UNIVERSITY OF ALBERTA

SEAMLESS SIP MULTIMEDIA SESSION TRANSFER ON IPV6 NETWORK VIA DEVICE SWITCHING

MINT 709-INTERNET PROJECT

RAMYA ANTONY CRUZ

8/19/2013

This project focuses on the performance of location tracking system and also describes the performance of SIP during multimedia session transfer between mobile clients in an IPv6 network.

ACKNOWLEDGEMENT

I thank, Lord Almighty for showering his blessings upon me and making this project a success.

I take this opportunity to express my profound gratitude and deep regards to Dr. Mike MacGregor, Professor & Chairman, Computing Science, University of Alberta for his constant encouragement and guidance throughout the MINT Program. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Mr. Dennis Egyedy, CAO, Municipal District of Opportunity 17 for his acceptance of my Internship. I also would like to extend my sincere thanks to Mr. Jagdeep Sandhu, System Administrator, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am obliged to the Professors of MINT program, for the valuable information provided by them in their respective fields. I am always grateful to them for giving me a strong base in the field of networking.

Lastly, I thank, my parents, brother and friends for their constant encouragement without which this project would not have been possible.

TABLE OF CONTENTS

CHAPTER 4

4. INTERNET PROTOCOL VERSION 6

CHAPTER 5

5. SIP ON IPV6

CHAPTER 6

6. ASTERISKS

CHAPTER 7

7. CODECS

CHAPTER 8

8. METHODS & TECHNIQUES

CHAPTER 9

9. WORKING MODEL

CHAPTER 10

10. CONCLUSIONS & FUTURE WORK

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 OVERVIEW

Wireless networks have become very important for users that have to be on the move most of the time. Mobility provides user freedom to be connected anywhere and everywhere. When wireless connections have been established, there would always be times when user would desire to switch the session from the current device to another without having to terminate the session. This is called device switching.

Location based services and multimedia communication over mobile devices have become more popular. Modern researches on location tracking are not only focused on calculating the distance but also on developing communications between the moving nodes. It is always important for the clients to have seamless multimedia communication with automatic transfer of the session to other clients. This is taken care of by SIP.

The SIP protocol is a transport layer signaling protocol used for carrying voice over internet. SIP is used to create, maintain and terminate a session. SIP has been successfully implemented in IPv4 networks; however its implementation in IPv6 environment will have more benefits over IPv4 (Discussed in Chapter 5).

This project focuses on multimedia communication and seamless transfer of a session using device switching.

1.2 PROJECT DESCRIPTION

In this project, Raspberry Pi has been used as the server. It performs three major roles of being the location server, SIP server and web server. Location tracking system not only focuses on calculating distances but also improves communication between the mobile clients. The distance between mobile clients can possibly be calculated by the location server using wireless connections. Embedded SQLite database has been used in the location server to store the calculated distances of the mobile clients.

SIP server takes care of each session and it is the backbone behind each session. Asterisk has been installed on the Rpi to support multimedia communication and it performs the job of a communication server.

In this project, the mobile clients communicate through the Web Browsers (Google Chrome). An improvisation (rather than using softphones) has been made by configuring Web RTC on the browsers to make the clients communicate with each other and with the server. Web RTC uses HTML5 to provide Peer-to-Peer connectivity between browsers. Web RTC allows interactive real-time multimedia communication (Voice, Video). It also supports video conferencing. Web RTC allows the browsers of a network to make voice calls, video calls, voice and video calls or instant messaging on the same session.

Web RTC has been enabled on the RPi by installing various packages like WebRTC, Webrtc2sip, and SIPML5, Doubango, FFMPEG (Discussed in Chapter 8). An effort has been made to demonstrate the working of Web RTC in an IPv6 environment using dual-stack approach.

An IPv6 cloud has been formed with the use of an IPv6 capable Dlink router (Dlink 815). The router has been configured with both IPv4 and IPv6 addresses. It has been set up with a scope 2000:c0a8:173::1/64 and any device that is connected to the router obtains dynamic address from this scope. Wi-Fi has been enabled to establish wireless connectivity with the clients. The server and the clients are configured with static IPv6 addresses and are connected to the router wirelessly.

Multimedia communication and the session control have been demonstrated using Asterisk and SIP. Device switching is controlled by the location server that is programmed on the Rpi based on the randomly generated signal strengths for different mobile nodes.

1.3 OBJECTIVES

- To design a multimedia system using asterisk.
- To design a SIP server.
- To design a location server.
- To design a SIP web client.
- To design a HTML5 web client for a multimedia system using Web RTC without any plugins, add-ons or installations.
- To form an IPv6 network.

1.4 DOCUMENT ORGANIZATION

- Chapter 2 outlines the hardware description of Rpi, uses and its capabilities.
- Chapter 3 outlines the working of the new technology, Web RTC with HTML5.
- Chapter 4 gives a clear understanding of IPv6 addresses, features and advantages over IPv4. It also talks about different ways of making IPv6 compatible with IPv4.
- Chapter 5 outlines how SIP works on IPv6 and its advantages.
- Chapter 6 outlines the working of Asterisk.
- Chapter 7 outlines the various codecs used in Web RTC.
- Chapter 8 outlines the methods and techniques used in accomplishing the objectives of this project.
- Chapter 9 gives an idea of the working model.
- Chapter 10 outlines the conclusions and future work.

CHAPTER 2

2. SYSTEM DESIGN- RASPBERRY PI (RPI)

2.1 INTRODUCTION [1]

Raspberry Pi was developed by Raspberry Pi Foundation, UK. It is a very compact, little computer that can perform many of the tasks that a regular computer can do. It supports high definition video, audio. It makes use of an ARM processor (ARM1176JZF-S, 700MHz) and its design is based on the Broadcom BCM2835 SoC. It neither has inbuilt hard drive nor solid state hard drive, instead it boots from a SD card where the operating system is stored. This board runs Linux kernal based operating system.

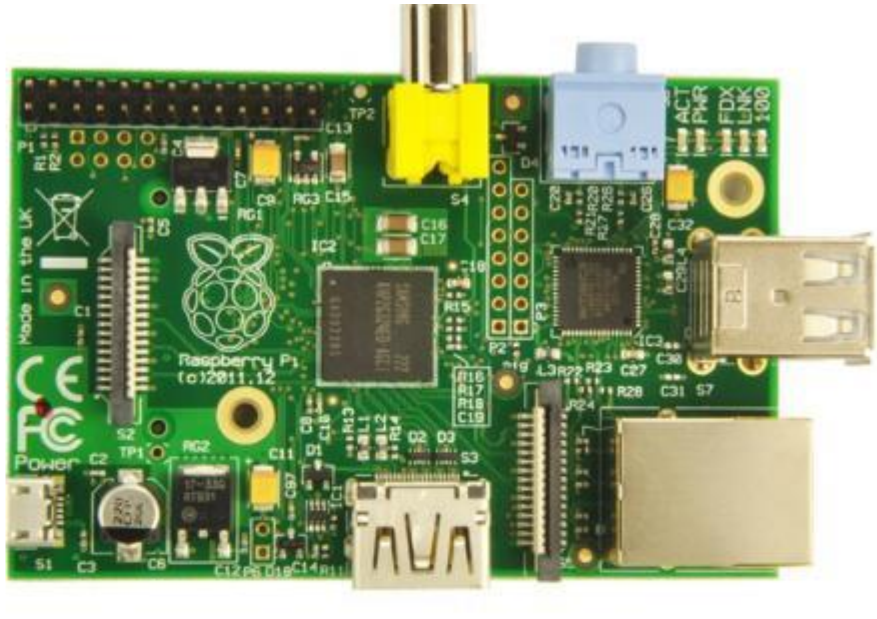


Figure 2.1 Raspberry Pi Board

The Raspberry Pi foundation developed two versions of the board called Model A & Model B.

MODEL A: The Raspberry Pi Model A uses Broadcom BCM2835 as the System-on-a-chip and it has a memory of 256MB (SDRAM) and provides one USB port and there is no Ethernet port.

MODEL B: Changes have been made on Model A like addition of an Ethernet port using LAN9152 and forming an USB hub using two LAN9152 or four LAN9154. The memory has been increased to 512MB (SDRAM). [1]

In Model A there is no provision for Ethernet connectivity but still it can be connected to the Internet via Wi-Fi adapter. So, the functionality of both the models is more or less similar.

2.2 SPECIFICATIONS [2]

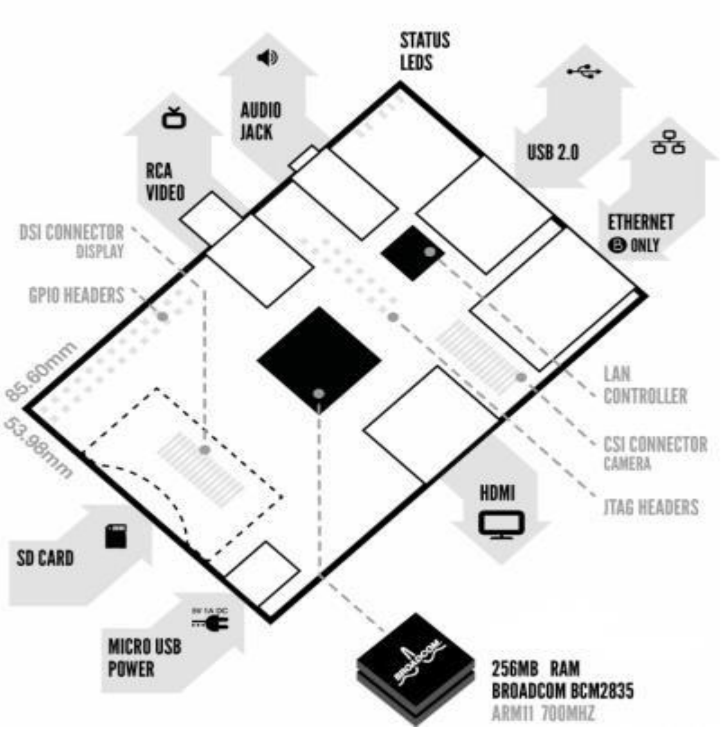


Figure 2.6 On-Board Specifications

- **SoC:** Broadcom BCM2835.
    - ARM11 700MHz processor (CPU).
    - Broadcom VideoCore IV, OpenGL ES 2.0 (supports graphics), 1080p30 h.264/MPEG-4 AVC high-profile decoder (GPU). GPU is capable of BluRay quality playback using H.264 at 40Mbps and also supports 3D core.

    - 256/512MB memory (SDRAM).
    - Digital signal processor to process the audio and video signals, but not a public API (DSP).

- **NETWORK:** 10/100 Ethernet support RJ45 using LAN9152 (LAN Controller).
- **LAN9152:** Provides a fully integrated, high speed USB 2.0 hub with two LAN9152, high performance 10/100 Ethernet and EEPROM controller. It provides USB to Ethernet and multiport USB connectivity. It has a separate power supply unit (PSU). There are only two USB ports on the Rpi and the BCM2835 USB is connected to LAN9152, so there is no separate USB port provided by BCM2835 on the board. [4]
- **USB:** Two USB 2.0 ports in Model B and one USB 2.0 port in Model A.
- **WI-FI:** Using Wi-Fi dongle.
- **VIDEO OUTPUT:** Composite RCA, HDMI (type A HDMI 1.3out).
- **HDMI:** High definition multimedia interface is used to transfer data (uncompressed video and compressed/uncompressed digital audio) from a HDMI compatible device to a compatible screen like monitor, projector, TV.
- **AUDIO OUTPUT:** 3.5mm jack, HDMI
- **COMPOSITE VIDEO & AUDIO:** RCA connector is an electrical connector that is used to carry composite video and stereo audio signals. Yellow cable is used for video, red and white/black are used for audio.
- **STORAGE:** SD, MMC, SDIO cards.
- **SD CARD:** Secure Digital card is a non-volatile memory storage device where the operating system of the Rpi is stored.
- **GENERAL PURPOSE INPUT/OUTPUT:** GPIO 17 pins do not have any special usage and they can be programmed by the user at any time. They do not have any usage by default. Some pins can be configured to work on SPI mode, I2C bus mode or Serial I/O (Tx/Rx) communication.
- **MIPI CSI-2:** Camera serial interface is a 15-way flat flex connector used to connect to a camera.
- **DSI:** Display serial interface is a 15-way flat flex connector used to connect to a Display.

- **JTAG (Joint Test Action Group):** Generally used for testing the printed circuit boards (PCB) and also for debugging purposes.
- **POWER REQUIREMENTS:** 1mA at 5V micro USB power. A capacitor is used behind the power supply to stabilize the DC power on the board.
- **STATUS LED:** Five status LEDs. One each for power and SD card, three for LAN.
- **REAL TIME CLOCK:** No real time clock, time has to be setup during boot.
- **DIMENSION:** 85.60mm x 53.98mm.
- **BOOT:** Boots from SD card.

2.2.1 MODEL B ENHANCEMENTS [1]

- ARM JTAG support included by making modifications to GPIO pins.
- JTAG debug support has been added.
- RESET A reset circuit has been included.
- The resettable fuses that were used to protect the USB output have been removed so as to provide power to RPI from USB hub.
- Version identification feature has been removed (4 GIPO signals) as it was never read by the software and the removed GIPO signals are used in additional I/O Expansion.
- A fix has been implemented for +5V leakage from HDMI because when connected to the TV this was interfering in the Customer Electronic Control (CEC) operation of the connected devices.
- SMSC +1V8 power has been removed.
- Mounting holes have been provided for automatic test equipment (ATE) testing.
- Minor LED changes have been implemented.
- Memory has been increased.

2.3 OPERATING SYSTEM [3]

An Operating System is a set of basic programs and utilities that makes the Rpi work. Raspberry Pi runs Linux kernel-based operating system.

Raspberry + Debian = Raspbian. Raspbian is a Debian-based free software (OS) based on ARM hard-float-Debian7 Wheezy architecture port with LXDE GUI desktop support, specially developed for Raspberry Pi. It is not just a simple OS; it comes with over 35,000 packages of pre-compiled software put together in a user friendly format. Raspbian is optimized for ARMv6 instruction set of Pi. The OS is stored on a SD card and it gives a text based menu on boot. It can be configured as per user's requirements. It gives options to enable SSH.

Raspberry Pi also supports other OS like RISC OS, Arch LINUX, Android and Plan9. RISC OS and Plan9 boot into a user friendly GUI, Arch LINUX into a terminal environment and Android into its own official android unlock screen.

Previous experience is required to work on RISC OS and additional cost is needed to make it fully functional with Rpi. To work on Arch LINUX, knowledge about terminal commands is required and knowledge in depth would make it one of the best OS. Android supports media applications but not office applications. Space requirements are more; it will not work with 256MB RAM but a little better with 512MB RAM which make the system slow. Plan9 boots into a nice desktop but learning process is tedious.

Raspbian tops all other OS because it is user friendly, has the best look and has loads of default software that will allow even a beginner to work easily on this box.

2.4 RASPBERRY PI INITIAL SETUP

2.4.1 REQUIREMENTS

- SD card 4GB/8GB with OS stored on it.
- HDMI to HDMI/HDMI to DVI cable.
- Keyboard and Mouse.

2.4.2 SD CARD SETUP

- Insert an 8GB card into the Laptop.
- Format the SD card using SD Association's Formatting tool.
- Download the OS (.img file) from the raspberry website (http://www.raspberrypi.org/downloads) and save it on the laptop and flash the card using wins32diskimager.
- Now the SD card is ready for use.

2.4.3 POWERING ON THE BOARD

- Insert the SD card in the SD card slot provided on the board.
- Connect the Pi to the display and connect the keyboard and mouse
- Power up the board using +5V 700mA micro USB power supply.
- Set the time zone and set up the initial configurations.
- Set the root password using ***sudo root password*** command. Now the board is ready to be configured.

CHAPTER 3

3. VOICE, VIDEO & DATA

3.1 INTRODUCTION [1]

3.1.1 WEB RTC

Web RTC is a web based real time communication using the web browser. The communication is directly between the browsers (Peer-to-Peer) and it is called real time because of its interactive nature. This is mainly developed for interactive voice, video and data communication. Unlike, Youtube which allows us to watch video, Web RTC allows us to communicate video (video chat) with no latency. It is interoperable with existing voice and video systems.

3.1.2 HTML5

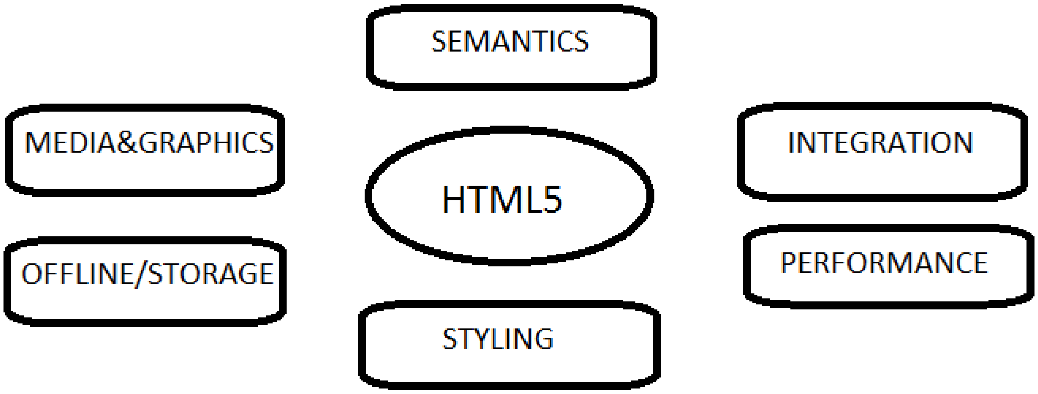


Figure 3.1 HTML5

Web RTC is a part of HTML framework. HTML5 almost looks like HTML4 (It describes the layout of a document or used for rendering a web page) and its backwards compatible. HTML5 HTML5 supports desktop applications as well as web applications and also adds databases, 3D graphics and real time collaboration communication. It’s not just used to display a web page but it’s used for interactive real time communication.

3.2 APPLICATIONS [4]

Consumer based applications include social networking, interactive gaming, personal email, social entertainment etc. Consumer's queries can be handled by an expert using voice and video calls. Meeting can be organized using standards HTML5 and H.264 in web RTC.

Business based applications include virtual meetings, employee to employee calls, public interaction, and social collaboration. For example, if an employee wants to call someone in middle of a presentation, he can use this application rather than using a phone.

It is the world's first HTML5 SIP Client.

3.3 ARCHITECTURE [4][3]

3.3.1 BROWSER TO NON-BROWSER ENDPOINT

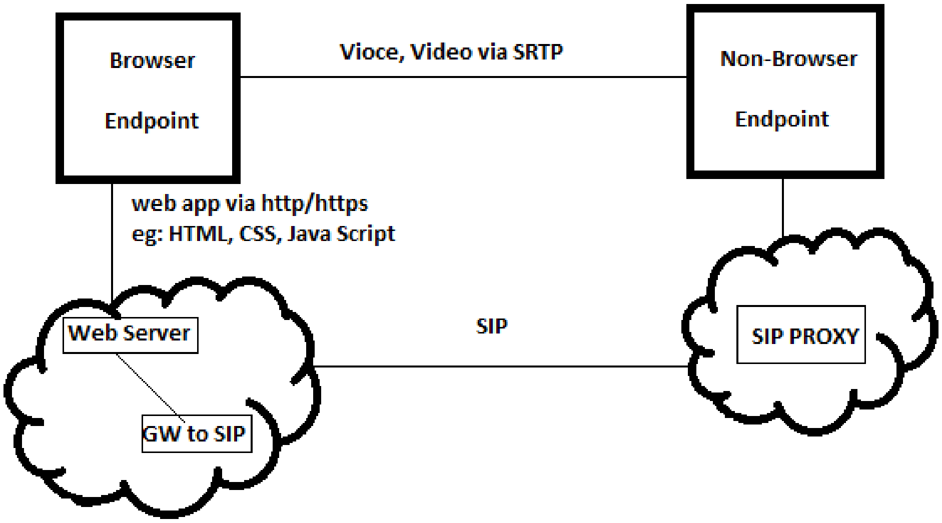


Figure 3.2 Browser to Non-Browser Architecture

In any communication system where lots of people are involved, the value of the new system depends on how many people can be reached and its interoperability with the existing system. Web RTC is designed in such a way that it is interoperable with the existing VOIP, predominantly SIP based system, if not it would be much harder and slower to deploy.

The basic idea behind web RTC is that it uses the same media (voice & video) as used in many of the existing SIP devices but the signaling is different (Gateway). The signal from the browser is passed to the web server and across to SIP proxy and over to the non-browser endpoint but the media is transferred directly between the endpoints. This ensures reduced latency and real time experience. Media transfer uses RTP.

3.3.2 BROWSER TO BROWSER ARCHITECTURE

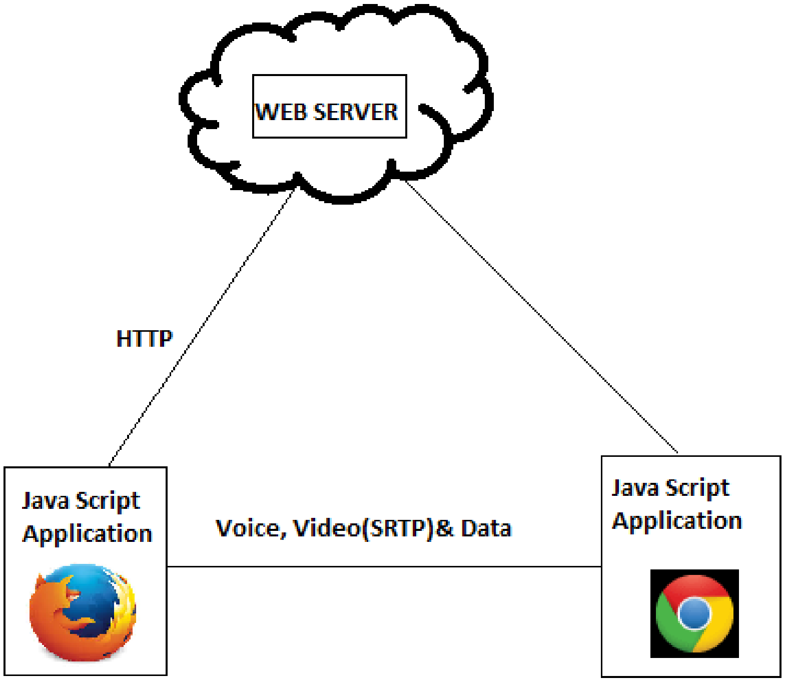


Figure 3.3 Browser to Browser Architecture

Browser actually sends and receives data from the web server. This is based on the Client-Server architecture where browser is the client and web server is the server.

For example, if a message is sent from a browser (Facebook) on one side to another browser (Facebook) on the other side, the message is not directly sent between the browsers. The web server gets the data from one browser and passes it on to the other browser. So, there is no direct communication between the browsers and the latency is more which is tolerable in case of Facebook , whereas could not be tolerated in real in time communication like voice and video conversations.

Therefore Web RTC has made a big change of introducing communication directly between the browsers. Browser sends information directly to another browser. This can be voice, video or even data.

3.4 NETWORK PROTOCOLS [2]

3.4.1 MEDIA LAYER CODECS

- Narrowband Audio: G.711, iLBC
- Wideband Audio: Opus, iSAC
- Video: VP8/ H.264 AVC

G.711 and Opus are mandatory codecs for audio. Other codecs can also be used by the browser but these are basic and work good with the browser application.

There is a debate going on between VP8 and H.264. Mozilla and Chrome prefer VP8, whereas Microsoft and Apple prefer H.264.

3.4.2 MEDIA TRANSPORT

The protocol used at the transport level is RTP. RTP is primarily used in SIP devices and H.323. The secure version of RTP is called SRTP, both encrypted and non-encrypted version can be used and it lets the user choose.

The encryption is done by keying and by setting up keying materials. The protocol used for this purpose is called as DTLS, TLS stands for transport layer security. It secures all https connections. Version that runs over UDP is called Datagram TLS and it’s used here.

This uses public keying approach to set up secure channel and keying materials to encrypt SRTP.

3.5 ISSUES [4]

3.5.1 FIREWALLS & NATS

ICE protocol is the protocol used by Web RTC that allows two different browsers behind various NATS to communicate directly with each other. ICE protocol is a combination of sub-protocols STUN and TURN. It’s also capable of detecting changes in the network and it can switch an interface from Wi-Fi to LTE. On media side, it makes sure that no unwanted traffic is sent to the devices.

TURN: This is basically a tunneling protocol that is used to tunnel data to and from a public relay server.

STUN: This protocol is used to get client’s IP address from the public server.

3.5.1.1 INTERACTIVE CONNECTIVITY ESTABLISHMENT (ICE)

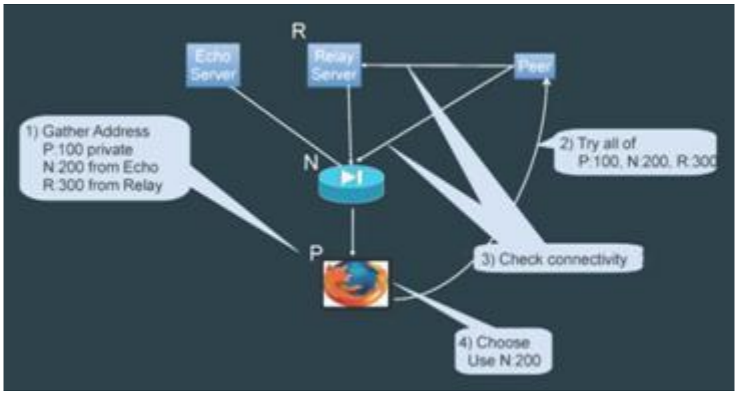


Figure 3.4 ICE Protocol

ICE gathers the address from one side and passes it on to the other side to establish a connection. It uses the local address of the machine (IPv4/IPv6), Global Public address, or the address behind the same NAT. Sometimes, NATs cannot communicate with each other even if they know their public addresses. For this type of environment we make use of a relay server, where traffic is sent from one side up to the rely server and relay server passes it on to the other side. This is done using TURN protocol and a device gets its public address from the relay server, where the other devices can reach at.

STUN protocol is used to discover the outside address of the NAT. The way STUN works is that it sends a packet to the public STUN server and the server will give the address where the packet appeared to have come from. It will give the outside address and port number of the NAT.

Browser, after gathering the private address, NAT address, relay server address, it exchanges the addresses to the other browser through the web server. ICE does a connectivity check. Connectivity check is done by ping (not ICMP) using STUN protocol which sends request and gets acknowledgement. It also uses STUN ID and password to do the check. The connectivity check is done by both the sides and connection is established based on one ground.

3.5.2 ARCHITECTURE MODIFICATION

Browser architecture needs to be modified to allow the browser to talk to things other than the web server, because traditionally the browsers were designed in a way that they can communicate only with the web server and not with the browsers.

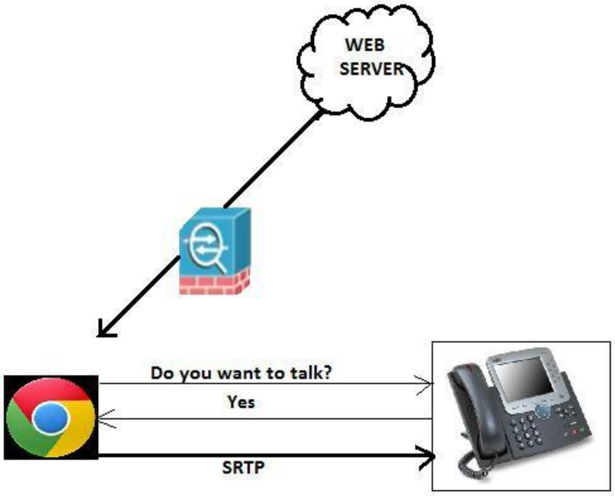


Figure 3.5 Modification of Browser Architecture

The browser sends request to the device (IP phone) asking if it wants to talk to the browser. If it gets a reply then it will know that it is the actual device and not a database server. SRTP and dataa communication will be setup. This request and reply mechanism is supported by STUN protocol in Web RTC. Any device that supports STUN and ICE can talk to the browser. This authentication is required to avoid malicious access.

3.5.3 IDENTITY

Identity providers are very few. Every single application on the internet depends on the identity providers to get authenticated. In next few years this would be a problem. Web RTC recognized that the identity on the internet is changing rapidly and there are various ways for getting authenticated, so the most important thing to be considered is that the protocol used for this purpose must be foreseeing the future.

OAuth or OpenID Connect is the authentication protocol used in Web RTC and this would be very successful in the next five years.

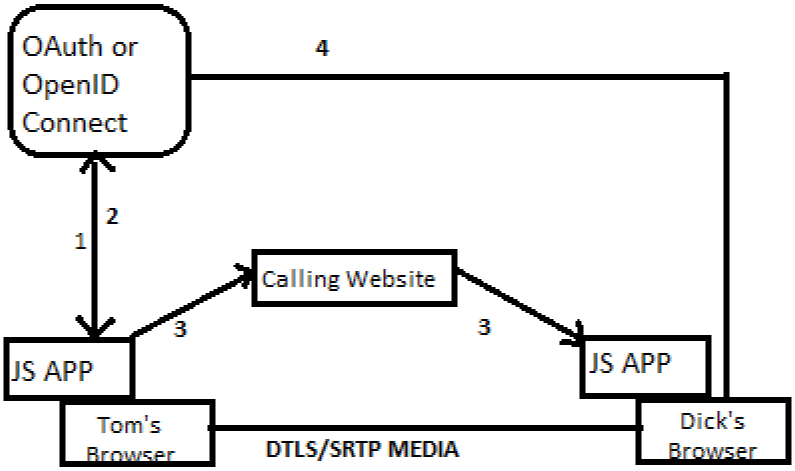


Figure 3.6 Browser Authentication

The basic concept here is it uses an encrypted connection. An encrypted connection has finger prints associated with it. The browser on one side must be able to communicate with the other browser that here is the finger print and you can verify the encrypted media communication with this finger print and start the session.

For example,

In the above example, Tom`s browser logs in to the OpenID connect and gets authenticated. The Identity provider signs an assertion saying that Tom wants to talk to Dick and sends the assertion with the finger print and a URL at which the Identity provider can be reached at to Tom. Now Tom sends this to Dick. Now Dick`s browser will contact its Identity provider to authenticate the connection and finally checks if the finger print of the DTLS/SRTP media connection and the finger print provided by Tom`s browser matches.

3.6 FUTURE

Web RTC is an advanced technology evolving at a faster rate. It has got the potential to do great thing in the near future. Due to various issues like QoS, Congestion control, Codec, etc., it is not completely ready to hit the market. The development team is working to fix all the issues and will be given out to people as one of the most exciting technologies in the near future. It will also be made an in-built technology in the web browser which will entice many users. It works based on Java Script. So it absolutely needs nothing (No installation, account setup, add-ons etc.,). Once its drawbacks are overcome, it will be one of the amazing, user-friendly technologies in real time communication.

CHAPTER 4

4. INTERNET PROTOCOL VERSION 6

4.1 INTRODUCTION

The unique number identifiers assigned to every device or computer that is connected to the internet are called IP addresses. Two versions of IP addresses prevail and they are IPv4 and IPv6.

Internet protocol version 4 was developed in 1980s and it has been serving the internet community globally for more than thirty years. IPv4 are 32-bit addresses and they have a capacity of about four billion IP addresses. With emerging new networks across the world, these addresses are being used up at an overwhelming rate. Of the four billion IP addresses only 3.7 billion are used by devices for internet access, whereas the remaining are being used for special purposes like IP multicasting. The interesting news here is all of the 3.7 billion IPv4 addresses are being used up.

An effort was made by Internet Engineering Task Force in 1990s to develop a successor to the IPv4 addresses. Therefore, Internet Protocol Version 6 was developed (IPv6). Compared to IPv4 addresses, IPv6 are 128-bit addresses with a capacity of 340 undecillion addresses.

4.2 REPRESENTATION

The IPv4 addresses are 32-bit and look like 192.168.0.1, whereas IPv6 are 128 bit addresses written in hexadecimal with each segment being separated by colon instead of periods as in IPv4. It can accommodate more information into fewer digits compared to Ipv4.

- 16 digit hexadecimal numbers.
- They are not case sensitive.
- Segments separated by colon (:).
- Double colon appears only once.
- Abbreviations are possible. Leading zeros in the adjoining block could be represented by (::).

For example, 2001:0df8::53 can be expanded as 2001:0df8:0000:0000:0000:0000:0000:0053.

4.3 IPv6 OVERVIEW

IPv6 has been designed to take a leap over IPv4. It is not designed in a way that it is totally different from IPv4 but it has got additions and deletions done on it. IPv6 retains the features that worked perfectly in IPv4 but the features that didn't work properly had been removed.

Changes implemented;

- More address spaces available.
- Simplification of header format.
- Improved support for options.
- Routing capabilities.
- QoS
- Authentication & Privacy.

4.3.1 HEADER FORMAT

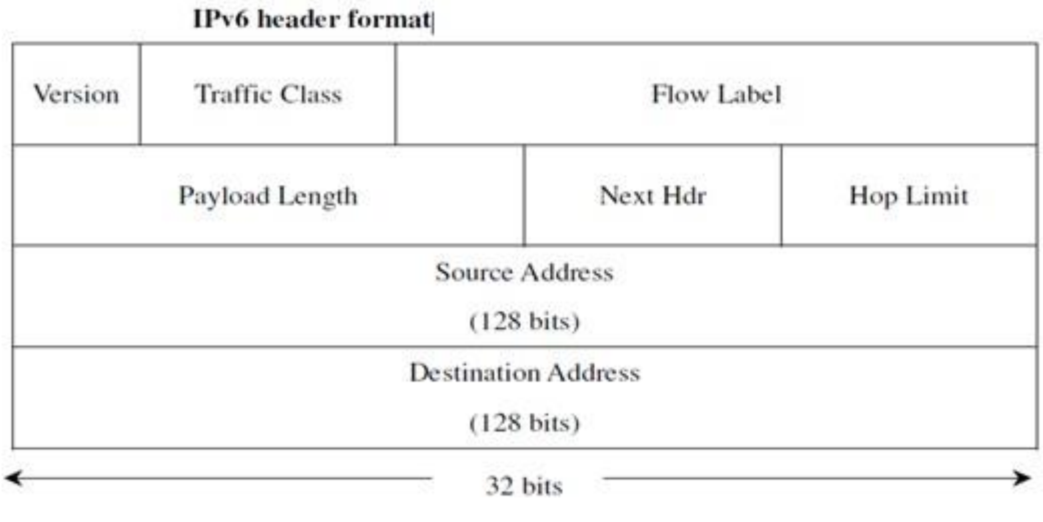


Figure 4.1 IPv6 Header

- **Version:** 4-bit field used to identify the version. It’s 6 for IPv6.
- **Traffic Class:** 8-bit field similar to ToS field in IPv4, used to assign priority to packets in a flow.
- **Flow Label:** 20-bit field used to identify a flow of datagram.
- **Payload Length:** 16-bit field (unsigned integer) gives the number of bytes in an IPv6 datagram.
- **Next Header:** 8-bit selector used to identify the type of the immediate header that follows the IPv6 header.
- **Hop Limit:** 8-bit unsigned integer. This field is decremented by 1 each time a node sends out a packet and the packet is discarded when the field reaches 0.
- **Source Address:** 128-bit address that defines the source.
- **Destination Address:** 128-bit address that defines the destination.

Increasing the size from 32-bit to 128-bit ensures more address spaces for the future. Unused or less used fields from IPv4 are either removed or kept optional. This helps in faster processing of datagram.

4.3.1.1 FIELDS REMOVED

- **Fragmentation & Reassembly:** Unlike at routers in IPv4, this is done at the source and destination. This increases the speed at which IP forwarding takes place within a network and it reduces the time consumed.
- **Checksum:** This is taken care of by layer 2 and even if the router forwards a packet to an unintended recipient, the host at the destination will disregard it.
- **Options:** Options field has become a part of the Next Header and it no longer a part of standard IP header.

4.3.2 ENHANCEMENTS

4.3.2.1 IPv6 EXTENSIONS: Unlike IPv4, options are placed in separate extension header between IPv6 header and transport layer. The options are improvised over IPv4 and are processed at the final destination rather than at the routers which helps in increasing the router performance. Another improvement is that the options carried in a packet are not limited to 40 bytes.

4.3.2.2 IPv6 EXTENSION HEADERS

- **Hop-by-Hop Options Header:** This is mandatory header and it is processed at the intermediate router.
- **Destination Options Header:** This could be for the intermediate destination or the final destination. If routing header is present it specifies packet delivery parameters for intermediate destination, if not it specifies the parameters for final destination.
- **Routing Header:** Specifies the list of intermediate destinations (loose source route) while a packet is on its way to the final destination.
- **Fragment Header:** Used only at the source.
- **Authentication Header:** Authentication and Integrity of data is provided by this header and also provides anti-replay protection.
- **Encapsulating Security Payload Header:** Confidentiality, Authentication and Integrity to the encapsulated payload.

4.3.3 IPv6 ADDRESSING[1]

- **Unicast:** An identifier defined for a single interface. A packet addressed to a unicast address will reach the interface identified by that particular address.
- **Anycast:** An identifier defined for a group of interfaces. A packet addressed to an anycast address will reach any of the interface (could be nearest or defined by any routing protocols) identified by that address.
- **Multicast:** An identifier defined for a group of interfaces belonging to different nodes. A packet addressed to a multicast address will reach all of the interfaces identified by that address.

- **Broadcast:** Not available in IPv6.

4.3.4 IPv6 ROUTING

Routing is done in the same way as it is done in IPv4 environment under CIDR. The only difference is the size of the addresses which is 32-bit in IPv4 and 128-bit in IPv6. The routing algorithms (OSPF, RIP, IDRP, ISIS, EIRP, BGP, etc.,) that are used in IPv4 network can be used in IPv6 network too. IPv6 network have additional routing capabilities such as,

- Provider Selection based on various parameters like policy, performance, cost, etc.
- Host Mobility; can be routed to current location.
- Auto-Readdressing; enables routing to new address.

These routing functions can be enabled using IPv6 routing option that is present in the IPv6 extension headers.

4.3.5 QoS CAPABILITIES

IPv6 is capable of handling real time services such as multi-media applications (voice and video communication). A host can identify the kind of packets that need special attention at the router side by using Flow Label and Traffic Class Fields. This capability of IPv6 enables it to support multi-media applications that require some degree of consistent delay, jitter and throughput and also to provide Quality of Service.

4.3.6 IPv6 SECURITY

A number of security problems prevail in the internet. There is no privacy or authentication below the application layer. These problems are being addressed by IPv6 environment. They make use of the Authentication Header and the Encapsulating Security Header to resolve this problem.

Authentication Header provides data authentication (verifies the sender node), data integrity (verifies if the data is not modified during transit) and anti-replay protection (do not permit re-transmission). It supports different types of authentication mechanism but the use of keyed MD5 is suggested just to ensure interoperability with the global internet.

Encapsulating Security Header provides data confidentiality, data authentication and data integrity to the datagram. Again this supports various security protocols but the most preferred is DES CBC for interoperability reasons.

4.4 TRANSITION FROM IPv4 TO IPv6

The public internet is completely based on IPv4 and almost all the existing networks are based on IPv4 and moreover they are not compatible with IPv6.

IPv6 is backwards compatible but the existing systems are not compatible with IPv6. For this reason we have two different approaches to make different networks compatible. They are,

- Dual-Stack approach
- Tunneling

4.4.1 DUAL-STACK APPROACH

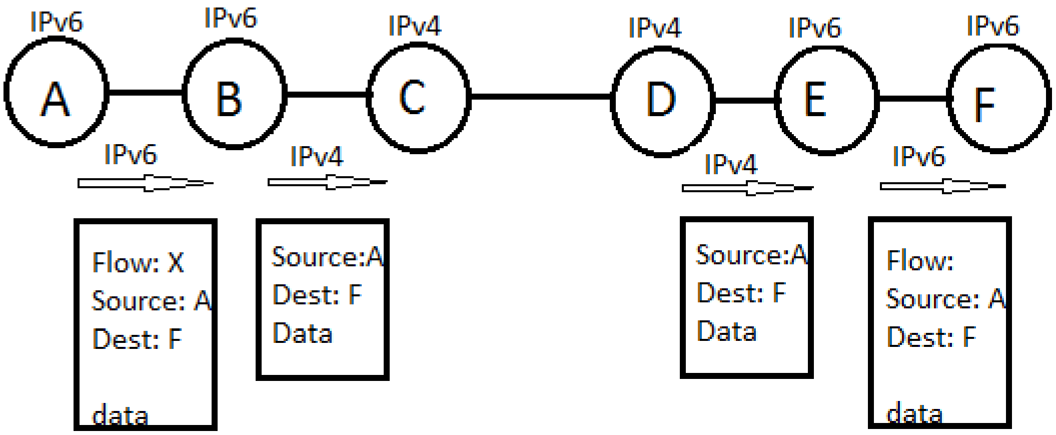


Figure 4.2 Dual-Stack Approach

IPv4 is completely implemented on an IPv6 node and this type of a node is called IPv4/IPv6 node and it will have both IPv4 and IPv6 addresses assigned to it and they are also capable of handling both types of datagram. This node when working with IPv4 node will use IPv4 datagram and when working with IPv6 node will use IPv6 datagram. It must also be capable of determining if a particular node supports IPv6 or not. DNS helps in resolving this problem by returning an IPv6 address if the node supports IPv6 or returns an IPv4 address if the node doesn't support IPv6.

If any one of the end is capable of using only IPv4 then the system must only use IPv4 datagram, which might result in an IPv6 capable node sending IPv4 datagram.

In the above diagram, node A sends IPv6 datagram and node B receives IPv6 datagram, whereas node C is not capable of receiving IPv6 datagram therefore B converts it into IPv4 datagram and sends it to node C. During this conversion some of the fields in IPv6 will be lost and the lost fields cannot be regained even if the destination nodes (E and F) are IPv6 capable. This is a disadvantage in using this approach.

4.4.2 TUNNELING

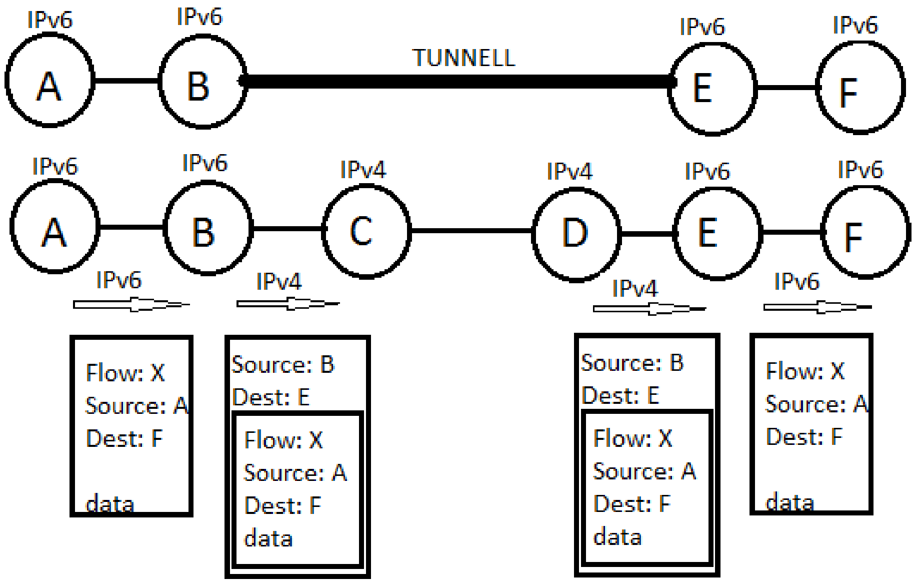


Figure 4.3 Tunneling

Due to the disadvantage in Dual-Stack approach, an alternative has been introduced and it is called Tunneling. A tunnel (two IPv4 routers) is formed between the two IPv6 routers. The sender side IPv6 node (node B) encapsulates the entire IPv6 datagram and adds it in the IPv4 data field and this IPv4 datagram is addressed to the receiver side of the tunnel (node E). The intervening IPv4 routers route these IPv4 datagram without knowing anything about the data field. On receiving this datagram the IPv6 node extracts the data field and then sends it to the final destination without losing any field in the datagram.

4.5 FUTURE

The use of IPv6 networks have been growing rapidly in the last few years, which ensures that Internet can meet all the user's requirements. The key factor for this is the availability of ample address spaces which allows all the devices to be connected online. IPv6 which is the successor to IPv4 has technological advancements over IPv4 in the fields of network security and business development.

In Future,

- IPv6 will be a part of all major operating systems and routers.
- Getting the right to configure IP addresses automatically will decrease the cost of ownership.
- The backbone route-table size is reduced due to improvisation in route aggregation.

Smooth transition is one of the major reasons for its success and there is no doubt that IPv6 will be the de-facto routing protocol of the 21st century.

CHAPTER 5

5. SIP ON IPV6

5.1 INTRODUCTION [1]

The telecommunication industries have been going through expeditious changes in communications. These changes are due to the improvement in the Internet and IP based applications. Internet has become a universal means of communication, therefore packet based traffic has been exceeding the circuit switched (traditional voice) traffic.

Due to rapid improvisation in technology, it is clear that VOIP will lead all other IP based applications in the telecommunication field.

5.2 SIP ADVANTAGES IN VOIP [1]

- **Reduced Cost:** Charges incurred in transporting calls across PSTN can be reduced. Due to the distributed nature of VOIP, providers and users can conserve BW by adding features only when it is absolutely necessary. Operation costs can be reduced by combining voice and data in the same network.
- **Open Standards & Interoperability:** VOIP applications allows us to use open standards, which in turn allows us to purchase equipment (that are interoperable with other equipment) from various vendors.
- **Integrated Networks:** If voice is treated as another IP application then integrated network for voice and video can be formed. This provides the same quality as PSTN.

The growth in packet telephony networks has also increased the requirements and dependencies in the telecom industry. Therefore the industries have to adopt VOIP protocols to meet all the requirements. The various call-control and signaling VOIP protocols are,

- H.323
- MGCP (Media Gateway Control Protocol)
- H.248/MEGACO
- SIP (Session Initiation Protocol)

5.3 PREFERRED SIGNALLING PROTOCOLS [1]

5.3.1 H.323

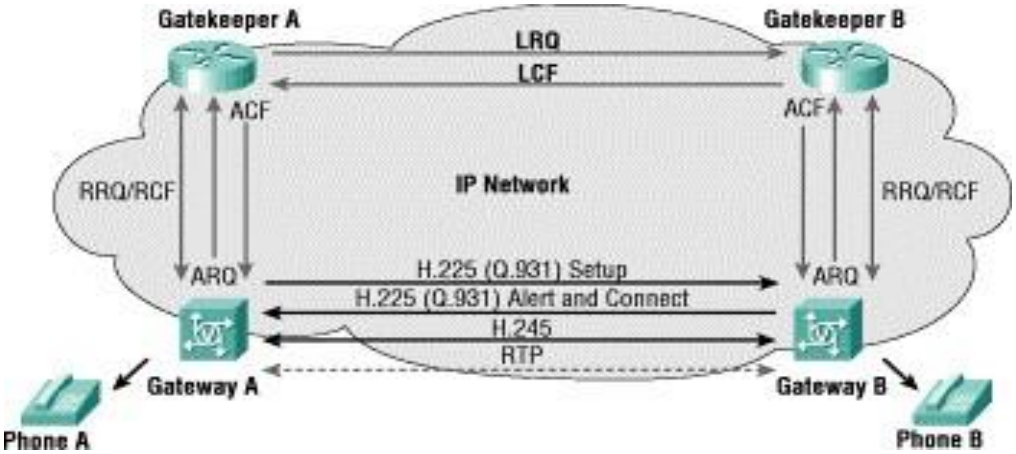


Figure 5.1 H.323 Architecture

H.323 was intended for transporting multi-media traffic over LAN (local area network). Due to its readiness and advancements it is the most extensively used signaling and call-control protocol. It is also used for videoconferencing applications by many vendors.

H.323 describes all the aspects of transmitting a call from establishment to exchange of available network resources. Hence it is called an ``Umbrella Protocol``.

Call Routing: H.323 defines the following protocols,

- Registration
- Admission
- RAS (Status Protocol)

Call Setup: H.225 protocols.

Exchange Capabilities: H.245 protocols.

H.323 is based on ISDN (Integrated Services Digital Network) Q.931 protocol that makes it compatible with the PSTN and SS7 (Signaling System 7).

H.323 lets the industries to build large, scalable, redundant and resilient networks. It is interoperable with other VOIP systems. Network intelligence is supported at the endpoints/gatekeepers.

5.3.2 MGCP/H.248/MEGACO

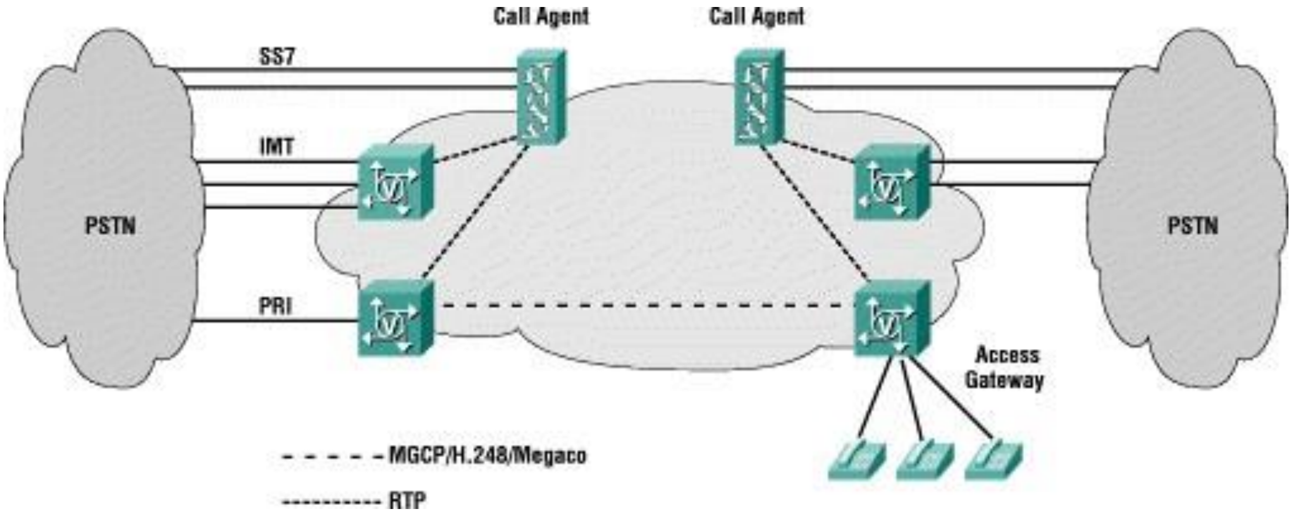


Figure 5.2 MGCP & H.248/MEGACO Architecture

The architecture of MGCP and H.248/MECAGO resembles PSTN architecture and allows us to centrally add the call control and other services to a VOIP network.

Packages are the key model used for signaling in MGCP and H.248/MECAGO. The packages support various functions such as signaling in PSTN, line side connectivity and also supports call transfer and hold features.

Due to its centralized architecture, MGCP and H.248/MECAGO let the industries to build large, scalable, redundant and resilient networks. It adds intelligence feature and provides options for interconnecting with other VOIP systems.

5.3.3 SESSION INITIATION PROTOCOL [2]

Session initiation protocol is a signaling protocol that is used to establish, maintain and tear down a connection. A session could be any multimedia application or a simple phone call. SIP is the most preferred option for any VOIP based application.

Though SIP is growing in many aspects, its main function is to set up a connection and control the session. It doesn't take care of the data exchange and other protocols are required for this.

The four major operations performed by SIP are,

- **User Location Establishment:** It establishes a user connection by translating the name into a network address.
- **Negotiation:** It provides negotiation feature, which allows the users in a session to agree upon common terms.
- **Call Management:** It manages the established connection and also adds features like holding a call, transferring to the next user, adding a user and dropping a user.
- SIP allows changing the features even on an on-going session.

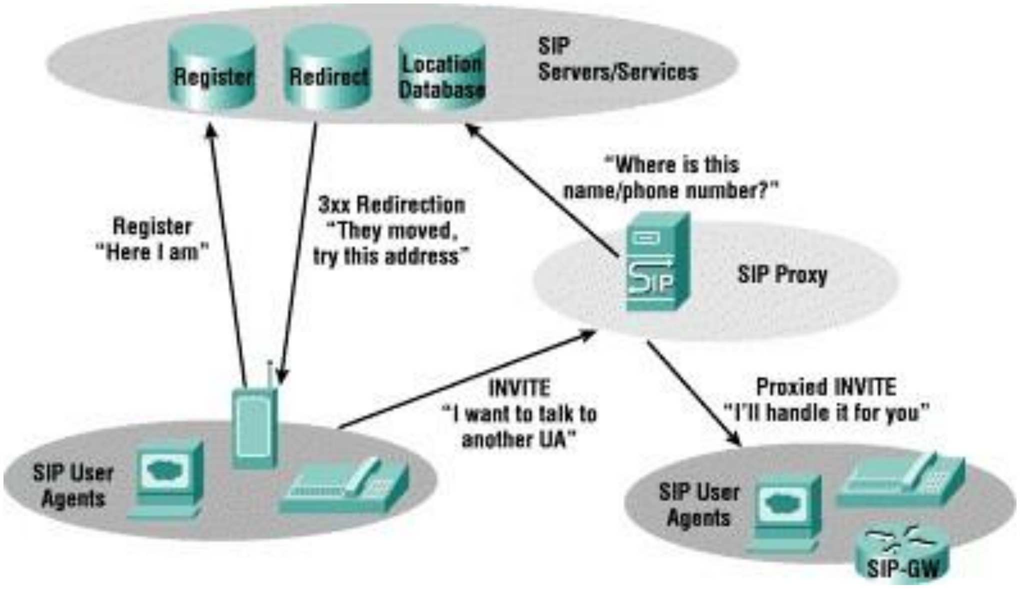

Register
Redirect
Location Database
SIP Servers/Services
Register "Here I am"
3xx Redirection "They moved, try this address"
"Where is this name/phone number?"
SIP
SIP Proxy
INVITE "I want to talk to another UA"
Proxied INVITE "I'll handle it for you"
SIP User Agents
SIP User Agents
SIP-GW

Figure 5.3 SIP Architecture

User Agents (UA) are the entities in a SIP environment. The UA can function as both client and server. User Agent Client (UAC) creates requests and passes it on to the server. User Agent Server (UAS) receives requests, processes them and gives replies to the clients.

CLIENTS: The end users are often referred to as clients. It could be a softphone on a laptop or a messaging device on an IP phone. The client generates a request and sends it to the server, usually a SIP proxy server, where the request gets processed to produce a response.

SERVERS: Servers are integral part of the network and they work according to the set protocols. SIP environment has various types of servers namely,

- **Proxy Server:** When a client generates a request it doesn't know the exact address of the recipient so it forwards its request to the proxy server, which on behalf of the client processes it and sends to the recipient or another proxy server.
- **Redirect Server:** This server sends the client`s request back to the client asking it to try a different route. This happens when the intended recipient has moved out of its place.
- **Registrar Server:** It detects the location of all the users in the network. Users inform the registrar about their location each time they make a move by sending special type of messages.
- **Location Server:** Stores all the information gathered by the registrar server.

5.4 SIP BENEFITS FROM IPV6 [1]

The main reason for using SIP with IPv6 is the availability of ample IPv6 addresses. Taking 3G networks into account, numerous cell phones that work on SIP requires IP addresses. Not only cell phones, even other devices or applications based on SIP and that connects to the internet requires IP addresses. So IPv6 will be a platform to provide more and more IP addresses with increasing SIP based user. Other advantages include,

- Dynamic configuration and Load Balancing
- Anycast
- NAT problem in SIP

5.4.1 DYNAMIC CONFIGURATION

When a SIP UA tries to establish a SIP session with another UA, besides the IP address of the recipient, it also require some specific details like the address of the proxy/registrar server and the home domain name. These addresses cannot be static as it changes as per the UA`s location (which changes rapidly). The dynamic configuration capabilities in IPv6 will be very helpful in resolving this problem.

5.4.2 ANYCAST

When a UA wants to start a session, it must send all of its SIP messages to the SIP server to get authenticated and also to stay away from firewall problems. Tracking the location of these servers is difficult. Therefore if the servers with same functionality are grouped and given an anycast address then the UA can send its request to the anycast address and the server that is connected to the nearest interface will respond and this speeds up the process and helps in load balancing.

5.4.3 NAT PROBLEM IN SIP

One of the biggest problems in SIP telephony is NAT traversal. The non-availability of enough address spaces and security features made it a problem in IPv4 environment. NAT/ Firewall servers were not designed for SIP communication or to accommodate real time traffic. However in IPv6 environment NAT shouldn't be a big problem because IPv6 provides numerous IP addresses and security features. Therefore NAT problems in SIP are eliminated by introducing IPv6 extended address spaces and IPSec function. IPSec enables authentication, confidentiality, integrity of data at the network layer. By using SIP in an IPv6 environment will eradicate most of the problems that were prevailing in IPv4 networks.

CHAPTER 6

6. ASTERISK

6.1 INTRODUCTION [2]

In today`s world asterisk is the most prosperous PBX (Private Branch exchange) system. It has made a revolution in the telecommunication industry by replacing the traditional PBX system.

Twenty years back, telephones were the only means of voice communication. But now due to rapid advancement in the telephony people have started to migrate towards IP based telephony system. Audio calls, video calls, instant messaging, using internet is preferred over traditional voice calls.

Asterisk is a free platform to build different kinds of communication applications. It is possible to make a PC work as a communication server by using asterisk. It can be used in IP based PBX systems and VOIP systems. It is backwards compatible with all the existing voice systems, protocols and codecs that are used in voice systems.

It supports,

- IVR (Interactive Voice Response)
- ACD (Automatic Call Distributor)
- Audio/Video conferencing
- Voicemail
- Call Recording
- Fax Termination
- CDR (Call Detail Record)

6.2 ARCHITECTURE [3]

Asterisk follows a modular architecture. The building blocks of asterisk are modules. Each module is a loadable component and each one performs a special function. Asterisk could be started even without loading any of the modules but it will not be able to perform its tasks. So the modules are very important for its proper functioning.

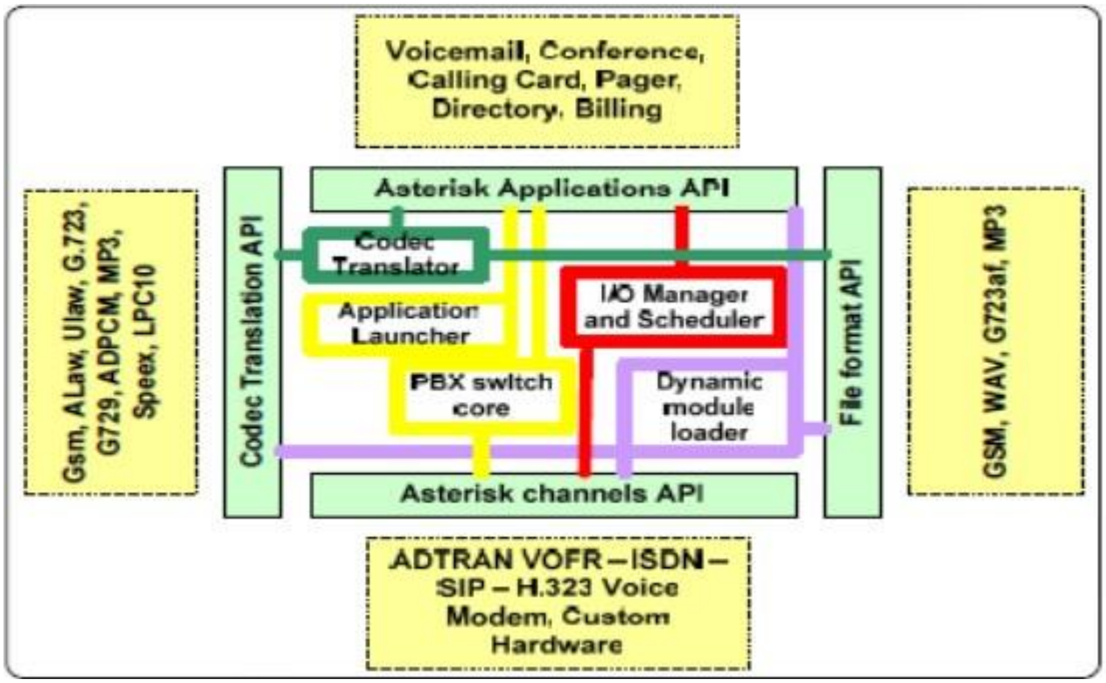

Voicemail, Conference, Calling Card, Pager, Directory, Billing
Gsm, ALaw, Ulaw, G.723, G729, ADPCM, MP3, Speex, LPC10
Codec Translation API
Asterisk Applications API
Codec Translator
Application Launcher
I/O Manager and Scheduler
PBX switch core
Dynamic module loader
Asterisk channels API
File format API
GSM, WAV, G723af, MP3
ADTRAN VOFR – ISDN – SIP – H.323 Voice Modem, Custom Hardware

Figure 6.1 Asterisk Architecture

6.2.1 MODULES

- **Applications:** Dialplan application defines the actions that are needed for a call. It takes care of the outgoing calls that are intended to reach an external party.
- **Bridging:** The process of bridging channels in a new API is performed by the bridging modules.
- **CDR:** Call detail recording module defines the method of storing the call logs. It can be stored in a file by default, database, syslog.

- **CEL:** Channel event logging gives more control over call reporting. This can be enabled by properly configuring the dialplan and it is not available automatically, it needs to be configured.
- **Channel Drivers:** No channel drivers, No calls. This is the gateway to asterisk. Each channel/protocol needs a driver to function properly.
- **Codec Translator:** The audio stream formats are translated (between calls) using Codec translator.
- **Format Interpreters:** It performs the same function as a codec translator but at the file level and not at the channel level. For example, Format Interpreter is needed to play a recording (stored as GSM) to any channel without using GSM codec.
- **Dialplan Function:** Complements dialplan applications. Provides many enhanced features like ODBC connectivity, string and time/date handling.
- **PBX Module:** PBX module is used for controlling and configuring the dialplan. Default asterisk dialplan is loaded using this module.
- **Resource Module:** Allows asterisk to communicate with the external resources.
- **Add-on Module:** Add-ons are developed by the communities and these are not installed by default.
- **Test Module:** For validating new codes that are developed by the asterisk team.

6.2.2 ASTERISK & TRADITIONAL PBX [1]

Asterisk is different from traditional PBX systems. All the incoming calls are treated in the same manner in the dialplan of asterisk.

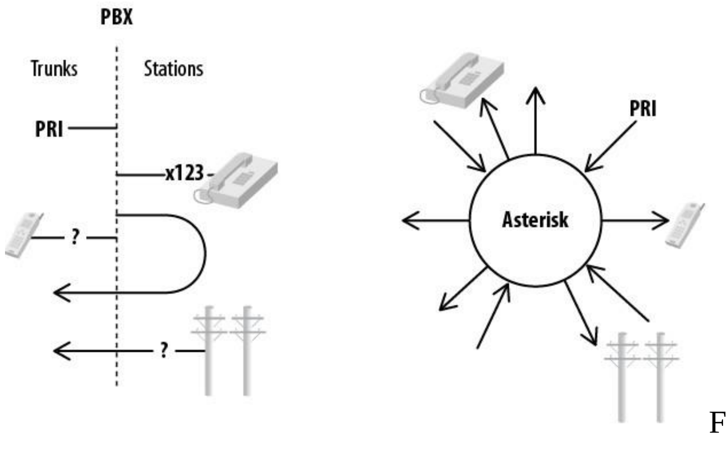


Figure 6.2 Asterisk & PBX

In PBX, there is a difference between the telephone sets (stations) and the resources connecting the outside world (trunk).

For instance, An external call cannot be routed to the destination without the user actually entering the party`s extension and an external gateway cannot be installed on a station port.

Off-site resource concept cannot be implemented in PBX because external resources cannot access the internal features.

On the other hand, asterisk doesn’t have anything called trunks/stations and uses the same type of channel to process both incoming and outgoing calls. Even different channels get treated in the same way. An internal user calling from a cell phone (external trunk) will deserve the same treatment as an internal user actually on the extension.

6.2.3 OPERATING SYSTEM

Asterisk was originally designed for LINUX. It can also run on various platforms namely, MAC OS X, FreeBSD, CentOS, Ubuntu, NetBSD, OpenBSD and Solaris.

6.2.4 PROTOCOLS

Asterisk supports various VOIP protocols like SIP, MGCP, H.323 and it is interoperable with almost all the SIP telephones, traditional and VOIP telephony systems.

6.2.5 WORKING

The most inexpensive way of calling a phone is using asterisk. In a communication that uses asterisk, a phone or a computer is the client and asterisk is the server. When the communication is established using SIP then the client must be a SIP phone. First step in using asterisk to make calls is to get registered to the asterisk server, the client devices must be configured correctly and connected to the server. Once you get registered a username and password will be given, this can be used for further communication purposes.

6.3 ADVANTAGES

- Open source framework that can be customized. It is stable and reliable.
- There is no hardware involved so it’s cost effective.
- Flexible and Interoperable.
- Acts as a server which enables the client devices to communicate easily.
- Cheaper solution for VOIP based server communication.

6.4 APPLICATIONS

- Asterisk could be the base for IP based PBX. It can control all the calling operations and can connect to the outside world over IP, POTS (analog) and T1/E1 connections (digital).
- Asterisk could act as a gateway bridging PSTN and IP telephony.
- It can make a conference bridge, automated attendant, advanced voicemail system, unified messaging and can be an interface to the web.
- Asterisk is used in Call Centers. It supports advanced routing, bulk dialing, remote IP agent capabilities and more.
- It could be a base for building real-time voice and video application, just like apache to web applications.
- It can be used in business phone systems.

CHAPTER 7

7. CODECS

7.1 INTRODUCTION

Coder-Decoder / Compressor-Decompressor are collectively known as a CODEC. It is a software or hardware used to encode and decode a digital signal (media file) during transmission. An encoder compresses the digital data during transmission, whereas a decoder decompresses the data and sends it out. Two major types of Codecs are,

- Audio Codec
- Video Codec

7.2 AUDIO CODEC

Audio Codec could be either software or hardware based; it is used to process any digital audio signal. In software it uses a set of rules to produce high-accuracy output (reproducing the input) with minimizing the number of bits and increasing the quality. It processes (encodes & decodes) the digital audio to produce a defined file or media format.

In hardware, a codec performs both encoding and decoding. It encodes an analog signal into digital signal using ADC (Analog to Digital Converter) and decodes it back to analog signal using DAC Digital to Analog Converter) at the output.

7.2.1 SUPPORTED AUDIO CODECS

- G.711 Mu-Law (Used in North America and Japan)
- G.711 A-Law (Used in rest of the world)
- G.722
- G.723.1
- G.726
- G.729
- GSM

- LPC10
- iLBC
- Speex
- ADPCM

Note: The most preferred and used audio codec for Web RTC is Speex.

7.2.2 SPEEX [1]

Speex is an open source speech codec developed for audio compression. It is based on CELP (Code Excited Linear Prediction). Its compression bit rates range from 2Kbps to 44Kbps.

7.2.2.1 FEATURES

- It supports compression in Narrowband (8Kbps), Wideband (16Kbps) and Ultra-Wideband (32Kbps) and it takes place in the same bit stream.
- It provides intensity stereo encoding.
- It allows mixing of multiple sampling rates in one bit stream, called as embedded coding.
- It supports Packet loss concealment (PLC) and hides losses in transmission.
- It uses the mechanism of Discontinuous Transmission (DTx) to reduce the BW usage during periods of silence using VAD (Voice Activity Detection) and CNG (Comfort Noise Generation).
- It support VBR (Variable bit rate) mode.
- It has the capacity to cancel the acoustic echo production at the remote end even in the presence of multiple speakers and microphones.
- It is capable of suppressing noise.

7.2.2.2 APPLICATIONS

- VOIP and used in Asterisk.
- Streaming of audio in the internet.
- Voicemail (archival).
- Audio Books.

7.3 VIDEO CODEC

Video Codec could be either software or hardware based; it is used to process any digital video signal. The goal of video compression is to deliver a video with high accuracy that closely resembles the input, simultaneously compressing it into a smallest file possible.

7.3.1 SUPPORTED VIDEO CODECS

- H.261
- H.263
- H.264
- VP8

Note: The most preferred and used video codec for Web RTC is VP8

7.3.2 VP8 [2][3]

VP8 is an open source video codec developed by On2 Technologies and released by Google. VP8 is the video codec used in HTML5, supported by Firefox and Chrome for streaming video in the web browser. After the introduction of WebM by Google, VP8 became the center of attraction because of its support to WebM. It enticed many developers and researchers because of its distinctive features.

7.3.2.1 FEATURES

- Provides high compression at a low computational complexity.
- It is designed to use the available network bandwidth, so the video quality ranges from watchable to lossless.
- It is designed to support web video format (420 color samples per image, 8-bit color depth per channel, dimension: 16383x16383 pixels and progressive scan without mixing).
- It supports both VBR (Variable Bit Rate) and CBR (Constant Bit Rate).
- It supports parallel processing which improves the decoder performance in a multi-processor environment.
- Compared to other codecs, VP8 is considered to give twice the quality by consuming half the network bandwidth.

- The reference frame called the ‘Golden Frame’ is used to improve the performance of the encoder. It enhances the coding efficiency and error recovery in video conferences.
- The ‘Alternate Reference’ Frame is used to remove the noise-reduced prediction.
- Even though the buffer size is limited to three reference frames, effective de-correlation is achieved in video motion compensation.
- The entropy coding at frame level is adaptive which results in a complete balance between compression efficiency and the complexity in computation.

7.3.2.2 APPLICATIONS

- Media Calls Server like Skype.
- Streaming of Video in the internet.
- Youtube.
- Flash.
- Video mail.
- Supports Nvidia.

CHAPTER 8

8. METHODS & TECHNIQUES

8.1 SYSTEM ARCHITECTURE [1]

Raspberry Pi is used as the Location Server in this project. Three laptops with Web RTC enabled browsers (Chrome or Firefox) are used as the mobile clients. The IPv6 cloud is generated by connecting the server to an IPv6 capable Dlink router. The server and the three clients are in the same IPv6 network. The location server computes the distance of each mobile node and stores it in the SQLite database for reference purposes (transferring calls to the nearest neighboring node). In this project, Rpi works as the location server, SIP server and web server and the laptops with browsers running Web RTC are the SIP clients.

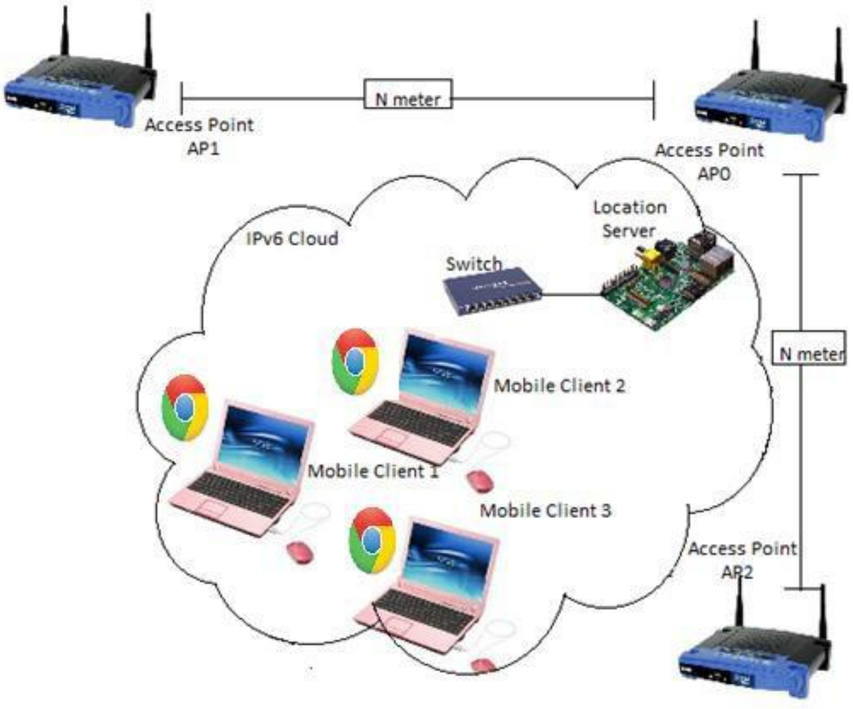


Figure 8.1 System Architecture

8.2 DISTANCE CALCULATION [1][2][3]

The location server uses triangulation method to calculate the distances of the clients.

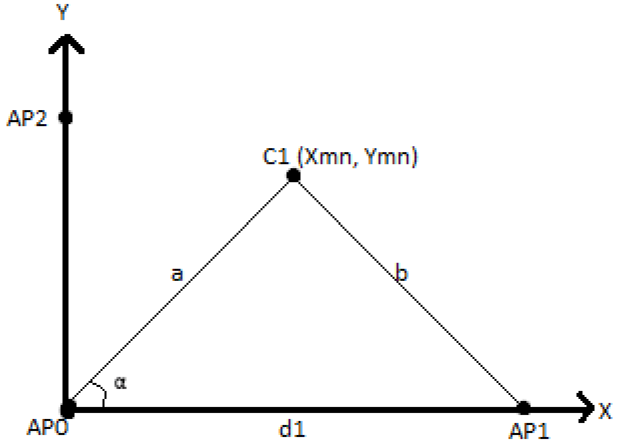


Figure 8.2 Triangulation Method

From the Figure 8.2, α is given by

$$\alpha = cos^{-1}\left(\frac{a^2-b^2+d_1{}^2}{2*d_1*a}\right) \qquad \text{———— (1)}$$

Where, a= distance between AP0 and mobile client C1.

b= distance between C1 and AP1.

d1= distance between AP0 and AP1.

Xmn and Ymn of the particular client node C1, can be calculated by the below formula,

$$X_{mn} = a * \cos(\alpha) \qquad \text{———— (2)}$$

$$Y_{mn} = a.\sin(\alpha) \qquad \text{———— (3)}$$

The indoor signal path loss delay obeys distance power law, which is

$$P_r(d) = \Pr(d_0) - 10 * n * log\left(\frac{d}{d_0}\right) + X_\sigma \qquad \text{———— (4)}$$

Where, Pr(d)= received power.

Pr(d0)= received power at a distance d0.

n= path loss component.

d0= reference distance (1m).

X_σ = Zero mean Gaussian random variable with standard deviation σ

Equation (4) is modified to have wall attenuation factor (WAF),

$$\mathrm{P_r}(d) = \mathrm{Pr}(d_0) - 10 * n * log\left(\frac{d}{d_0}\right) - T * WAF \quad (5)$$

Where, T= number of walls between the Tx and Rx.

From equation (5) we can derive an equation to calculate the distance,

$$d = e^{\left(\frac{P_r(d_0) - P_r(d) - T * WAF}{10 * n}\right)} \quad \text{———} \quad (6)$$

Where‘d’ gives the distances between access points and mobile client.

When the location of the mobile client is calculated, the below equation is used to calculate the distance of every device.

$$Dist = \sqrt{((X_{mn1} - X_{mn2})^2 + (Y_{mn1} - Y_{mn2})^2)} \quad \text{———} \quad (7)$$

The location server computes the distance of every mobile device using equation (7) and the compares all the distances to find out the nearest neighbor.

Note: Since the project is experimented in a small room the received signal strength (RSSI) from all the client nodes will be the same, which would cause difficulties in computing the nearest neighbor. So due to this inconsistency I have managed to use the above concept (Ref 8.2) with pre-assumed distance values values for each mobile client.

8.3 WORKING OF SIP [1]

SIP1 and SIP2 are the two mobile clients. SIP1 requests SIP2 to join the session and SIP2 joins the session by accepting the invite. Once the session starts the clients start exchanging the data through the Web RTC enabled browser. The session could be a voice, video, voice & video or instant messaging. Any time during the session, one of the client could transfer the entire session over to another client and this is will be the nearest neighbor decided by the location server.

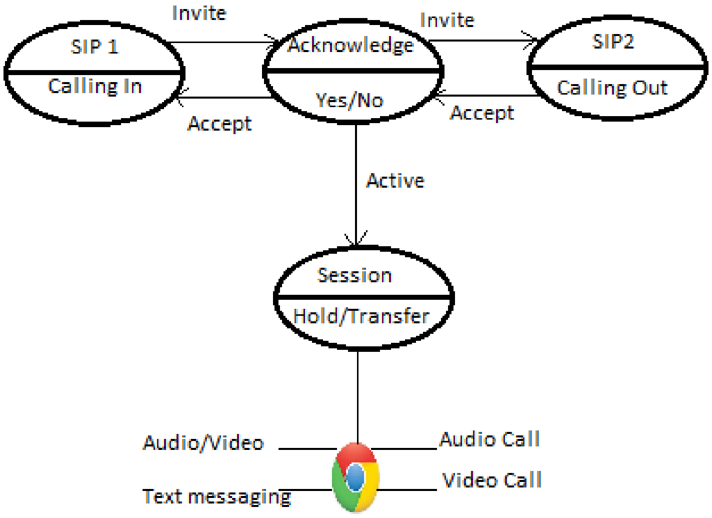
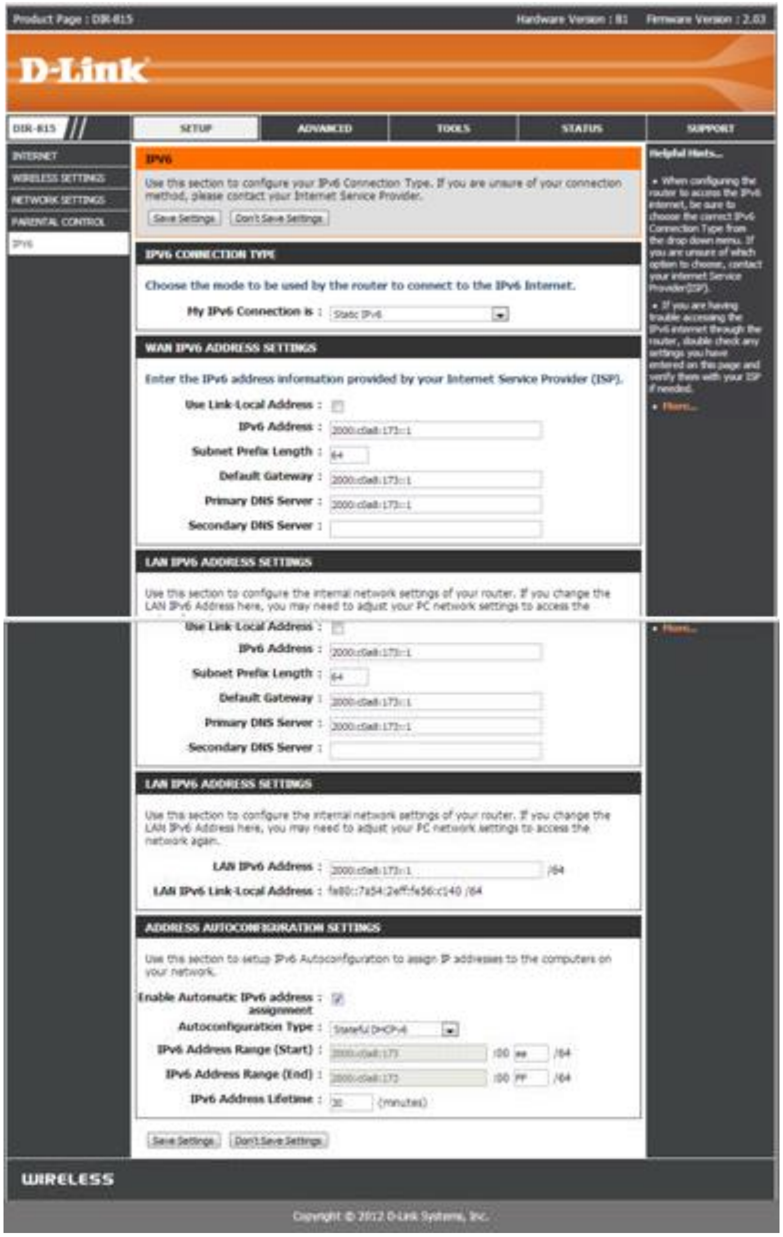


Figure 8.3 SIP Activity

8.4 CONFIGURATIONS

8.4.1 IPV6 NETWORK

- **Router Configuration:**



Product Page : DIR-815 Hardware Version : B1 Firmware Version : 2.03

IPV6

Use this section to configure your IPv6 Connection Type. If you are unsure of your connection method, please contact your Internet Service Provider.

Save Settings | Don't Save Settings

IPV6 CONNECTION TYPE

Choose the mode to be used by the router to connect to the IPv6 Internet.

My IPv6 Connection is : Static IPv6

WAN IPV6 ADDRESS SETTINGS

Enter the IPv6 address information provided by your Internet Service Provider (ISP).

Use Link-Local Address : □
IPv6 Address : 2000:c0a8:173::1
Subnet Prefix Length : 64
Default Gateway : 2000:c0a8:173::1
Primary DNS Server : 2000:c0a8:173::1
Secondary DNS Server :

LAN IPV6 ADDRESS SETTINGS

Use this section to configure the internal network settings of your router. If you change the LAN IPv6 Address here, you may need to adjust your PC network settings to access the

Use Link-Local Address : □
IPv6 Address : 2000:c0a8:173::1
Subnet Prefix Length : 64
Default Gateway : 2000:c0a8:173::1
Primary DNS Server : 2000:c0a8:173::1
Secondary DNS Server :

LAN IPV6 ADDRESS SETTINGS

Use this section to configure the internal network settings of your router. If you change the LAN IPv6 Address here, you may need to adjust your PC network settings to access the network again.

LAN IPv6 Address : 2000:c0a8:173::1 /64
LAN IPv6 Link-Local Address : fe80::7a54:2eff:fe56:c140 /64

ADDRESS AUTOCONFIGURATION SETTINGS

Use this section to setup IPv6 Autoconfiguration to assign IP addresses to the computers on your network.

Enable Automatic IPv6 address assignment : ☑
Autoconfiguration Type : Stateful DHCPv6
IPv6 Address Range (Start) : 2000:c0a8:173 ::00 ee /64
IPv6 Address Range (End) : 2000:c0a8:173 ::00 FF /64
IPv6 Address Lifetime : 30 (minutes)

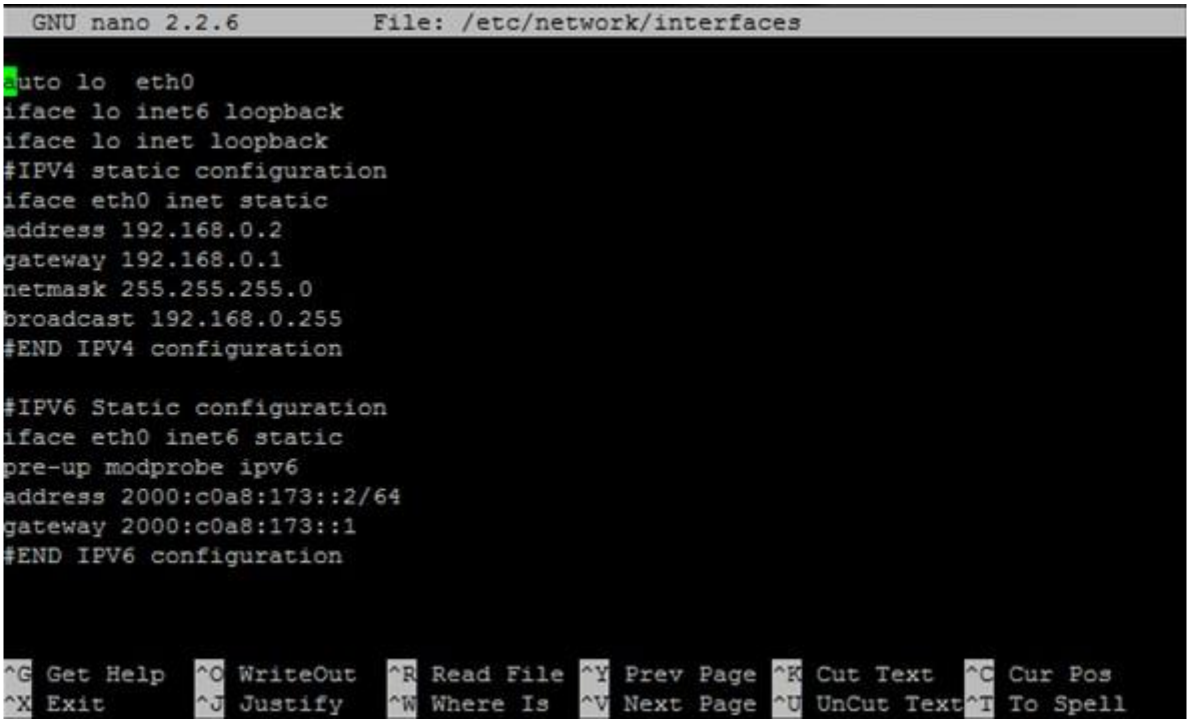
Save Settings | Don't Save Settings

Helpful Hints...

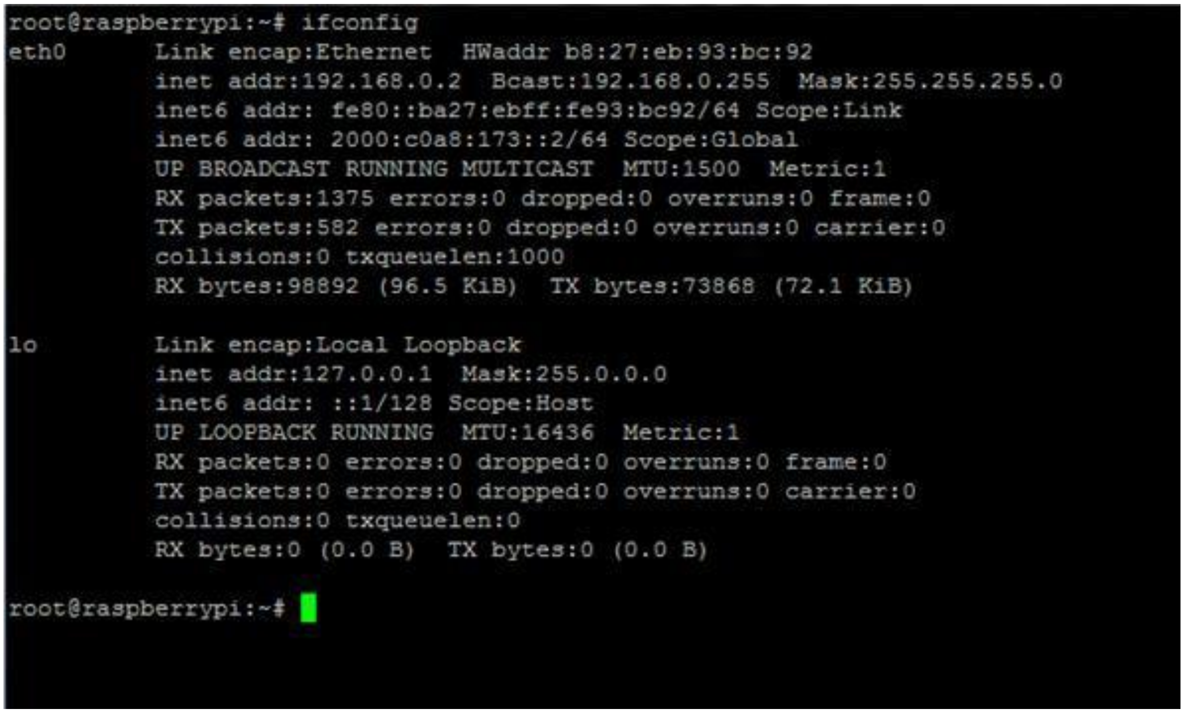
- When configuring the router to access the IPv6 internet, be sure to choose the correct IPv6 Connection Type from the drop down menu. If you are unsure of which option to choose, contact your internet Service Provider(ISP).
- If you are having trouble accessing the IPv6 internet through the router, double check any settings you have entered on this page and verify them with your ISP if needed.
- More...

Copyright © 2012 D-Link Systems, Inc.

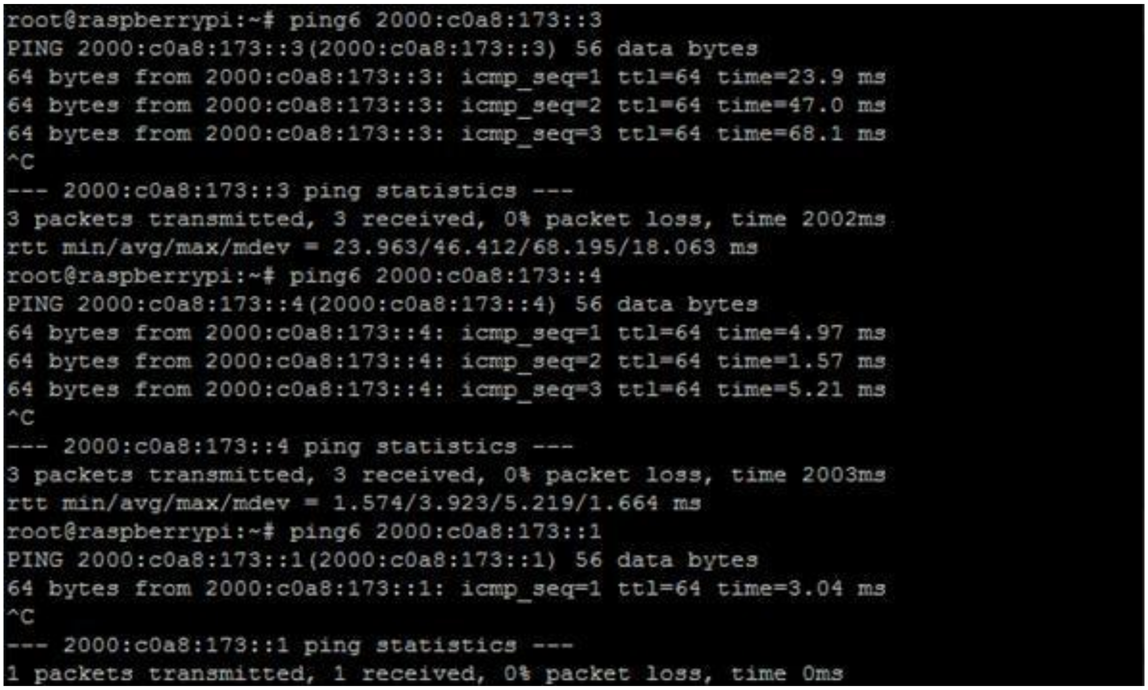
- **Server Configuration**



```
  GNU nano 2.2.6          File: /etc/network/interfaces

auto lo  eth0
iface lo inet6 loopback
iface lo inet loopback
#IPV4 static configuration
iface eth0 inet static
address 192.168.0.2
gateway 192.168.0.1
netmask 255.255.255.0
broadcast 192.168.0.255
#END IPV4 configuration

#IPV6 Static configuration
iface eth0 inet6 static
pre-up modprobe ipv6
address 2000:c0a8:173::2/64
gateway 2000:c0a8:173::1
#END IPV6 configuration



^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

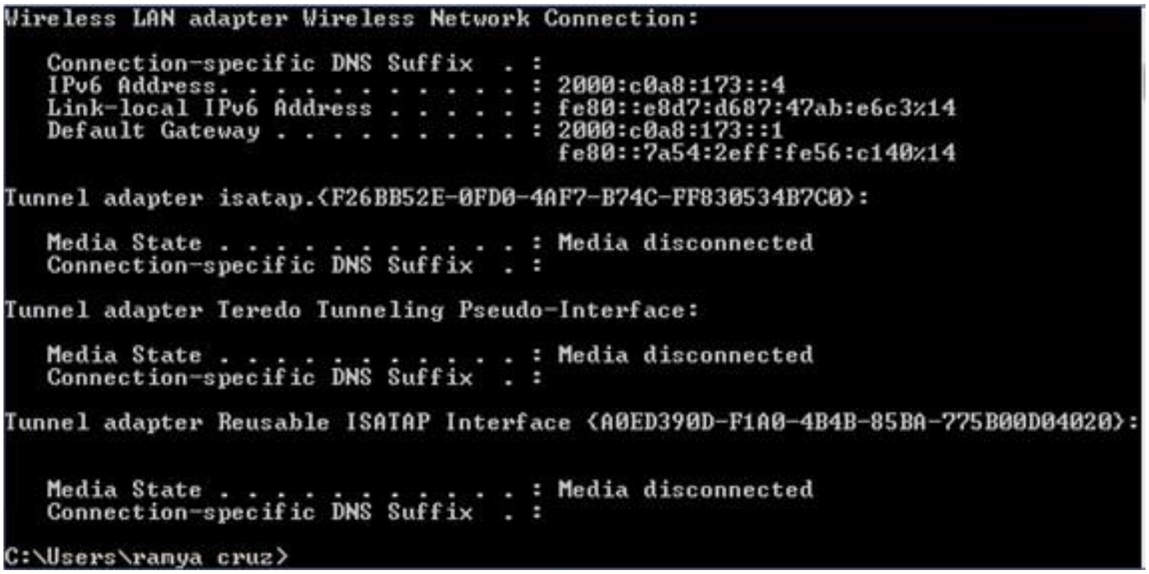


```
root@raspberrypi:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:93:bc:92
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe93:bc92/64 Scope:Link
          inet6 addr: 2000:c0a8:173::2/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1375 errors:0 dropped:0 overruns:0 frame:0
          TX packets:582 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:98892 (96.5 KiB)  TX bytes:73868 (72.1 KiB)

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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@raspberrypi:~#
```

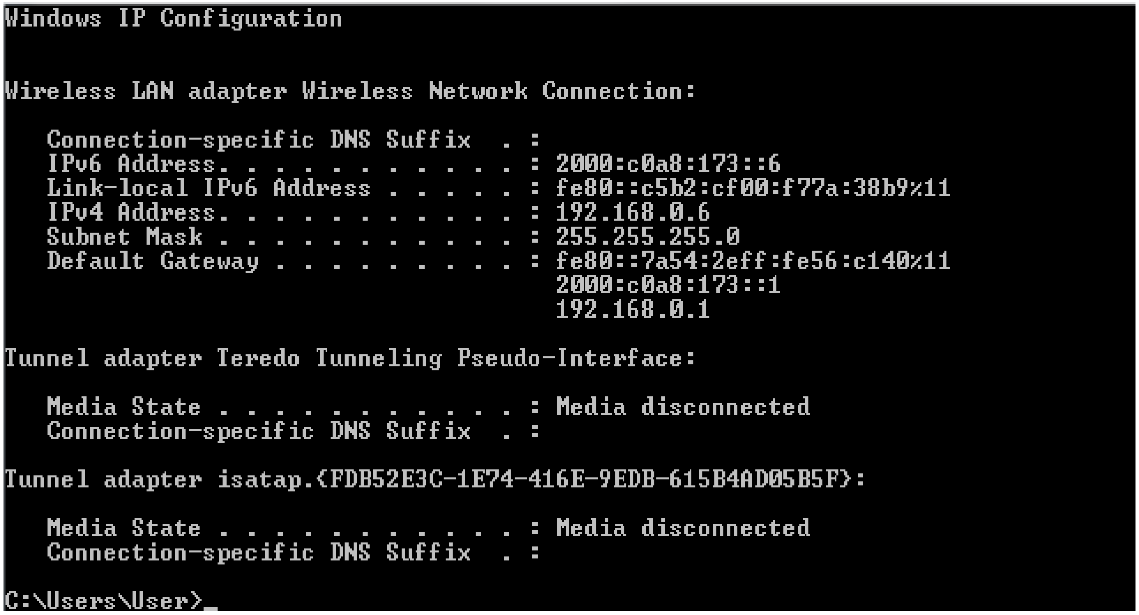


```
root@raspberrypi:~# ping6 2000:c0a8:173::3
PING 2000:c0a8:173::3(2000:c0a8:173::3) 56 data bytes
64 bytes from 2000:c0a8:173::3: icmp_seq=1 ttl=64 time=23.9 ms
64 bytes from 2000:c0a8:173::3: icmp_seq=2 ttl=64 time=47.0 ms
64 bytes from 2000:c0a8:173::3: icmp_seq=3 ttl=64 time=68.1 ms
^C
--- 2000:c0a8:173::3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 23.963/46.412/68.195/18.063 ms
root@raspberrypi:~# ping6 2000:c0a8:173::4
PING 2000:c0a8:173::4(2000:c0a8:173::4) 56 data bytes
64 bytes from 2000:c0a8:173::4: icmp_seq=1 ttl=64 time=4.97 ms
64 bytes from 2000:c0a8:173::4: icmp_seq=2 ttl=64 time=1.57 ms
64 bytes from 2000:c0a8:173::4: icmp_seq=3 ttl=64 time=5.21 ms
^C
--- 2000:c0a8:173::4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.574/3.923/5.219/1.664 ms
root@raspberrypi:~# ping6 2000:c0a8:173::1
PING 2000:c0a8:173::1(2000:c0a8:173::1) 56 data bytes
64 bytes from 2000:c0a8:173::1: icmp_seq=1 ttl=64 time=3.04 ms
^C
--- 2000:c0a8:173::1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

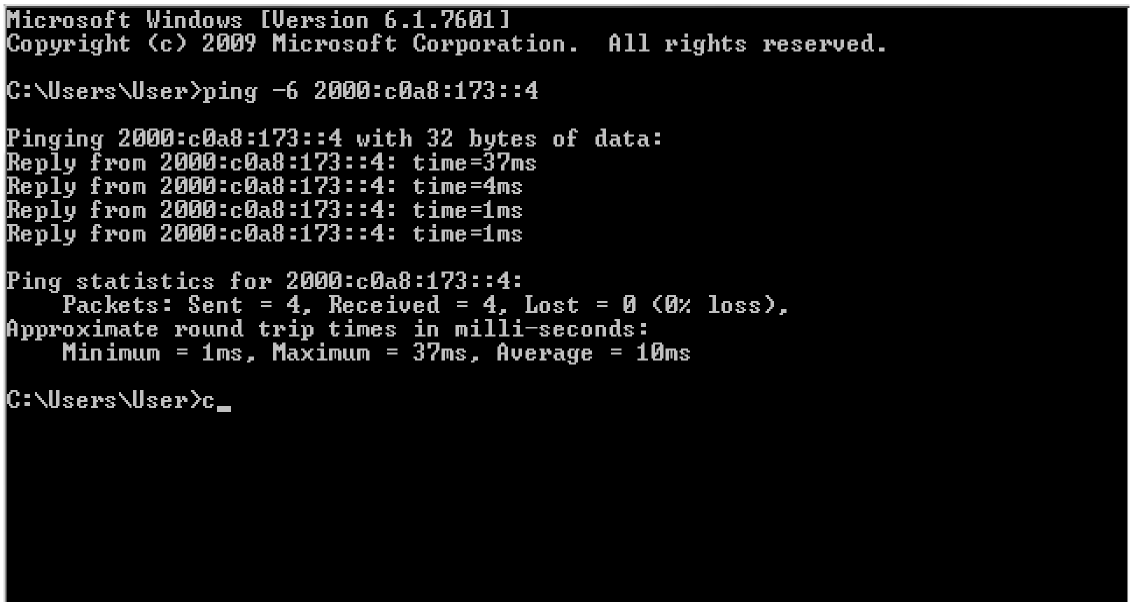
- **Client Configuration:**



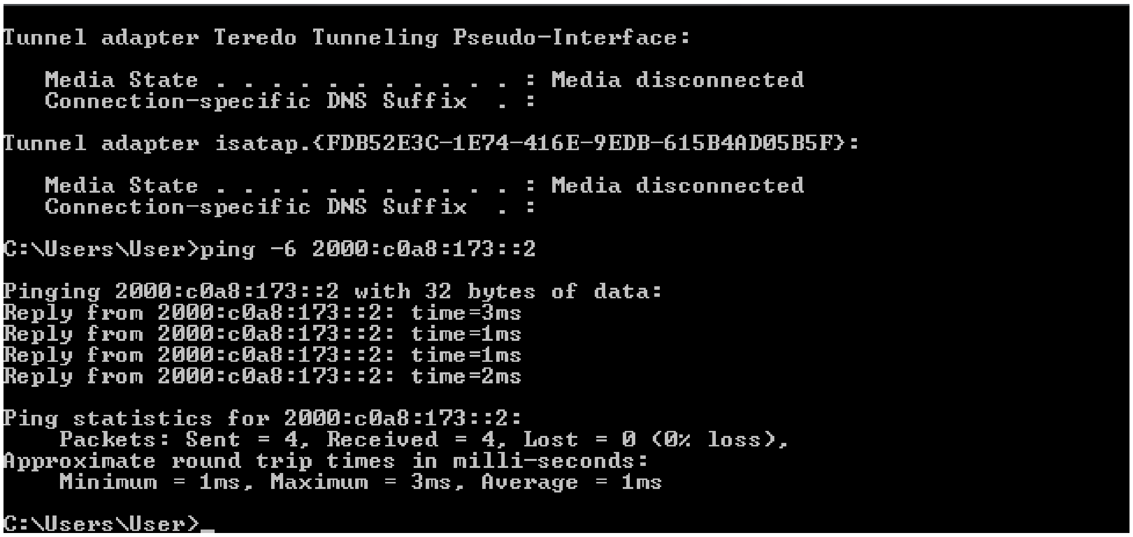
```
Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2000:c0a8:173::4
   Link-local IPv6 Address . . . . . : fe80::e8d7:d687:47ab:e6c3%14
   Default Gateway . . . . . . . . . : 2000:c0a8:173::1
                                       fe80::7a54:2eff:fe56:c140%14

Tunnel adapter isatap.{F26BB52E-0FD0-4AF7-B74C-FF830534B7C0}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Reusable ISATAP Interface {A0ED390D-F1A0-4B4B-85BA-775B00D04020}:


   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\ramya cruz>
```



```
Windows IP Configuration


Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2000:c0a8:173::6
   Link-local IPv6 Address . . . . . : fe80::c5b2:cf00:f77a:38b9%11
   IPv4 Address. . . . . . . . . . . : 192.168.0.6
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::7a54:2eff:fe56:c140%11
                                       2000:c0a8:173::1
                                       192.168.0.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{FDB52E3C-1E74-416E-9EDB-615B4AD05B5F}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\User>_
```



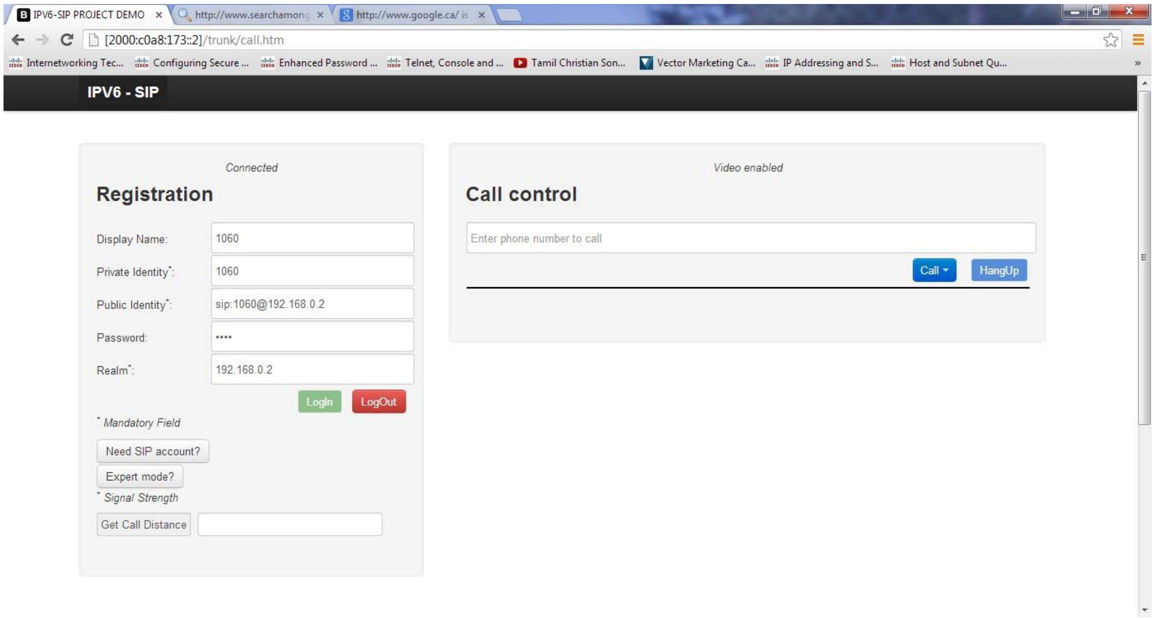
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\User>ping -6 2000:c0a8:173::4

Pinging 2000:c0a8:173::4 with 32 bytes of data:
Reply from 2000:c0a8:173::4: time=37ms
Reply from 2000:c0a8:173::4: time=4ms
Reply from 2000:c0a8:173::4: time=1ms
Reply from 2000:c0a8:173::4: time=1ms

Ping statistics for 2000:c0a8:173::4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 37ms, Average = 10ms

C:\Users\User>c_
```



```
Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{FDB52E3C-1E74-416E-9EDB-615B4AD05B5F}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\User>ping -6 2000:c0a8:173::2

Pinging 2000:c0a8:173::2 with 32 bytes of data:
Reply from 2000:c0a8:173::2: time=3ms
Reply from 2000:c0a8:173::2: time=1ms
Reply from 2000:c0a8:173::2: time=1ms
Reply from 2000:c0a8:173::2: time=2ms

Ping statistics for 2000:c0a8:173::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms

C:\Users\User>_
```

- **Web RTC Setup**

❖ **Client 1060**



[2000:c0a8:173::2]/trunk/call.htm

Connected

Registration

Display Name: 1060

Private Identity*: 1060

Public Identity*: sip:1060@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* *Mandatory Field*

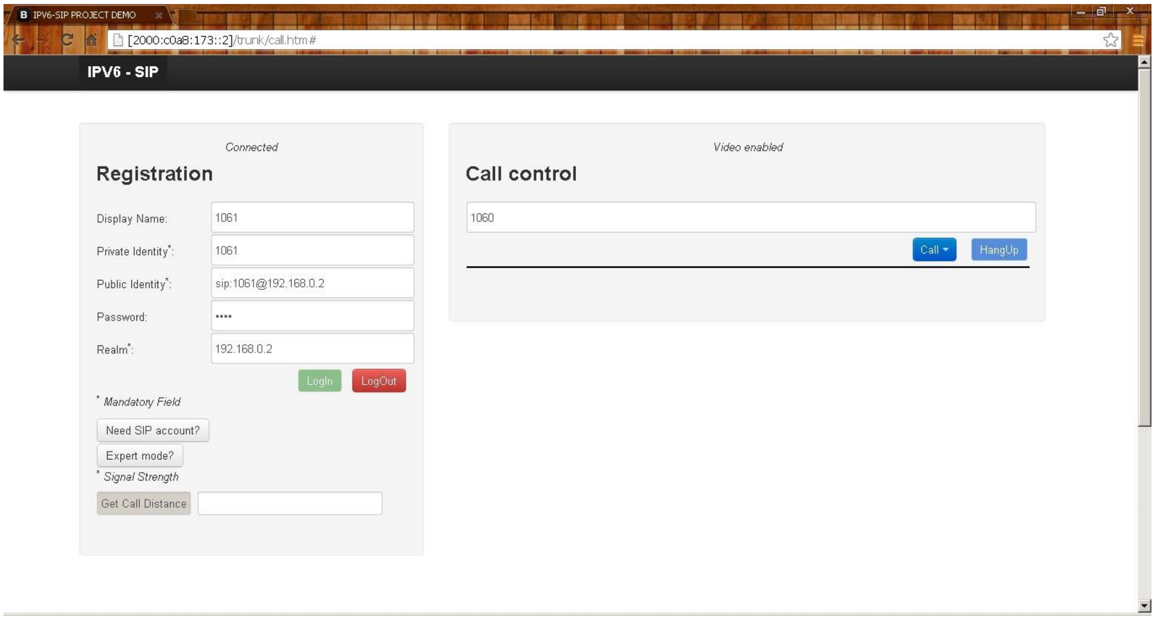
* *Signal Strength*

Video enabled

Call control

Enter phone number to call

❖ Client 1061



Connected

Registration

Display Name:	1061
Private Identity*:	1061
Public Identity*:	sip:1061@192.168.0.2
Password:	••••
Realm*:	192.168.0.2

* *Mandatory Field*

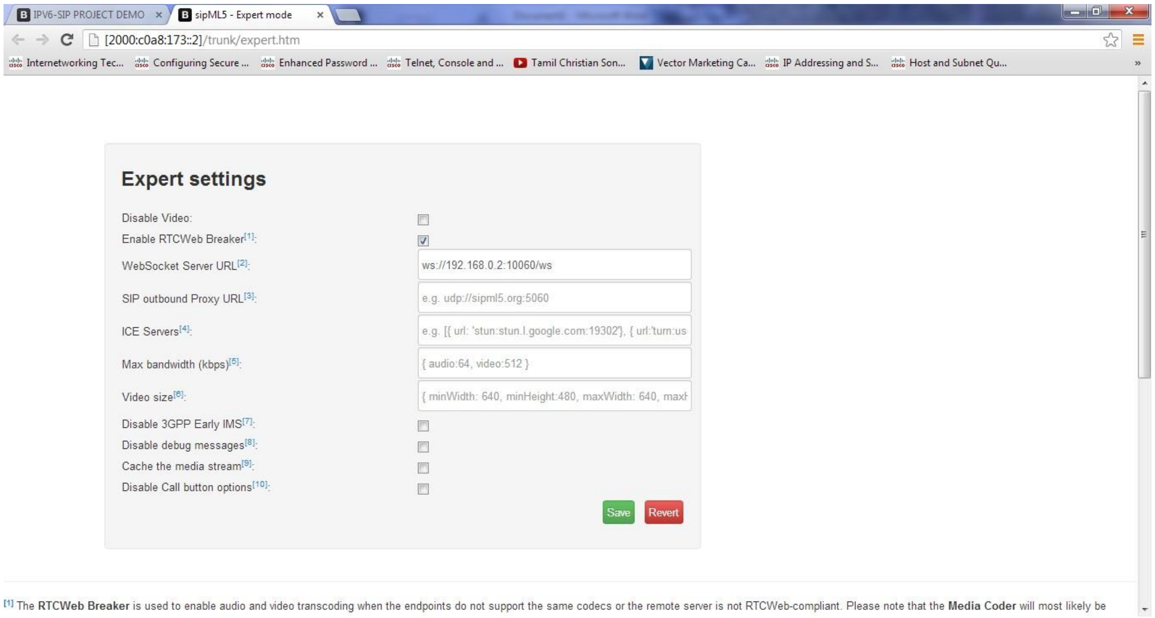
* *Signal Strength*

Video enabled

Call control

1060

❖ Expert Settings



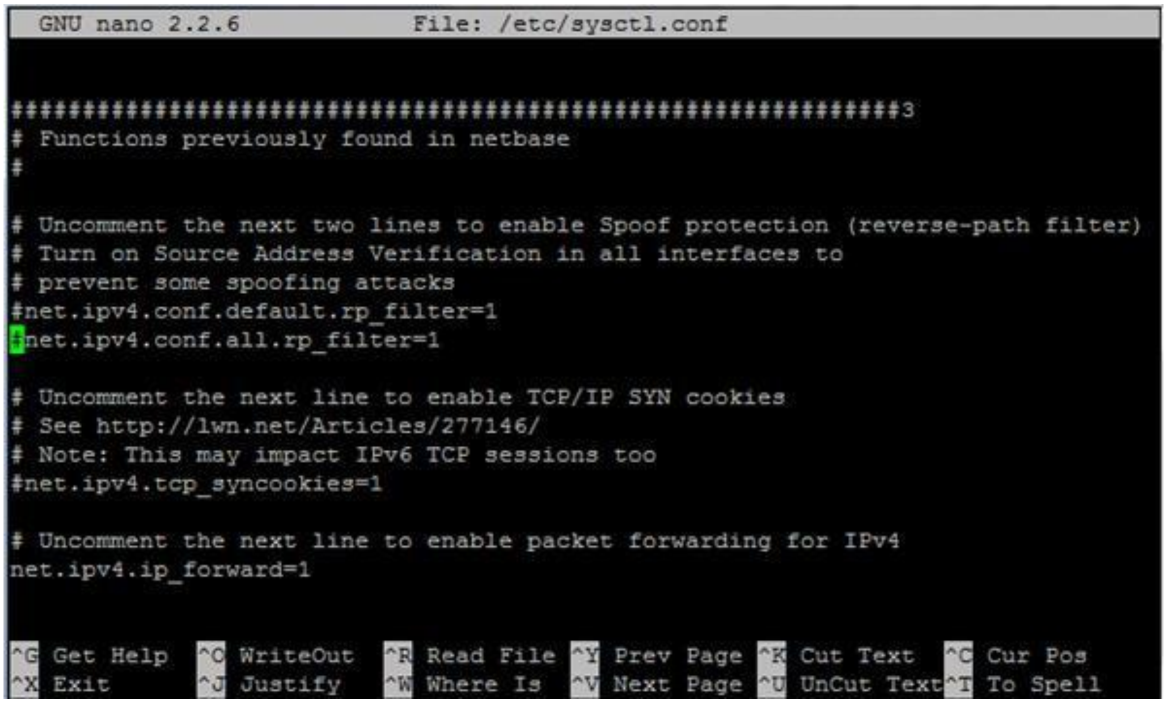
Expert settings

Disable Video:	□
Enable RTCWeb Breaker[1]:	☑
WebSocket Server URL[2]:	ws://192.168.0.2:10060/ws
SIP outbound Proxy URL[3]:	e.g. udp://sipml5.org:5060
ICE Servers[4]:	e.g. [{ url: 'stun:stun.l.google.com:19302'}, { url:'turn:us
Max bandwidth (kbps)[5]:	{ audio:64, video:512 }
Video size[6]:	{ minWidth: 640, minHeight:480, maxWidth: 640, maxh
Disable 3GPP Early IMS[7]:	□
Disable debug messages[8]:	□
Cache the media stream[9]:	□
Disable Call button options[10]:	□

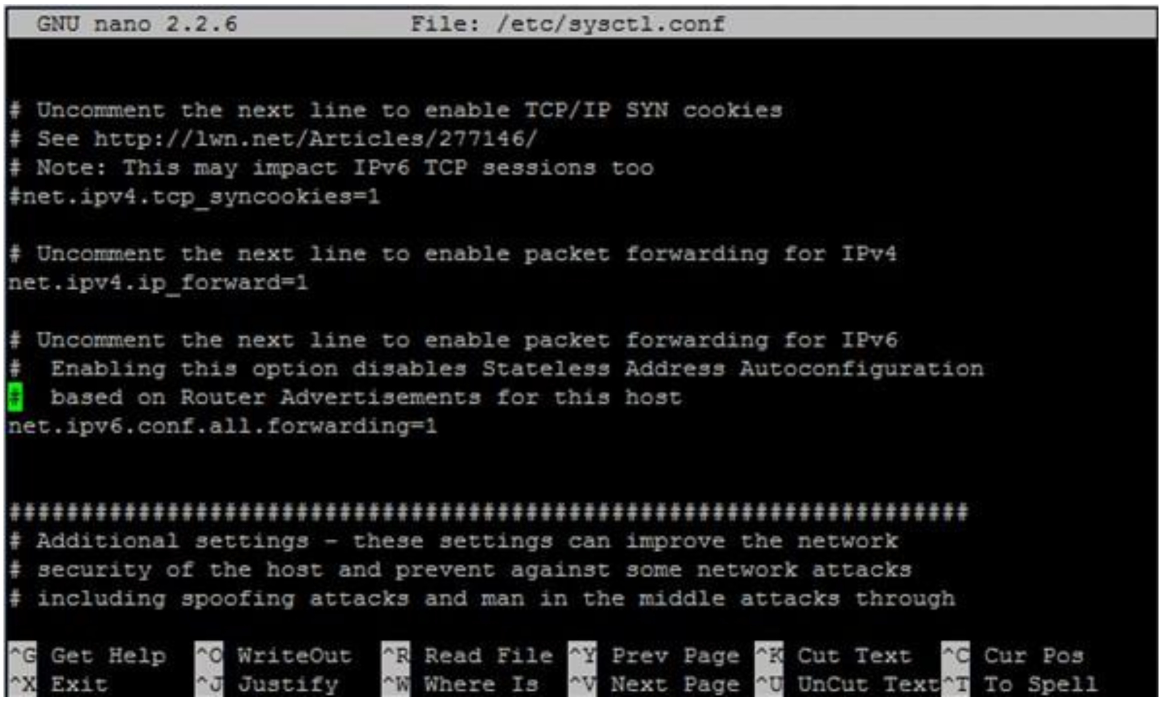
[1] The **RTCWeb Breaker** is used to enable audio and video transcoding when the endpoints do not support the same codecs or the remote server is not RTCWeb-compliant. Please note that the **Media Coder** will most likely be

8.4.2 SCREEN SHOTS

- **SYSCTL.CONF**

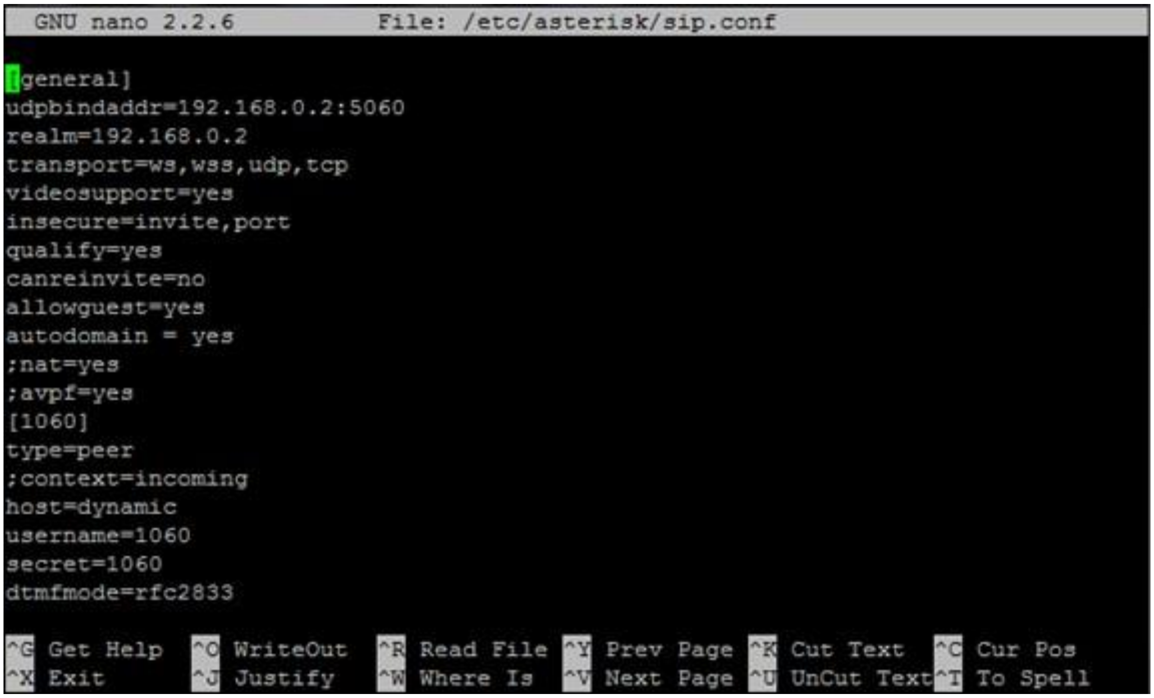

GNU nano 2.2.6 File: /etc/sysctl.conf
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```
##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

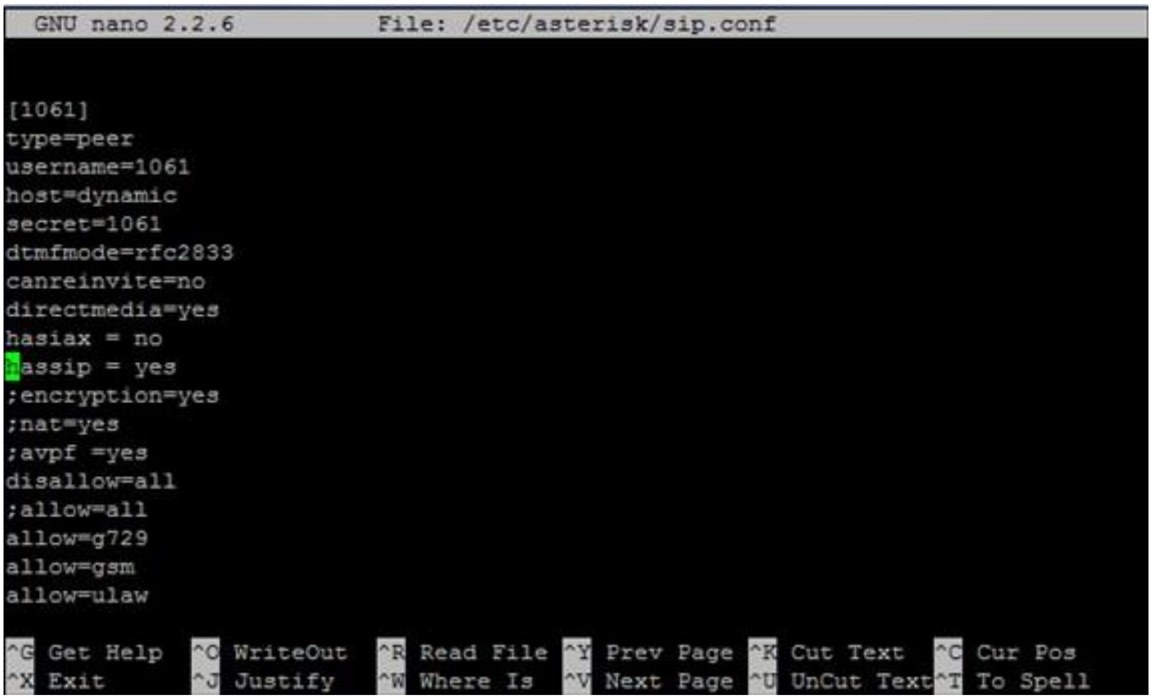

GNU nano 2.2.6 File: /etc/sysctl.conf
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
```

- **SIP.CONF**

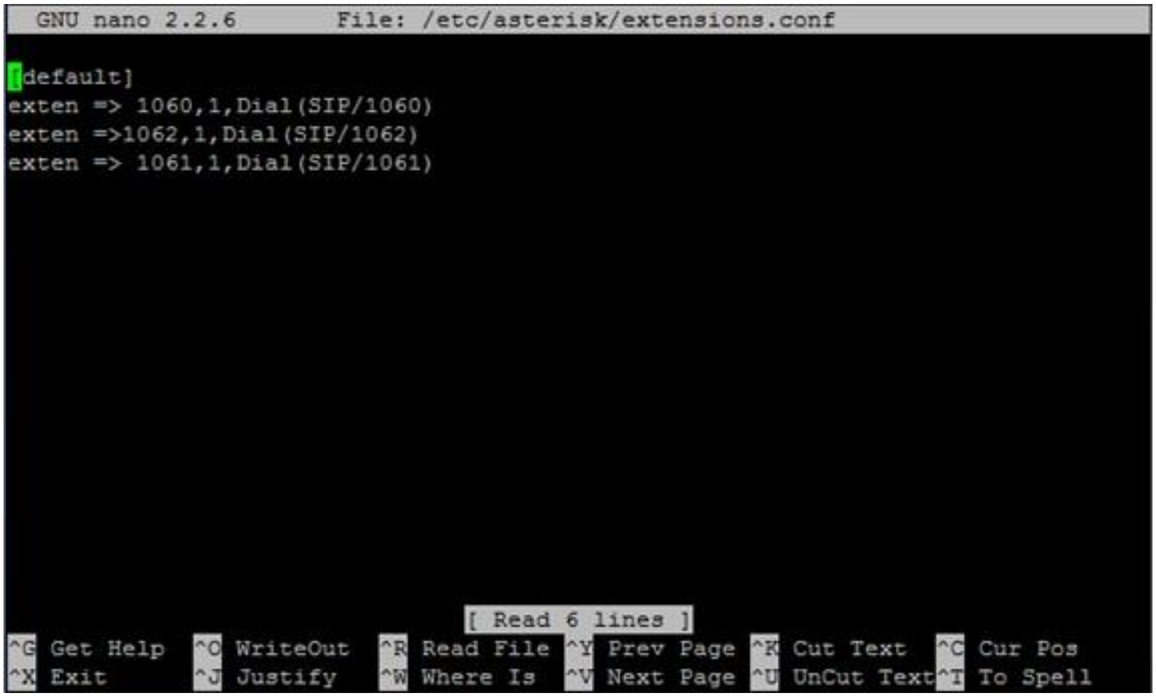


```
[general]
udpbindaddr=192.168.0.2:5060
realm=192.168.0.2
transport=ws,wss,udp,tcp
videosupport=yes
insecure=invite,port
qualify=yes
canreinvite=no
allowguest=yes
autodomain = yes
;nat=yes
;avpf=yes
[1060]
type=peer
;context=incoming
host=dynamic
username=1060
secret=1060
dtmfmode=rfc2833
```



```
[1061]
type=peer
username=1061
host=dynamic
secret=1061
dtmfmode=rfc2833
canreinvite=no
directmedia=yes
hasiax = no
hassip = yes
;encryption=yes
;nat=yes
;avpf =yes
disallow=all
;allow=all
allow=g729
allow=gsm
allow=ulaw
```

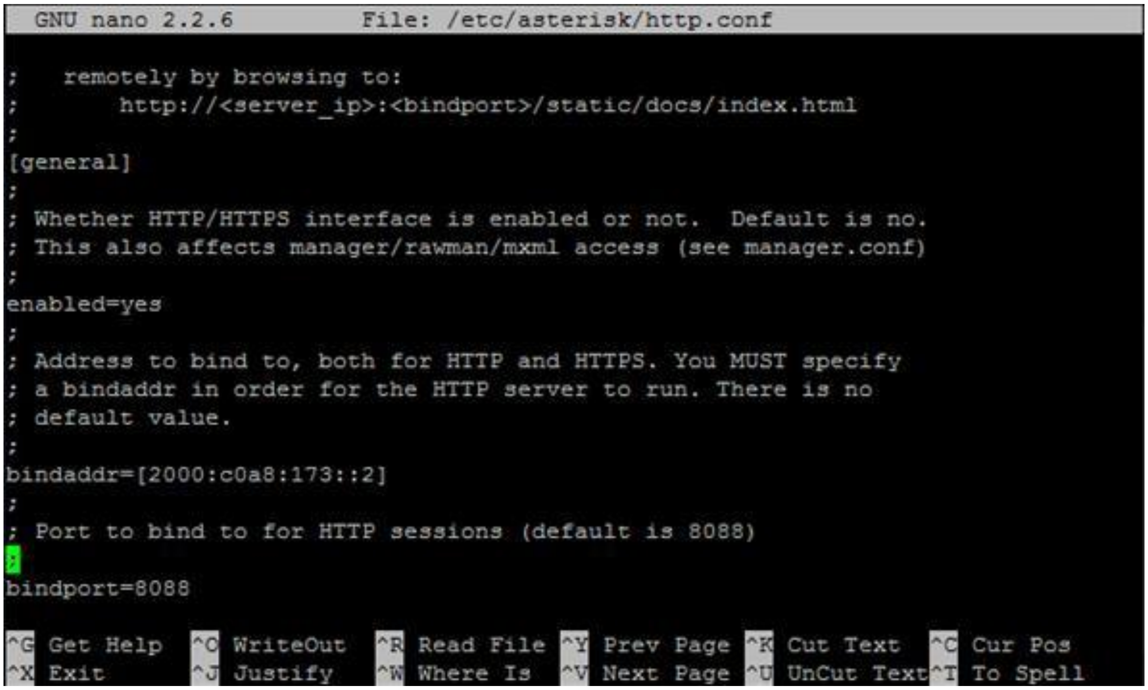
- **EXTENSIONS.CONF**



```
  GNU nano 2.2.6          File: /etc/asterisk/extensions.conf

[default]
exten => 1060,1,Dial(SIP/1060)
exten =>1062,1,Dial(SIP/1062)
exten => 1061,1,Dial(SIP/1061)
```

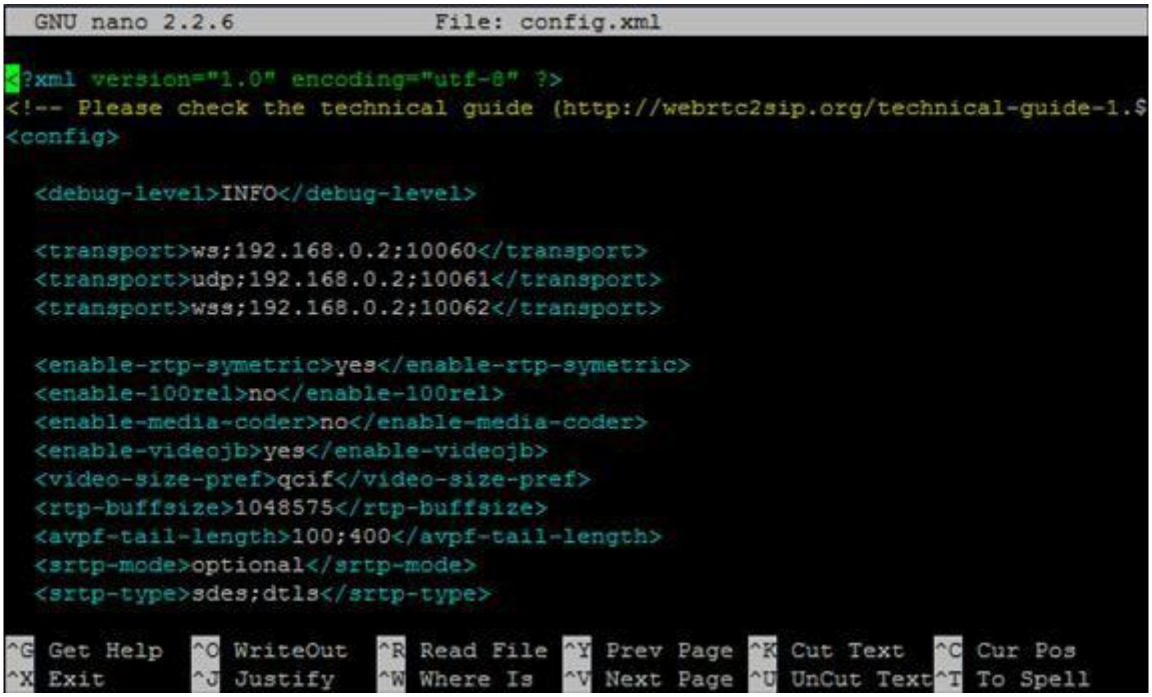
```
                               [ Read 6 lines ]
```

- **HTTP.CONF**

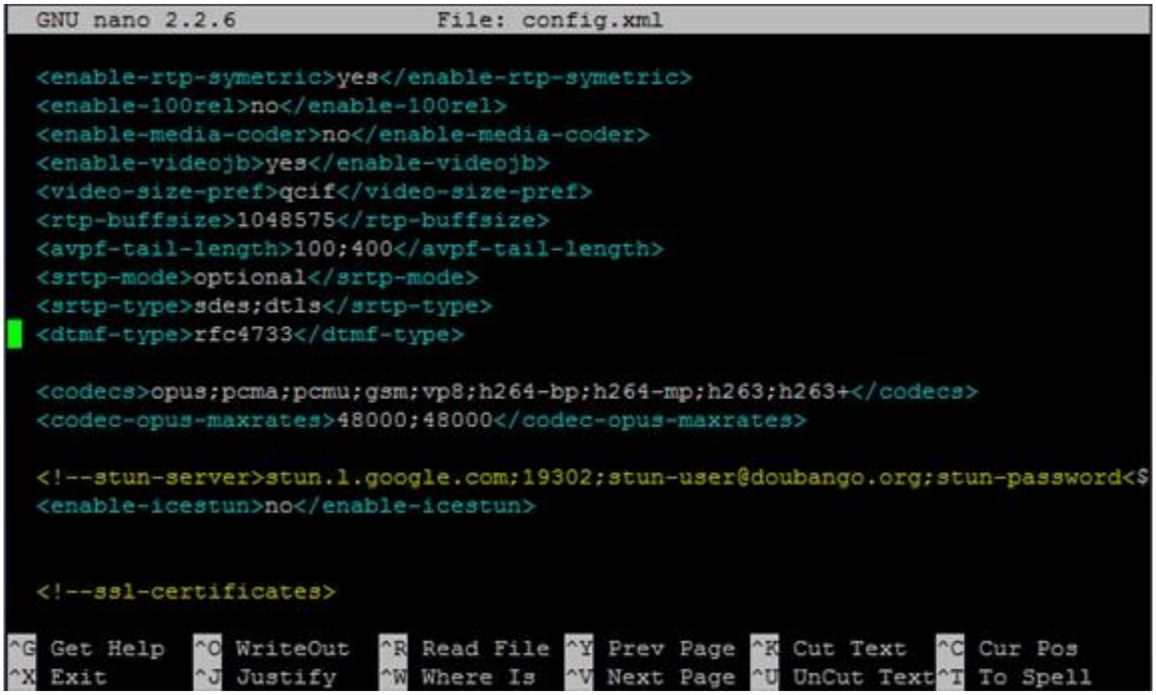


```
  GNU nano 2.2.6            File: /etc/asterisk/http.conf

;   remotely by browsing to:
;       http://<server_ip>:<bindport>/static/docs/index.html
;
[general]
;
; Whether HTTP/HTTPS interface is enabled or not.  Default is no.
; This also affects manager/rawman/mxml access (see manager.conf)
;
enabled=yes
;
; Address to bind to, both for HTTP and HTTPS. You MUST specify
; a bindaddr in order for the HTTP server to run. There is no
; default value.
;
bindaddr=[2000:c0a8:173::2]
;
; Port to bind to for HTTP sessions (default is 8088)
;
bindport=8088
```

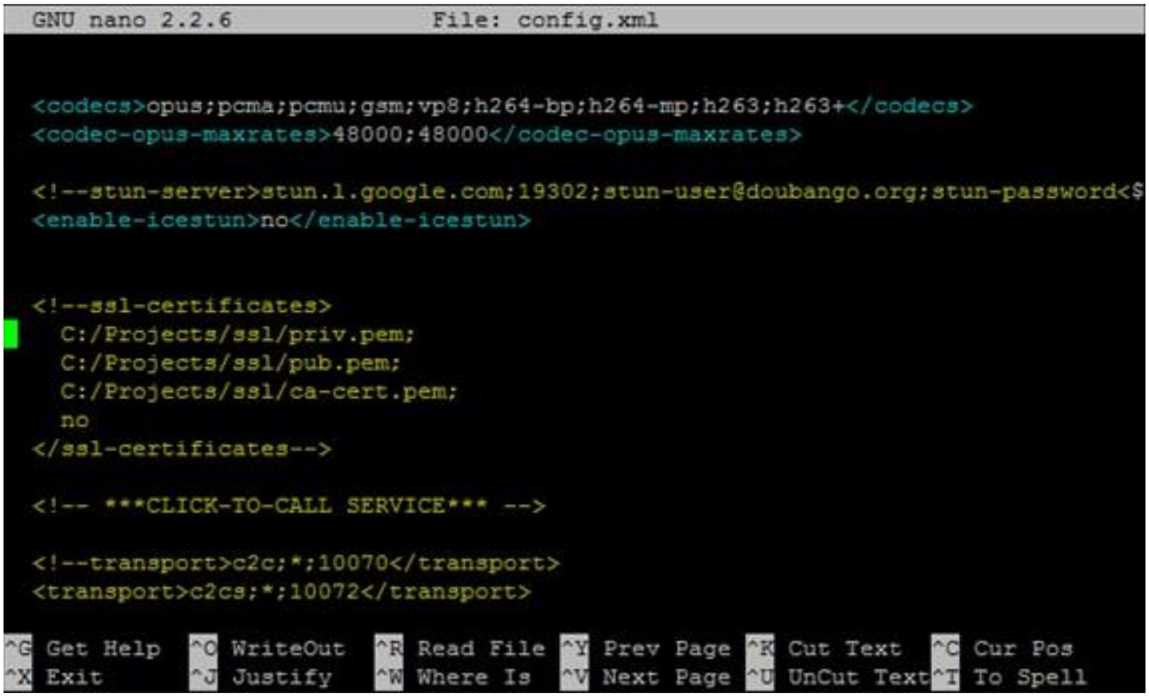
- **CONFIG.XML**



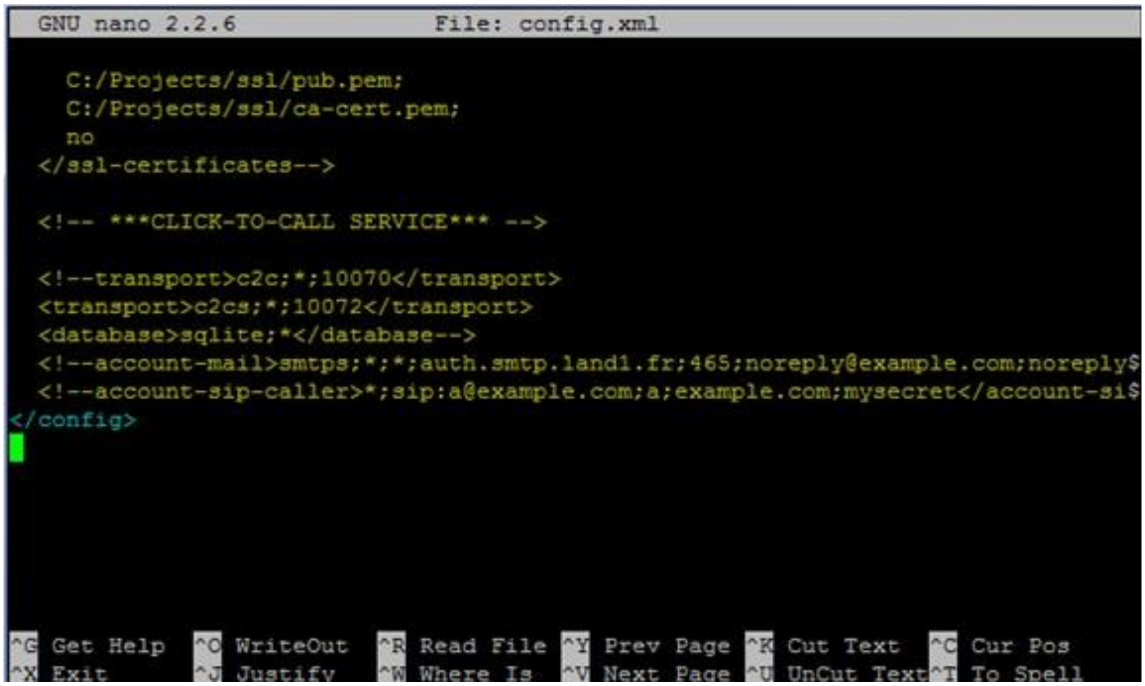
```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Please check the technical guide (http://webrtc2sip.org/technical-guide-1.$
<config>

  <debug-level>INFO</debug-level>

  <transport>ws;192.168.0.2;10060</transport>
  <transport>udp;192.168.0.2;10061</transport>
  <transport>wss;192.168.0.2;10062</transport>

  <enable-rtp-symetric>yes</enable-rtp-symetric>
  <enable-100rel>no</enable-100rel>
  <enable-media-coder>no</enable-media-coder>
  <enable-videojb>yes</enable-videojb>
  <video-size-pref>qcif</video-size-pref>
  <rtp-buffsize>1048575</rtp-buffsize>
  <avpf-tail-length>100;400</avpf-tail-length>
  <srtp-mode>optional</srtp-mode>
  <srtp-type>sdes;dtls</srtp-type>
```



```
  <enable-rtp-symetric>yes</enable-rtp-symetric>
  <enable-100rel>no</enable-100rel>
  <enable-media-coder>no</enable-media-coder>
  <enable-videojb>yes</enable-videojb>
  <video-size-pref>qcif</video-size-pref>
  <rtp-buffsize>1048575</rtp-buffsize>
  <avpf-tail-length>100;400</avpf-tail-length>
  <srtp-mode>optional</srtp-mode>
  <srtp-type>sdes;dtls</srtp-type>
  <dtmf-type>rfc4733</dtmf-type>

  <codecs>opus;pcma;pcmu;gsm;vp8;h264-bp;h264-mp;h263;h263+</codecs>
  <codec-opus-maxrates>48000;48000</codec-opus-maxrates>

  <!--stun-server>stun.l.google.com;19302;stun-user@doubango.org;stun-password<$
  <enable-icestun>no</enable-icestun>


  <!--ssl-certificates>
```



```
<codecs>opus;pcma;pcmu;gsm;vp8;h264-bp;h264-mp;h263;h263+</codecs>
<codec-opus-maxrates>48000;48000</codec-opus-maxrates>

<!--stun-server>stun.l.google.com;19302;stun-user@doubango.org;stun-password<$
<enable-icestun>no</enable-icestun>


<!--ssl-certificates>
  C:/Projects/ssl/priv.pem;
  C:/Projects/ssl/pub.pem;
  C:/Projects/ssl/ca-cert.pem;
  no
</ssl-certificates-->

<!-- ***CLICK-TO-CALL SERVICE*** -->

<!--transport>c2c;*;10070</transport>
<transport>c2cs;*;10072</transport>
```



```
  C:/Projects/ssl/pub.pem;
  C:/Projects/ssl/ca-cert.pem;
  no
</ssl-certificates-->

<!-- ***CLICK-TO-CALL SERVICE*** -->

<!--transport>c2c;*;10070</transport>
<transport>c2cs;*;10072</transport>
<database>sqlite;*</database-->
<!--account-mail>smtps;*;*;auth.smtp.land1.fr;465;noreply@example.com;noreply$
<!--account-sip-caller>*;sip:a@example.com;a;example.com;mysecret</account-si$
</config>
```

8.4.3 PACKAGES INSTALLED [4][5][6][7]

- **ASTERISK**

  Asterisk is used for creating communication applications. It has got all the modules to create a PBX system. The installation steps are given below,

  ```
  sudo apt-get install libncurses5-dev libnewt-dev libsqlite3-dev
  wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11-current.tar.gz
  tar zxvf asterisk-11-current.tar.gz
  cd asterisk-11.1.2
  ./configure
  make menuselect # optional step
  make -j `getconf _NPROCESSORS_ONLN`
  make install
  make config
  make samples
  ```

- **FFMPEG**

  FFMPEG consists of the leading audio/video codec library, libavcodec. It is used to record, convert and stream audio/video. The installation steps are given below,

  ```
  cd /usr/local/src
  wget -c http://ffmpeg.org/releases/ffmpeg-1.0.2.tar.gz
  tar zxvf ffmpeg-1.0.2.tar.gz
  cd ffmpeg
  ./configure --extra-cflags="-fPIC" --extra-ldflags="-lpthread" --enable-pic \
  --enable-memalign-hack --enable-shared --disable-static --disable-network \
  --disable-protocols --disable-pthreads --disable-devices --disable-filters \
  --disable-bsfs --disable-muxers --disable-demuxers --disable-parsers \
  --disable-hwaccels --disable-ffmpeg --disable-ffplay --disable-ffserver \
  --disable-encoders --disable-decoders --disable-zlib --enable-gpl --disable-debug \
  --enable-encoder=h263 --enable-encoder=h263p --enable-decoder=h263 \
  --enable-encoder=mpeg4 --enable-decoder=mpeg4 --enable-libx264 \
  --enable-encoder=libx264 --enable-decoder=h264
  make -j `getconf _NPROCESSORS_ONLN`
  make install
  ldconfig
  ```

- **DOUBANGO**

  Doubango is an experimental, open source, 3GPP IMS/LTE framework for both embedded and desktop systems. The installation steps are given below,

  ```
  cd /usr/local/src
  svn co http://doubango.googlecode.com/svn/branches/2.0/doubango doubango
  ```

```
cd doubango
sed -i '1,/==/s/==/=/' autogen.sh
./autogen.sh
./configure --with-ssl --with-srtp --with-vpx --with-speex --with-speexdsp \
--enable-speexresampler --enable-speexjb --enable-speexdenoiser --with-gsm \
--with-ffmpeg --with-h264 --prefix=/usr/local
make -j `getconf _NPROCESSORS_ONLN`
make install
ldconfig
```

- **WEBRTC2SIP**

  Webrtc2sip uses RTCweb and SIP to turn the browser into a smart phone to support audio, video and messaging capabilities. This allows the browser to make or receive calls from any SIP network or PSTN. The installation steps are given below,

```
cd /usr/local/src
svn co http://webrtc2sip.googlecode.com/svn/trunk/ webrtc2sip
cd webrtc2sip
sed -i '1,/==/s/==/=/' autogen.sh
./autogen.sh
./configure --with-doubango=/usr/local --prefix=/usr/local
make -j `getconf _NPROCESSORS_ONLN`
make install
mkdir -p /usr/local/etc/webrtc2sip
cp config.xml /usr/local/etc/webrtc2sip/
```

- **SIPML5**

  SIPML5 is an open source HTML5 SIP client. It requires absolutely nothing except Java Script to make it compatible with social networks. The media stack relies on Web RTC.

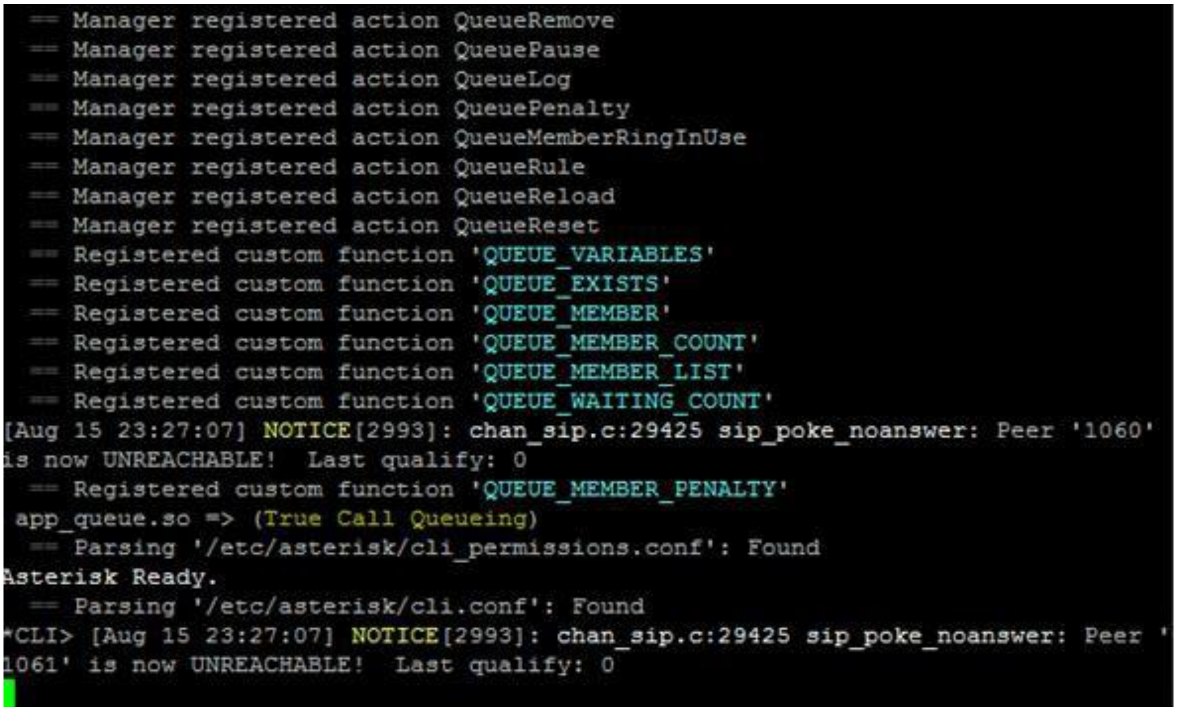
  Download and install the latest version of Google Chrome.

CHAPTER 9

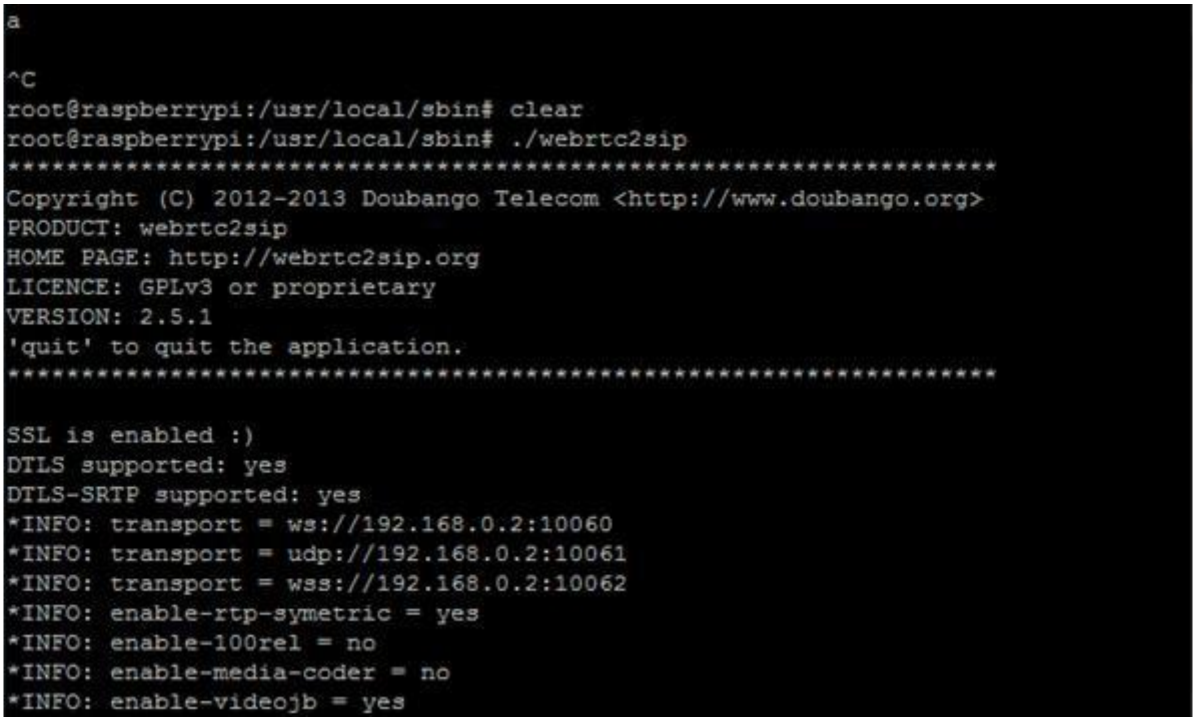
9. WORKING MODEL

9.1 PROCEDURE

- Open two putty sessions, one for Asterisk and the other for Web RTC. Run asterisk and webrtc with *asterisk –vvvvvvc* and *cd /usr/local/sbin* followed by *./webrtc2sip* respectively.



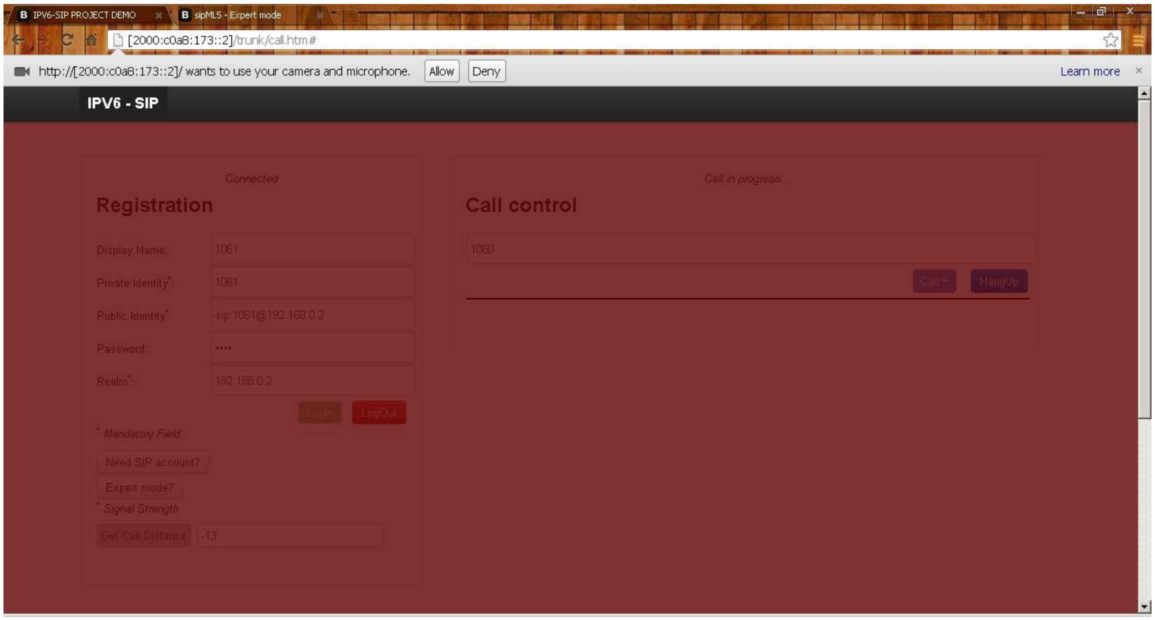
```
  == Manager registered action QueueRemove
  == Manager registered action QueuePause
  == Manager registered action QueueLog
  == Manager registered action QueuePenalty
  == Manager registered action QueueMemberRingInUse
  == Manager registered action QueueRule
  == Manager registered action QueueReload
  == Manager registered action QueueReset
  == Registered custom function 'QUEUE_VARIABLES'
  == Registered custom function 'QUEUE_EXISTS'
  == Registered custom function 'QUEUE_MEMBER'
  == Registered custom function 'QUEUE_MEMBER_COUNT'
  == Registered custom function 'QUEUE_MEMBER_LIST'
  == Registered custom function 'QUEUE_WAITING_COUNT'
[Aug 15 23:27:07] NOTICE[2993]: chan_sip.c:29425 sip_poke_noanswer: Peer '1060'
is now UNREACHABLE!  Last qualify: 0
  == Registered custom function 'QUEUE_MEMBER_PENALTY'
 app_queue.so => (True Call Queueing)
  == Parsing '/etc/asterisk/cli_permissions.conf': Found
Asterisk Ready.
  == Parsing '/etc/asterisk/cli.conf': Found
*CLI> [Aug 15 23:27:07] NOTICE[2993]: chan_sip.c:29425 sip_poke_noanswer: Peer '
1061' is now UNREACHABLE!  Last qualify: 0
```



```
a

^C
root@raspberrypi:/usr/local/sbin# clear
root@raspberrypi:/usr/local/sbin# ./webrtc2sip
*******************************************************************
Copyright (C) 2012-2013 Doubango Telecom <http://www.doubango.org>
PRODUCT: webrtc2sip
HOME PAGE: http://webrtc2sip.org
LICENCE: GPLv3 or proprietary
VERSION: 2.5.1
'quit' to quit the application.
*******************************************************************

SSL is enabled :)
DTLS supported: yes
DTLS-SRTP supported: yes
*INFO: transport = ws://192.168.0.2:10060
*INFO: transport = udp://192.168.0.2:10061
*INFO: transport = wss://192.168.0.2:10062
*INFO: enable-rtp-symetric = yes
*INFO: enable-100rel = no
*INFO: enable-media-coder = no
*INFO: enable-videojb = yes
```

- Now connect the SIP clients to the server and start calling.

❖ Call is initiated by client 1061



http://[2000:c0a8:173::2]/ wants to use your camera and microphone.

Connected

Registration

Display Name: 1061

Private Identity*: 1061

Public Identity*: sip:1061@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* *Mandatory Field*

* *Signal Strength*

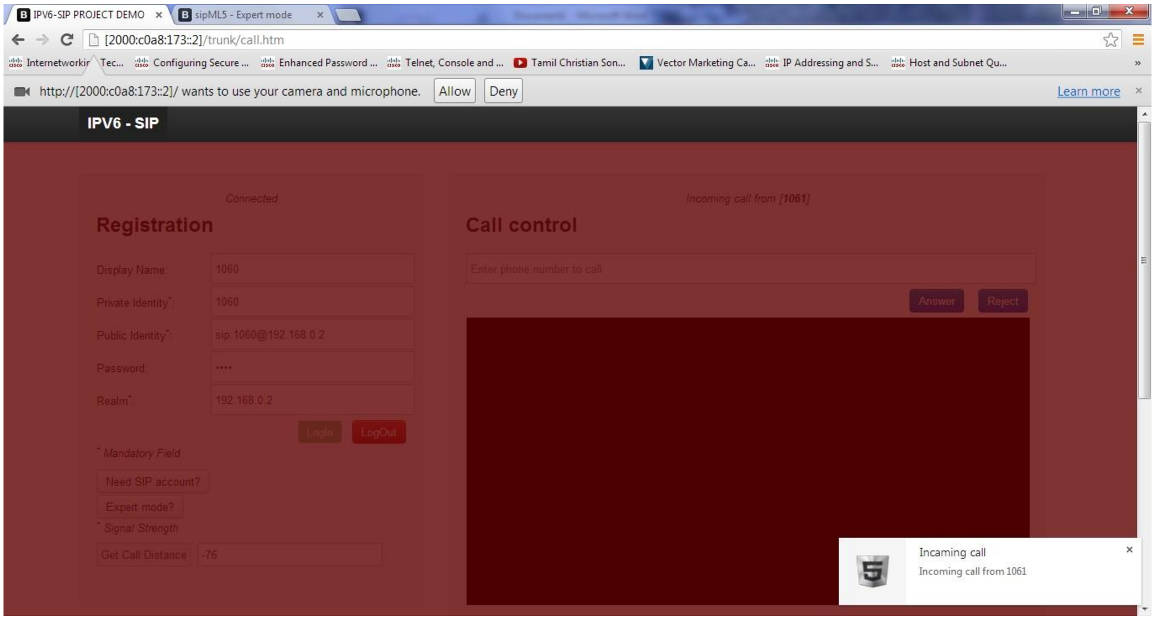
Get Call Distance: -13

Call in progress...

Call control

1060

- Call received by 1060



[2000:c0a8:173::2]/trunk/call.htm

http://[2000:c0a8:173::2]/ wants to use your camera and microphone.

IPV6 - SIP

Connected

Registration

Display Name: 1060

Private Identity*: 1060

Public Identity*: sip:1060@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* Mandatory Field

Need SIP account?

Expert mode?

* Signal Strength

Get Call Distance -76

Incoming call from [1061]

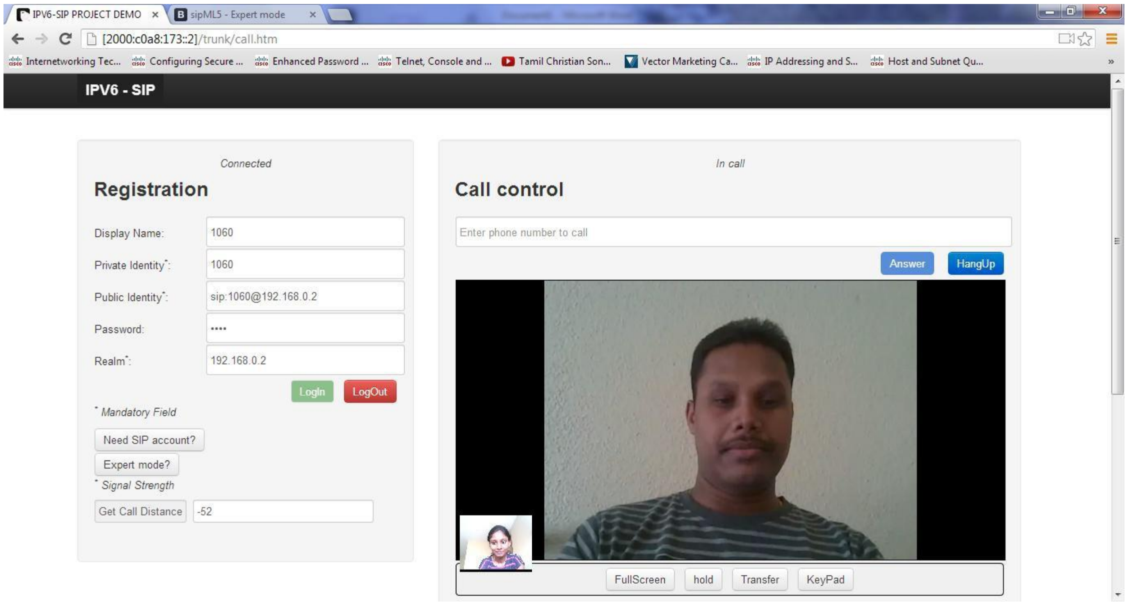
Call control

Enter phone number to call

Incoming call

Incoming call from 1061

- Video Capture on 1060



[2000:c0a8:173::2]/trunk/call.htm

IPV6 - SIP

Connected

Registration

Display Name: 1060

Private Identity*: 1060

Public Identity*: sip:1060@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* Mandatory Field

Need SIP account?

Expert mode?

* Signal Strength

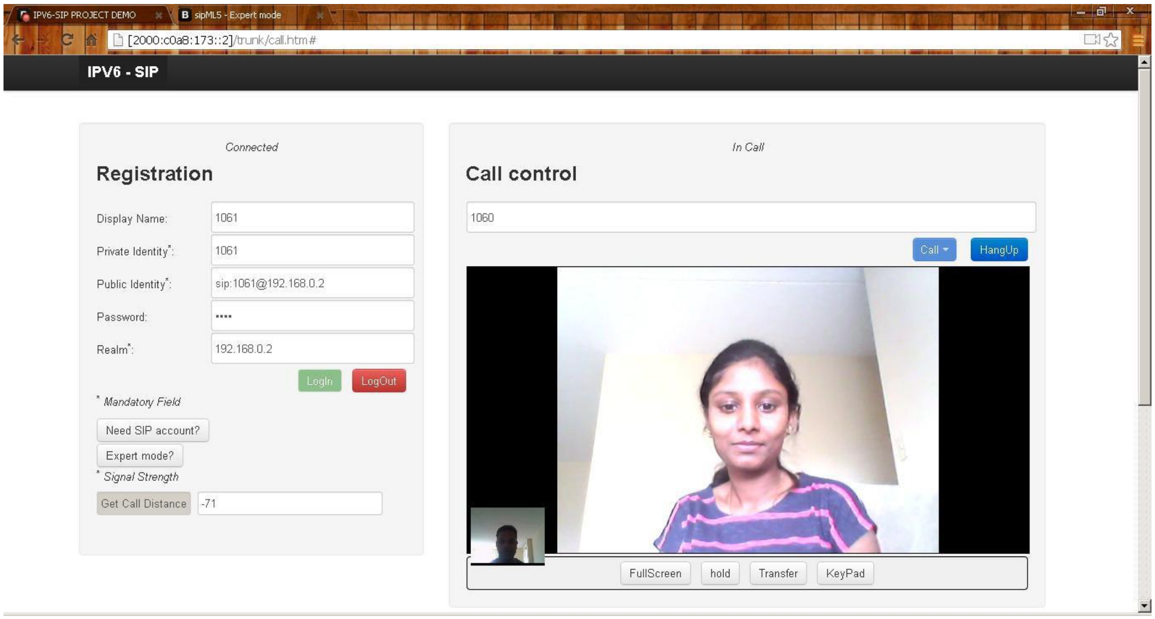
Get Call Distance -52

In call

Call control

Enter phone number to call

❖ Video capture on 1061



Connected

Registration

Display Name: 1061

Private Identity*: 1061

Public Identity*: sip:1061@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* *Mandatory Field*

* *Signal Strength*

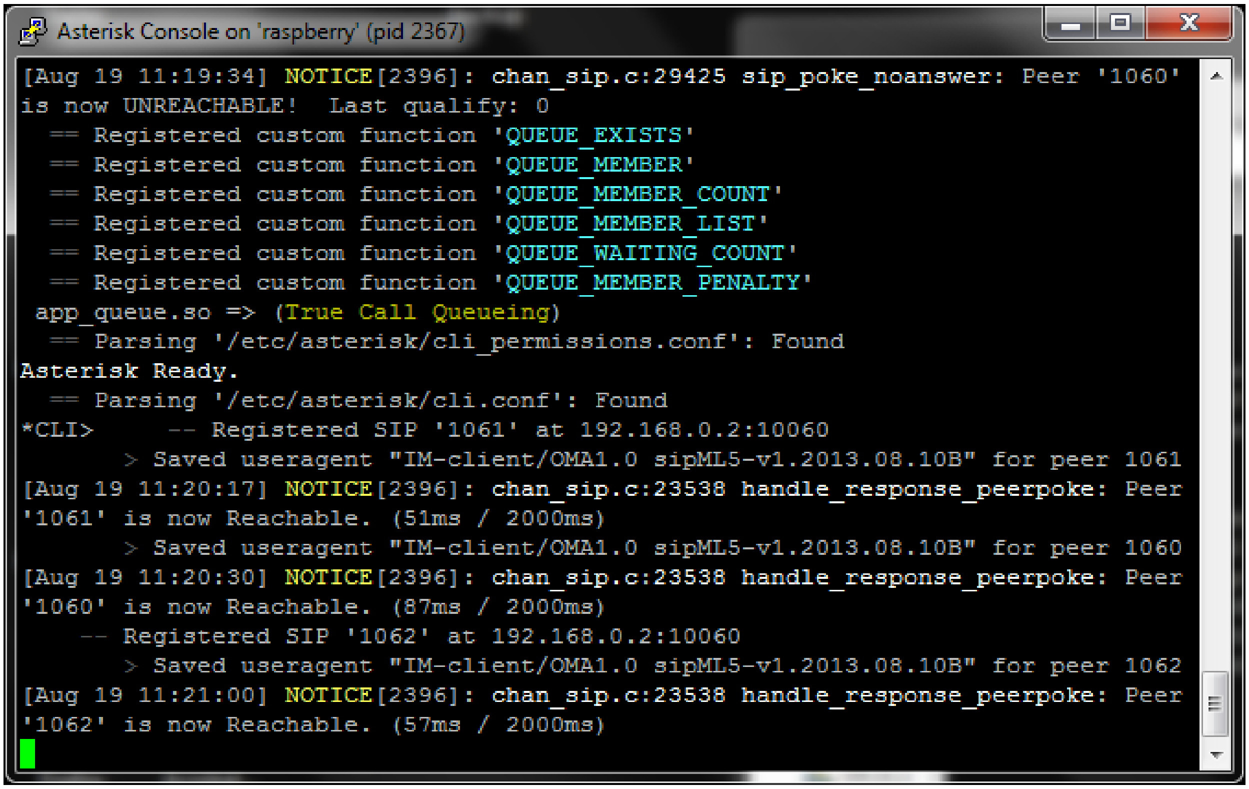
Get Call Distance: -71

In Call

Call control

1060

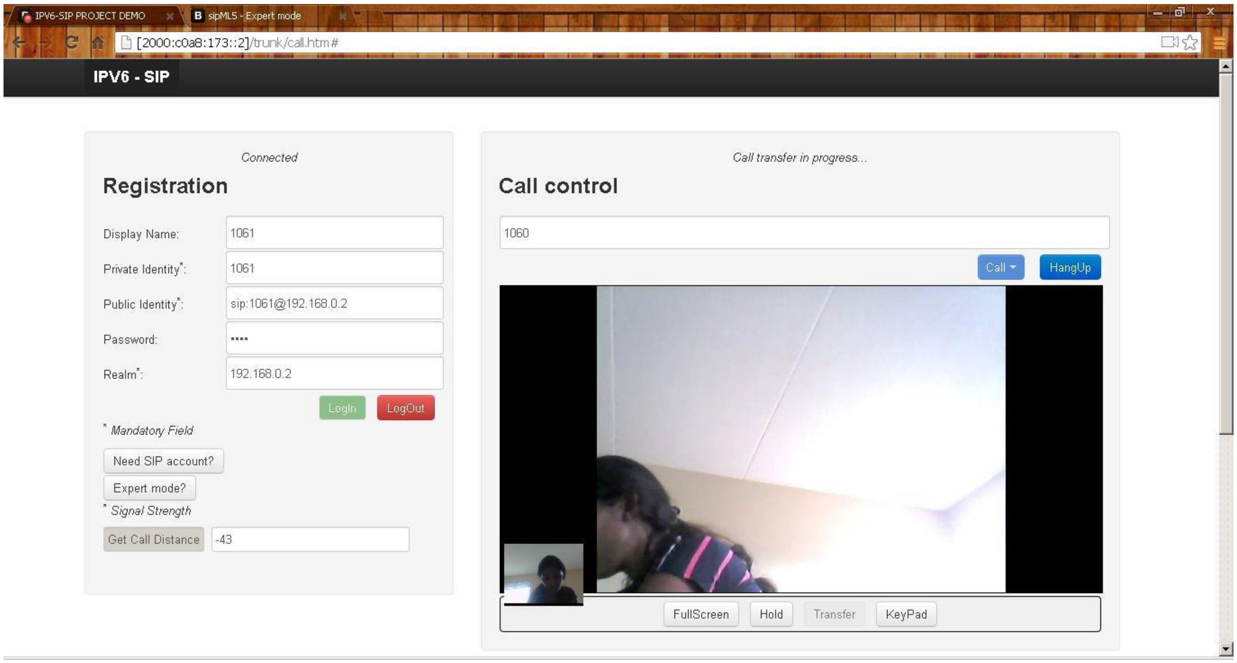
- **Transferring a call between clients**



Asterisk Console on 'raspberry' (pid 2367)

```
[Aug 19 11:19:34] NOTICE[2396]: chan_sip.c:29425 sip_poke_noanswer: Peer '1060'
is now UNREACHABLE!  Last qualify: 0
  == Registered custom function 'QUEUE_EXISTS'
  == Registered custom function 'QUEUE_MEMBER'
  == Registered custom function 'QUEUE_MEMBER_COUNT'
  == Registered custom function 'QUEUE_MEMBER_LIST'
  == Registered custom function 'QUEUE_WAITING_COUNT'
  == Registered custom function 'QUEUE_MEMBER_PENALTY'
 app_queue.so => (True Call Queueing)
  == Parsing '/etc/asterisk/cli_permissions.conf': Found
Asterisk Ready.
  == Parsing '/etc/asterisk/cli.conf': Found
*CLI>     -- Registered SIP '1061' at 192.168.0.2:10060
       > Saved useragent "IM-client/OMA1.0 sipML5-v1.2013.08.10B" for peer 1061
[Aug 19 11:20:17] NOTICE[2396]: chan_sip.c:23538 handle_response_peerpoke: Peer
'1061' is now Reachable. (51ms / 2000ms)
       > Saved useragent "IM-client/OMA1.0 sipML5-v1.2013.08.10B" for peer 1060
[Aug 19 11:20:30] NOTICE[2396]: chan_sip.c:23538 handle_response_peerpoke: Peer
'1060' is now Reachable. (87ms / 2000ms)
    -- Registered SIP '1062' at 192.168.0.2:10060
       > Saved useragent "IM-client/OMA1.0 sipML5-v1.2013.08.10B" for peer 1062
[Aug 19 11:21:00] NOTICE[2396]: chan_sip.c:23538 handle_response_peerpoke: Peer
'1062' is now Reachable. (57ms / 2000ms)
```

- ❖ Video Capture on 1061



[2000:c0a8:173::2]/trunk/call.htm#

Connected

Registration

Display Name: 1061

Private Identity*: 1061

Public Identity*: sip:1061@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* Mandatory Field

* Signal Strength

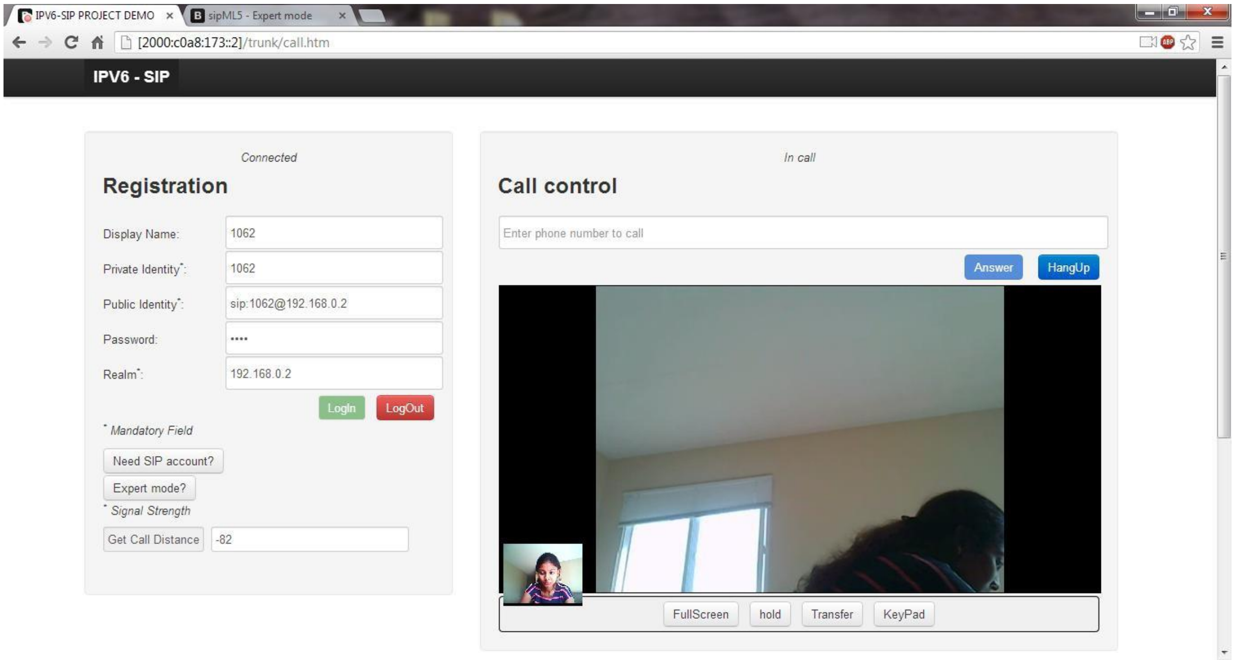
-43

Call transfer in progress...

Call control

1060

- ❖ Video Capture on 1062



[2000:c0a8:173::2]/trunk/call.htm

Connected

Registration

Display Name: 1062

Private Identity*: 1062

Public Identity*: sip:1062@192.168.0.2

Password: ••••

Realm*: 192.168.0.2

* Mandatory Field

* Signal Strength

-82

In call

Call control

Enter phone number to call

Note: During this capture there was no one to help me so I had to be the local and remote client.

CHAPTER 10

10. CONCLUSIONS & FUTURE WORK

10.1 CONCLUSION

This project demonstrates the performance of location based services and seamless transfer of SIP multimedia sessions in an IPv6 environment. It would be an ideal solution, where tracking of devices and making them communicate with each other would be of great importance. Making Raspberry Pi the location server, SIP server and web server is a cost-effective solution with maximum benefits. It also demonstrates the working of Web RTC for browsers which would be of great interest in the near future with several advantages and security features.

BENEFITS:

- Data traffic load on the network can be reduced.
- Multimedia communication between the clients could be enhanced.
- Over loading of the server with numerous requests can be avoided.
- Session transfers are done by the clients themselves upon server's instructions.
- Cost-effective.

APPLICATIONS:

- Can be used in moving devices.
- Childcare Applications.
- Patient Monitoring.
- Super malls/ Hyper malls.
- Indoor Organizations.

10.2 FUTURE WORK

Developments can be made in the future as more and more developers are indulged in developing open source codes. Once completely developed, this model will be modeled to meet universal standards and will be a breakthrough in SIP industry providing secure communication.

EXPERIENCES

This project gave me an opportunity to work on the world's cheapest mini PC. It was really exciting to work with Rpi and when connected to a LCD display it was just fantastic. The way it works would definitely entice many children to learn programming and to use LINUX OS. I would strongly recommend Rpi for educational purposes in schools. It's got games in it, which is another attraction for children. It would an ideal solution for anyone who would like to experiment new projects.

I also had a chance to configure IPv6 network. I was able to figure out how IPv6 addresses work and how it can be made backwards compatible with IPv4 (Tunneling process). Availability of unlimited address spaces is one of its best qualities in the current world where we are running out of IPv4 addresses. Its security features and extensions give an opportunity for the user to configure it as per the needs and it has got more advantages over IPv4. I was able to test the dual stack concept and see how IPv4 and IPv6 networks communicate with each other. Theoretical knowledge about IPv6 helped in the practical implementation in my project and also gave me a clear idea of what I am doing. In years together IPv6 will be a wide spreading technology with lot of advantages over IPv4

MINT 708 gave me a great exposure to networking concepts because of which I was able to do the networking, especially in configuring IPv6 network in my project. MINT 719 (MPLS & VOIP) helped me in understanding the concepts of VOIP and SIP. These labs were an inspiration for my project. Previous experience working on elastix server in MINT 719 was an added value and encouraged me to experiment asterisk server. The knowledge that I had gained through these labs helped me in completing my project successfully.

Web RTC is an exciting technology that is being recently talked about in the world. This could be one of the best ways to provide interactive real-time communication to the people in the near future. Its best quality is that it needs absolutely nothing (no installation, plug-ins, add-ons, and user account) except the Java Script (to make it compatible with other social networking apps) at the back end. Since it is fresh in the market I had opportunities to explore new things about it. I am sure it will be an amazing technology when people start using it because of it user friendly nature.

REFERENCES

CHAPTER 1:

[1] IEEE 978-1-4577-0005-7/11 Seamless SIP Multimedia Session Transfer on IPv6 Network via Device Switching. Telematic Research Group (TRG), UTM SPACE and MIMOS Berhad, Malaysia.

CHAPTER 2

[1]http://www.element14.com/community/docs/DOC-42993/l/raspberry-pi-single-board-computer#anchor3
[2] http://www.myraspberry-pi.org/specifications/
[3]http://www.techradar.com/news/software/operating-systems/raspberry-pi-operating-systems-5-reviewed-and-rated-1147941
[4]http://www.smsc.com/Products/USB/USB_to_Ethernet_Controllers/USB_to_Ethernet_Controller_Hub_Combos/LAN951x/Description
http://www.raspbian.org/
http://elinux.org/RPi_Hardware
http://www.raspberrypi.org/faqs

CHAPTER 3

[1]http://www.html5rocks.com/en/tutorials/webrtc/basics/
[2]http://www.webrtc.org/reference/architecture
[3]http://www.youtube.com/watch?v=p2HzZkd2A40
[4]http://www.youtube.com/watch?v=Yf3eNciKddc
http://code.google.com/p/sipml5/

CHAPTER 4

RFC 2460, “Internet Protocol, Version 6 Specification”, December 1998
RFC 2373, “IP Version 6 Addressing Architecture”, July 1998
Robert M. Hinden, “IP Next Generation Overview”, May 14, 1995
James F. Kurose, “Computer Networking: A Top-Down Approach Featuring the Internet”, 2001 (ISBN 0-201-47711-4)
[1]http://publib.boulder.ibm.com/infocenter/zvm/v5r4/index.jsp?topic=/com.ibm.zvm.v54.kijl0/hcsk7b3014.htm

CHAPTER 5

[1]Deliverable D5.7 Report on Integration of SIP and IPv6, IST-2001-32603
[2]http://www.siptutorial.net/SIP/index.html
http://audio.icann.org/icann-start-05-ipv6-20100429-en.mp3

CHAPTER 6

[1]http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/index.html
[2]http://www.asterisk.org/get-started
[3]http://ofps.oreilly.com/titles/9781449332426/asterisk-Arch.html

CHAPTER 7

[1]http://www.speex.org/
[2]Analysis of packet loss and delay variation on QoE for H.264 and WebM/VP8 Codecs-Yeshwanth Alahari & Buddhiraja Prashant, Blekinge Institute of Technology, Sweden
[3]Implementation, Performance Analysis & Comparison of VP8 by Keyur Shah, University of Texas.
http://www.itu.int/net/itu-t/sigdb/speaudio/Gseries.htm#
http://www.radio-electronics.com/info/cellulartelecomms/gsm_technical/audio-codecs-vocoders-amr-celp.php

CHAPTER 8

[1] IEEE 978-1-4577-0005-7/11 Seamless SIP Multimedia Session Transfer on IPv6 Network via Device Switching. Telematic Research Group (TRG), UTM SPACE and MIMOS Berhad, Malaysia.
[2] Performance Analysis Indoor Location Tracking Framework with SIP on IPv6 Island. Telematic Research Group (TRG), UTM SPACE and MIMOS Berhad, Malaysia.
[3] Indoor Seamless Roaming for VoIP Using IPv6 Location Assisted Network. Management and Science University, UTM MIMOS Center of Excellence, Universiti Teknologi Malaysia and MIMOS Berhad, Malaysia.
[4] http://webrtc2sip.org/technical-guide-1.0.pdf
[5] http://linux.autostatic.com/asterisk-and-sipml5-interoperability
[6] http://linux.autostatic.com/installing-webrtc2sip-on-ubuntu-1204
[7] http://sipml5.googlecode.com/svn/trunk/