries or even a connection to other users and multimedia devices – the so called backend processing on the server. Only the simplest applications communicate solely with the development Web server (Deel, D. & Nelson, M., 2004).

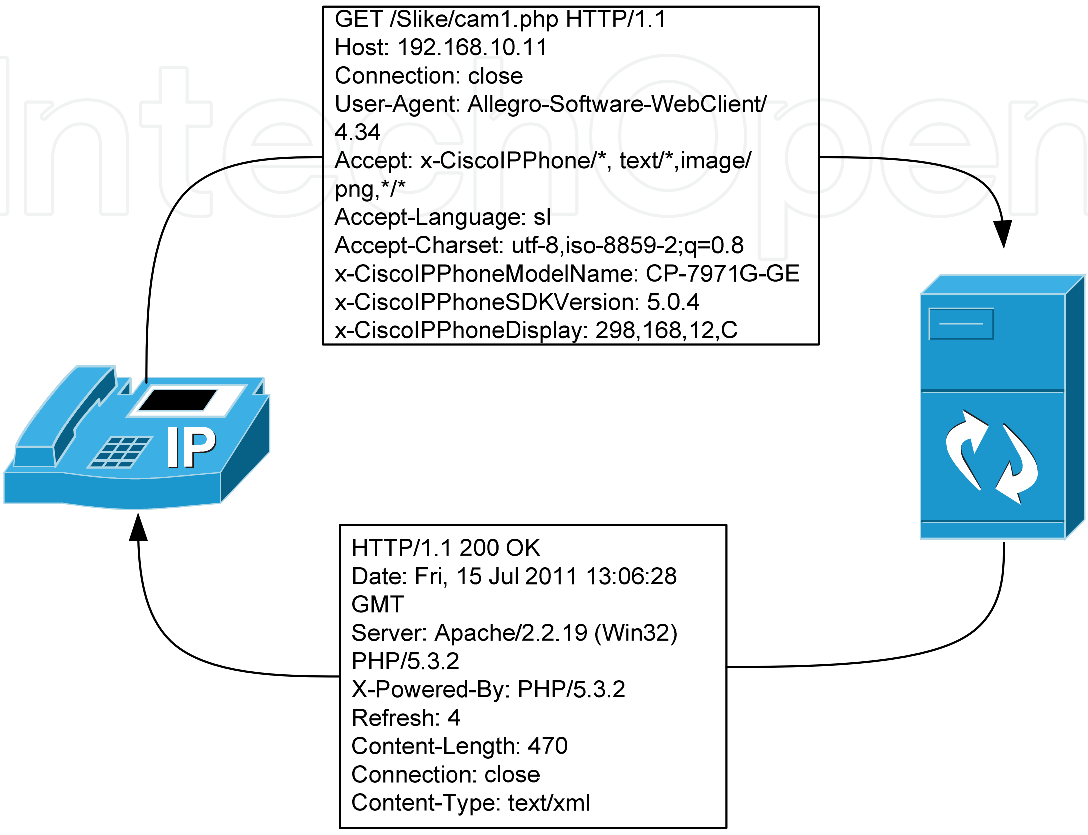
5.2 The IP phone and application server interaction

There are three major topics to emphasize in communications between the server and the IP phone: The HTTP Protocol, the customized XML language and the applications on the IP PBX.

5.2.1 The HTTP protocol

IP phones use a HTTP client, which communicates with the HTTP server. Nevertheless the IP phone does not operate the same as a Web browser, because it is not capable of processing demanding complex Web pages built with HTML. This is also the reason that it does not understand the HTML language. IP phone is limited because of its lack of memory and processor resources which is common for all embedded systems. The IP phone also has an embedded HTTP server for sending information about the configuration, firmware, name and status of the device. For more advanced functions it uses CGI, which enables external program to change the configuration on the IP phone. Both client and the server embedded in the IP phone use HTTP version 1.1 for communication with other entities in the network (Cisco Systems, 2011a).

The communication between the IP phone and the server is accomplished with HTTP messages. HTTP requests are sent by the phone. The request includes a header, a method, a body of request and status messages about the capabilities of the phone. HTTP response is sent back to the phone with similar content messages. The IP phone supports only a few from the huge variety of HTTP headers because it is not capable of processing them all. It would be too difficult to implement that variety into the embedded system. IP phone uses solely the HTTP GET method. This method has an URI identifier, which is a path to the server folder containing the data, a process which returns static or dynamic content as depicted in Fig. 5.



```
GET /Slike/cam1.php HTTP/1.1
Host: 192.168.10.11
Connection: close
User-Agent: Allegro-Software-WebClient/
4.34
Accept: x-CiscoIPPhone/*, text/*,image/
png,*/*
Accept-Language: sl
Accept-Charset: utf-8,iso-8859-2;q=0.8
x-CiscoIPPhoneModelName: CP-7971G-GE
x-CiscoIPPhoneSDKVersion: 5.0.4
x-CiscoIPPhoneDisplay: 298,168,12,C
```

```
HTTP/1.1 200 OK
Date: Fri, 15 Jul 2011 13:06:28
GMT
Server: Apache/2.2.19 (Win32)
PHP/5.3.2
X-Powered-By: PHP/5.3.2
Refresh: 4
Content-Length: 470
Connection: close
Content-Type: text/xml
```

Fig. 5. HTTP message exchange between the server and IP phone

We also have to mention the headers that the IP phone can read:

- *Content-type*: MIME type of data, so the phone knows what to process and parse. This has to be set to Text/Xml so the phone starts to parse XML objects and correctly displays them.
- *Refresh*: Page refresh in seconds. Enables the phone to refresh the application. Manually or automatically.
- *Expire*: Expiration of the requested URL. With this header one determines how long certain content is available. IP phone also has a history stack, which can store up to ten pages. With this header the history can also be erased.
- *Location*: Intended for redirection.
- *Set-Cookie*: Saved information about the session. When the phone uses the application again, the Web server can recognize that particular IP phone.
- *Accept*: Informs the Web server about the capabilities of the IP phone, its language and charset.

One can define or read header parameters in the Web programming language code (Deel, D. & Nelson, M., 2004).

5.2.2 The IP phone customized XML language

XML language is used to display the content on IP phones. It is a hierarchical language, it has one main element and many sub elements with different attributes. An XML object is predefined and IP phone proprietary. Inside of the XML object there is dynamic content which is processed on the server and sent to the phone via the IP network. The Web server does all the operational work related to the application. The contents of the application is packed into XML objects and sent to the phone which parses and displays the contents.

One can only use predefined proprietary XML objects, tags and attributes. Those predefined elements are the only elements the IP phone can parse. They are defined in the XML schema, which is integrated with the phone through its firmware. This schema is defined by the vendor (Cisco) and cannot be changed by the user. The schema is being changed and updated regularly by the vendor. New functionalities are added and new ways of checking the XML with phone firmware updates are possible. Rules and a dictionary of behaviours and display of the objects are defined in XML schema. Therefore an overall knowledge of XML for the development of applications on Cisco IP phones is not necessary. Only the rules and Cisco custom XML objects knowledge is needed. XML objects are an envelope for the content of applications (Web based programming language).

One is limited with the usage of objects with different phone models. Some models do not accept all the available objects; they have a limited scope of functionality (Cisco Systems, 2011a). The server must have the "text/xml" MIME extension enabled otherwise the IP phone will display just plain text. For the Web server files navigation URLs and URIs are used.

The IP phone also has soft keys with which we normally navigate through application. They use proprietary XML tags to describe their functionality. With the properly designed soft keys we enhance the user experience. We can set the order, function and the text they show. Every Cisco XML object also has its own default soft key defined which can be arbitrary altered. Internal URI can be used on soft keys. Those URIs use various functions already built in the phone, for example Dial or Transfer Call. External URI is actually a URL for a Web server. Soft keys are defined inside Cisco XML objects.

All Cisco XML objects, elements, tags and attributes, their definition and usage, are described in references (Deel, D. & Nelson, 2004; Cisco Systems, 2011a). It is not our intention to describe them explicitly as that would exceed the topics of the article.

5.2.3 Enabling applications on the IP PBX

To make the application available to the IP phones we have to configure the IP PBX correctly. IP-PBX serves as a proxy between the IP phone and the server and also sends all parameters to the phone. The phone cannot be directly configured. IP-PBX is managed with the proprietary Call manager software package. All one has to do is determine the path to the Web server where the application resides. URL services parameter has to be changed in the Call manager configuration. The Call manager then sends this parameter to the IP phone which then knows where to find the developed applications. IP-PBX is also important because IP phones and its users are registered to it. By registering they get an IP address and consecutively a connection to the IP network. With the determination of the URL one is not limited to the local network (the introduced network schema) but can choose any Web server on the internet that hosts appropriate applications. The only condition is that the IP-PBX is connected to the internet. That means that one can share the locally resident applications with any Cisco IP-PBX and Cisco IP phones on the internet.

In the URL services parameter one can just write a single URL. Therefore the phone has limited access to that particular application. To solve this problem we created a menu with Cisco XML tag IPPhoneMenu where we defined many URLs that point to different applications on different locations. Therefore the user can choose between different applications (UC500.com, 2011; Cisco Systems, 2011).

5.3 The service architecture

As already mentioned Cisco IP phone service is not running on the IP phone locally despite it could seem so. The phone is actually reduced to an I/O device, being used to render screens of data and retrieve input from the user. The main processing usually takes place on a Web server where the applications reside. The location of such a server can be locally but can be also anywhere in the Internet. Most IP Phone services are centralized on a Web server, but generally involve many distributed processes. The "locally positioned" Web server receives the request from the IP phone and generally does some local parsing. Nevertheless it relies on additional components, programs, and processes to do the back-end work (such as database access, scripting, security, and interfacing with scripts and CGI). Only the simplest services do all their processing on just one server.

5.4 Developing a phone service

To develop IP phone services for a Cisco IP Telephony environment, one needs access to the proprietary CallManager application on the suitable networking device (usually a Cisco router or a unified communication device). With XML over HTTP, one can provide targeted functionality designed specifically for the device, while still keeping with the desire to maintain but expand the core phone functionality. XML provides the horsepower through Cisco IP Phone objects and tags to push the content. Additionally XML is user-readable, machine-readable, and low-cost, plus it expands the use and value of the phone without diminishing the phone functionality. Instead it enhances the phone to the extent that it can be considered munch more than a legacy phone. The key to the concept is XML, a user- and machine-readable way of structuring and encapsulating data; this simply provides the data to the phone to use with its existing menu structure.

An application can send simple text to the IP phone, but when developing real services for the IP phone system, the odds are that XML objects have to be created that the phone understands. The extra flexibility and power we get with these objects make them almost mandatory. XML enables the design engineer to provide data to the native functionality of the device. With tailored sophisticated XML objects including the CiscoIPPhoneGraphicMenu and CiscoIPPhoneImage enhanced services in IP phones have been enabled.

The main purpose of the introduced VoIP framework (Fig. 4) is the development of IP phone applications which add extra value and usability. A user does not need a desktop computer to browse the internet or read local intranet information. Even stock markets and other dynamic changing data can be checked from an IP phone. Various applications can be implemented with the use of XML, PHP and JavaScript and can then be tested for responsiveness and usability. All applications are served from an http server which is directly or indirectly connected to the UC appliance. One has to configure the IP address and server folder location on the UC appliance and the applications are ready to be accessed on IP phones. There are various solutions for server implementation e.g. Microsoft IIS or Apache most frequently but also other free PHP servers.

As already mentioned services on IP Phones are applications that run rather on a dedicated WEB server than on IP Phones and can be very useful in a home environment, branch offices, faculties, etc. The range of possible application domain use is very large, from straightforward applications e.g. unit conversion, messaging systems, opinion polls etc. to more entertaining applications as quizzes, games, bulletin boards, photo albums. Nevertheless severe applications for smoke detection or other alarms can be deployed as well as support and control of other applications.

IP Phones have a HTTP compliant client that is used for services and directories functionality. The HTTP client enables the phone to use a simple standard mechanism for retrieving data and providing output to and from standard Web servers. Therefore the method for providing content to the phones is straightforward. All IP Phones services used in our platform use the client functionality for every request; the client is used to request URLs and provides the HTTP GET functionality.

The main framework we have used for designing such applications is based on use of XML i.e. IP Phone XML objects and tags, scripting language PHP and SQL for data management. The proprietary IP Phone XML objects can be:

- ProprietaryIPPhoneMenu,
- ProprietaryIPPhoneText,
- ProprietaryIPPhoneInput,
- ProprietaryIPPhoneImage,
- ProprietaryIPPhoneError,
- ProprietaryIPPhoneResponse,
- ProprietaryIPPhoneDirectory,
- ProprietaryIPPhoneGraphicMenu,
- ProprietaryIPPhoneIconMenu,
- ProprietaryIPPhoneExecute, etc.

The listed proprietary objects are providing functionality to soft keys. When pressed, the soft key invokes the associated action, with exceptions to ProprietaryIPPhoneResponse and ProprietaryIPPhone-Execute (UC500.com, 2011). We have to emphasize what each of these objects is capable of and what kind of functionality it can provide, e.g. how many characters can be in a single ANSI string, how many instances a menu can have or maximum of pixels in a bitmap. We decided to use PHP (Still, 2005; Holzner, 2007). PHP is necessary for more advanced applications development. Most of dynamic applications need some data managing system to deal with different data, for that we used an open source data base management system (Gilmore, 2010).

Under the services button on the IP Phone we can install more than one application at the same time using menus.

6. Designing practical applications

When designing applications special care must be taken to adapt the application to the right phone model. This can be done with decisive structures in the application code. Also the phone firmware can be helpful for adapting the content to the physical capabilities of the IP phone. For demonstrational purposes we will present a multimedia application of improvised video surveillance.

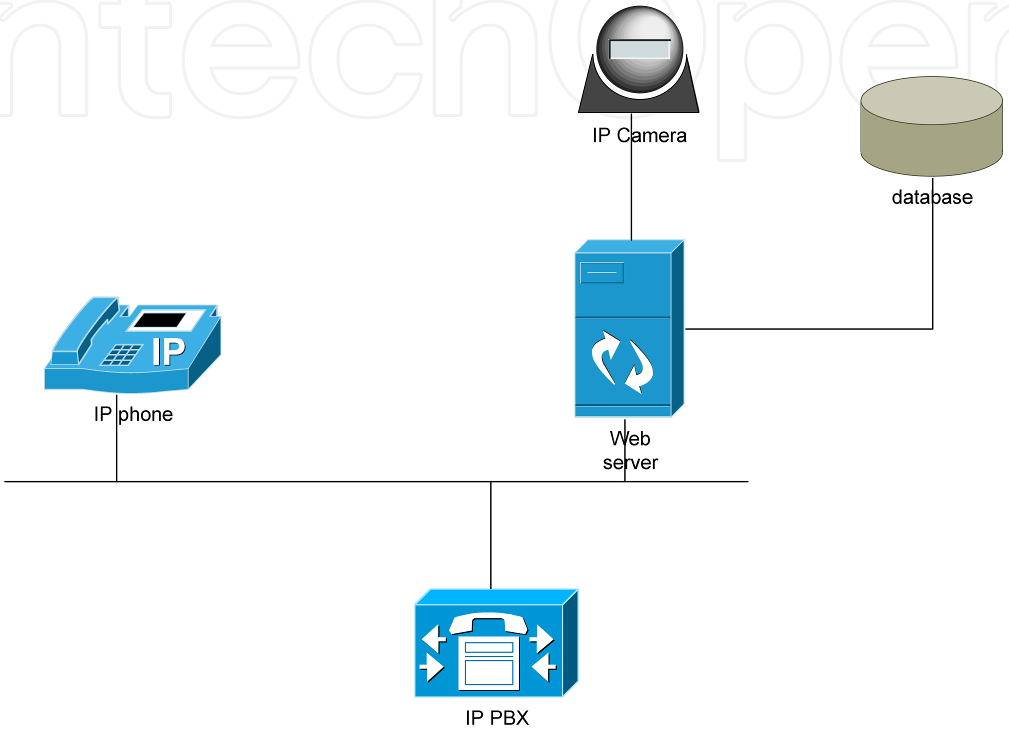


Fig. 6. The test environment for video surveillance application

6.1 The video surveillance prototype

The video surveillance is improvised because the IP phone is not capable of directly video streaming via display. We have to take snapshots from the Web camera and send them to the IP phone. Some models of IP phones have video call functionality built in but not for application development purposes (Cisco Systems, 2011b). Many distributed process are involved in the operation of the application. Here included are IP phone, IP-PBX, Web server, database and an IP/Web camera (Fig. 6).

For the application we used a Cisco IP phone with color display, which has a 298x168 pix space for displaying the picture. This is the biggest area display available on Cisco IP phone models. We have to consider that the phone only displays pictures in PNG format. The picture can only have 12 bits of color depth which means we have to resize and correctly alter the picture to display it correctly. Because video cannot be streamed to the phone we have to take picture snapshots at certain intervals. For that purpose we need a designated program that will be able to snapshot the picture from the IP/Web camera, resize it and send in on request to the IP phone. We used two programs for that purpose. The picture can then be automatically or manually refreshed locally on the phone.

For acquiring snapshot pictures from the Web camera we used an open source program Fwink. The settings of the program enabled us to take the picture at certain time intervals. The picture is saved locally on the disk and is overwritten with every new snapshot. The format of the picture snapshot is JPEG (Lundie, 2011). Then the Web server comes into play – it converts the picture and sends it to the phone. Server is an Apache type with PHP and MySQL (Apache Server, 2011). PHP script does the work on converting pictures to PNG format and correctly altering the picture to be displayed on the phone (display capabilities). Then its sends the picture in XML objects to the IP phone. MySQL database adds functionality to the users with the option of saving certain pictures for a later review (Gilmore, W. J., 2010, UC500.com, 2011).

The main goal of the server is converting the picture to the expected format. For that PHP language is used. PHP needs an extension for a successful conversion. This extension enables, that new objects are added to enhance the capabilities of writing applications. Therefore we used the ImageMagick software suite. The package enables creation, editing and conversion of bitmap pictures. It can read, write and convert in 100 different picture formats (Still, 2005). In PHP ImageMagick is used as an extension through ImageMagick API. We included the extension for the PHP and also for the ImageMagick software suite (Powers, 2008; Holzner, 2007). With the extension selection we have to be careful using the correct DLL library version for the used PHP installation version. With the appropriate implementation the conversion of the picture can be done with a few lines of code. That is the positive part of PHP language. It gives us an opportunity to write applications quickly and do a lot of processing with a few lines of code. With that in mind PHP offers a big leverage (Deel, D. & Nelson, M., 2004). The actual operation of the application is depicted in Fig. 7.

In the following we will describe a code example to give an insight how applications can be written. A script can be designed to automatically refresh the pictures on the phone. On top of the script we added a header for refresh.

```
header("Refresh:3");
```

First we need to read the content from the folder where Fwink program saves the acquired pictures from the Web/IP camera.

```
$image = file_get_contents($campath);
```

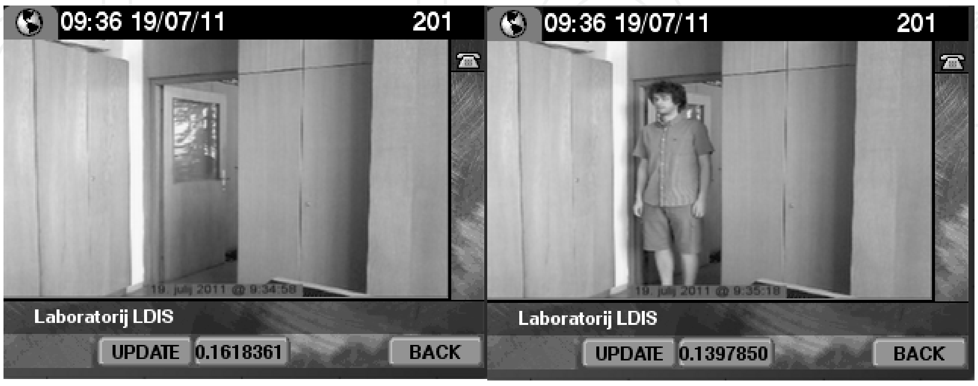


Fig. 7. The test environment for video surveillance application

Then we use the ImageMagick extension. First we create an object, and then we save the picture in that object. Then we resize and convert the picture.

```
$im = new Imagick();
$im->readimageblob($image);
$im->resizeImage(296,166,Imagick::FILTER_LANCZOS,1);
$im->setImageFormat("png");
```

After that the picture is saved back to the disk into a server folder enabling the IP phone to have access to the acquired picture.

```
$fp=fopen($imagepath,'w');
fwrite($fp,$im);
fclose($fp);
```

Now we have a converted and saved the picture on the Web server. Next we need to send it back to the phone. Proprietary XML language objects now come into play, as the phone has to display the image correctly. We used the CiscoIPPhoneImageFile XML object for that purpose. This object is used for displaying PNG pictures on IP phones with a color display. It is necessary that we include a header that instructs the phone of the content in XML format. Only if that header is included in the script the phone will correctly parse XML objects and display the picture.

```
header("Content-type: text/xml");
```

We use a “Title” tag for the name of the picture that will be displayed on the top of the screen and “Prompt” for displaying information about the picture. With the tag “Location” we determined the position of the picture on the phone display. With the “URL” tag we specify where the application resides on the IP phone and where the picture is located. Inside of our CiscoIPPhoneImageFile XML object we also define softkeys. As already mentioned, softkeys are created for user application interaction; updates, application closing, etc. (UC500.com, 2011).

```
<CiscoIPPhoneImageFile>
   <Title>LDIS-G215</Title>
    <Prompt>Laboratorij LDIS</Prompt>
   <LocationX>0</LocationX>
   <LocationY>0</LocationY>
   <URL>http://192.168.10.11/image.png</URL>
 <SoftKeyItem>
   <Name>UPDATE</Name>
   <URL>http://192.168.10.11/cam1.php</URL>
   <Position>2</Position>
   </SoftKeyItem>
   <SoftKeyItem>
   <Name>BACK</Name>
   <URL>SoftKey:Exit</URL>
   <Position>5</Position>
   </SoftKeyItem>
</CiscoIPPhoneImageFile>
```

6.2 Application testing

Some tests were executed to determine the capabilities of the introduced improvised video surveillance application. Multiple parameters had to be considered – the speed of the Web camera snapshots and the speed of application execution. Additionally we had to consider the speed of the PHP script – it executes the conversion of the image and also reads/writes from/to the file. The last but most important was the IP phone processing time. We had to evaluate the time of picture processing and XML objects parsing. The IP phone is an embedded system and because of that limited with the memory and processor power (Deel, D. & Nelson, M., 2004; Cisco Systems, 2011b). That also means the IP phone cannot refresh a new picture as fast as a Web browser on a PC would. We had to consider all the limitations to determine the optimal time values when the script is refreshed and had to evaluate the minimum time interval between the sampled snapshots.

For the test reference we used a 10 seconds picture refresh interval with a 5 seconds refresh of the script. This is how we were able to display every picture acquired. With this reference we also evaluated the time limit of the script execution and picture refresh. In the following table (Table 1) the results of the performed test acquired with the packet sniffer software are presented (Wireshark, 2007).

Resolution	Average time for loading the picture[s]	Average time of script execution [s]
640x480	2,83	0,12
320x240	2,40	0,11

Table 1. Average application loading time

The average time of picture loading and all delays that contribute to the system are covered. With the lower resolution the execution time is app. half a second (0,43 s). From the column Average time of script execution, we see that the conversion itself on the server adds only a small portion of time to display the picture. It was evident that the speed of application execution on the IP phone is solely limited with its capability of image processing.

There is also some unexpected delay in the system. In our examination we concluded that this is contributed by the TCP protocol accessed with the upper layer HTTP. TCP offers reliable transfer and consequently for every request from the IP phone a session is established with the generally tree way handshake. Between the sessions acknowledgments are sent for the segments of the picture between the Web server and the IP phone. Additionally the session has to be closed after the communication ends. Two separate sessions are established during the transfer of one image from the Web server – the first one for the request of script execution and the second one for actually transferring a single picture to the IP phone. That procedure contributes approximately a 0,7 s delay for every single refresh on the IP phone display.

After all the tests in the introduced test environment we conclude that one can use a minimum time interval of 5 seconds between the snapshots if every picture has to be displayed on the phone. In a production environment careful testing is required. Because the pictures need approximately 3,5 s (average time for loading the picture in addition to TCP delay) to load a time higher than 3,5 s must be chosen for a refresh-rate. That corresponds to 320x240 resolution pictures. For the higher 640x480 resolution we have chosen 4 seconds because of the slightly increased load time (0,4 s). If we had chosen a time lower than 3,5 s not every snapshot is displayed, some are skipped. Because of that a new refresh happens before the previous one has finished and therefore the application may occasionally crash. This is because TCP sessions take too long to complete. Consequently the application times out (MaFiRa, 2011). The best scenario conducted in our test is shown in the following table (Table 2).

Resolution	Optimized sampling time [s]	PHP Script optimized refresh time [s]
640x480	5	3,5
320x240	5	4

Table 2. Optimal time interval settings

If required we can still use lower than optimal time intervals between the snapshots. In that case not all the sampled pictures will be displayed. Nevertheless more precise movement detection is possible.

7. Conclusion

Our application works for all types of Web/IP cameras. With a help of an open source program and Apache Web server we changed a Web camera into an IP camera.

Our application can be improved in many ways. Because there are many different types of IP phones available with different capabilities we have to adapt applications to the receiving phone model. For example some models do not have a color display. So we can write a decisive structure to choose a different conversion for a different phone type. We implement that with the help of the header the IP phone sends when initiating a session with the Web server.

Pictures can also be saved by the user or automatically into a database. We can also implement a movement detection feature some IP cameras offer and for example an image on the phone only appears when motion in front of the camera is detected. Then only changes and appropriate time tags are saved with the image to the database.

In the presented application we had to evaluate different values for the snapshots acquiring time interval and refresh rate to correctly display every picture acquired. With the PHP script we can synchronize those parameters which now operate individually. With that we could achieve shorter intervals for displaying the snapshot images on the IP phone.

We presented an example of a multimedia application that can be written for the IP phone. We tried to demonstrate that an IP phone application involves many distributed processes. The application is intended as a demo and together with the Web server, IP-PBX and other equipment in our local IP & VoIP network forms a basis for educational application design approach. It enables numerous possibilities in application development and proof of context for prototype development in business environments.

8. References

Au, D.; Cho, B.; Haridas, R.; Hattingh, C.; Koulagi, R.; Tasker, M. & Xia, L., (2005). *Cisco IP Communications Express: CallManager Express with Cisco Unity Express* (1. edition), Cisco Press, ISBN 1-58705-180-X, Indianapolis, IN 46240 USA

Cisco Systems, (2006). (n.d.). *Cisco Unified IP Phone 7970 Series for Cisco Unified CallManager* 4.2(3), Document ID: OL-10776-01, 2006. September 2011, Available from: http://www.cisco.com/en/US/products/hw/phones/ps379/products_user_guide_list.html

Cisco Systems, (2011c). (n.d.). Configure and Manage the CUE system auto Attendant, Document ID: 63986, September 2011, Available from: http://www.cisco.com/en/US/products/sw/voicesw/ps5520/products_configuration_example09186a00803f82eb.shtml

Cisco Systems, (2011). *Cisco Unified Communications Manager Express System Administrator guide*, Cisco Press, San Jose, CA 95134-1706 USA, 2011.

Cisco Systems, (2011a). (n.d.). *IP phone services –Resources*, In: *Cisco Developer Community – Resources*, June 2011, Available from: http://developer.cisco.com/web/ipps/resources

Cisco Systems, (2011b). (n.d.). *IP phone services –Forum*, In: *Cisco Developer Community-Forums*, June 2011, Available from:, Access: http://developer.cisco.com/web/ipps/forums

Deel, D. & Nelson, M. and A. Smith, (2004). *Developing Cisco IP Phone Services*, ISBN: 978-1587050602, Cisco Press, Indianapolis, IN 46240

Gilmore, W. J., (2010). *Beginning PHP and MySQL: From Novice to Professional*, 4.th Edition, ISBN: 978-1430231141 Springer, New York, NY 10013

Hatting, S., Sladden, D., & Swapan, Z.A., (2010). *SIP Trunking, Migrating from TDM to IP for Business Communications,* Cisco Press, ISBN 1-58705-944-4, Indianapolis, IN 46240 USA

Holzner, S., (2007). *PHP: The Complete Reference*, McGraw Hill, ISBN: 978-0071508544, New York, NY 10121-2298

Kaza, R.; & Asadullah S., (2005). *Cisco IP Telephony – Planning, Designing, Operation and Optimization,* Cisco Press, ISBN 1-58705-157-5, Indianapolis, IN 46240 USA

Lundie, C. (2011), Free webcam software, In: *Fwink*, July 2011, Available from: http://www.lundie.ca/fwink/

MaFiRaWiki TCP/IP, (n.d.). July 2011, Available from: http://wiki.fmf.uni-lj.si/wiki/TCP/IP

Powers, S., (2008). *Painting the Web*. O'Reilly Media, Inc., ISBN: 978-0596515096, Sebastopol, CA 95472

Still, M., (2005). *The Definitive Guide to ImageMagick*, 1. Edition, APress, ISBN: 978-1590595909, Berkeley, CA 94710

The Apache Software Foundation. (n.d.). Apache Usage Statistics, July 2011, Available from: http://trends.builtwith.com/Web-Server/Apache

The Apache Software Foundation. (n.d.). The Apache HTTP Server project. July 2011, Available from: http://httpd.apache.org/, july 2011

UC500.com. (n.d.). Custom XML IP phone services, March 2011, Available from: http://uc500.com/en/custom-xml-ip-phone-services-uc500-and-cisco-unified-call-manager-express

Wireshark (2011). (n.d.). *Go Deep*, July 2011, Available from: http://www.wireshark.org/

NEW TECHNOLOGIES
TRENDS, INNOVATIONS AND RESEARCH
Edited by Constantin Volosencu
INTECH

New Technologies - Trends, Innovations and Research
Edited by Prof. Constantin Volosencu

ISBN 978-953-51-0480-3
Hard cover, 396 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

The book "New Technologies - Trends, Innovations and Research" presents contributions made by researchers from the entire world and from some modern fields of technology, serving as a valuable tool for scientists, researchers, graduate students and professionals. Some practical applications in particular areas are presented, offering the capability to solve problems resulted from economic needs and to perform specific functions. The book will make possible for scientists and engineers to get familiar with the ideas from researchers from some modern fields of activity. It will provide interesting examples of practical applications of knowledge, assist in the designing process, as well as bring changes to their research areas. A collection of techniques, that combine scientific resources, is provided to make necessary products with the desired quality criteria. Strong mathematical and scientific concepts were used in the applications. They meet the requirements of utility, usability and safety. Technological applications presented in the book have appropriate functions and they may be exploited with competitive advantages. The book has 17 chapters, covering the following subjects: manufacturing technologies, nanotechnologies, robotics, telecommunications, physics, dental medical technologies, smart homes, speech technologies, agriculture technologies and management.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Janez Stergar, Janez Klanjšek and Sibila Vadlja (2012). A Framework for VoIP Testability and Functionality Extension with Interactive Content Delivery, New Technologies - Trends, Innovations and Research, Prof. Constantin Volosencu (Ed.), ISBN: 978-953-51-0480-3, InTech, Available from: http://www.intechopen.com/books/new-technologies-trends-innovations-and-research/a-framework-for-voip-testability-and-functionality-extension-with-interactive-content-delivery

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821