VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Sanjeeb Karki

SETTING UP LINUX BASED NETWORK SYSTEM

Faculty of Information Technology
2012

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Programme in Information Technology

ABSTRACT

Author	Karki Sanjeeb
Title	Setting Up Linux Based Network System
Year	2012
Language	English
Pages	65
Name of Supervisor	Gao Chao

The goal of the project was to build a Linux based network system where clients get full benefit of various services. The project was implemented at Isokyrö 61500, Finland. The project contains various network system implementations such as File server, Webcam server and Asterisk server.

The thesis implements the understanding of the network systems into real use. Configuration of File server consist the security of each user based on the users need and handling their permissions. The network also have webcam server for security reasons. For the VOIP calls within the network Asterisk server was used. So the main focus of the thesis is to build a complete stable network where we are more cautious to the quality than the quantity.

In the mean time frame of implementing the project, configuration of ADSL router, creation of the DHCP subnets for dynamic IP configuration, handling the data stored by webcam server, implementation of SIP with Asterisk and securing the network with firewall configurations were completed which gives the clear overview of the network operation. After each configuration proper tests were performed to verify the result's stability.

The completion of projects results the clients to use secure and reliable network where clients can use the benefits from the different network components installed during the project.

Keywords File premissions, VOIP, SIP

CONTENTS

ABBREVIATIONS

ADSL	Asymmetric Digital Subscriber Line
CD-ROM	Compact Disk, read-only-memory
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DSL	Digital Subscriber Line
FDD	Frequency Division Duplex
FTP	File Transport Protocol
GNU	Gnu's Not Unix
GPL	General Public License
HTML	HyperText Markup Language
ISO	International Organization for Standardization
LAN	Local Area Network
MAC	Media Access Control
MD5	Message-Digest Algorithm
NAT	Network Address Translation
NetBIOS	Network Basic Input/Output System
RAM	Random Access Memory
RTP	Real-time Transport Protocol
SIP	Session Initiation Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
UDP	User Datagram Protocol
Ufw	Uncomplicated Firewall
VoIP	Voice over Internet Protocol
WAN	Wide Area Network

1. INTRODUCTION

The main purpose of the project was to set up a Linux Server for a residential property, which needs a complete vision of theory and practice. This project has involved a very high level of network system, which will be used by different clients. Clients are distributed in one office room and residential apartments. This project involves setting up a network which uses security web cameras as well. So as a whole, this project is a complete implementation of not only setting up a Linux Server but also installing the different devices desired by the clients.

1.1 Background Information

This project involves many different tasks, which as a whole can be called as Installation of a Private Linux based Network System. This involves the installation of computer systems using a topology, installation of web based security cameras, setting up a DHCP Server, File Server and a secured firewall.

1.2 Working environment of the project

Project has been done in Oltermannintie 7, 61500, Isokyro, Finland. The network was implemented in a residential building of nearly 40-50 users.

1.3 Purpose of the project

The main purpose of the project was to setup Linux server for networking because the owner of the house want to rent his apartment for office purpose and other apartment for rent to family members. The owner wants file sharing and networking to shares data between computers and to play games.

The owner wants security in this building, especially in parking area because it is open. The owner had some problems related to this before. So, he wants to put security camera to be more secure.

1.4 Objectives of project

The main objective of the project was to configure Ubuntu server which includes the configuration of DHCP server, File server, Asterisk server for VoIP, Security web cam server, and firewall to protect the system.

This thesis was implemented step-by-step which can be organized as follows:

- ADSL Configurations
- DHCP Implementation
- Samba File Server Configurations
- Security Cam Server Implementations
- Asterisk Server Configurations and
- Firewall Implementations

2. PROJECT DESCRIPTION

2.1 Ubuntu Server

A server is a software program or a computer on which programs run which provides specific kind of services to the client running on a computer or other computers on a network. The main task of a server is to manage network resources and organize access to these resources for other computers linked to it. And Ubuntu server is free of charge to install./1/

The minimum hardware requirement for Ubuntu 11.10 and the hardware of the system being used consists as follows.

Install Type	RAM	Hard Drive
Ubuntu 11.10	128 megabytes	1 gigabytes
Own system	512megabytes	60 gigabytes

Table 1. Ubuntu 11.10 Installation requirement and actual server hardware

2.2 ADSL Router:

ADSL, Asymmetric Digital Subscriber Line is a data communication technology over copper telephone lines and provides faster data transmission compared to voice band modem. Both ADSL services and voice calls services can be obtained at the same time since it uses the different frequency than that of voice telephone call which is obtained by a special filter called micro filter installed on a subscriber's telephone line. Generally the data carried by ADSL are routed over telephone's company data networks to Internet Protocol network.

ADSL is named asymmetric because the downstream and upstream rates are not same. In ADSL the downstream rate is high than that of upstream rate which provides higher downloads speed from internet to the costumers. Nowadays ADSL communications are full-duplex which can be achieved by FDD which uses two different frequency ranges for upstream and downstream. The communication from end user to telephone central office is over upstream band and the communi-

cation from telephone central office to end user is over downstream. There are different types of ADSL standards available like standard ADSL, ADSL2 and ADSL2+.

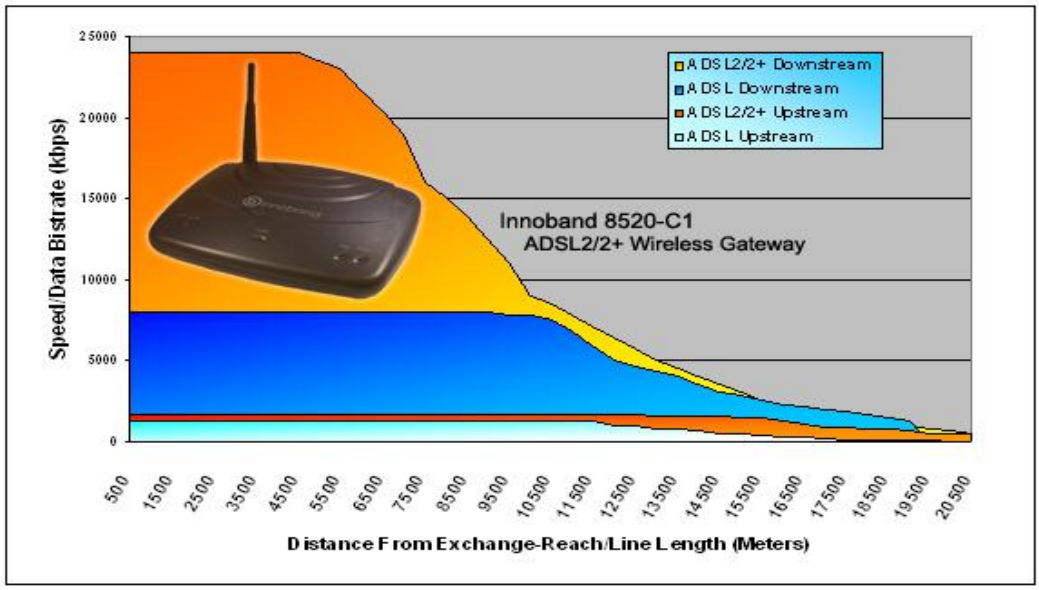


Figure 1. Upstream/downstream speed comparison of different ADSL standards /3/

The Figure 1 shows the supported upstream and downstream by ADSL and ADSL2+ standards and also shows how the speeds are affected by the distance of separation from the nearest telephone exchange and end user. From Figure we can say the downstream for ADSL2+ decrease faster when the distance of separation increases than that of ADSL. In standard ADSL case, the band from 25.875 kHz to 138 kHz is used for upstream communication while 138 kHz to 1104 kHz is used for downstream communication. The ADSL router used in the project was ASDL2+ because the upstream rate is 12795 kbps and the downstream rate is 2556 kbps.

2.2 DHCP Server

DHCP stands for dynamic host configuration protocol. It is used to allocate IP address to the clients. IP address can be either statically or dynamically allocated to the clients. DHCP assigns IP address to the clients from the pool of the address defined for dynamic allocation and for static allocation it matches the Ethernet mac address from where the request is coming and assigns the static address defined. The Figure 2 shows the process of assigning the IP address:

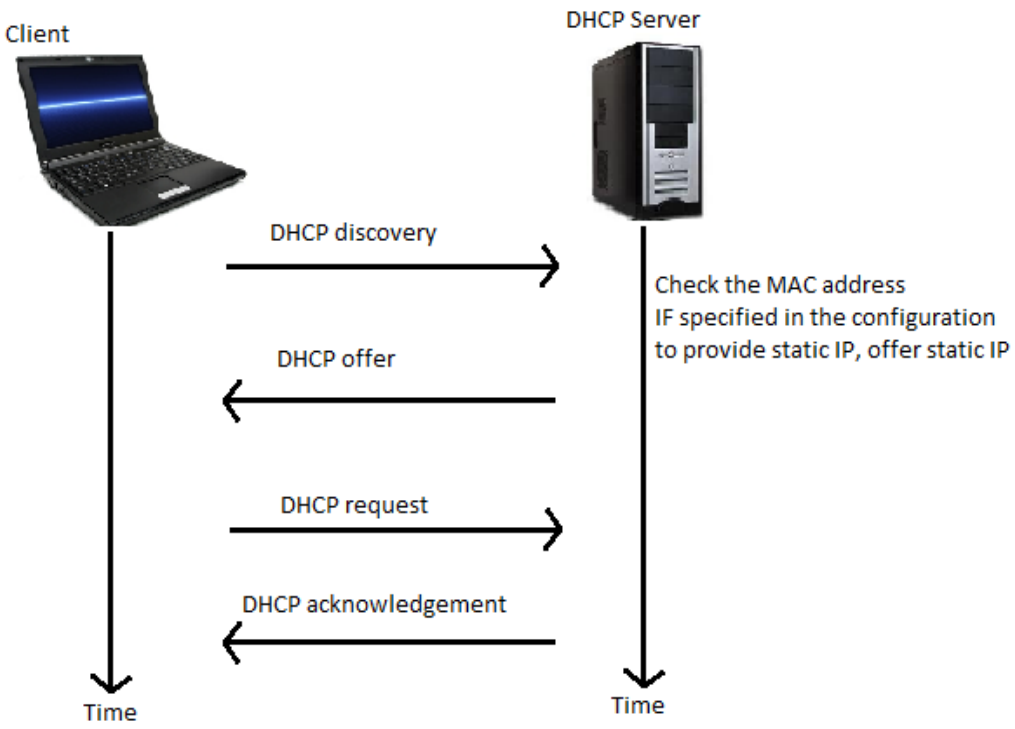


Figure 2. DHCP Communication

2.2.1 DHCP Discovery

First, whenever the client is connected to the network it sends DHCP discovery to discover the available DHCP servers in the network. The client broadcast the discovery message with source IP of 0.0.0.0 and the destination of 255.255.255.255 or the specific subnet broadcast address. The broadcast message uses UDP which

has source Port 68 with protocol “bootpc” and destination port 67 using protocol “bootps”. Once the server gets DHCP discovery message it replies with DHCP offer.

2.2.2 DHCP offer

After receiving DHCP discovery message the server or servers on the network segment offers the client with the unused IP from the pool if needed to be assigned dynamically and a static IP if the mac address is assigned static IP in the configuration. This message contains the mac address of the client, IP being offered, subnet mask, lease duration and IP of the server making the offer. In case if the DHCP is configured to provide the additional information like DNS servers address, it is also included in the offer.

2.2.3 DHCP request

Once the client receives the DHCP offer it replies to the server with the DHCP request. Usually it is unicast to the server but might be broadcast message if the client haven’t got and IP. If there are multiple DHCP servers in the network, a client gets multiple IP offer but in turn the client only accept one DHCP offer. The DHCP server or servers are informed whose offer the client has accepted based on the transaction ID field in the request. So other DHCP servers with return the IP offered to the pool as unused IP.

2.2.4 DHCP acknowledgement

After receiving DHCP request from client, the server sends DHCPACK packet to the client which includes the lease duration for which the IP offered is valid and any other information requested by the client. This is the last message included in this process. Till now the client has successfully configured the IP and the IP offered is moved to the “dhcp.leases” file from the pool of available address in server database./4/

2.3 File Server

File server is a storage device dedicated to stores files on a system. And make possible for multiple users to share the files on network. File server is different from personal computer, which provides permission to access to use the file to the networked computers. User can save his/her work and have access to the files without saving it in the external devices such as USB, floppy or any other storage devices. In a file server it is possible to restrict the access privilege to guest and other registered users. And the most important things are that it backup the file on a regular basis. If somehow the file is deleted, then it can be restored from the backup. Even if there is hard disk failure the file which have been saved in the network server will not have any affect.

Samba is free software available in Linux which provides the interoperability between Linux/Unix servers and Windows client. Samba is not available in windows platform but using samba a windows client can have access to samba file server as if it is a hard disk drive in windows itself since samba protocol are carried over TCP/IP and name lookup through DNS. The important fact that should be noted is that samba consists of two programs namely “smbd” and “nmbd” daemon. File and print services are provided by ”smbd” daemon which also handles the share and user mode authentication and authorization. The “nmbd” daemon is used to map NETBIOS computer names to the IP address of TCP/IP network. The samba configuration file is located at “smb.conf”. It should be kept in mind that the firewall configuration should be made to allow samba since samba protocol uses TCP/IP for communication. It is also possible to use host based, user based and interface based protection. Ports used by Samba:

PORTS	PROTOCOL	DAEMON
137	UDP	Nmbd
139	TCP	Smbd
445	TCP	Smbd
138	UDP	Nmbd

Table 2. Ports used by Samba /5/

2.4 Security Cam Server

For security considerations sometimes it is necessary to have webcam server installed in our systems. Before installing a webcam server first we should check the hardware comp ability of the webcam. It is so rare that the webcam manufactures develop the driver for Linux distribution. The webcam will consistently capture the video and the recoded video is saved on the server and can be accessed anytime. The motion software is also installed so the webcam starts recording whenever it detects motion in its range. Apache server is also installed so that clients inside network can easily access the webcam through a web browser. The webcam will capture the field all the time but the data is only recorded whenever the motion is detected in the range like sensor cameras. The recorded videos are saved at /temp/motion/. The recoded videos are saved in .swf format but can be changed according to our needs. The TCP port used by the webcam server is 8081 by default. Webcam servers also have additional options like display mode of the stream, date, frame rate, image capture size and also specify the foreground and background colors.

Using the stream of image required the use of Java and HTML pages which are provided with the webcam server packages by default. In our case the files are located at /usr/share/doc/webcam-server/applets/. To be able to view in HTML pages editing of the file webcam.html is required where we can also specify the layout of the page. Motion software is very flexible and reduces the power consumption also because it allows the webcam to remain in sleep mode until it detects a noise or movement and starts recording.

The Figure 3 shows the client connecting to the webcam server through webpages through the servers IP address. Since we are not going to register any domain name for the cam webpage it can only be viewed through its IP address. Also our server is behind the ADSL router the servers have private IP address so is not accessible through the external network which also improves security but the clients inside the network can easily access the server since the security is for them.

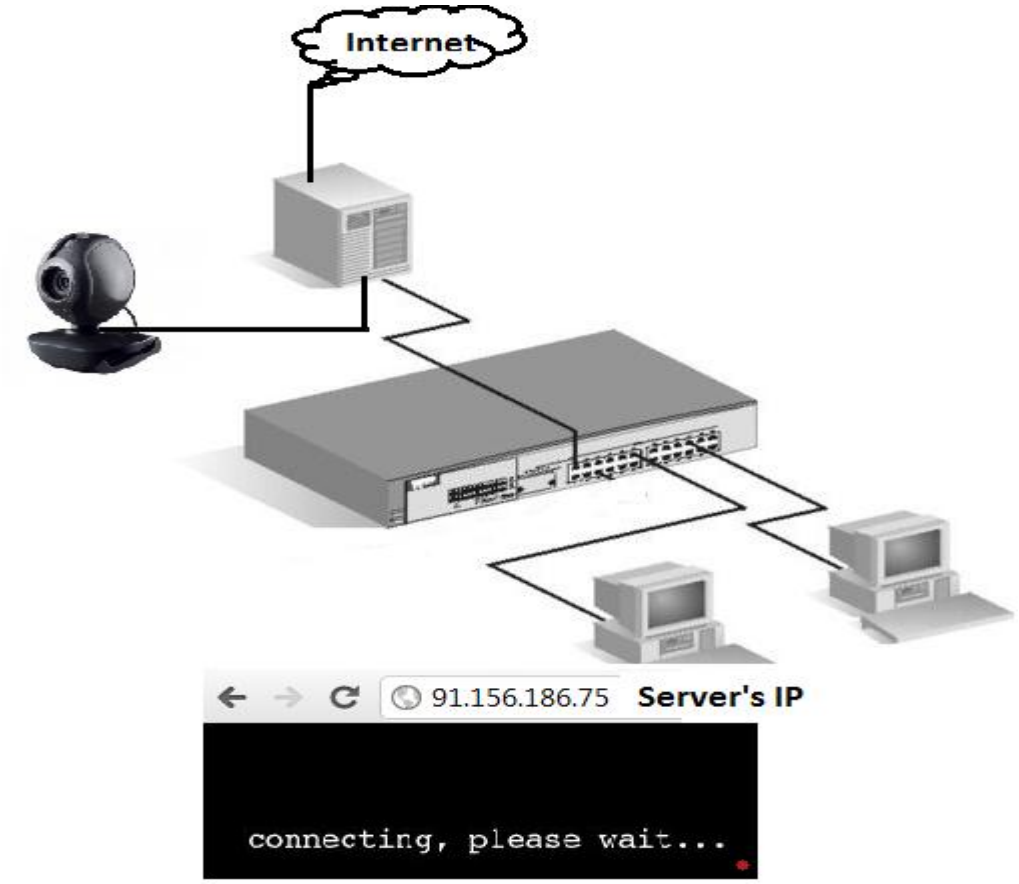


Figure 3. Webcam server

2.6 Asterisk server

Asterisk is free software that turns a normal computer system into a communication server. It was released as open source under the GNU and also acts as a media gateway. Asterisk allows us to create a customized phone system according to our requirements. It gives the flexibility to create custom modules that extends our phone system and is more cost effective than leasing a telephone line for each telephone required. Here we are going to use asterisk as a VoIP system since asterisk works with Internet Protocol. Even though asterisk is not a SIP proxy it supports SIP for VoIP and calls can be made and received with SIP using Asterisk. SIP is a method of signaling VoIP and is a part of default installation of Asterisk. Every device using SIP are registered with a SIP server and are allowed to establish communications. Since our project is not intended for large customers asterisk

server will be fine. So our SIP devices will be registered with Asterisk but if the number of SIP devices increases asterisk is not able to scale very well. /7/

Nowadays most of the VoIP devices supports SIP protocol because the code is smaller but SIP only supports basic features and all the advanced features are supported through the separate Internet Standards. However in this project we are going to use our server for simple phone calls so SIP is the best choice in this case.

First when a caller calls a SIP ID, SIP sends an INVITE message to the caller which includes the SDP (session description protocol), audio endpoint IP address say 1.1.1.1, port say p1 and codecs that can be used. This information is then forwarded to the calle. When the calle receives the information or call it replies with 200 saying OK and saying its audio endpoint IP say 2.2.2.2, port say p2 and codecs chosen. At this point the caller start sending the RTP packets from 1.1.1.1:p1 to 2.2.2.2:p2 while the calle sends from 2.2.2.2:p2 to 1.1.1.1:p1 i.e. to say that the audio path is direct and SIP messages are transferred through intervening proxies.

But this does not applies when we are using Asterisk because the default behavior is to set up the two legs as two separated audio streams. While using Asterisk, the caller sends INVITE informing the IP say 3.3.3.3 and codecs it can use. Now the Asterisk server decides where the another leg is and if it found that the calle device is SIP it sends the fresh SIP INVITE to the calle with its own IP address say 5.5.5.5 and the port it has chosen say p5. Now this message is received by the second phone which replies OK and IP say 4.4.4.4 and the codec chosen. Now the asterisk forwards this message to the caller with its own IP 5.5.5.5, port used say p7 and choose codec. By this way the communication is setup, the caller send packets from 3.3.3.3 to 5.5.5.5:p7 and the calle sends packets form 4.4.4.4 to 5.5.5.5:p5. In this way the Asterisk acts as a media path, all RTP packets are received by Asterisk and then resent in both the direction. It is also possible to set up media path directly between two SIP devices or endpoints where SIP message

sent by Asterisk to calle will contain all the IP of the caller and vice versa. This process is also call native bridging i.e. two compatible endpoints are connected directly instead of relaying to Asterisk for audio. However this bridging process sounds almost like SIP proxy. /9/ Figure 4 describes the SIP process.

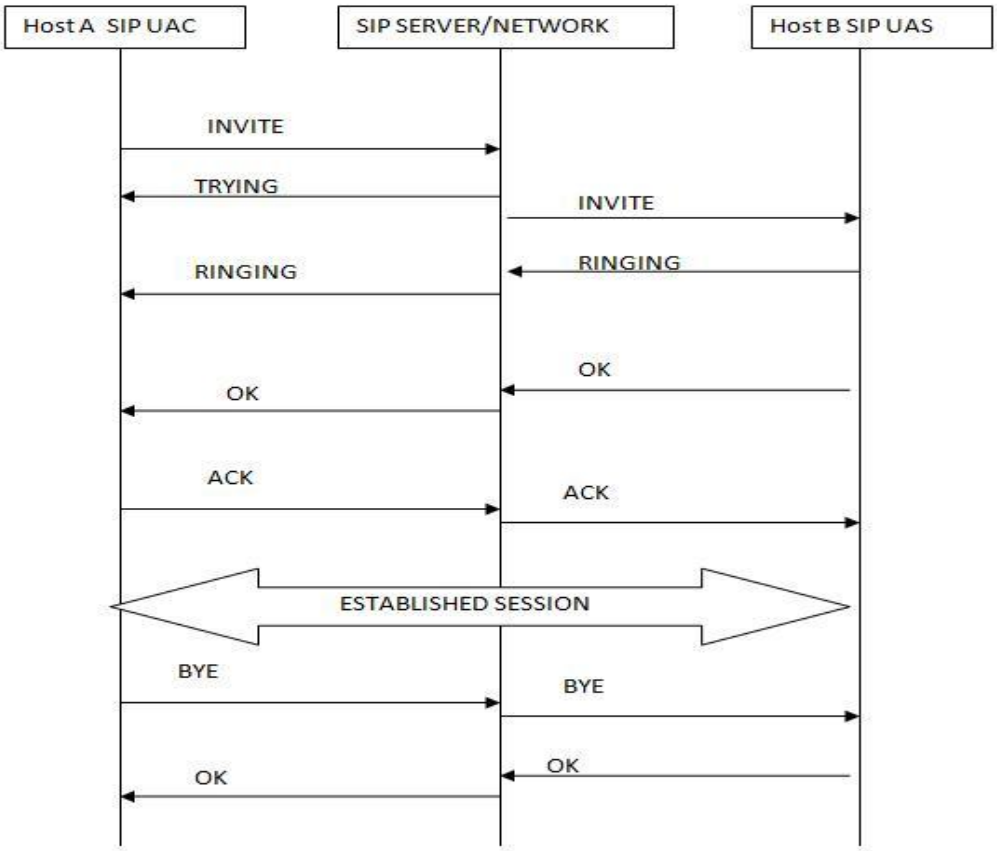


Figure 4. SIP session /8/

2.5 Firewall

A **firewall** is a part of a computer system or network that is designed to block unauthorized access while permitting authorized communications. It is a device or set of devices that is configured to permit or deny network transmissions based upon a set of rules and other criteria. There are several types of firewall techniques:-

- Packet filter: - In packet filter firewall it take cares of each packet which is entering or leaving the network whether it has to accept or rejected by user define rules. And it is effective.
- Application gateway: - It applies security mechanisms to specific applications like FTP server. this is very effective
- Circuit-level gateway: -It applies security mechanisms after a TCP or UDP connection is made. After having connection, packet can flow between the hosts without further checking.
- Proxy server: - It intercepts messages entering or leaving the network. And it hides the true network addresses.

The Linux kernel by default includes the net filter firewall subsystem which is a type of packet filtering. The filtering rules are defined in iptables. The packets are handed to the netfilter subsystem to decide whether to accept or drop the packets. But in our case we will use Shorewall firewall since it is regarded as good firewall for small business because of its efficiency and flexibility. Shorewall is a shoreline firewall and more commonly known as ”Shorewall” is a high level firewall for configuring firewall/Netfilter in Linux. This firewall is effective because we can define our firewall requirement entries in a set of configuration file. While we run Shorewall it reads the configuration file and practices those as a firewall rules.

3. IMPLEMENTATION OF THE PROJECT

3.1 Overview of the design

The overview of the network structure is shown below in the Figure. The ADSL router is the main gateway to the internet which is connected to our Linux server which is acting as DHCP server, File server, Web Cam server, Asterisk server and Firewall. So a single machine is working as a multiple servers. The Linux machine is then connected to the switch and the clients are connected to the switch.

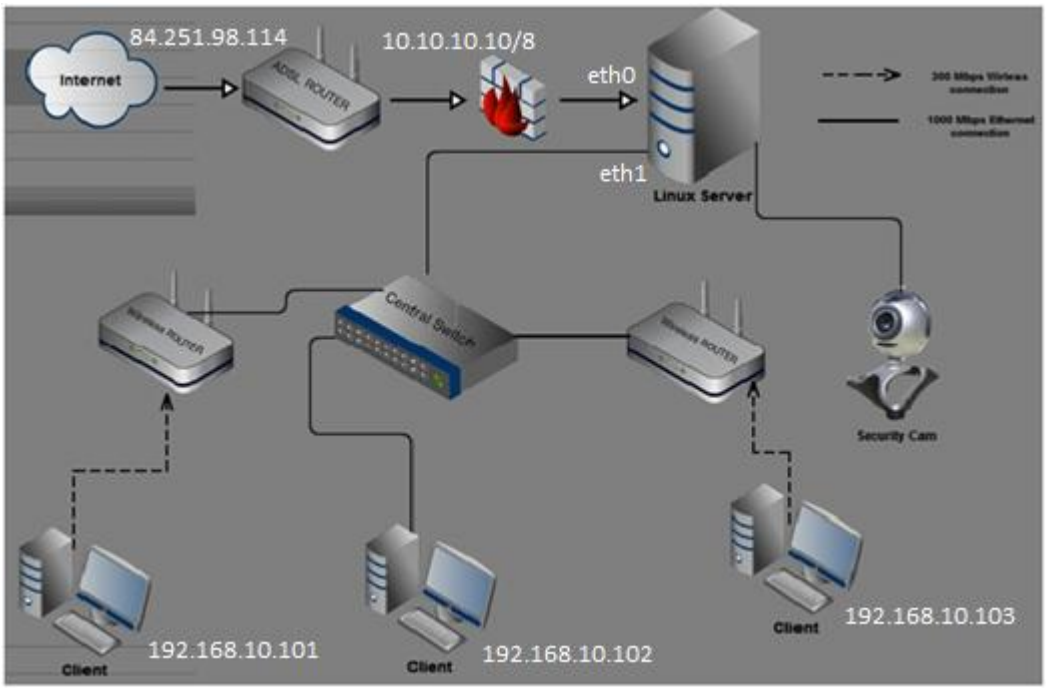


Figure 5. Overview of design

Based on the overview of design our network topology will be star topology. All the client computers are connected to one central switch or hub which acts as a channel for message transmission. The central node is the common connection point for all the nodes. All the incoming and outgoing data passes through the switch which also acts as a repeater and controls all network functions. Each node, which can be file server, asterisk server, peripheral devices or workstations are directly connected to the central switch. The main advantages of using star topology is to decrease in the chances of network failures, better inspection of the traffic through the network and the network is easily expandable. Its disadvantages

are network failure if the central switch fails and the reliability the system highly depends on the functioning of the switch.

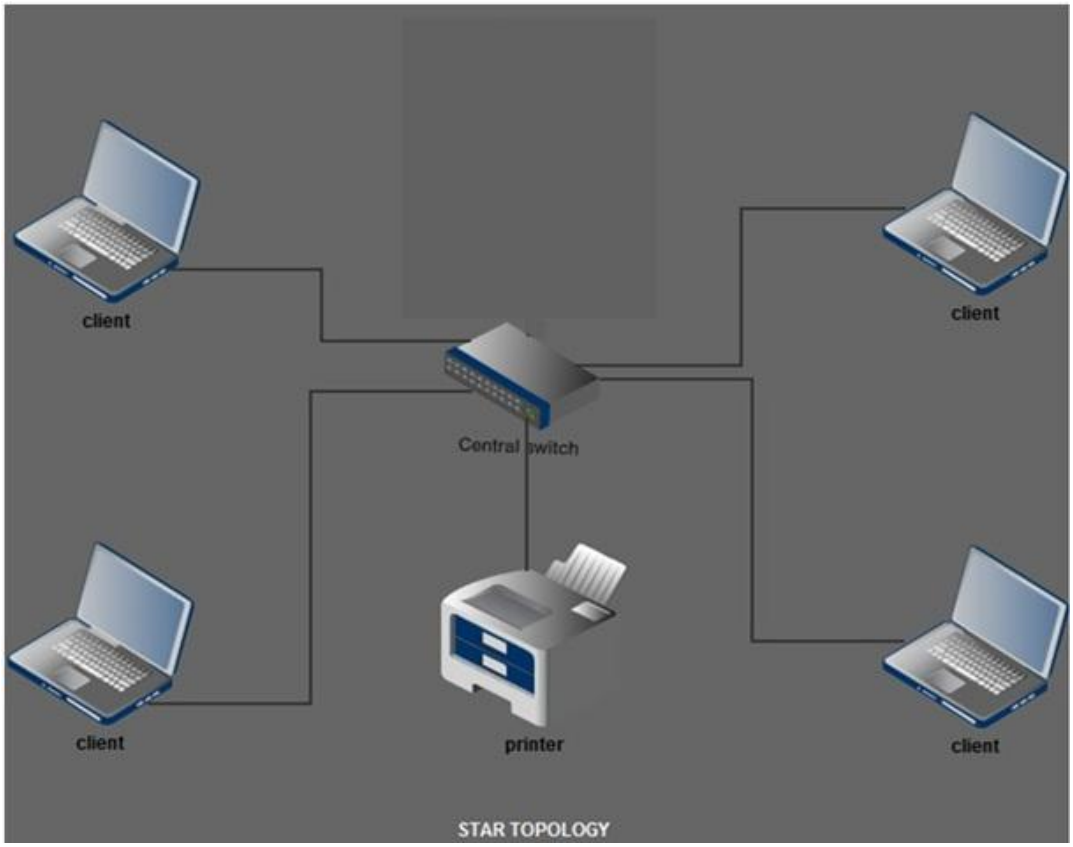


Figure 6. Network topology

3.2 Environment of the project

Before implementing in the real environment it is necessary to set up an environment similar to it and perform testing there. The Figure 7 shows our working or testing environment or network. We are provided a public IP from the service provider which is directly connected to our Linux machine, so it is also possible for remote login from external network in the lab environment. The screen shot in this report are taken from the lab sessions. The ADSL router was configured at the end in working location and then our system were tested again.

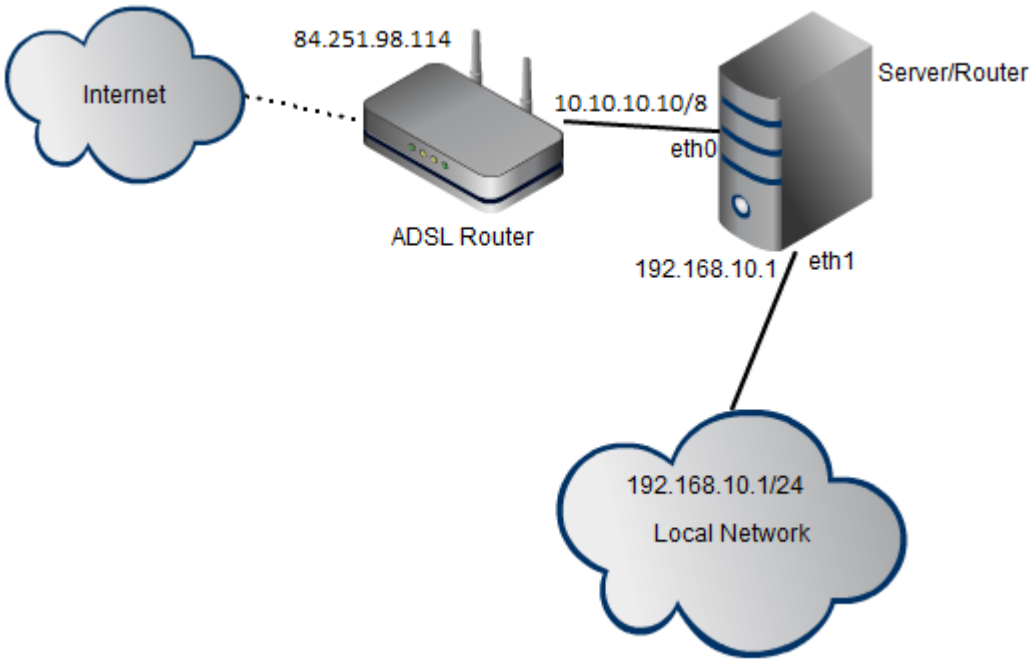


Figure 7. Working Environment

3.3 Configuration of ADSL router

The ADSL router used was ZyXEL P-600HW-D1 (F). The default gateway is 192.168.1.1 from where we can start our configurations. After connecting to the router, open the webpage with the IP of default gateway which will prompt a login window. After login ***Go to Advance setup*** was clicked.

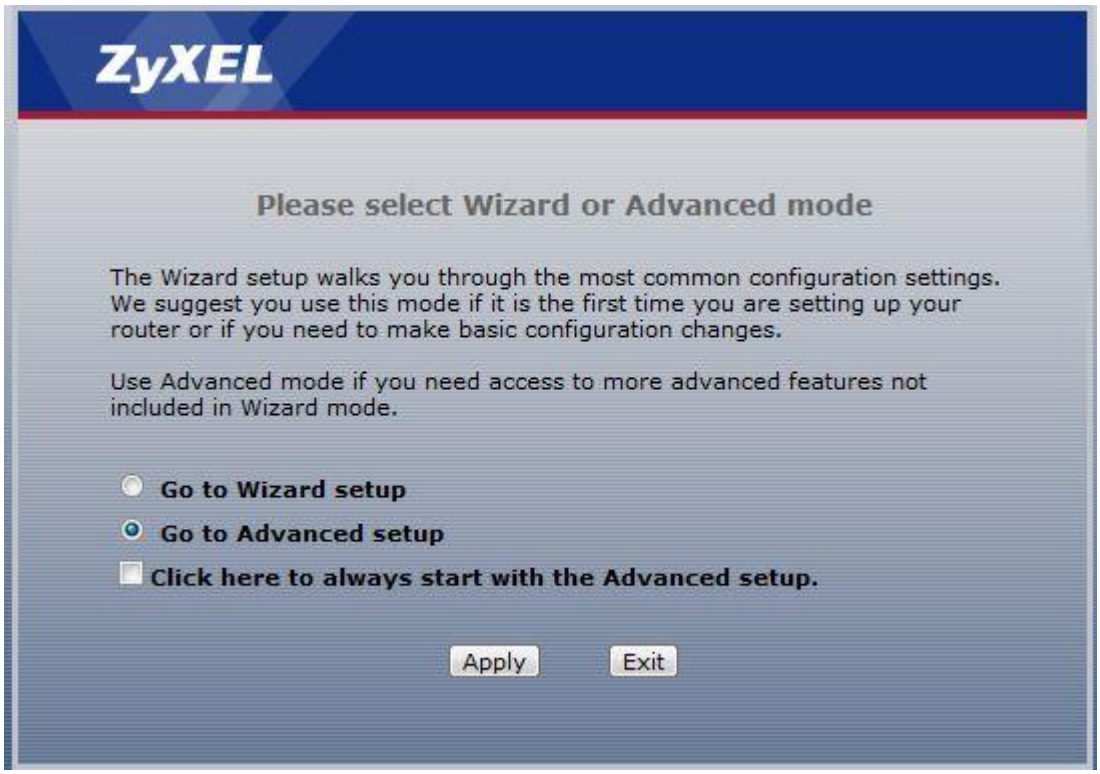


Figure 8. Advance setup mode

The changes made must be followed by ***apply*** button click to make it working. The configuration steps are described below:

- ***Network > WAN > Internet connection***:-Defines how the router will get IP address.

Figure 9. IP assignment for Router

- ***Network > LAN > IP***:-Define the IP to be allocated to clients. The new configuration page will open at 10.10.10.10 IP address which is also the gateway IP for clients.

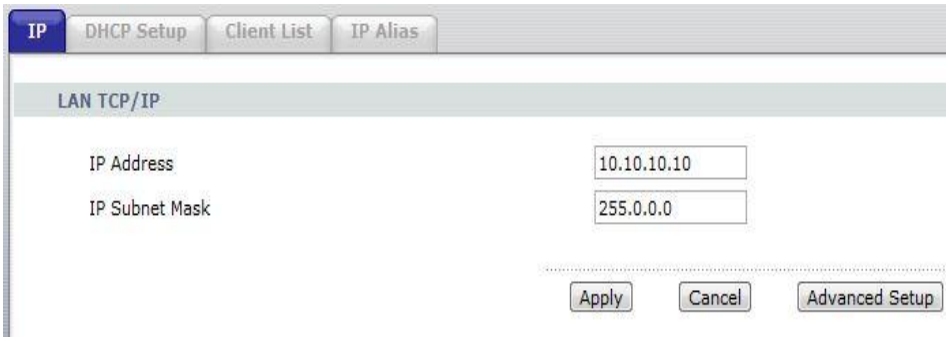


Figure 10. IP for LAN networks

- ***Network > LAN > DHCP Setup***:- Allocate pool of address

Figure 11. Define pool of address available

- ***Network > Advanced > NAT***:- Enable port Forwarding

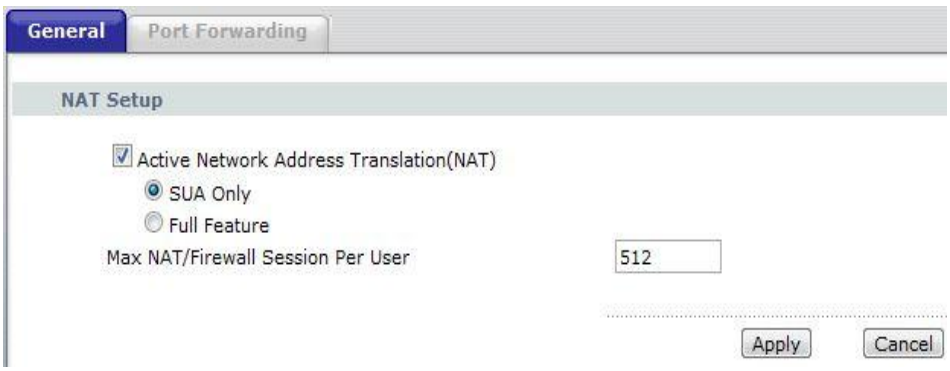


Figure 12. NAT

- ***Security*** **>** ***General*** :- Enable Firewall.

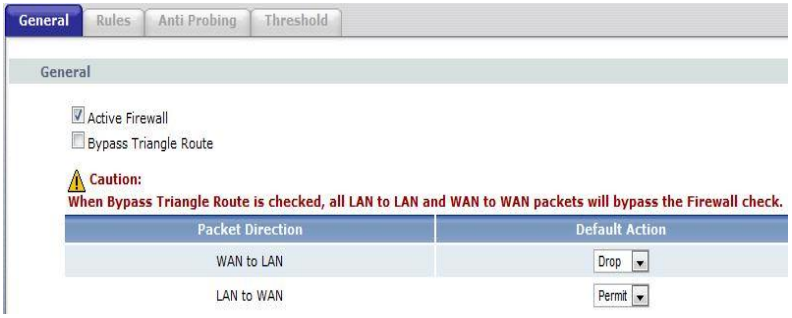


Figure 13. Enable Firewall

- ***Network*** **>** ***Wireless LAN*** **>** ***General***:- Securing Wireless Connection.

Figure 14. Enable password

The ADSL router is set to be working in Router mode. It will convert the public IP to private IP. Our Server will also be a Router meaning ADSL router connected to our Router through network interface ”eth0”. The local network is connected to network interface ”eth1” form the server. So the interface ”eth0” will get the pri-vate IP form DHCP of ADSL router and is DHCP server and default gateway for our local network. The private IP are made different i.e. ADSL router will assign IP 10.10.10.0/24 and our DHCP server will assign 192.168.10.0/24 to the clinets.

3.4 Configuration of DHCP Server

The configuration file is located at /etc/dhcp/dhcpd.conf. This file includes necessary configuration for the DHCP sever. The configuration are shown below:

```
Max-lease-time 7200;
Default-lease-time 7600;
Option-subnet-mask 255.255.255.0;
option broadcast-address 192.168.10.255;
option routers 192.168.10.1;
option domain-name-servers 192.168.10.1, 8.8.8.8;
option domain-name "sanjeeb_Server";
subnet 192.168.10.0 netmask 255.255.255.0 {
option netbios-name-servers 192.168.10.1;
range 192.168.10.100 192.168.10.254;
}
host AP0024A514ADE0 {
  hardware ethernet 00:0A:E4:E6:62:21;
 fixed-address 192.168.10.5;
}
```

The syntax in the configuration is described below:

Syntax	Description
Max-lease-time	Maximum time duration the IP offered is valid.
Default-lease-time	Default duration.
Option subnet-mask	Maximum no. of hosts that can be in that network.
Option broadcast-address	Defines the broadcast address to be used by the client to broadcast the message in the network.
Option routers	Routing, defines the address where the traffic generated by the clients to be sent.
Option domain-name-servers	Informs the address to be used for name resolution.
Option domain-name	Name of the domain.
Subnet.....netmask.....	Defines the subnet and its netmask.
Option netbios-name-servers	Network basic input output system.
Range	Range of the available IP. Defines the pool of address.
Host.....	Host name.
Hardware ethernet	MAC address of the host.
Fixed-address	Static address to be assigned to the mac address.

Table 3. Description of the syntax configured

In the configuration above we can see that DHCP server will assign IPs from the range i.e. from 192.168.10.100 to 192.168.10.254 while in the second configuration the client with the specified MAC address will receive the fixed address 192.168.10.5 whenever it is connected. So assuming that the client is connected broadcasts discovery address which includes the MAC address of the client, the DHCP server will first look in the "dhcpd.leases" files to check the assigned address and then offers the IP from the free available address but if in case the MAC address is defined to be given the static it offers the static IP address to the client. Configuring the static IP is necessary in certain host like webserver, mail servers etc. The disadvantage of using static IP is that if the client is rarely connected to the network, the use is for small time and the DHCP server always reserves the IP address.

3.5 IP Forwarding and NAT

Even though we have ADSL router connected to UNIX server acting in router mode, we will enable IP forwarding and NAT in our server so that it will act as router. Now we have ADSL router connected to our router. To enable IP forwarding we must configure the /etc/sysctl.conf file and add the following line:

net.ipv4.ip_forward=1

We also need to configure the /etc/network/interfaces to set where the DHCP Server will listen. The configuration file is show below:

auto lo

iface lo inet loopback

auto eth0

auto eth1

iface eth1 inet static

address 192.168.10.1

netmask 255.255.255.0

broadcast 192.168.10.255

network 192.168.10.0

The loopback interface"lo" set to auto i.e. it is automatically 127.0.0.1 and also the eht0 which is interface to the ADSL router. The interface eth1 is set static which is the server's router's address. Network, broadcast and netmask are also defined there. Here the rules are also set to eth0 interface for network address translation purpose. The rules are configured in /etc/rc.local file as shown:

/sbin/iptables -P FORWARD ACCEPT

/sbin/iptables --table nat -A POSTROUTING -o eth0 -j MASQUERADE

exit 0

Network address translation is special technique to modify the IP address information in IP packet headers while it passes across a traffic routing device. Here we have used is many-to-one NAT i.e. many private IP behind the network are translated to one public IP address.

To start/stop DHCP server: /etc/init.d/isc-dhcp-server start/stop

Or we can use service isc-dhcp-server start/stop

After starting the DHCP daemon, it invokes OK output, if properly configured. In case if nothing is seen or it yields Fail message then there exists certain errors in the configuration. The error message can be checked through /var/log/syslog files. The DHCP server will save the lease of IP offered in /var/lib/dhcp/dhcpd.leases which must be manually created before starting the dhcp server.

3.6 Configuration of File Server

Running the command below will install Samba file server:

sudo apt-get install samba libpam-smbpass

The configuration file of Samba file server is located at /etc/samba/smb.conf. Now to start configuration of samba, edit the file in the editor and changes the configuration to fulfill our requirements. In the global section as shown in the picture, workgroup is the name of the domain and security level is User. In the domain section, Domain logon value is yes and users can logon to the domain. Default profile path and user home directory path machine script and logon script is configured. Logon script runs when a user is logged on into the domain and machine script runs when a machine is added to the domain. After that share folder has to be configured, Users home folder shared only for the user own access and a public folder are shared for all the users to access. After that the netlogon directory, public share directory, a logon.cmd script file and a group named machines have to be created to work samba properly. The configuration is shown in Appendix1.

The following actions were performed to edit or create suitable files for samba configurations:

sudo mkdir -p /srv/samba/netlogon
sudo mkdir -p /etc/public
sudo touch /srv/samba/netlogon/logon.cmd
sudo addgroup machines
Now, edit the logon script file with the following command:

nano -w /srv/samba/netlogon/logon.cmd
net use Y: \\192.168.10.1\public
After all the files had been configured, now restart the samba services to work properly.

3.7 Configuration of cam server

Running the command below will install the webcam server:

apt –get install webcam-server

However installing the webcam server only is not enough; we should manually create a file and add the script where we can adjust our settings. The file is created as below and the scripts are also shown.

"touch /etc/init.d/webcam-server" and write the script.

The webcam-script is shown below:

```
#!/bin/sh
SERVER_BIN=webcam-server
LOCK_FILE=/var/lock/$SERVER_BIN
RTRN=0
OPTIONS="-v –g 320*240  -p 8081 –c 192.168.10.1"

start() {
   [ -f $LOCK_FILE ] && echo "$SERVER_BIN already started"
   [ -f $LOCK_FILE ] && return
   echo -n "Starting $SERVER_BIN: "
   export LD_PRELOAD=/usr/lib/libv4l/v4l1compat.so
   nohup $SERVER_BIN $OPTIONS > /dev/null 2>/dev/null &
   RTRN=$?
   [ $RTRN -eq 0 ] && echo Started! || echo FAIL
   [ $RTRN -eq 0 ] && touch $LOCK_FILE
}

stop() {
   [ -f $LOCK_FILE ] || echo "$SERVER_BIN is not running"
   [ -f $LOCK_FILE ] || return
   echo -n "Stopping $SERVER_BIN: "
   pkill -f "$SERVER_BIN $OPTIONS"
   RTRN=$?
   rm -f $LOCK_FILE
   [ $RTRN -eq 0 ] && echo Stopped! || echo FAIL
}

case "$1" in
   start)
      start
      ;;
```

```
   stop)
      stop
      ;;
   restart)
      stop
      start
      ;;
   *)
      echo "Usage: $0 {start|stop|restart}"
      RTRN=1
esac

exit $RTRN                    /11/
```

In this configuration file we can define the resolutions, ports and the hostname or IP address.

*OPTIONS=”-v –g 320*240 -p 8081 –c 192.168.10.1”*

The resolution is set to 320*240 because the camera we are using is not designated for High Definition Videos or pictures. The option –p 8081 defines the port number which is used for the communication and the option –c is for the IP address. This configuration is not active for the motion sensor and if we want to make the videos being captured to be save whenever the motion is detected we need to install motion which can be obtained by following command:

apt –get install motion

To view the webcam live from internet pages apache server should be installed as below:

apt –get install apache2

Now the default files from webcam-server were copied to apache server so that we are able to view the webcam. The files to be copied are:

cp /usr/share/doc/webcam-server/applet/* /var/www/

The file named webcam.html in /var/www/ now can be used to modify the webpage. The html configuration are show below:

<html>

<head>

<title>Sanjeeb Security WebCam</title>

</head>

* *

* *

* *

<APPLET CODE = "WebCamApplet.class" archive="applet.jar" WIDTH = "320" HEIGHT = "240">

<param name=URL value="http://91.156.186.56:8888/">

<param name=FPS value="60">

<param name=width value="320">

<param name=height value="240">

</APPLET>

* *

</body>

</html>

The configuration is plain html where we define the height and width of the webcam screen. If needed extra information can also be added so that the client will be informed about the location and angle of the camera from where the captures are being taken. To run the webcam server and enable motion we simply run the following commands:

#/etc/init.d/webcam-server start and #motion

3.8 Configuration of Asterisk server

Installing an Asterisk needs a computer with Linux installed. To install Asterisk the following commands are used:

apt-get install build-essential libxml2-dev ncurses-dev

apt-get install asterisk

This will install all the necessary files we need to operate an Asterisk server 1.8. Now we need to configure file named sip.conf located at /etc/asterisk/sip.conf. The configurations are shown below:

[general]

bindport = 5060

bindaddr = 0.0.0.0

tcpbindaddr = 0.0.0.0

tcpenable = yes

On the general part we have defined the “bindport” that is used during SIP sessions. Here “bindaddr” and “tcpbinaddr” are set to 0.0.0.0 i.e. unspecified address meaning that the Asterisk server will accept connection from any IP in TCP sessions. This gives the broad range for the users. Specifying a specific IP means that TCP sessions originated by that host is allowed. Since the broadcast address for 0.0.0.0 IP is 255.255.255.255 connection from any IP to any IP i.e. from any host to any host are allowed. The file continues as below:

[sip104xxxx]

type=peer

username=sip104xxxx

secret=yyyyyyy

context=default

host= dynamic

fromuser=sip103xxxx

auth=md5 ; maybe, maybe not

insecure=very ; also maybe, maybe not

disallow=all

allow=gsm

allow=ulaw

; add other "allow" lines as needed

[sip103xxxx]

type=peer

username=sip103xxxx

secret=yyyyyyy

context=default

host= dynamic

fromuser=sip103xxxx

auth=md5 ; maybe, maybe not

insecure=very ; also maybe, maybe not

disallow=all

allow=gsm

allow=ulaw

; add other "allow" lines as needed

In this part we have defined the SIP users. SIP user accounts are created and features for the users are defined. The initiation of session are encrypted by the line auth=md5. The GSM standard is used for the users. Here each SIP users are manually added, so a user cannot use Linux account to make calls through Asterisk server. All users ID are started form word “sip”. Since the type defined is peer we need to create pair for the SIP users independently and allow the permissions. The configuration is shown below and the file is located at /etc/asterisk/extensions.conf.

[default]

exten => sip103xxxx,1,Answer()

exten => sip103xxxx,n,Dial(SIP/sip103xxxx,20,tr)

exten => sip103xxxx,n,Hangup

exten => sip104xxxx,1,Answer()

exten => sip104xxxx,n,Dial(SIP/sip104xxxx,20,tr)

exten => sip104xxxx,n,Hangup

In this file the permission for each user are defined explicitly. In the configuration above the SIP user sip103xxxx have permission to answer, dial and Hang-up. But the problem arises when the number of SIP user’s increases because each SIP user should be explicitly added to the configuration files and assigned the permission. The asterisk server can be started using this command: ***asterisk –r.***

3.9 Configuration of Firewall

The step by step configurations of firewall are described below:

- ***Setting up Shorewall >Ubuntu Installation***

  Shorewall can be found suitable to be installed in two ways. Either we can install directly from the Ubuntu repository or we can download a setup file i.e. (.deb / Debian). Best way for installing Shorewall is by execution of following commands.

  Sudo apt-get install Shorewall

  To install the file we should be a super user which can be done by "**sudo"** command. After executing this command our server automatically search for the repository, locate the Shorewall configuration file and install. After successful installation of Shorewall we have to do some further things to make Shorewall fully working. Below I am describing those steps for setting up Shorewall as a firewall.

- ***Making Running on startup***

  First we have to set our Firewall to be executed every time we start our server. However our server will run 24/7/365 but if it has been stopped or shut down by any mean then in next start our Firewall also should execute automatically. For these setting we need to change simply a line of a configuration file **/etc/Shorewall/Shorewall.conf.**

  And change the following line.

  STARTUP_ENABLED=Yes

  We have changed the **STARTUP_ENABLE = No** to **STARTUP_ENABLE = Yes.** This setting will be saved and in every next restart our firewall will start automatically.

- ***Copying default configuration file from Shorewall source location to /etc/Shorewall/***

  Now we have to copy all the configuration files from the source location of Shorewall to our default location of Shorewall (/etc/Shorewall). Since

we are using two interfaces, we need to copy all the files which are made for two interfaces. Shorewall provided default configuration files for at least three interfaces system.

- ***Setting up Interfaces and save its configuration file > /etc/Shorewall/interfaces***

  Now we have to setup the entire interfaces available in our server and set a record in a Shorewall interfaces configuration file. By default while we re-start after a fresh installation of Shorewall it automatically refresh the interfaces and make a default entry but we can also manually enter the needed interfaces. Net defines the interface connected to ADSL router and loc defines the local interface.

  #ZONE INTERFACE BROADCAST OPTIONS

  net eth0 detect dhcp,tcpflags,nosmurfs,routefilter,logmartians

  loc eth1 detect tcpflags,nosmurfs,routefilter,logmartians

- ***Setting up policy and save its configuration file > /etc/Shorewall/policy***

  Policy is a setup which works a backup rules if no others rules are not applied in our firewall configuration. This is a default rules for Shorewall firewall. We define our policy as below.

  #SOURCE DEST POLICY LOG LIMIT: CONNLIMIT:
  # LEVEL BURST MASK
  loc net ACCEPT
  net all ACCEPT info
  all all ACCEPT info

- ***Setting up rules and save its configuration file > /etc/Shorewall/rules***

  Rules are most important and main configuration in Shorewall. We define our individual firewall configuration in this file. Every time while we need to add and remove some firewall setting then this configuration file is used. This have carries all the firewall setting and execute while Shorewall

is running. If there is any connection came up through our server then it pass through this rules and if the firewall setting match for the connection then the firewall takes effect as accordance. Our default rules configuration file is as below:

#ACTION SOURCE DEST PROTO DEST COMMENTS

ACCEPT net loc icmp echo-request

ACCEPT loc:192.168.10.102 $FW tcp http

ACCEPT loc:192.168.10.103 $FW icmp echo-request

ACCEPT net loc:192.168.10.102 tcp smtp

ACCEPT net loc:192.168.10.102 tcp pop3

ACCEPT loc:192.168.10.0/24 $FW tcp 22

ACCEPT loc:192.168.10.0/24 $FW tcp 5060

ACCEPT loc:192.168.10.0/24 net tcp 80

ACCEPT loc:192.168.10.0/24 $FW tcp 80

ACCEPT loc:192.168.10.1 $FW tcp 80

ACCEPT loc:192.168.10.1 net tcp 80

REJECT net tcp 80

In above firewall rules configuration file we have setup access for **ping, http, pop and smtp** from an local interface 192.168.10.102 and only grant access of **ssh** for all local computer connected to our server.

- ***Shorewall Start / Stop***
  /etc/init.d/Shorewall start (to start)
  /etc/init.d/Shorewall stop (to stop)
- ***Shorewall Log***
  # tail –f /var/log/syslog | grep (keyword)

4. TEST, RESULTS AND ANALYSIS

4.1 Testing ADSL router

The ADSL router itself provides some testing and diagnostic tools. The steps used for testing are described below:

- ***Maintenance > Diagnostic > General*** :- Ping testing

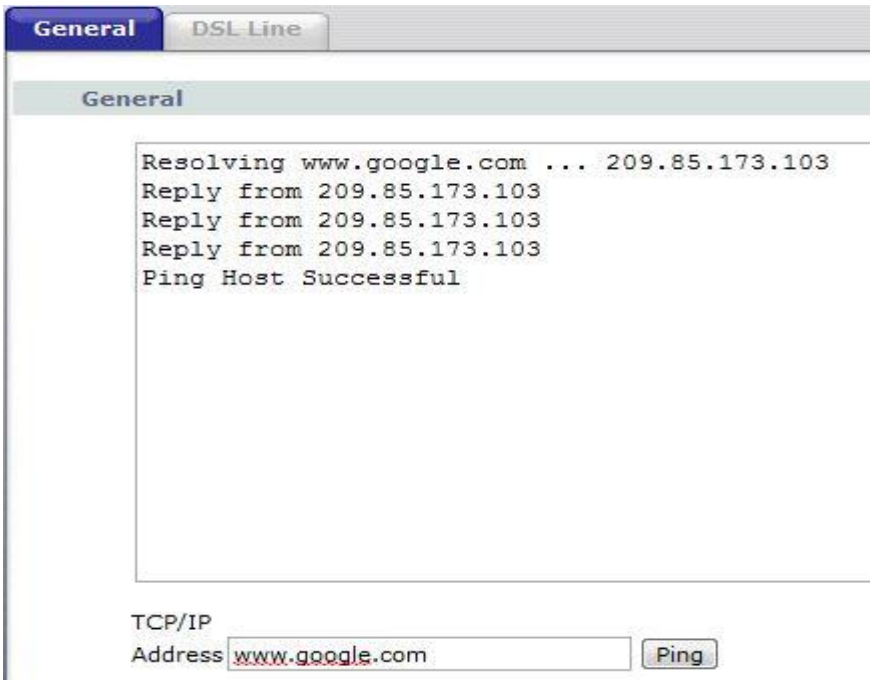


Figure 15. Pinging website

- Checking the client IP address and default gateway.

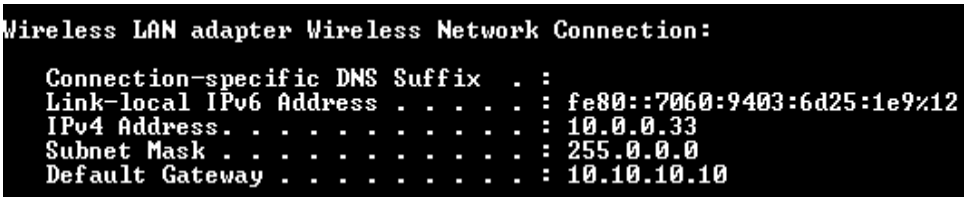


Fig 16: IP verifying

- ***Network > LAN > Client List***:- View the clients connected.

Figure 17. Client list connected to the router

- ***Security > Firewall >*** :- Check packets dropped and allowed.

#	Time	Message	Source	Destination	Notes
1	01/01/2000 00:41:08	Packet without a NAT table entry blocked: ICMP(Echo)	10.14.253.5	80.223.39.94	ACCESS DROPPED
2	01/01/2000 00:41:08	board 0 line 0 channel 0, call 10, C01 Incoming Call 1500000			CALL DETAIL RECORD
3	01/01/2000 00:41:07	Router reply ICMP packet: ICMP(Host Unreachable)	10.10.10.10	10.0.0.33	ACCESS PERMITTED
4	01/01/2000 00:41:06	board 0 line 0 channel 0, call 9, C02 Call Terminated			CALL DETAIL RECORD
5	01/01/2000 00:40:30	Firewall default policy: TCP (W to W/PRESTIGE)	113.140.39.136:6000	80.223.39.94:1433	ACCESS DROPPED
6	01/01/2000 00:40:02	board 0 line 0 channel 0, call 9, C01 Incoming Call 1500000			CALL DETAIL RECORD
7	01/01/2000 00:40:01	board 0 line 0 channel 0, call 1, C02 Call Terminated			CALL DETAIL RECORD
8	01/01/2000 00:37:32	Firewall default policy: TCP (W to W/PRESTIGE)	94.23.237.143:28665	80.223.39.94:80	ACCESS DROPPED
9	01/01/2000 00:34:37	Firewall default policy: TCP (W to W/PRESTIGE)	116.11.252.195:1330	80.223.39.94:110	ACCESS DROPPED
10	01/01/2000 00:31:21	Successful WEB login			User:admin

Figure 18. Firewall

- Clicking the ***status*** will display the current status and configurations of the router:

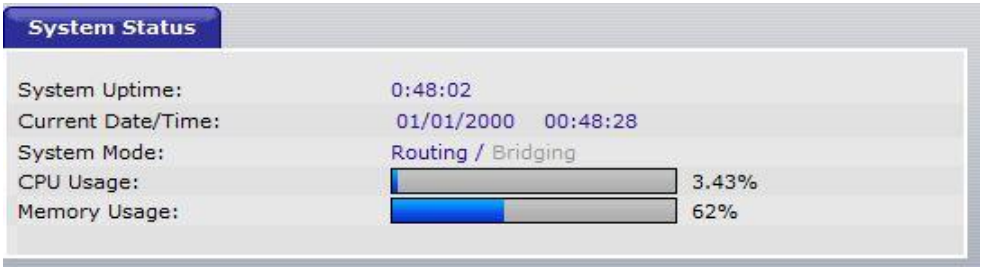
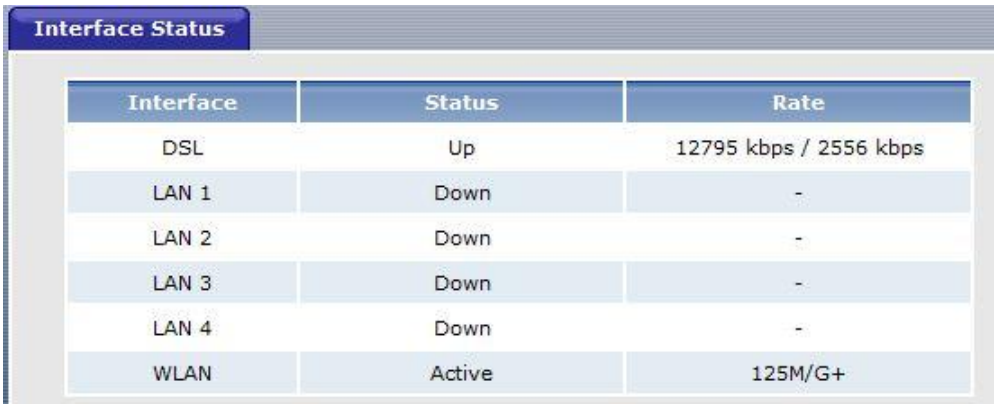


Figure 19. System status

This will provide the information regarding Memory and CPU usages and the time duration for how long it has been working or up. It also shows the mode, in our case it is Routing.

From the Figure 20 we can say that the ADSL router we are using is ASDL2+ because the upstream rate is 12795 kbps and the downstream rate is 2556 kbps.



Interface Status

Interface	Status	Rate
DSL	Up	12795 kbps / 2556 kbps
LAN 1	Down	-
LAN 2	Down	-
LAN 3	Down	-
LAN 4	Down	-
WLAN	Active	125M/G+

Figure 20. Interface status

Device Information

Host Name:	
Model Number:	P-660HW-D1
MAC Address:	50:67:f0:8d:e9:61
ZyNOS Firmware Version:	V3.40(AGL.9) | 12/07/2009
DSL Firmware Version:	TI AR7 08.00.03.00
WAN Information	
- DSL Mode:	Error
- IP Address:	80.223.39.94
- IP Subnet Mask:	255.255.240.0
- Default Gateway:	80.223.32.1
- VPI/VCI:	0/33
LAN Information	
- IP Address:	10.10.10.10
- IP Subnet Mask:	255.0.0.0
- DHCP:	Server
WLAN Information	
- SSID:	ZyXEL
- Channel:	6
- Security:	Disable
Security	
- Firewall:	Enabled
- Content Filter:	Disable

Figure 21. General overview

The Figure 21 gives the brief overview of the system current mode. We can view the public IP assigned at that moment, the IP address for LAN, default gateway for LAN and the role of router is DHCP. We can see that the DSL mode state is shown error because in the upstream and down-stream rate are not same. Further we can view the channels used by WLAN and the security status also.

4.2 Testing DHCP Server

The client successfully configures the IP address in the defined range when connected to the server. The Figure 22 shows the “ifconfig” command results and we can see eth1 have received the correct IP address.

```
root@sanserver:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:44:09:a4:04
          inet addr:91.156.185.185  Bcast:91.156.187.255  Mask:255.255.252.0
          inet6 addr: fe80::202:44ff:fe09:a404/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:548563 errors:0 dropped:0 overruns:0 frame:0
          TX packets:357995 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:760629958 (760.6 MB)  TX bytes:28642883 (28.6 MB)
          Interrupt:22 Base address:0xd800

eth1      Link encap:Ethernet  HWaddr 00:e0:4c:e0:00:48
          inet addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:4cff:fee0:48/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:345744 errors:0 dropped:0 overruns:0 frame:0
          TX packets:530953 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26548766 (26.5 MB)  TX bytes:745747151 (745.7 MB)
          Interrupt:23 Base address:0xd400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:346 errors:0 dropped:0 overruns:0 frame:0
          TX packets:346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:34310 (34.3 KB)  TX bytes:34310 (34.3 KB)
```

Figure 22. ifconfig output

For to make sure that our traffic is passing through the DHCP interface, we use tracert command from windows client, output shown in Figure 23.

```
Tracing route to google.com [173.194.32.34]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  SANSERVER [192.168.10.1]
  2    33 ms     2 ms     1 ms  10-2-1.vaahov1.fi.elisa.net [139.97.18.217]
  3     2 ms     2 ms     2 ms  ge1-2-0.seisei-p1.fi.elisa.net [139.97.18.218]
  4     7 ms     7 ms     7 ms  ae2.helpa-gw1.fi.elisa.net [139.97.6.250]
```

Figure 23. Traceroute

From Figure 23 we can see any request from client first goes to the router address i.e. 192.168.10.1, so we can say that the traffic is passing through default gateway.

We can verify the client and server are connected. The verification was done by pinging client to server and vice versa as shown below:

Clinet to Server

```
Pinging 192.168.10.1 with 32 bytes of data:
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
```

server to client

```
sanjeeb@sanserver:~$ ping 192.168.10.5
PING 192.168.10.5 (192.168.10.5) 56(84) bytes of data.
64 bytes from 192.168.10.5: icmp_req=1 ttl=128 time=0.605 ms
64 bytes from 192.168.10.5: icmp_req=2 ttl=128 time=0.523 ms
64 bytes from 192.168.10.5: icmp_req=3 ttl=128 time=0.652 ms
64 bytes from 192.168.10.5: icmp_req=4 ttl=128 time=0.578 ms
```

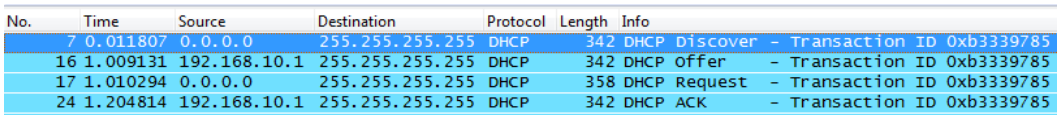
Figure 24. ping-pong

We can see the leases of assigned IP in file /var/lib/dhcp/dhcpd.leases. In this file we can see the assigned IPs by the DHCP server, the time duration till the IP is valid for the client and date and time when the lease started and when it will end.

```
lease 192.168.10.103 {
  starts 2 2012/02/28 08:30:29;
  ends 2 2012/02/28 10:30:29;
  tstp 2 2012/02/28 10:30:29;
  cltt 2 2012/02/28 08:30:29;
  binding state free;
  hardware ethernet 00:0a:e4:e4:e1:10;
  uid "\001\000\012\344\344\341\020";
}
lease 192.168.10.102 {
  starts 2 2012/02/28 15:06:12;
  ends 2 2012/02/28 17:06:12;
  cltt 2 2012/02/28 15:06:12;
  binding state active;
  next binding state free;
  hardware ethernet 00:24:a5:14:ad:e0;
  uid "\001\000$\245\024\255\340";
  client-hostname "AP0024A514ADE0";
}
```

Figure 25. dhcpd leases file

The process of obtaining IP address from DHCP server is captured from client Ethernet adapter interface as shown below:



No.	Time	Source	Destination	Protocol	Length	Info
7	0.011807	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb3339785
16	1.009131	192.168.10.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0xb3339785
17	1.010294	0.0.0.0	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 0xb3339785
24	1.204814	192.168.10.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xb3339785

Figure 26. DHCP communication

Here we can verify that the client first sends the broadcast message with the source IP of 0.0.0.0 and destination IP 255.255.255.255. Then the DHCP server replies to the client as a broadcast message which includes the IP address offered, default lease time and mac address of the client, domain-name, subnet mask, router address and DNS servers address if requested. In response to the reply from DHCP server the client then sends message back informing the IP address offered is accepted or rejected in case multiple DHCP servers were present and client have already configured IP address from another DHCP server. If the client accepts the offer the DHCP server place the offered IP to dhcpd.leases files and remove it from the available free IP address pool. In case the client rejects the offer it places the IP address offered to the available pool for another client. In response the DHCP server sends the final message that the configuration process has ended. Transaction ID is unique identifier which identifies from which server the client has configured the IP address. This also informs the other DHCP servers if present in the network that the client with that specific MAC address has obtained the IP address from that specific DHCP server. This way 4-way hand shake is completed while configuring the IP address. The DHCP offer is shown below:

```
⊞ User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
⊟ Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xb3339785
    Seconds elapsed: 0
  ⊞ Bootp flags: 0x8000 (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.10.107 (192.168.10.107)
    Next server IP address: 192.168.10.1 (192.168.10.1)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Sony_0e:a7:6f (f0:bf:97:0e:a7:6f)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ⊟ Option: (t=53,l=1) DHCP Message Type = DHCP Offer
      Option: (53) DHCP Message Type
      Length: 1
      Value: 02
  ⊟ Option: (t=54,l=4) DHCP Server Identifier = 192.168.10.1
      Option: (54) DHCP Server Identifier
      Length: 4
      Value: c0a80a01
  ⊟ Option: (t=51,l=4) IP Address Lease Time = 2 hours
      Option: (51) IP Address Lease Time
      Length: 4
      Value: 00001c20
  ⊟ Option: (t=1,l=4) Subnet Mask = 255.255.255.0
      Option: (1) Subnet Mask
      Length: 4
      Value: ffffff00
  ⊟ Option: (t=15,l=13) Domain Name = "sanjeebserver"
      Option: (15) Domain Name
      Length: 13
      Value: 73616e6a656562736572766572
  ⊟ Option: (t=3,l=4) Router = 192.168.10.1
      Option: (3) Router
      Length: 4
      Value: c0a80a01
  ⊟ Option: (t=6,l=8) Domain Name Server
      Option: (6) Domain Name Server
      Length: 8
      Value: c0a80a0108080808
      IP Address: 192.168.10.1
      IP Address: 8.8.8.8
  ⊟ Option: (t=44,l=4) NetBIOS over TCP/IP Name Server = 192.168.10.1
      Option: (44) NetBIOS over TCP/IP Name Server
      Length: 4
      Value: c0a80a01
    End Option
    Padding
```

Figure 27. DHCP offer

4.3 Testing File Server

Testing of the file server is straight forward. A user must be able to login to the domain and should be able to access personal drive and the public drive or share. The Figure 28 shows a user being logged to the domain.

Figure 28. Login to SANJEEB.COM

After login to the domain the user can observer a Personal share and a Public share. Personal or Home share is only accessible by the defined user.

Figure 29. Home and Public share in Sanjeeb File server

The public share is accessible by all users in the domain and has write access to it also.

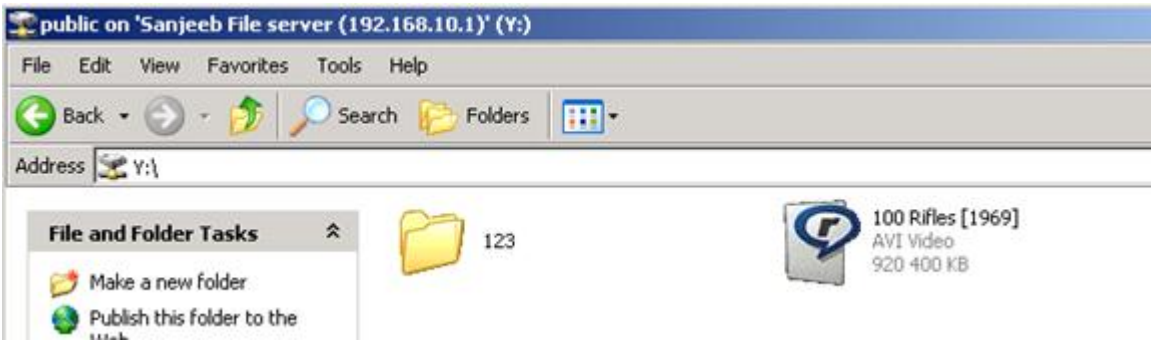


Figure 30. Creating files in Public share

The Figure 30 shows a user is creating a folder and share a video file in share directory. We can observe the router's IP address at the top bar which is 192.168.10.1. The Public directory is named Y: drive. Then we login using different user and test the accessiblity or rights of the next user in Public directory shown in Figure 31:

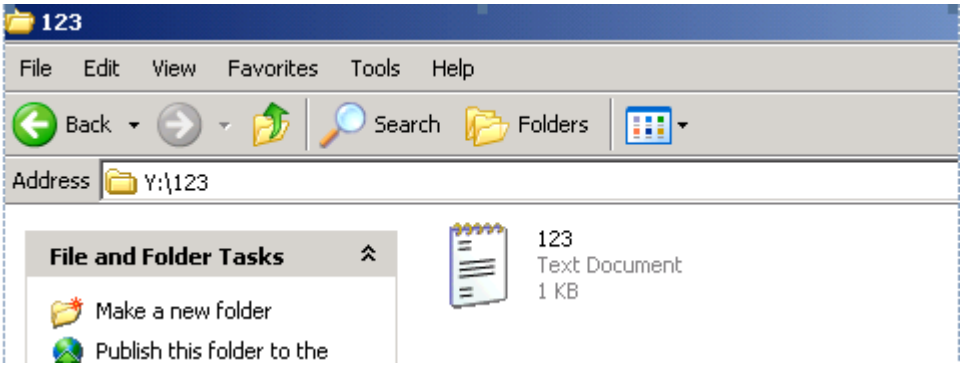


Figure 31. Using public directory

Next Home directory is tested by creating and saving files in corresponding user's directory as shown. The Home direcrtory is named Z:\ drive.

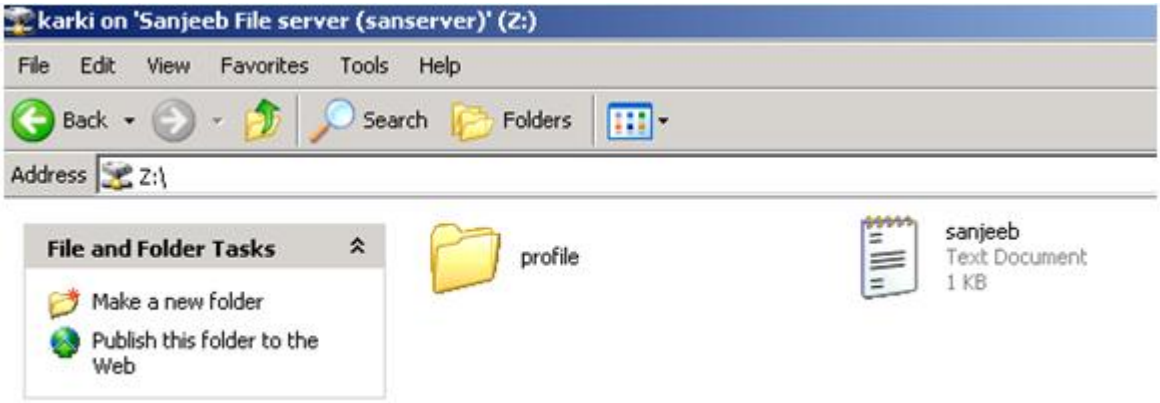


Figure 32. Home directory (Z:\ drive)

Creating file in Home directory is shown in Figure 33:

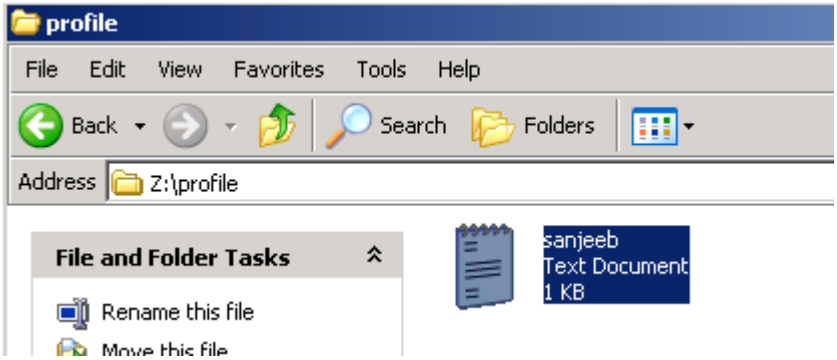


Figure 33. Create file in Home directory

Eventhough the samba documentaion says that all we need is to wait a minute or restart the deamons to see the effect of the changes we made to the samba configuraitons, it usually takes more than a minute to see the changes. In order to see all of the changes to a set of Samba permissions, we have to log off and then log on again./12/ So the easiest way is to restart the samba services.

4.4 Testing Web-Cam Server

The output of running the motion and the files being saved when motion is detected is shown below. The files are being saved on /tmp/motion/ directory.

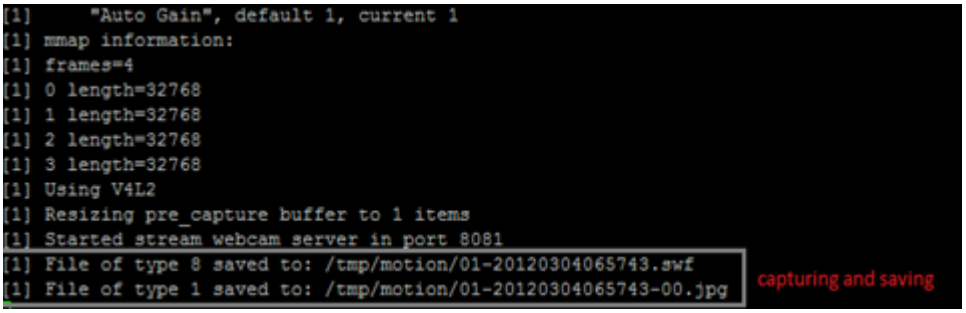


Figure 34. Capturing and saving by webcam-server

The videos are saved on .swf format and images in .jpg format and the stream is acquired from port 8081. The sample image saved is shown in Figure 35.

Figure 35. Sample image saved

4.5 Testing Asterisk Server

The Figure 36 shows the TCP packets passing through interface eth1. The packets shows calling state:

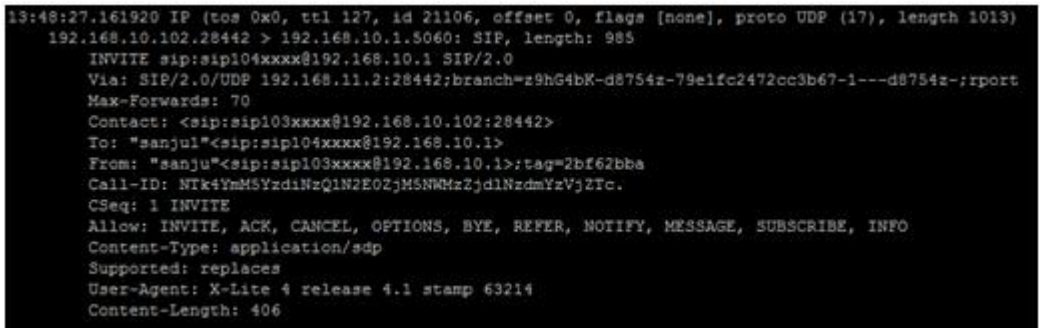


Figure 36. SIP session while calling

We can see the INVITE message being sent from SIP user sip104xxxx from IP 192.168.10.1 which is the server's IP to SIP user sip103xxxx with IP 192.168.102. The allow line shows the permission that the sip104xxxx have which are INVITE, ACK, CANCLE, BYE and so on. In the Figure 36 Cseq shows the message type sent i.e. INVITE.

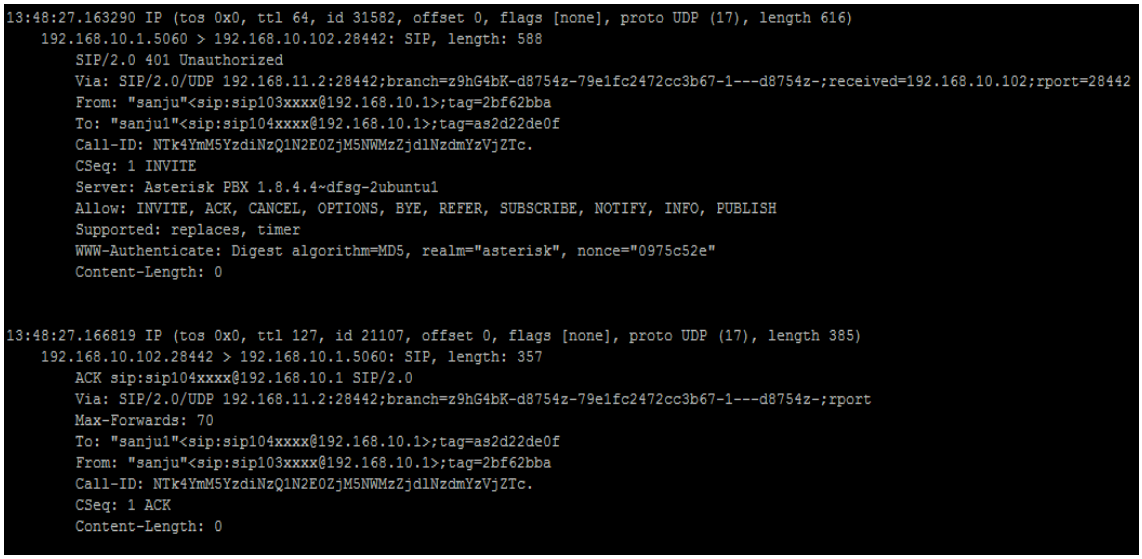


Figure 37. During call time

The Figure 37 was captured when the session was established. The audio voice are encoded and sent to the server and from server to the target host. Here we can see the acknowledgement message, the algorithm used to encrypt data"MD5" and relem"asterisk".

```
13:48:35.156596 IP (tos 0x0, ttl 64, id 31588, offset 0, flags [none], proto UDP (17), length 542)
    192.168.10.1.5060 > 192.168.10.102.56112: SIP, length: 514
        BYE sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97 SIP/2.0
        Via: SIP/2.0/UDP 192.168.10.1:5060;branch=z9hG4bK114a69b9
        Max-Forwards: 70
        From: "sanju" <sip:sip103xxxx@192.168.10.1>;tag=as380a054d
        To: <sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97>;tag=a4188d4f
        Call-ID: 01a583447175f6b674cbc6594c643347@192.168.10.1:5060
        CSeq: 103 BYE
        User-Agent: Asterisk PBX 1.8.4.4~dfsg-2ubuntu1
        X-Asterisk-HangupCause: Normal Clearing
        X-Asterisk-HangupCauseCode: 16
        Content-Length: 0


13:48:35.239183 IP (tos 0x0, ttl 127, id 23962, offset 0, flags [none], proto UDP (17), length 463)
    192.168.10.102.56112 > 192.168.10.1.5060: SIP, length: 435
        SIP/2.0 200 OK
        Via: SIP/2.0/UDP 192.168.10.1:5060;branch=z9hG4bK114a69b9
        Contact: <sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97>
        To: <sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97>;tag=a4188d4f
        From: "sanju"<sip:sip103xxxx@192.168.10.1>;tag=as380a054d
        Call-ID: 01a583447175f6b674cbc6594c643347@192.168.10.1:5060
        CSeq: 103 BYE
        User-Agent: X-Lite 4 release 4.1 stamp 63214
        Content-Length: 0
```

Figure 38. Call terminated

When a party or hosts terminate the session the BYE message is sent. In the Figure 38 the user sip103xxxx have terminated the call. The BYE message from user sip103xxxx is sent to user sip104xxxx through the Asterisk server and the user sip104xxxx sends the message 200 OK confirming the termination of the call. The wire shark capture shown below shows the detailed conversation way.

3	1.186912	192.168.10.1	192.168.11.5	SIP/SDP	975	Request: INVITE sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97, with session description
4	1.239068	192.168.11.5	192.168.10.1	SIP	485	Status: 180 Ringing
20	6.468544	192.168.11.5	192.168.10.1	SIP/SDP	992	Status: 200 OK, with session description
21	6.472232	192.168.10.1	192.168.11.5	SIP	528	Request: ACK sip:sip104xxxx@192.168.10.102:56112;rinstance=c40c9412a7c12e97
188	9.835435	192.168.11.5	192.168.10.1	SIP	558	Request: BYE sip:sip103xxxx@192.168.10.1:5060
189	9.905002	192.168.10.1	192.168.11.5	SIP	574	Status: 200 OK

Figure 39. Wireshark caputre of SIP protocol

The first packet shows the request INVITE message sent by user sip103xxxx to sip104xxxx with its IP and session description which includes the codecs used. The second packet is sent back to caller with the status "Ringing". When the user

sip104xxxx accepts the call the”200 OK” message with its session description which includes the codecs choose shown in the third packet above. The fourth packet shows the established sessions where the audio packets are transferred form and to. The fifth message shows the termination of call from user sip104xxxx and the requested message is BYE. The last packet with the message”200 OK” is call clearing. At this point it should be noted that the wire shark capture captured does not resemble the traffic captured in eth1 the previous Figures but was captured with entire different session where the caller is sip103xxx and is also the call terminator.

Figure 40. Calling

The Figure 40 shows SIP user sip102xxxx calling SIP user sip103xxxx. The Figures 41 and 42 shows the call established between two SIP users sip102xxxx and sip103xxxx. X-lite 4.0 was used to make calls.

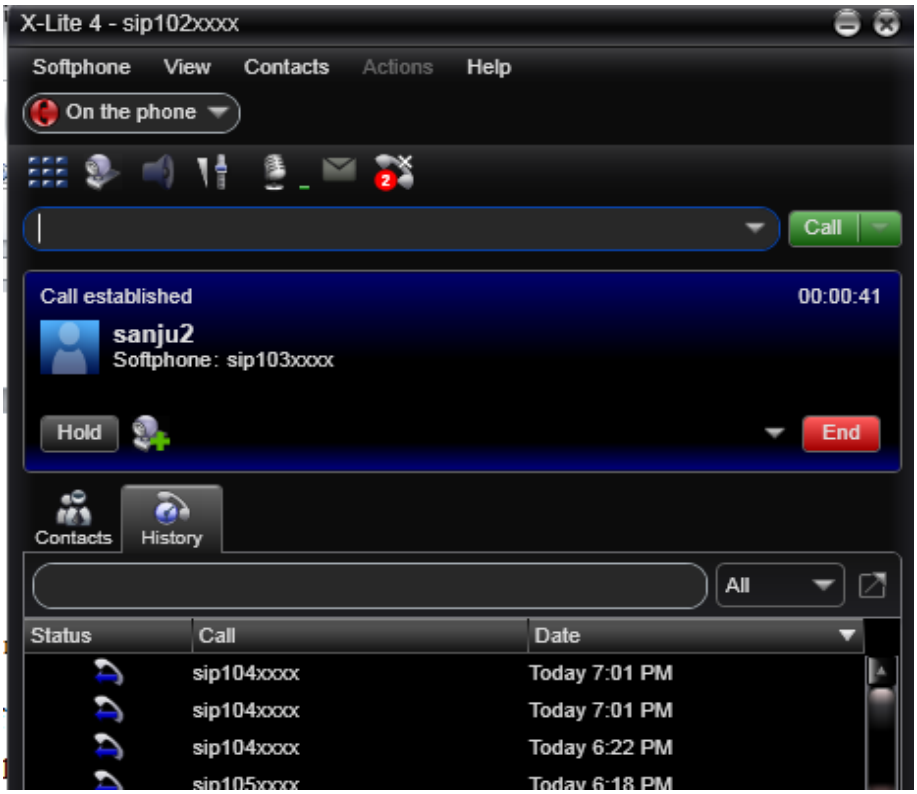


Figure 41. Call established between SIP users

Figure 42 . Call established between SIP users

4.6 Testing Firewall

We can use ICMP echo request-reply to verify whether our firewall rules are working or not. The Figure 43 shows the ICMP requests are being dropped by our firewall. Similar, other rules were tested.

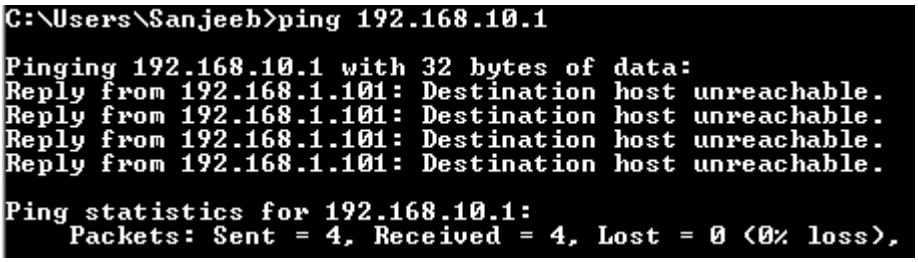


Figure 43. ICMP echo request

During this time we can also observer our /var/log/syslog file to see the rejected packets by the firewall as shown in Figure 44.

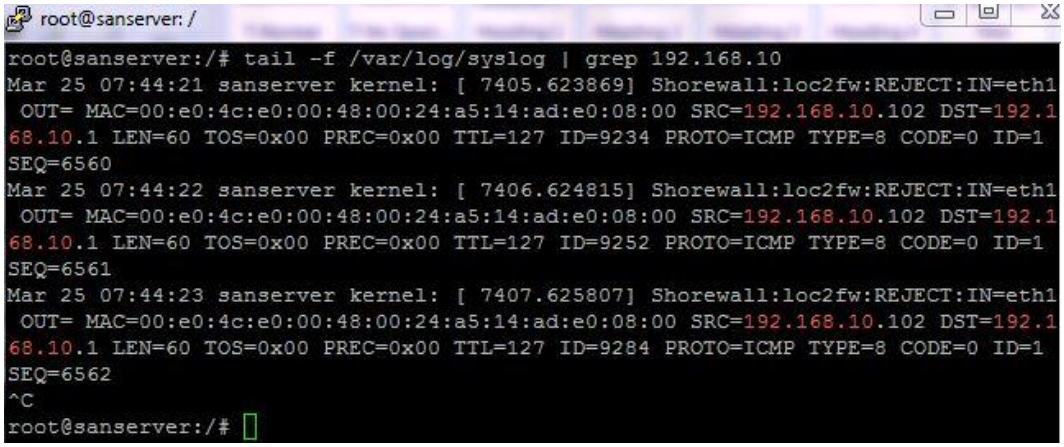


Figure 44. Checking syslog file

If the firewall is stopped or disabled then it operated according to the default policy i.e. allow all outgoing connections and block all incoming packets.

5. CONCLUSION

In this project, a Linux (Ubuntu) server has been configured. All the required objective of this project has been achieved within the time frame. In this project work installing and configuration of Samba file server, Asterisk server, DHCP server and security cam server have been done.

In these project, we have implemented theoretical knowledge of Linux operating system which we have gain during my studies. Before this project, we were familiar with the configuration of Windows server but installation and configuration of Linux based server was completely new task for us. During this project, we have configured Samba server as a domain controller, Samba server was also acting as a file server to manage network public and private shares, DHCP server configured to assign IP's for LAN, Asterisk server provides the VoIP calling facility over the LAN, Security cam server records the video for security purposes and firewall configured to secure over network from hackers and from unauthorized access. As a whole we can say that we have installed secure private home networking system, from which many people can get benefits. They are now able to access the fast internet services, share the files and documents, able to do VoIP call, have a look around the houses through web cam for security reason and running secure system.

We can conclude that, we have achieved the goals of the project which we were supposed to do. With our practical implementation of the project, provides an easiest and step by step guide to peoples who wants to configure a private home network system.

Linux is a free distribution so nobody has to worry about licensing. Linux users can get free software from many repositories such as OpenOffice Firefox, Linux users has reported uptime more than a year, Linux has less than hundred viruses while windows has more than hundred thousand known viruses and Linux online forums provides round the clock support.

REFERENCES

/1/ Server definition, Available in www-form:

<URL: http://www.linfo.org/server.html>

/2/ Prepraing to install, Available in www-form:

<URL:https://help.ubuntu.com/11.10/serverguide/preparing-to-install.html#idp1832816>

/3/ Innoband Technologies, Available in www-form:

<URL: http://innoband.com/technologies.php>

/4/ Vugt van S. ”Begining Ubuntu Server Administration” USA: Apress, 2008. (pg. 269)

/5/ SAMBA- opening windows to a wider world, Available in www-form:

<URL: http://www.samba.org/samba/docs/server_security.html>

/7/ Gomillion David and Dempster Barrie. ”Building Telephony Systems with Asterisk” 32 Lincoln Road: Packet Publishing, 2005. (pg. 11)

/8/ Open Source Platforms for interconnected virtual worlds, Available in www-form:

<URL: http://wiki.realxtend.org/index.php/Communications:_IM,_voice,_video>

/9/ Asterisk sip canreinvite, Available in www-form:

<URL: http://www.voip-info.org/wiki/view/Asterisk+sip+canreinvite>

/10/ Get Ubuntu now, Available in www-form:

<URL: http://www.ubuntu.com/>

/11/ Webcam-server, Available in www-form:

<URL: http://www.hacktivision.com/files/webcam-server>

/12/ Minasi Mark and York Dan. ”Linux for Windows Administrators” USA: SYBEX Inc., 2003. (pg. 439)

APPENDIX 1

Samba Configurations:

#============= Global Settings ======================

[global]

Browsing/Identification

 # Change this to the workgroup/NT-domain name your Samba server will part of

workgroup = Sanjeeb.com

 # server string is the equivalent of the NT Description field

server string = Sanjeeb File server

 # WINS Support - Tells the NMBD component of Samba to enable its WINS Server

wins support = yes

############## Networking ##############

 # interface names are normally preferred

interfaces = 127.0.0.0/8 eth1

bind interfaces only = yes

Debugging####

 # This tells Samba to use a separate log file for each machine that connects

log file = /var/log/samba/log.%m

 # Cap the size of the individual log files (in KiB).

max log size = 1000

####### Authentication #######

security = user

passdb backend = tdbsam

obey pam restrictions = yes

pam password change = yes

########## Domains ###########

domain logons = yes

logon path = \\%N\%U\profile

logon drive = Z:

logon home = \\%N\%U

logon script = logon.cmd

add machine script = /usr/sbin/useradd -g machines -c "%u machine account" -d /var/lib/samba -s /bin/false %u

########## Printing ##########

load printers = yes

printing = bsd

printcap name = /etc/printcap

#========== Share Definitions ========================

[homes]

comment = Home Directories

browseable = no

read only = no

create mask = 0700

directory mask = 0700

valid users = %S

[netlogon]

comment = Network Logon Service

path = /srv/samba/netlogon

guest ok = yes

read only = yes

[public]

comment = Public Share

path = /etc/public

read only = no

guest ok = yes