A Reliable System Integration Study Using Open Source VoIP Software

*[1]Ahmet Özmen and [2]Veysel Harun Şahin
*[1]Faculty of Computer and Information Sciences, Department of Computer Engineering, Sakarya University, Sakarya, Turkey
[2] Faculty of Computer and Information Sciences, Department of Software Engineering, Sakarya University, Sakarya, Turkey

Abstract

In this work, an open source VoIP software is customized for private needs. This software package is called Asterisk and available at GNU sites for free. In fact, our university has a commercial VoIP PBX for more than ten years; however, the institutional needs are increasing both in the volume e.g. the number of end users and the functionality such as smart phone integration and video conferencing etc. Although the commercial counterpart provides much functionality at some expenses, working on an open source software helped us to learn and teach VoIP systems to our colleagues and students in detail. Moreover, we get the freedom of selection any brand of telephone apertures. In the beginning, we have carefully investigated existing network and VoIP infrastructure. It was very large like more than 2400 end user were already connected to the commercial brand PBX and multiple IVR systems were serving. In this study, we have extracted the needs and the open source software package capabilities. To meet the demands, we have designed a fault tolerant system with two servers. If one fails the other server takes over the load automatically and transparently. We have developed an end user registration software to make it easy to use. So, a new user record can be easily created in the server and the user information is stored in an XML based files in the server. This xml based files are transferred to the newly connected telephones by TFTP server automatically. Many new features and services have created and started on the VoIP server since then. Currently more than 500 end user is attached on the system and the system has been working successfully more than two years.

Key words: Asterisk, Internet, Open Source Software, PBX, SIP, Voice over IP (VoIP), XML

1. Introduction

Voice over Internet Protocol (VoIP) is a stack of protocols which enables the realization of different real-time communication schemes like audio, video, fax etc. over Internet Protocol (IP) instead of traditional public switched telephone networks (PSTNs). Currently VoIP standards are handled by The Internet Engineering Taskforce (IETF) [1] and International Telecommunications Union (ITU) [2]. Besides these standards bodies, some companies and communities like Digium-Asterisk, Microsoft and XMPP also develop proprietary and open VoIP standards and protocols. In addition there are several applications either on the server side or on the user side that helps us create and use VoIP systems. One of these is Asterisk [3] is an open source communications framework which is used to create and manage VoIP systems.

In this work we customized the open source Asterisk framework for the voice, video and fax

*Corresponding author: Address: Bilgisayar Mühendisliği Bölümü, M5 Blok 5302, Esentepe Campus, Serdivan Sakarya 54187 Turkey. E-mail address: ozmen@sakarya.edu.tr

communication needs of Sakarya University. Although Sakarya University has a commercial VoIP system already in use, the increasing needs like the number of users, end points etc., and new functionality demands like video conferencing, smartphone and tablet integration, led us to create a more functional, flexible and manageable VoIP solution in a more cost effective way.

2. Voice over IP

Increasing speed on both computer systems and computer communication devices replaced analogue systems with digital counterparts at all sites in our life. VoIP telephony systems are also result of this improvements. Although VoIP end user components are still expensive comparing the analogue counterparts, their operating expenses are quite cost effective. They allow getting service from several different providers from within the same end user hardware. VoIP systems reduces Private Branch Exchange (PBX) count dramatically. For example, large companies or universities where some buildings are scattered in a campus and some branches may be in long distance locations. In this case, inside the campus all buildings and long distance locations must also contain a PBX system. In addition to this, a service must be bought for communication with long distance branches. As it can be seen, this increases PBX management complexity dramatically, and calling long distance branches will not be the same call pattern.

VoIP systems can be examined in hardware and software levels.

2.1. Hardware Components

2.1.1. IP Telephony

These devices are actually heterogeneous multi-processor systems, they include: 1) a special purpose processor such as DSP, and 2) a general purpose processor.

The DSP processor is dedicated only getting digital samples from human voice. A compression process takes place to reduce the data size without losing information. Compressing mechanisms are called as *codec*s which is short for **co**der and **dec**oder. Some very well-known codecs are G.711, G.726 etc. By the help of these codecs compressed voice bandwidth can be reduced down to 4 Kbit/s. Some of compression algorithms work as lossless and some of them reduces the bandwidth segregating from voice quality which is very difficult to understand. However, bandwidth is a very important parameter for VoIP systems and almost all telephony devices are supporting many compressing algorithms.

After obtaining the compressed voice data it should be put into a packet. All of these tasks must be accomplished between two samples. Sample interval is changing from 5ms to 30ms based on the codec used. DSP injects the packets to the network. For voice data DSP is usually selected from TI-5500 series processor, and if visual data is considered then TI-6500 series processor is used.

In the meantime general purpose processor runs a monitor software which is a kind of small operating system. It scans the key pad, show information in the LCD screen. These tasks are also very complex, hence this processor should have a moderate processing power.

2.1.2. Switches

Normal market switches would be sufficient to accomplish the task form small size environments, however, Quality of Service (QoS), and VLAN management is preferred. For large business applications manageable and high performance switches gain importance.

2.1.3. Server Computer

This can be entry level server computer because the server does not accomplish heavily calculations. The server just matches two ends. That is all; all processing tasks are done in end user components. However, if trans-codec is required then this real time job should be done in the server. A simple server can handle thousands of end user if trans-coding is not required.

2.2. Software Components

The most important software component of VoIP systems is the server. The server software matches caller and the target entities just by searching its table first. If searched number is not on the list then the task is forwarded to the other servers. Links between servers are called as trunks.

Asterisk is the most popular VoIP server system. Like Linux operating system, it is very widely used and frequent improvements are done by the community. Asterisk server provides all properties supported by commercial counterparts. Universal Cisco Call Manager (UCM) is proprietary server developed by Cisco.

3. VoIP Application at Sakarya University

In May 2102, we began to search alternative solutions to the IP telephony system of Sakarya University which was based on Cisco products. Although the Cisco system was functionally sufficient for us we were having issues with the rising license and operating costs. There were mainly two problems: The first one was the high costs for extending the system and the second one was the high expenses of the software updates (license update fees). The main source for high costs was the requirement to pay an additional license fee for each telephone which will be added to the system. The second source was the necessity to use specific telephone models from a specific manufacturer.

After some research on the subject it has been realized that Asterisk open source framework could be utilized for our needs. Asterisk, was an open source and free software; however, to make it work together with the existing Cisco system was requiring the use of extra plug-ins and configurations. Although these requirements seem to be serious obstacles, Asterisk had many benefits. One, who uses Asterisk, did not have to pay for software updates and license fee per phone. Also it would be possible to use any IP telephone from any manufacturer. There were mainly two problems to adapt Asterisk in our university. The first one was the steep learning curve to integrate Asterisk with Cisco systems and to maintain it. And the second one was that Asterisk did not have an official support company in Turkey, so we would not be able to get support in case of an emergency. After examining these two problems, we decided that this subject is worth investigating and began to work on this system in May 2012.

To first step of our work was to configure and run an experimental IP telephone PBX service on servers which were donated by Ford Turkey. This work is achieved in Open Source Academy (AKA) laboratory. We tested the system in itself by using software telephones like X-Lite without touching and thus compromising the current system and got satisfactory results. Then we integrated the newly created system to our university's Cisco infrastructure (we created a trunk between to servers). After this integration we carried out comprehensive tests on the integrated system by the help of 10 users who are connected the system through the Asterisk software for 2 months. After getting successful results, in July 2012 at the Information Meeting of our university we reported that we could expand our tests from 10 users to a department. In this meeting, for further testing, it has been decided to use the system for the medical school which in that time it was being newly built. Until that time, all our experiments were conducted by software telephones and thus we did not have any analysis data for real telephone models.

The first thing we did for the new department's work was to choose the brand of the telephone. We explored different telephone models from three different manufacturers (Cisco, Yealink and Grand Stream). We prepared a feature matrix for these models and then we bought two telephones from each of the three manufacturers. We realized extensive testing on each of the devices. We also randomly selected some testers from the involved department's employees and gave them the phones to use. After some usage, we got the opinions of testers. After these tests, we found out that for our needs Cisco SPA303 can be preferred model from the price and performance point of view. The physiological factors also affected this decision. Because our university were using Cisco systems for a while, our employees (end users) physiologically feel comfortable when using Cisco telephones. When using Cisco telephones the users would not recognize that a different server (Asterisk server) was servicing them. Besides this the selected model has more functionality than the currently used Cisco 7911G series telephones. At that time, Cisco SPA303 model telephone's price was 180 Turkish Liras for single purchases.

The second thing we did was to make search for the PBX server. Although the donated servers from Ford Turkey were used in test phase they were not suitable for production systems. Because, the planned system must be non-stop working. The present and possible future workloads of the server have been calculated and then the network design has been realized on the paper. In this work the requirements of a server system has been calculated which will run an PBX server that is capable of driving 2500 telephones. In this calculation we did not get into account, the encoding/decoding work, interactive voice response (IVR) system and voicemail system since we did not plan to give those services at the beginning. After these calculations, entry level rack server was preferred.

Service failures are unacceptable in telephone systems. Even if a server breaks down the service should continue. Because of this, we decided to use two servers for the service. We bought two servers and configured them to run symmetrically. They were configured in a way that even if one server fails, the other server continues to give the service. However, for load balancing issues, some workloads (provision process etc.) were given to the backup server, in an asymmetric way.

As a result, 2 entry level Dell rack servers and a power supply have been bought and the system has been configured to work with backup. The table below shows the devices and their costs.

Table 1. Hardware components and their properties.

No	Name	Specifications	Total	Unit Price (TL)	Total Price (TL)
1	Server	Dell R210 Intel Xeon X3430, 500 GByte HD, 4GByte Ram, Rack Server	2	1700	3400
2	Power Supply	3 KVA R Series Rack Cabin, 2 Pieces	1	1100	1100
	TOTAL				4500

In the next level, we concentrated on provision system. This system enables telephones to get their configuration information automatically. For this purpose we learned the detailed operating principles of the selected telephone model, designed and created a provision file (automatic configuration file) for it in accordance to our needs. We developed an application which automatically creates provision files for each user and telephone and then hooks the file to the system. There were no Turkish language support for SPA303 telephones and Cisco were not giving support about this situation. Because of this problem, we created an additional provision file which provides Turkish menu support. Besides this, we added "local directory search" property which was present on the other Cisco telephones, to the provision system.

After setting up the provision system we worked on user management side. We needed to edit several files and restart the service to be able to add a new user to the system. This manual operation was error prone and complex. Because of this we designed an IP server management application and implemented it in C# and PHP languages. By the help of this software, we managed to add new users to the system easily. Whenever a physical telephone setup has been realized in our campus, we got it working in a couple of minutes. Also this software allows us to monitor the state of the telephones thus enabling to notice and fix the faults in a short amount of time.

After these operations this new server has been integrated to the Cisco UCM server over a trunk. By doing this we enabled to make calls from the new system to all telephones present in the campus as well as in the reverse direction.

One other work was to create a trunk to the Karel DS-200 PBX which is located at Korucuk Research and Education Hospital. For this purpose we firstly looked at inter building communication lines and identified a single-mode fiber optic line. We planned to use this line also for VoIP. To make this connection operate, we needed to add an additional board to the Karel server. We found the person who is in charge of the Karel server and after making a few meetings we managed to attach the board to the server experimentally. After making communication configuration for both of the sides, we run the trunk. The price of the board has been paid by the hospital. Now, a telephone in the new system can make call to any telephone in the hospital (First 2 and then dialing 4 digit telephone number, like 2XXXX).

We provided, direct inward dialing (DID) call for the new telephones. Also we provided to make intercity and inter-country calls from new telephones. To make this happen we worked closely with the IT department. By using our software we enabled to activate or deactivate intercity and/or inter-country calls for any telephone. Also the new systems pricing policy is compatible with the existing system. Thus we can immediately calculate the correct pricing for the calls and transport that information to the accounts of users.

The operation of the system is as follows:

1. IT department gets the new telephone requests. Saves the MAC numbers of the telephones and introduces them to the network. After this, e-mails us the user information (name, surname, telephone number etc.)
2. We add the new user to the server, create the provision file for the user and upload the file to the system by using the developed software.
3. The user plugs the telephone to the Internet socket; the telephone becomes usable in a couple of minutes.
4. We also created a quick call button named “DSTK” on the telephone. This button is used to make direct calls to us in case of a faulty situation.

As of today, the system is running nearly 24 months. It gives service to more than 550 telephones. Some weeks, every day a few telephone records have been added and no serious problem has occurred since then. Only one time in October 2012, the system could not serve for a short time because of an error made in a provision file. This problem required in place intervention. In this incident, we went to the hospital and resolved the problem quickly.

Our VoIP system was configured to use Secure Real-time Transport Protocol (SRTP) for security purposes. In our system, different keys are generated for each call in the system and all audio data are transmitted after encoding them with the keys generated for that call. A key's life time is only one call. In other words whenever a call ends, the keys generated for that call (session) are deleted.

The new telephone system (telephones and PBX) provides all of the functionalities which Cisco system provides. In addition, a new functionality can be added to the system in a short amount of time. We employ 2 students have been educated about the system in our laboratory. These students get grant from our university and help us managing the system.

We have also developed an experimental server which has 100 lines capacity. This system mainly serves to the users who connect the system from their mobile phones. This system runs on an IBM server which is donated by Ford Turkey. The users on this server can make any in-campus call and can be called from any telephone inside the campus. However they cannot make off-campus calls which requires to connect to the third party operators. Also there is no direct inward call. The main purpose of this system is to give users the ability to make in-campus calls from their smartphones by using special programs like SipDorid, CsipSimple without paying any fee.

If any user of this system, has also a VPN account in the university, the campus becomes the whole world. To make the VPN work, the user also should install Cisco's AnyConnect software on to his/her smartphone. By doing this, the user can make in-campus calls from anywhere in the world with an Internet connection. Today, this server has 50 users. Users of this server get the service occasionally. In other words, they connect the service whenever they needed. This server also supports video calls.

Another experimental study includes 100 lines and has been configured at DMZ region and has a real IP. It runs on a Dell server which is also donated by Ford Turkey. This server does not have integration with the inside servers yet. The users of this server, can make voice and video calls by using internet connection without needing a VPN account. Server behaves as an intermediary between the users. This kind of communication mode is called peer-to-peer communication. It does not have any vulnerability but in some networks which does not know about this service, these ports could be closed. In this case the call cannot be established. In other words, the success of any call in this system depends on the structure of the network. For example, on some campus networks in which these ports are closed the calls cannot be made. However, we made successful results in video calls in the Türk Telekom network. Also, we have successfully carried tests at different regions by using TT-Net Internet service. There is no charge in these calls because these calls are made directly over internet without any operator intervention.

Conclusions

In this work, an open source VoIP software is customized for private needs. This software package is called Asterisk and available at GNU sites for free. In fact, our university has a commercial VoIP PBX for more than ten years; however, the institutional needs are increasing both in the volume e.g. the number of end users and the functionality such as smart phone integration and video conferencing etc. Although the commercial counterpart provides much functionality at some expenses, working on open source software helped us to learn and teach VoIP systems to our colleagues and students in detail. Moreover, we get the freedom of selection any brand of telephone apertures. In the beginning, we have carefully investigated existing network and VoIP infrastructure. It was very large like more than 2400 end user were already connected to the commercial brand PBX and multiple IVR systems were serving. In this study, we have extracted the needs and the open source software package capabilities. To meet the demands, we have designed a fault tolerant system with two servers. If one fails the other server takes over the load automatically and transparently. During this study, we have gained a lot of experiences and trained many students through the work

References

[1] The Internet Engineering Taskforce homepage [Online]. Available: http://www.ietf.org/ [10.04.2014]

[2] International Telecommunications Union homepage [Online]. Available: http://www.itu.int/ [10.04.2014]

[3] Asterisk homepage [Online]. Available: http://www.asterisk.org/ [10.04.2014]