

AALTO UNIVERSITY SCHOOL OF ELECTRICAL ENGINEERING
Department of Communications and Networking

Kapil Maheshwari

Core Network Design of Software Defined Radio Testbed

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, August 22nd, 2013

Supervisor: Docent Pasi Lassila

Instructor: M.Sc. (Tech) Mika Nupponen



AALTO UNIVERSITY SCHOOL OF ELECTRICAL ENGINEERING
Department of Communications and Networking
ABSTRACT OF THE MASTER'S THESIS

Author: Kapil Maheshwari	
Subject of the thesis: Core Network Design of Software Defined Radio Test bed	
Number of pages: 11+91	Date: 22.08.2013
Professorship: Networking Technology	Code of professorship: S-38
Supervisor: Docent Pasi Lassila	
Instructor: M.Sc. (Tech.) Mika Nupponen	
<p>The 4th generation of cellular system (LTE) does not inherit the traditional voice (circuit-switched) capabilities from its predecessors. Instead it relies on its high speed packet-switched core network with IMS (IP Multimedia Subsystem) for voice capabilities. Even though there are temporary solutions available until LTE gets its full deployment and coverage, operators are looking for a long term solution known as VoIMS which uses VoIP with SIP protocol for voice in the LTE network (VoLTE) through the IMS domain.</p> <p>The scope of this thesis work is to design, implement and verify the working of the core network for an LTE type software defined radio (SDR) testbed which is able to initiate, maintain and terminate voice and data connections. First step in this regard is to search and select the tools, programs and technologies that fulfil the network requirement in terms of network performance and user satisfaction. Next is to build, configure and verify the network operations of the designed network.</p> <p>As SDRs are used for testing purposes, the core network is also designed in correspondence to that, i.e., it is a test (lab) core network with configurations that are simple to implement and do not require coding implementation. The core network makes use of the virtualization technology and is realized with the help of open-source solutions, i.e., protocols and technologies that are customizable as required and does not require licensing for their use. These functionalities are implemented with the help of OpenSIPS, an open-source SIP server, DHCP and DNS servers.</p> <p>Demonstration of the core network verifies that successful voice and video call can be made between registered users on two different networks, running VoIP client software on different operating system platforms. The core network provides features such as voice, video, instant messaging, presence, dynamic IP assignment, IP address to name resolution and mobility.</p>	
Keywords: Core Network, DHCP, DNS, OpenSIPS Configuration, Open-source, VoIP, VoIMS, VoLTE.	Publishing language: English

Acknowledgement

First and foremost, I would like to thank almighty and then start by expressing my sincere gratitude to my supervisor Pasi Lassila for his directions, support and patience in this thesis work. I am very much great full to my instructor Mika Nupponen, it was his guidance, suggestions and motivation that made this thesis work possible.

I would like to present my deepest gratitude to my parents Mr. and Mrs. Nathu Ram and all the family members for their enduring love and encouragement. Their prayers, blessings and wisdom have guided me on every step of the way. I would like to extend my thanks to a very special friend and my fiancée Komal Karmani for her everlasting support all the way.

I would also like to seize this opportunity to thank my colleagues and friends in and out of the Aalto University, one in particular, Mr. Yeswanth Kumar for his throughout help and motivation.

Espoo, August 22nd, 2013

Kapil Maheshwari

Table of Contents

Acknowledgement	iii
Table of Contents	iv
List of Figures.....	vi
List of Tables	vii
Abbreviations and Terms.....	viii
1 Introduction	1
2 Call Control in Cellular Systems.....	3
2.1 Introduction to 3G - UMTS	3
2.2 Introduction to 4G – LTE	7
2.3 Comparison between 3G and 4G systems	10
2.4 Voice in 3G.....	11
2.5 Voice in LTE (VoLTE)	12
2.6 Summary	18
3 Test Network Technologies.....	19
3.1 What is VoIP	19
3.2 Virtualization	31
3.3 IP Addressing Scheme	35
3.4 Virtual LAN.....	38
3.5 IP Address Allocation.....	39
3.6 Domain Name System	41
3.7 Routing in IP networks	44
3.8 Summary	47
4 Test Network Planning and Configuration	48
4.1 Test Network Design	48
4.2 VirtualBox Settings	50
4.3 IP Addressing Scheme	50
4.4 Routing Scheme for Test Network	53
4.5 Configuring DHCP Server.....	53
4.6 Configuring DNS Server	55
4.7 VLAN Configuration.....	57
4.8 Configuring SIP Server	57
4.9 Ekiga Installation.....	59
4.10 Testing of Lab-based Core Network.....	59
4.11 Test Network Issues.....	62

5 Conclusion.....	64
References.....	66
Appendix A.....	71
Appendix B.....	75
Appendix C.....	78
Appendix D.....	80

List of Figures

Figure 2-1: UMTS Architecture.....	4
Figure 2-2: Release 99 Architecture of the UMTS	5
Figure 2-3: Release 4 Architecture of the UMTS	6
Figure 2-4: Release 5 Architecture of the UMTS	6
Figure 2-5: Core Network Components in All IP Structure of UMTS	7
Figure 2-6: LTE Network Architecture	8
Figure 2-7: Interfaces Between eNBs, E-UTRAN and EPC.	9
Figure 2-8: Call Connection Flow In UMTS Circuit-Switched Domain.....	12
Figure 2-9: CSFB Architecture.....	15
Figure 2-10: CSFB Mobile Terminating Call Procedure.....	15
Figure 2-11: Configuration of VoLTE Network.....	16
Figure 2-12: VoLTE Mobile Call Origination Process.....	17
Figure 3-1: Illustration of VoIP Concept.....	19
Figure 3-2: H.323 Network Components.....	20
Figure 3-3: H.323 Protocol Scope	20
Figure 3-4: SIP Network Entities.....	23
Figure 3-5: SIP Request and Response Messages During Call Flow	26
Figure 3-6: Bare metal and Hosted Virtualization Technologies	32
Figure 3-7: Ports Separated Into VLANs In Switch	39
Figure 3-8: DHCP Messages Sent Between Server and Client	41
Figure 3-9: DNS Hierarchical Structure	43
Figure 4-1: Test Network Topology	49
Figure 4-2: Network Adapter Setting For BS1	51
Figure 4-3: Sample Interface Configuration of BS-1	52
Figure 4-4: Sample Static Route Configuration of BS-1	53
Figure 4-5: Sample DHCP Server Configuration of BS-1.....	54
Figure 4-6: Sample DNS Configuration of BS-1.....	56
Figure 4-7: Sample Zone File Configuration of BS-1	56
Figure 4-8: HP Switch VLAN Configuration.....	57
Figure 4-9: OpenSIPS Main Configuration Menu.....	58
Figure 4-10: OpenSIPS-CP Login Screen	59
Figure 4-11: Registration of User2 on VoIP Client Ekiga on Windows Platform.....	60
Figure 4-12: Registration of User3 on Linux Platform.....	61
Figure 4-13: Call Initiation and Progress Between User2 and User3	61
Figure 4-14: A Video Call between User2 and User3	62

List of Tables

Table 3-1: SIP Supporting Protocols.	22
Table 3-2: IP Address Classes	36
Table 3-3: IP Address Blocks For IP Classes	36
Table 3-4: IP Prefix Notation for Private IP Addresses Block	37
Table 4-1: NIC card configurations on VirtualBox	50

Abbreviations and Terms

1G	First Generation
2G	Second Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
ABR	Area Border Router
ACK	Acknowledgement
AM	Amplitude Modulation
AS	Application Server
AUC	Authentication Center
B2BUA	Back-to-Back User Agent
BDR	Backup Designated Router
BIND	Berkeley Internet Name Daemon
BOOTP	Bootstrap Protocol
BS	Base-station
BSC	Base Station Controller
ccTLD	Country Code Top-Level Domains
CDMA	Code Division Multiple Access
CDR	Call Detail Records
CIDR	Classless Inter Domain Routing
CM	Connection Management
CN	Core Network
CR	Cognitive Radio
CS	Circuit Switch
CSCF	Call Session Control Function
CSFB	Circuit-Switched Fallback
DHCP	Dynamic Host Control Protocol
DNS	Domain Name Service
DR	Designated Router
DV	Distance Vector
EDGE	Enhanced Data-Rate for GSM Evolution
EECRT	End-to-End Cognitive Radio Testbed
EIR	Equipment Identity Register
eNB	Evolved Node B
ENUM	E.164 Number Mapping
EPC	Evolved Packet Core
EUTRAN	Evolved UTRAN
FM	Frequency Modulation

GGSN	Gateway GPRS Support Node
GPL	General Public License
GSM	Global System for Mobile Communication
GSMA	Global System for Mobile Communication Association
gTLD	Generic Top-Level Domain
GUI	Graphic User Interface
HD	High Definition
HDD	Hard Disk Drive
HLR	Home Location Register
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
I-CSCF	Interrogating Call Session Control Function
IEEE	Institute of Electrical and Electronics Engineers
IEFT	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPv4	Internet Protocol version 4
ITU	International Telecommunication Union
IVR	Interactive Voice Recording
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LSA	Link State Advertisement
LTE	Long Term Evolution
MGCF	Media Gateway Control Function
MGCP/MEGACO	Media Gateway Control Protocol
MGW	Media Gateway
MM	Mobility Management
MME	Mobility Management Entity
MSC	Master Switching Center
MX	Mail Exchange Record
NAPTR	Name Authority Pointer
NAT	Network Address Translation
NIC	Network Interface Card
NNI	Network-Network Interface
NS	Name Server
NSS	Network Switching Subsystem
OFDMA	Orthogonal Frequency Division Multiple Access
OS	Operating System
OSPF	Open Shortest Path First

PBX	Private Branch Exchange
PC	Personal Computer
PCEF	Policy and Charging Enforcement Function
PCRF	Policy Control and Charging and Rule Function
P-CSCF	Proxy – Call Session Control Function
PDN	Packet Data Network
P-GW	Packet Data Network Gateway
PS	Packet Switch
PSTN	Public Switched Telephone Network
PTR	Pointer Record
QEMU	Quick Emulator
QoS	Quality of Service
RADIUS	Remote Access Dial In User Service
RAN	Radio Access Network
RF	Radio Frequency
RFC	Request For Comments
RIP	Routing Information Protocol
RNC	Radio Network Controller
R-SGW	Roaming – Signaling Gateway
RSVP	Resource Reservation Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SAE	System Architecture Evolution
SC-FDMA	Single-Carrier Frequency Division Multiple Access
S-CSCF	Serving – Call Session Control Function
SDP	Session Description Protocol
SDR	Software Defined Radio
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
SIP	Session Initiation Protocol
SLD	Second Level Domain
SOA	Start of Authority
SPF	Shortest Path First
SQL	Structured Query Language
SRV	Service Record
SRVCC	Single Radio Voice Call Continuity
SVLTE	Simultaneous Voice over LTE
TCP	Transmission Control Protocol
TLD	Top Level Domains
TLS	Transport Layer Security
T-SGW	Transport Signaling Gateway

UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UNI	User-Network Interface
UTRAN	UMTS Terrestrial Radio Access Network
VLAN	Virtual Local Area Network
VLR	Visitor Location Register
VM	Virtual Machine
VMM	Virtual Machine Monitor
VoIMS	Voice over IP Multimedia Subsystem
VoIP	Voice over Internet Protocol
VoLGA	Voice over LTE via Generic Access
VoLTE	Voice over Long Term Evolution
W-CDMA	Wideband Code Division Multiple Access

1 Introduction

Like public switched telephone network (PSTN), cellular telephony has used the circuit-switched technology for voice communication but this trend has changed with the advent of broadband Internet connections. Users with high-speed mobile Internet connections frequently use VoIP (Voice over Internet Protocol) applications for voice communication rather than traditional circuit-switched voice call. Such changes have also been reflected in the network design and infrastructure of cellular networks. Among the generations of cellular communication 1G, 2G, 2.5G and 3G have relied on the circuit-switched voice. On the other hand, 4th generation of cellular communication is all set to utilize the services of packet-switched domain for voice, i.e., it has an all IP based core network.

Cellular network operators are warming up to the idea of implementing LTE (Long term Evolution) and its versions such as LTE-Advanced on full scale basis to facilitate their network users. But shifting to a new technology (4G) and replacing the current infrastructure (2G - 3G) takes time. However both the new and the old technologies can coexist during this phase, giving the network operators time to make their deployment strategies. Coexistence of both the technologies does not mean that user equipment will automatically switch to best available signal quality coming from either 3G or 4G cellular base station. In order to cope with this problem, the research community is working on something known as cognitive radio.

Cognitive Radio (CR) defines an intelligent radio system that can monitor and adapt to the changing surroundings in order to make the best use of the radio frequencies. In other words, it dynamically manages the spectrum to give the best possible connection quality for user experience [1]. EECRT (End-to-End Cognitive Ratio Test-bed) is a project at Aalto university that aims to study the whole cognitive radio system, i.e., from end user to the core network. This study endeavors to realize its findings with the help of testbed, i.e., the platform having necessary hardware, software and other necessary tools. Software-defined radio (SDR) [2] toolkits are the means to implement this test-bed. SDRs are useful toolkits since they implement and process the baseband signals in software and hardware is used for RF trans-receiving. This makes it easier for the research community to replicate different kinds of radio systems such as a cellular

base-station or a trans-receiver for an Amplitude Modulation (AM) or Frequency Modulation (FM) radio system. [3]

The focus of our research is to implement voice and data capable core network for the LTE type software defined radio testbed. As discussed earlier about the SDR toolkits, they can be programmed to simulate a trans-receiving LTE base station. An underlying core network is required to make the communications possible between two LTE SDR toolkits. Programming and the testing of SDR toolkits is not within the scope this thesis work. The scope of our thesis work is to design, implement and verify the core network operation. In the first phase, the task is to search and select the tools, programs and technologies that best suit the core network requirements and then the second phase is to build, configure and verify the operations of the core network.

Rest of this thesis is organized in the following manner.

Chapter 2 discusses the call control in 3G's UMTS and 4G LTE technology. This shows the trend that now the cellular operators are also moving towards the VoIP solution, since the mobile broadband is now accessible to almost all the users around the globe. Chapter 3 presents the network technologies that are related to this thesis work, e.g., the SIP (Session Initiation Protocol) used in setting up a VoIP call and so on. Chapter 4 explains the design and implementation of the core network. It describes how the multiple servers inside one single machine are installed and configured to make this network functional. Chapter 5 concludes with the results, issues and future work that can be done further.

2 Call Control in Cellular Systems

With the advancement in the technology over the decades, its impact on the wireless communication has been huge, i.e., people adapted very quickly to the idea of wireless/cellular communications because of its mobility feature. Unlike the previous generations of cellular communications 4G or 4th generation uses the packet-switching method for voice communication. This chapter focuses on the current (4th) and previous (3rd) generations' voice call methods.

2.1 Introduction to 3G - UMTS

This section gives a brief description of the 3G UMTS technology, architecture and its evolution, i.e., from its initial release with CS (circuit-switched) and PS (packet-switched) architecture to an all IP (Internet Protocol) based architecture (PS) with the IMS (IP Multimedia Subsystem).

2.1.1 Evolution towards 3G

As the research and development in technology sector never stops at a particular point, neither do the needs of the world. Voice communications over the cellular system was enough to provide an alternative for the fixed PSTN network but there was nothing to satisfy the needs of growing Internet usage trend on the mobile devices. 2nd generation (2G) cellular system was not providing enough bandwidth, whereas 2.5G was not satisfying the needs of bandwidth hungry applications.

The user behavior of using the mobile Internet has also significantly changed from just checking emails on their devices to listening, watching (streaming) and downloading music, movies and playing online games. Such type of demanding applications require more bandwidth and in order to accommodate more of such type users requires more capacity in the system. The answer was to develop a new technology that could satisfy all the new requirements.

The 3rd generation of cellular system known as the UMTS or Universal Mobile Telecommunication Systems came up with that answer, i.e., it provides the data rates that can match-up to the bandwidth hungry applications, provide real time multimedia services along with the mobility of international roaming. Wideband CDMA or WCDMA is to make sure that users are accommodated and are given the required

bandwidth for their real-time multimedia applications. The 3rd Generation Partner Project (3GPP) is responsible for developing and maintaining the standards of the UMTS technology.

UMTS technology can be looked upon as the upgraded version of its predecessor GSM technology. Since both provide the voice call capabilities and there was not much room left in the development in this area so what differentiates UMTS from GSM is the ability to provide much higher data rates for the mobile Internet users. [4]

2.1.2 UMTS network architecture

UMTS's network architecture can be divided into two main elements, Radio Access Network or RAN also known as UTRAN short for UMTS Terrestrial RAN and other element is CN for Core Network. 'Iu' interface is responsible for the interactions between these two, as shown in Figure 2-1 [5]

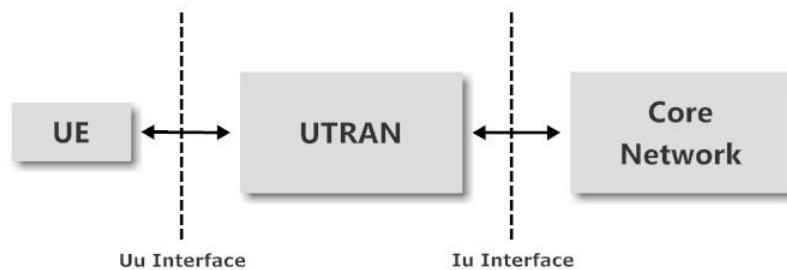


Figure 2-1: UMTS Architecture

UTRAN for (UMTS Terrestrial Radio Access Network) part of the infrastructure consists of the Node B and RNC (Radio Network Controller). Node B can be seen as the equivalent of the BTS in GSM technology, whereas the RNC can be compared to the BSC (Base Station Controller) in GSM. UTRAN is the point of access for the User equipment, i.e., where the user connects to the UMTS network and this is done over the Uu interface.

CN or Core Network in the context of GSM can be viewed as the NSS (Network Sub System) entity. As mentioned earlier that UMTS is also viewed as an update on the GSM technology. Thus, there are two possible scenarios in which UMTS core network can be implemented. The first one reuses the GSM infrastructure as in Release 99 of the UMTS infrastructure. In this architecture, the RNC is connected to the logical entities that are responsible for handling circuit switched calls and packet switched data, which

are in turn connected to PSTN and other IP networks. CS handles the circuit switched calls through the IuCS interface, whereas the PS handles the packet data through the IuPS interface connected, both connected to the MSC and SGSN respectively. This type of architecture is shown in Figure 2-2 [6]

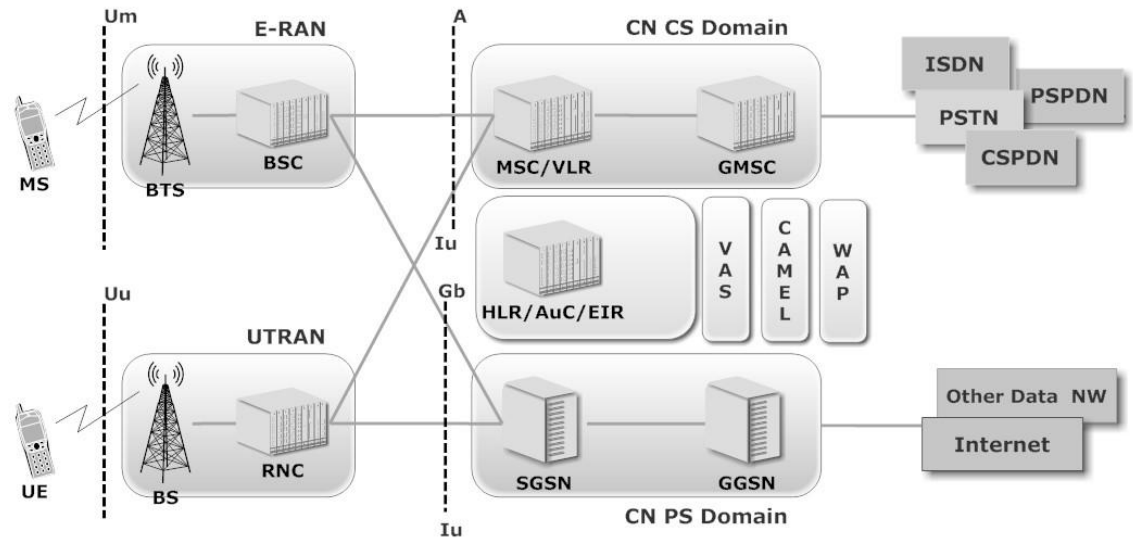


Figure 2-2: Release 99 Architecture of the UMTS

As can be seen from Figure 2-2 [6] UE (user equipment) connects to the Node B (base station) which is then connected to the RNC. RNC in turn is connected to the core of the network, i.e., both domains (PS and CS) which handle either circuit switched calls or packet data.

The other core network implementation approach involves a third new logical element called IMS or IP Multimedia Subsystem. As the name suggests, it involves giving services to the users that are multimedia centric through an enhanced PS core design. This type of network infrastructure is first given in Release 4 of the UMTS as shown in Figure 2-3 [6]. Later releases, i.e., Release 5 (Figure 2-4 [6]) and so on follow the all IP based structure and provide higher data rates than that of the Release 99 UMTS network. [5][7]

Figure 2-3 shows the Release 4 of the UMTS system. In this both the PS and CS domains are connected to a new domain IMS. CS domain in Release 4 introduces the MSC (Master Switching Center) Server and Media Gateway (MGW). MSC Server handles the call control and mobility management logic along with the switching

functions and media gateway. MGW is used for terminating the bearer channels, which are connections between two points that carry the user data. [8]

Release 5 and later versions of UMTS are all based on IP and shown in Figure 2-4. Enhanced version of the SGSN and GGSN are there to support the voice call (circuit switched) capabilities. In order to interact with the external system, R-SGW and T-SGW (roaming and transport signaling gateway) are there. To regulate media sessions CSCF is present. IMS media gateways are handled by MGCF. [8]

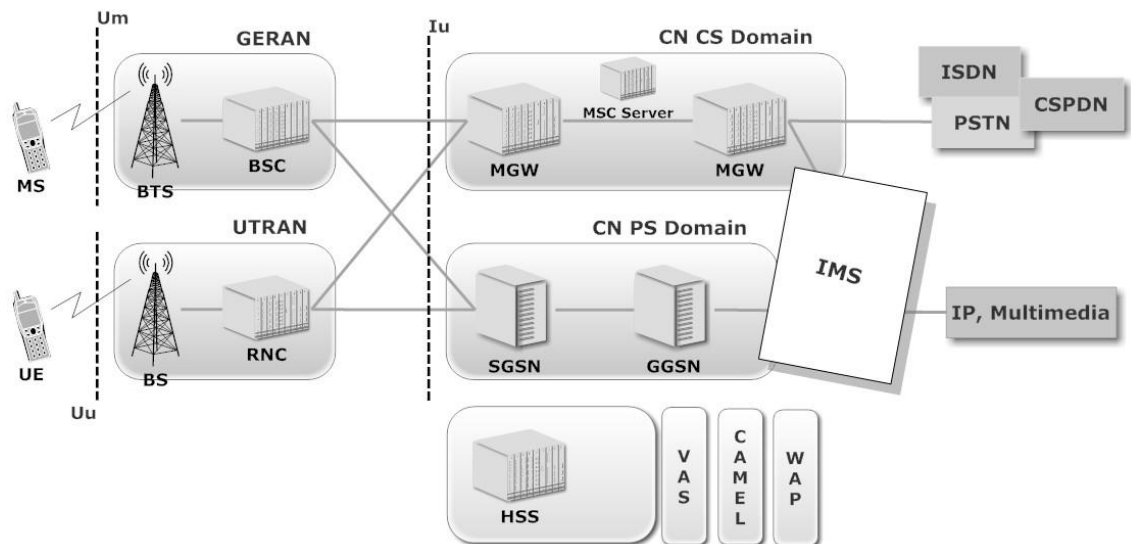


Figure 2-3: Release 4 Architecture of the UMTS

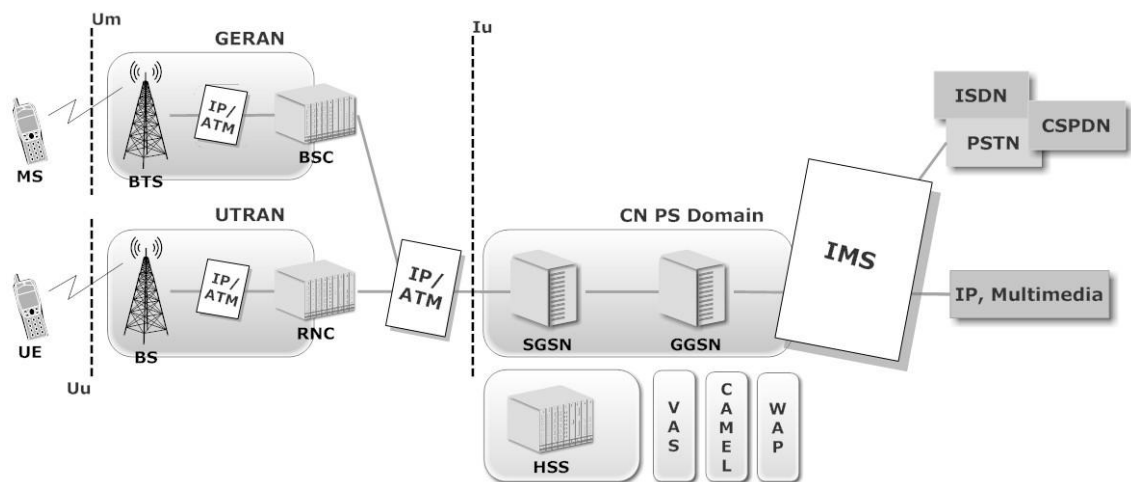


Figure 2-4: Release 5 Architecture of the UMTS

As the core network the 3G UMTS system is divided into domains of CS and PS there are role specific elements in respective domains such as, MSC or 3G MSC, VLR

(Visitor Location Register), HLR (Home Location Register) and MGW (Media Gateway) on the CS domain and SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node) on the PS domain. IMS also has such elements namely MRF (Media Resource Function), CSCF (Call State Control Function), MGCF (Media Gateway Control Function) and HSS (Home Subscriber Server) as shown in Figure 2-5 [8].

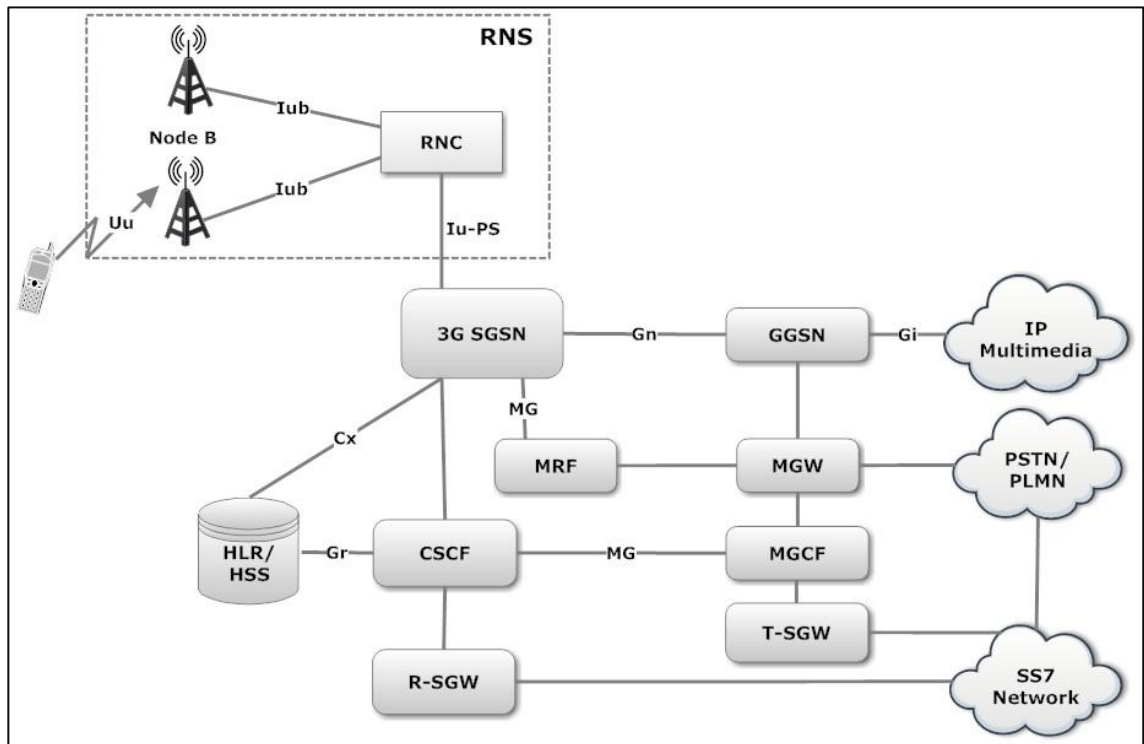


Figure 2-5: Core Network Components in All IP Structure of UMTS

Enhanced versions of the SGSN and GGSN are there to support the voice call (circuit switched) capabilities. In order to interact with the external system, R-SGW and T-SGW (roaming and transport signaling gateway) are there. To regulate media sessions CSCF is present. IMS media gateways are handled by MGCF. [8]

2.2 Introduction to 4G – LTE

Release 7 of the 3GPP was the end of the third generation of mobile communication, i.e., UMTS technology. Continuous research and break-throughs paved the way to introduce the world with yet another generation of the cellular communication, now known as the Long Term Evolution or LTE, i.e., the Release 8 from the 3GPP. LTE literally gets its name from the evolution that happened in cellular communication over

the decades of progress and research, i.e., from GSM to EDGE to 3G-UMTS and finally the IMS. [9]

In order to make full use of the 4G technology, already existing and implemented core infrastructure needed to be upgraded so as to meet the new demands and minimize the replacing/upgrading costs. Thus, System Architecture Evolution (SAE) was started by the 3GPP group to cope with this situation.

2.2.1 LTE network architecture

LTE is an evolution of the UMTS system, which is also reflected in its infrastructure. LTE has, as a part of its infrastructure, E-UTRAN (Evolved – UTRAN) in its so called RAN (not actually present in LTE architecture) and its CN is known as, EPC (Enhanced Packet Core) which is all IP based as can be seen in Figure 2-6 [10].

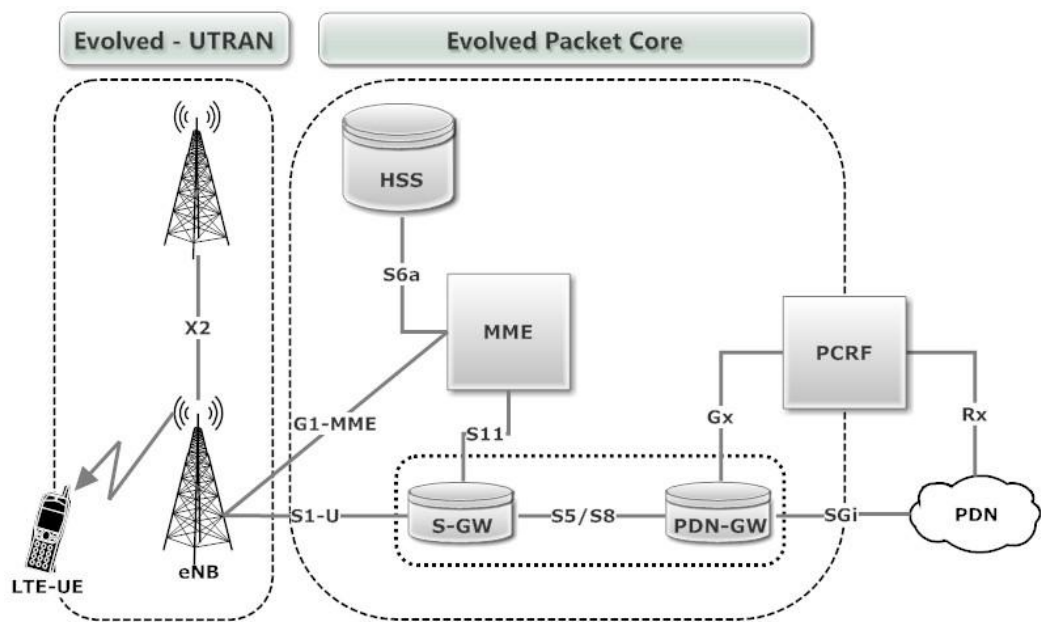


Figure 2-6: LTE Network Architecture

As shown in Figure 2-6, LTE's E-UTRAN only consists of eNBs short for Evolved Node B, which serves as the access point for the LTE supported UEs. eNBs are sole identities in the E-UTRAN, i.e., they operate independently and are connected with other eNBs over the X2 interface and to the Serving Gateway over S1-U and to the MME over the s1-MME interfaces. MME and SGW can support multiple eNBs connected to them, as shown in Figure 2-7 [11]. [12]

Since there is no RNC equivalent component in LTE's E-UTRAN infrastructure, functionalities of the user plane and control plane reside within eNBs along with radio related functionalities such as radio resource management, header compression, security and connectivity. [13]

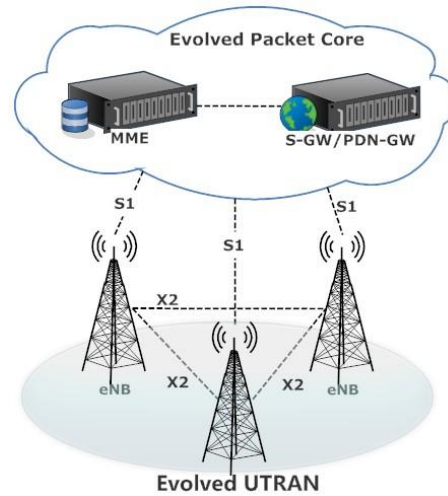


Figure 2-7: Interfaces Between eNBs, E-UTRAN and EPC.

2.2.2 EPC architecture

As mentioned earlier, the core network of LTE is known as EPC and is all IP based, i.e., only the packet switched domain is present in LTE that connects the LTE network to all other outside networks. Figure 2-6 also shows the main components that reside within EPC of the LTE network and they are briefly described below.

Mobility Management Entity (MME): As the name suggests mobility management tracks the location of the user, i.e., giving access to the CN, roaming and handover (within LTE network and also to the legacy networks, i.e., 2G and 3G). This entity allocates CN resources to the user, selection of the S-GW and P-GW also establishes, maintains and tears down the bearers. Basically, MME performs the signaling and control functionalities and serves as the control plane node for EPC. [13]

Serving Gateway (S-GW): Within S-GW reside the functionalities of the user plane node that serve as the mobility anchor for data connections, for both the users that are moving within the LTE network and to and from other systems, i.e., 2G/3G. S-GW has the capability to store/buffer the user's downlink data when the user has gone in idle mode until the user becomes available again. Among other functionalities S-GW also

performs the billing responsibilities, interception of data for filtering, data packets routing and forwarding. [9][13]

Packet Data Network Gateway (PDN-GW): P-GW connects the user in the LTE network with other packet domains, e.g., Internet, intranets and other service providers. Being gateway to other PDNs, its responsibilities extend to but are not limited to Policy and Charging Enforcement Function (PCEF), packet filtering and packet routing and forwarding among others. It also allocates IP addresses to the UEs and manages the mobility related functions between 3GPP and non-3GPP systems. [9][13]

Home Subscriber Server (HSS): HSS can be viewed as the evolved or advanced version of the HLR along with integrated functionalities of AUC (authentication) register. It is responsible for maintaining the database of the user (a profile) for the services such as roaming, access to P-GWs that a user is allowed to and not allowed to along with the QoS (Quality of Service) that a user is entitled to. HSS keeps track of the user location and MME that is serving the user in the home/visited network and updates the information as the serving MME changes for the user. [9][14]

Policy control and Charging and Rule Function (PCRF): As it is clear from the name, PCRF's responsibility is to make policies and charging rules, i.e., to control user's data flow or session in terms of requested QoS and charge them accordingly. It is also responsible for authorizing the QoS to the IP flows that a user is entitled to by enforcing the policies that are in PCEF. PCEF (Policy and Charging Enforcing Function) is logical entity that works in conjunction with PCRF. It may reside within PCRF or could be implemented at gateway. [15]

2.3 Comparison between 3G and 4G systems

It is evident from the discussion and network architecture presented above that 4G is not the upgraded version of 3G but it is built and designed with the future trends of the communications in mind. 4G is all about high speed and evolved packet core rather than splitting its services into two domains of CS and PS domains. This section further compares and differentiates the two generations/technologies of communications.

- First and the foremost difference between them is the use of the access technology, i.e., 3G is built around WCDMA. Whereas, 4G uses OFDMA/SC-FDMA (Orthogonal Frequency Division Multiple Access/Single Carrier-FDMA).
- One of the major differences between UMTS and LTE is the downlink and uplink speeds. LTE with its flat and simplified architecture provides higher data rates than UMTS could ever provide.
- Difference in layout of the architecture is also visible as 3G (UMTS) uses the old cellular concept of access node, edge node and the core, i.e., Node B, RNC and the core (CS and PS domains). Whereas, in 4G (LTE) functionalities of access and edge are combined into one unit, i.e., eNodeB which is called E-UTRAN, the core known as EPC and is all IP based.
- Change in core network architecture of a technology reflects the change in components used in core. UMTS's core is divided into domains of CS and PS with an additional domain of IMS from Release 5 and onwards. However, the use of IMS is to provide the rich multimedia services, it does not eliminate the CS domain. On the other hand, LTE is completely IP based and has updated/simplified its core network elements too.
- UMTS's core network involves MSC/VLR, GMSC, SGSN and GGSN for CS and PS domains in order to provide voice and data connection services. UTRAN (NodeB and RNC) is responsible for managing the radio resources of the network. However, LTE's network uses E-UTRAN comprised of eNBs (interconnected) for radio resource management. LTE's flat core network architecture, i.e., EPC, handles the data packets. Unlike in UMTS, LTE uses HSS, an advanced version of HLR/VLR to maintain the database for the LTE users. Other functions in LTE's EPC are handled by MME, S-GW, P-GW and PCRF. However for voice calls, LTE uses IMS or legacy networks, described in later sections.

2.4 Voice in 3G

Circuit-switched domain of the UMTS allows the user to make calls to the PSTN network. One such basic call flow diagram is shown below in Figure 2-8 [16]. Call is originated by the mobile user to the PSTN network.

In order to make a call, the mobile needs to establish a link with the RNC in the UTRAN and this is done by establishing the RRC (Radio Resource Control)

connection, having the underlying protocols MM (Mobility Management) and CM (Connection Management). After which the UE (user equipment) requests to access the CN and connection is established over the Iu-CS interface. Next step in making a call is to authenticate (optional step) and to secure (mandatory step) the established link done by ciphering or encryption of the information over the radio link.

As shown in Figure 2-8, next in line is the call progress information, i.e., the information about the called party is supplied to the 3G-MSC, after which the RAB (Radio Access Bearer) is assigned, i.e., the resources in the system are allocated for the voice call in question. Other information like QoS is also supplied to the UTRAN, which is then forwarded on to the Node B and then the UE to allocate proper resources for the call. RAB is the combination of the Radio Bearer from UE to the UTRAN and Iu-CS Bearer from UTRAN to the 3G-MSC. Once the acknowledgement is received, the call is put through. [16]

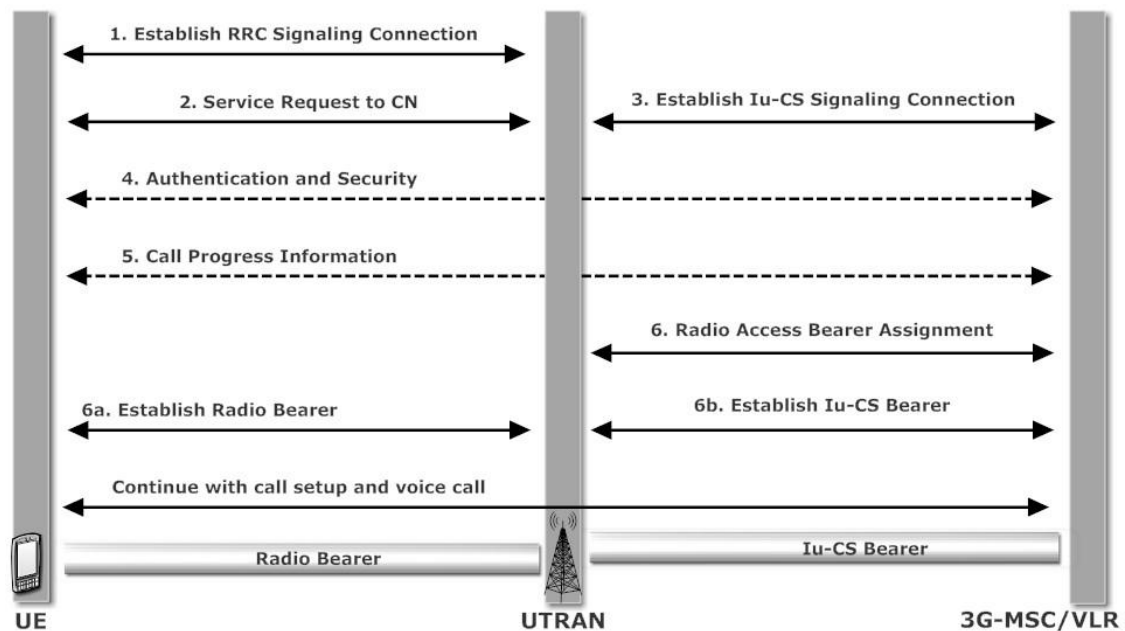


Figure 2-8: Call Connection Flow In UMTS Circuit-Switched Domain

2.5 Voice in LTE (VoLTE)

LTE was designed as the complete IP based system and with respect to the traditional cellular technologies this poses a problem that there is no support for the circuit-switched voice calls or SMS (short message service) or any voice complimentary services for that matter. According to the statistics in general, a major portion of the revenue for the telecom operators is generated from the voice and messaging packages.

This presents a dilemma for the operators whether to deploy LTE systems or not, despite its high data rates and flat architecture.

To end this dilemma, 3GPP and other organizations like GSMA (Global System for Mobile communication Association) provided interim and long term solutions, i.e., Circuit-Switched Fallback (CSFB) and Voice over LTE (VoLTE) using IMS also known as VoIMS respectively that enable an LTE network to support voice calls along with other voice related services. However, a couple of other alternative options are available as a result of independent studies done by research groups and organizations such as, Voice over LTE via Generic Access (VoLGA) and Simultaneous Voice over LTE (SVLTE). [17]

SVLTE and VoLGA ideas were dropped by the industry because of SVLTE's complexity and LTE device hardware support issues, whereas VoLGA was perceived as a step back as it relied on the previous technologies for call features rather than exploring the LTE's core network for voice options. [17]

2.5.1 Circuit-Switched Fallback (CSFB)

As we know that LTE is all IP based architecture and this architecture does not support the traditional circuit-switched voice calls. Thus, as an interim solution CSFB was introduced by 3GPP in TS 23.272 [18] specification.

As the name fallback implies, the device will switch to a legacy network (GSM/UMTS/CDMA) whenever there is voice call activity and as the activity ends it reverts back to the LTE network. Because of their (legacy system's) wide deployment and coverage this can prove to be a good temporary solution during the initial phase of the LTE deployment, i.e., until LTE (service provider) gets its own nationwide coverage and moves on to a better and permanent solution.

CSFB comes with its own flaws, i.e., the switching between legacy systems takes time (steps that are required) and this adds delays to the call setup time. Call setup time, strict synchronization of Location Area & Tracking Area mappings and upgrading of MSC Servers surrounding the LTE coverage area can be some other problems to deal with for CSFB. [19]

Figure 2-9 [20] shows the architecture through which CSFB is supported in an LTE network. As we can see in Figure 2-9, the MSC/VLR in the 3G network is connected to the MME in the LTE network over the SGs interface. SGs interface is based on the Gs interface, between the MSC/VLR and SGSN for signaling exchange. This interface is responsible for providing all the necessary signaling that is required for mobile terminating and mobile originating calls to and from the CS domain to the LTE network and also handles the mobility management between 3G-CS and evolved packet core. [20]

Figure 2-10 [20] shows the mobile terminating call process using CSFB. MSC/VLR on receiving the notification of the terminating call finds the MME in which the user can be located. MSC sends out a paging message to the concerned MME and in turn the MME sends out a paging message to the mobile user indicating that there is a voice call and is a CS service. UE acknowledges that and requests the MME for the fallback service. MME receives that request and a handover command is initiated to the 3G network. On 3G network mobile device notifies the concerned MSC/VLR via paging and is acknowledged thereafter for the call termination procedure to complete. [20]

However, in mobile originating call, the UE first requests for the fallback service from the MME as the CS call service resides in the 3G domain. MME initiates the handover process and in doing so, it also transfers the data bearer connection over to the 3G, to maintain the always connected state. UE that has moved on to the 3G network, requests the concerned MSC/VLR for the voice service and is acknowledged back by the MSC/VLR and thus, completing the fallback procedure. [20]

2.5.1 VoLTE using IMS (VoIMS):

As discussed earlier, many solutions have been presented to counter the fact that LTE does not support the circuit switch voice call and its features. One solution that has gained the consensus of the industry was studied, researched and is documented in GSMA's PRD (Permanent Reference Document) IR.92, known as "IMS Profile for Voice and SMS", i.e., it uses the IMS domain in order to achieve the voice call capabilities. GSMA's IR.92 was also known as the "One Voice" and it defines "a profile that identifies a minimum mandatory set of features which are defined in 3GPP specifications that a wireless device (the User Equipment (UE)) and network are

required to implement in order to guarantee an interoperable, high quality IMS-based telephony service over Long Term Evolution (LTE) radio access.” [21]

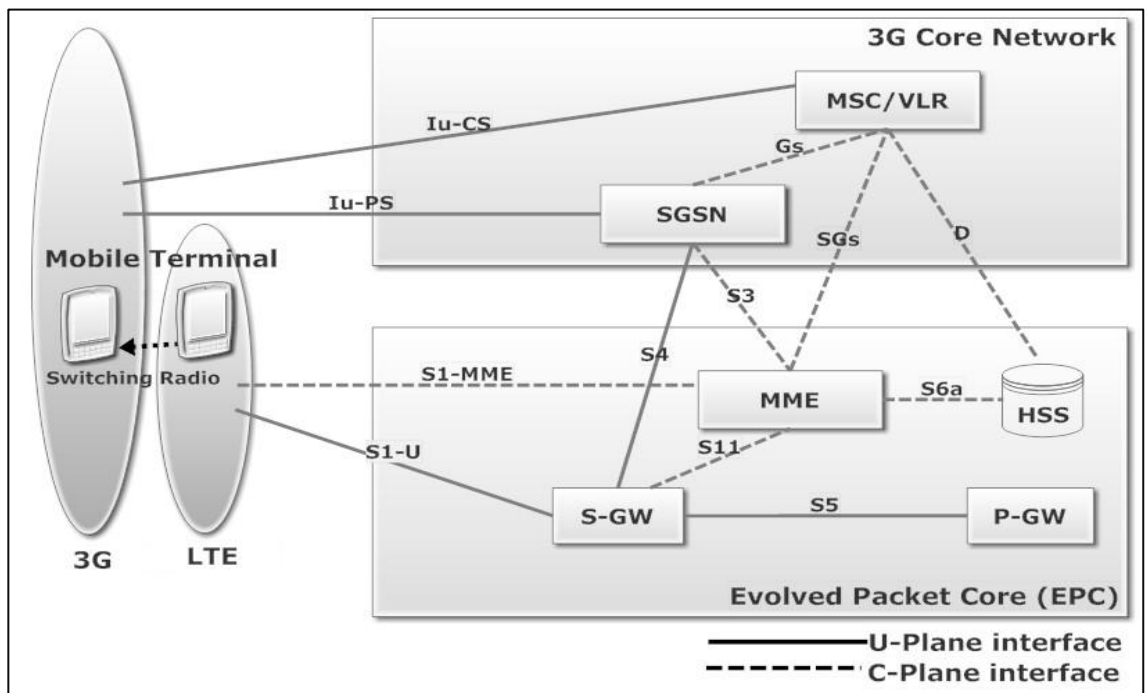


Figure 2-9: CSFB Architecture

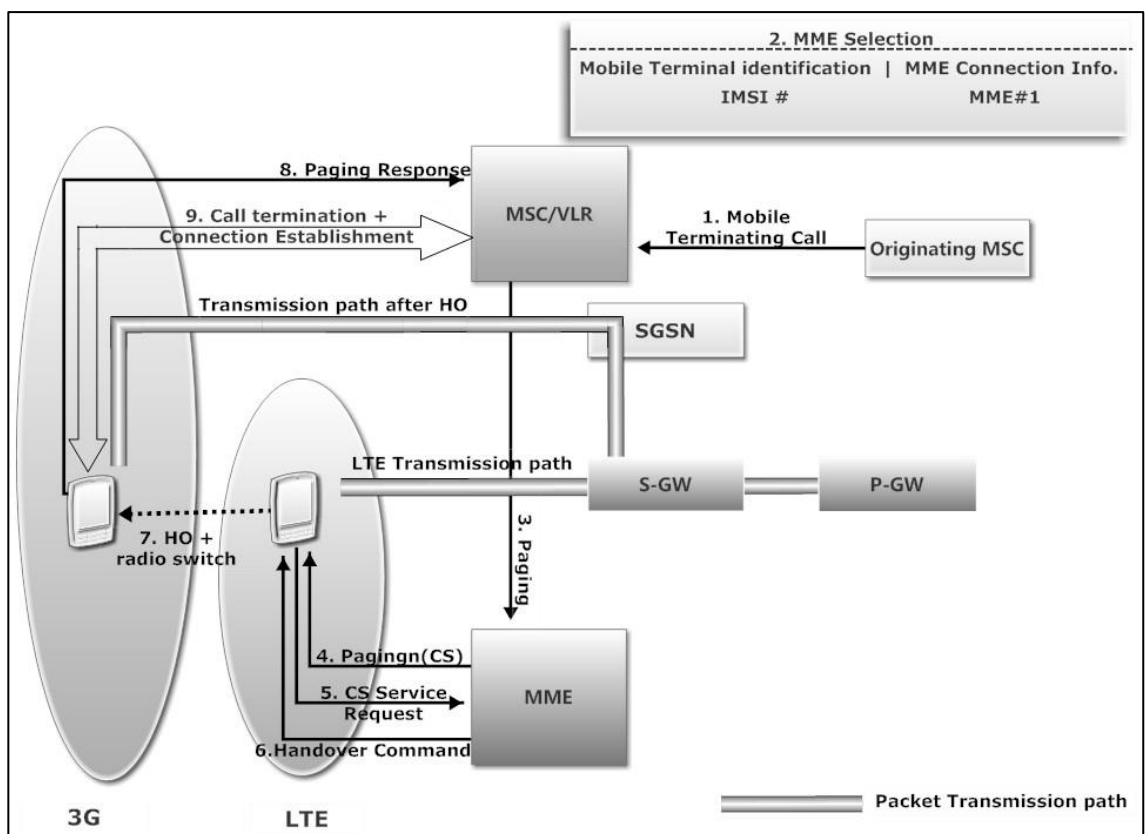


Figure 2-10: CSFB Mobile Terminating Call Procedure

Since this a long term solution, all the players in the industry have agreed upon and accepted to implement it as soon as LTE gains its nation-wide coverage. However, until that time, CSFB (discussed earlier) was chosen as the interim solution along with the support of SRVCC (Single Radio Voice Call Continuity). SRVCC standardized by 3PPG in TS 23.216 [22], defines that when a UE leaves the LTE coverage it should stick to one RAT (radio access technology) at a time, meaning the data connection from LTE should also be handed over to the switched over system. It also enables the transformation of the calls, i.e., continuity from IMS domain to the switched over network since the data connection is available. [22][23]

Figure 2-11 [24], shows the configuration of the VoLTE network and the interface connections between the UE (VoLTE terminal) and IMS ,i.e., UNI or User-Network Interface along with the NNI (Network-Network Interface) used between IMS and other networks. [24]

IMS is responsible for handling the connection path between UE and IMS, security features, call origination and termination functions. All such and related actions are handled through AS (Application Server), S-CSCF (Serving – Call Session Control), P-CSCF (Proxy – CSCF) and I-CSCF (Interrogating - CSCF) in the IMS structure.

Figure 2-12 [24] shows the call origination process in VoLTE using IMS. Since this solution does not involve the circuit-switching, UE uses the SIP (Session Initiation Protocol) signaling procedure to complete the call process.

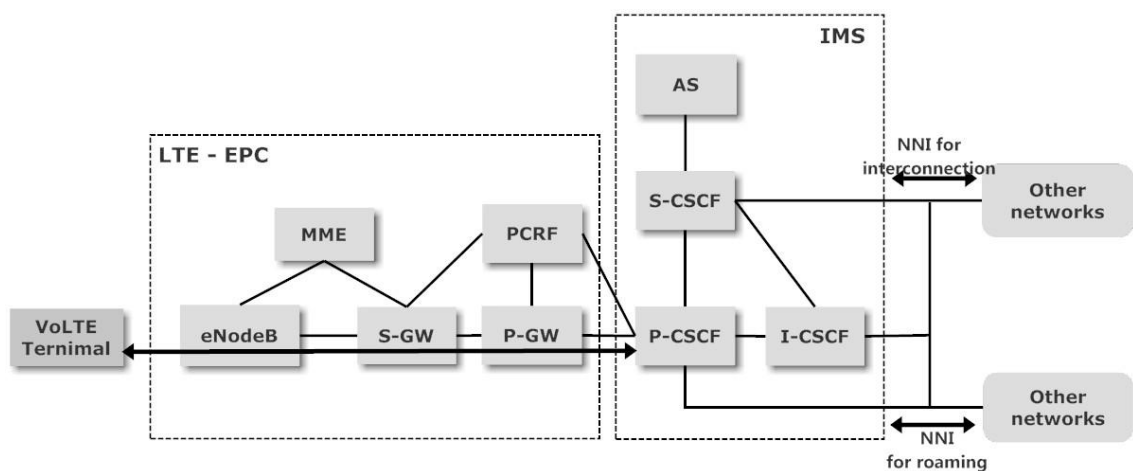


Figure 2-11: Configuration of VoLTE Network

As can be seen from Figure 2-12, first UE sends INVITE message carrying the necessary information that is required to setup this session (call) to the terminating IMS. This INVITE message travels through the P-CSCF, S-CSCF and AS of this call originating AS.

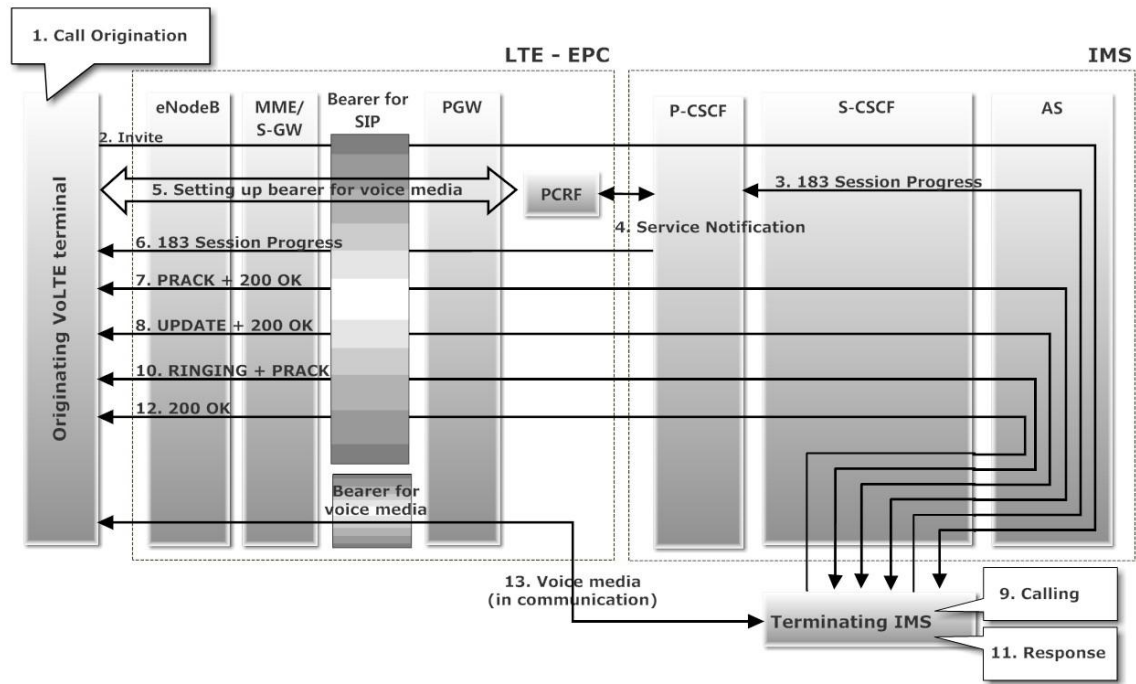


Figure 2-12: VoLTE Mobile Call Origination Process

The terminating IMS replies with a session progress message, with information (parameters), required for the both parties to consent on, in order for this session (call) to go through, to the originating IMS, more specifically P-CSCF. P-CSCF in turn asks PCRF to reserve the resources, i.e., setup a bearer for this call with required QoS. P-CSCF having received the information from terminating end, forwards that to the originating device so that it can compare and/or modify the parameters accordingly for the call progress.

New selection/modification/left unchanged parameters are conveyed to the other end using PRACK and an acknowledgement for that is received from the terminating end using '200 OK' message. Next the UPDATE message is sent from the originating terminal, when both the parties have agreed to go through this call process using the selected parameters and QoS and is again acknowledged for that from the terminating end using '200 OK' message.

Called party user receives a ringtone and when picked up and in response, an acknowledgement message (200 OK) is sent to the call originating end and the voice communication is ready to go through.

2.6 Summary

We have seen in this chapter how the voice calls take place in both UMTS and LTE networks. It is evident that technology drives the market trends and the current market trend that people opt for is high speed broadband mobile Internet. LTE fulfills all the current needs with its simplified and flat EPC architecture, which is all IP based. This brings up a new trend in voice communication, i.e., VoIP or more specifically VoLTE using IMS and eliminates an old one, i.e., voice over circuit-switched network.

3 Test Network Technologies

This chapter briefly discusses the technologies that are required in a small to medium sized SIP based network. These include, IP addressing, allocation of IP addresses, IP to name mapping, how IP packets are forwarded between different networks and finally what kind of technology allows to make calls over Internet and what kind of servers are required in order to implement all of the above.

3.1 What is VoIP

With the speed Internet grew over the years its uses and applications also grew with the same pace. Transporting voice over the Internet has been one of those useful applications. With end users having access to more reliable and fast connections, i.e., broadband connections VoIP gained overnight popularity among the Internet users.

In general, VoIP technology refers to the transmission of the digitized voice, encapsulated in packets over the IP network. VoIP calls are made through the use of software applications called soft-phone and are installed in computer and/or smart phones. Soft-phones are used to make and receive calls between the users of same application software. Calls can also be placed to the traditional network (if service provider allows) via the use of gateways that interconnect both systems. Figure 3-1 [25] depicts generic concept of VoIP calls.

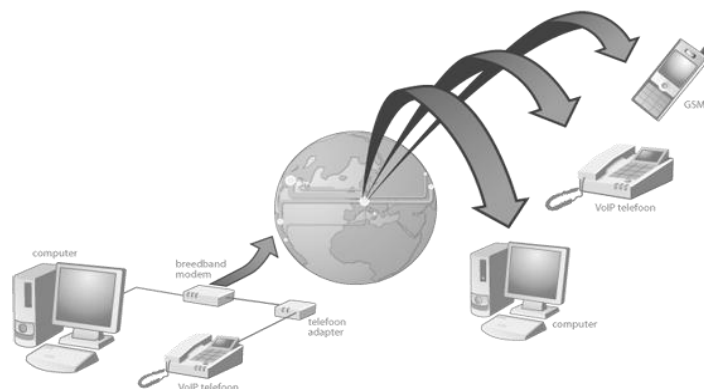


Figure 3-1: Illustration of VoIP Concept

Figure 3-1 shows that how an end terminal device (a computer, with VoIP client software) uses modem to connect to the ISP (Internet service provider) and can make voice calls, i.e., VoIP sessions to other end terminals (can be a computer running VoIP client software, a traditional phone or a cell phone).

VoIP is generally referred and confused with IP Telephony even though they both accomplish the same purpose, i.e., transmit voice over the network. IP Telephony was designed and developed to support video conferencing over a LAN segment, e.g., that of an organization but later on was modified to support voice over span of networks [26]. It also uses different protocol than that of the IEFT (Internet Engineering Task Force) standard of VoIP protocol, which was originally designed for Internet.

H.323 protocol is an ITU standard that is linked with IP Telephony. IP Telephony deals with all communication services, i.e., voice, text, video (conferencing) and fax that make use of IP and underlying protocols (UDP for signaling and TCP for reliable data transfer) for transportation that otherwise use traditional PSTN network for such services [27]. H.323 uses a different stack of protocols, different network entities and signaling architecture than that of an IETF's VoIP protocol. However, both H.323 and SIP (Session Initiation Protocol) IEFT's VoIP standard use some common set of protocols (briefly described in later section) to achieve their goal of voice transportation over IP network. [26]

H.323 network comprises of terminals, gatekeeper, gateway and MCU (Multipoint Control Unit) components. As can be seen from Figure 3-2 [28], 'Gateway' connects the H.323 network with the other outside networks. Whereas Figure 3-3 [29] depicts the terminal components, codecs and protocols required for real-time two way communications.

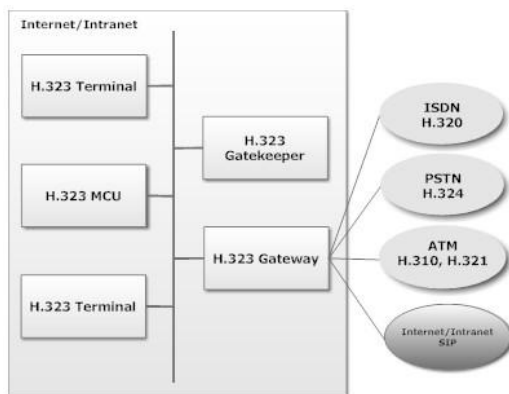


Figure 3-2: H.323 Network Components

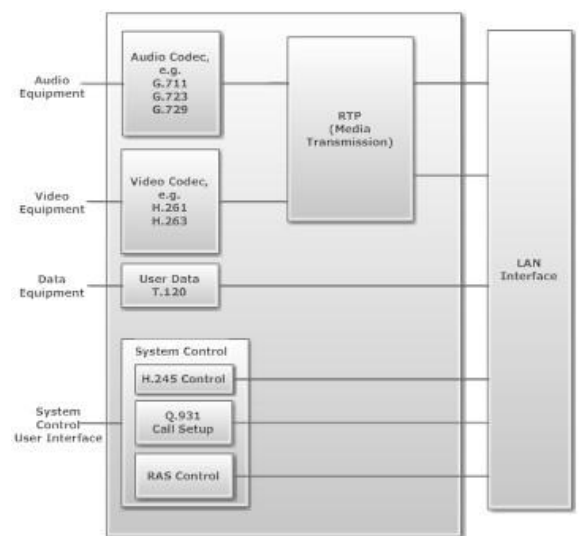


Figure 3-3: H.323 Protocol Scope

H.323 was designed to support multimedia communications over a LAN segment, but later on connectivity to Internet and other networks was added. Whereas SIP as a text based protocol was designed to support multimedia communications over Internet. Also, SIP protocol is relatively simple and faster to process for the network entities, and as such it will be the scope of our study here and not the H.323.

3.1.1 Session Initiation Protocol

This section briefly explains the SIP protocol in general, the network entities that make SIP communication possible along with signaling (request and response) architecture of SIP. As was seen in the previous Chapter 2, VoIMS solution for VoLTE makes also use of the SIP protocol for voice communication in the LTE network. A detailed study of SIP protocol can be read from RFC 3261 [30] and its extensions for other related RFCs.

Session Initiation Protocol short for SIP is an application layer communication protocol designed by IETF. It is a widely accepted and deployed VoIP protocol by almost every manufacturer of IP phones, Call Managers and IP PBX (Internet Protocol Private Branch Exchange). As the name indicates, SIP is responsible for initiating the session along with the support of maintaining and terminating the session. A session in this context can be described as the activity between two or more users that include but is not limited to multimedia interaction like audio and video call, an instant messaging session (chatting) or a conference call. SIP protocol concept has its legacy based on another text based protocol known as HTTP. SIP follows the similar concept of client-server model in this context.

Though the working principle of SIP is similar to end-to-end concept, i.e., peer-to-peer communication used in PSTN (Public Switched Telephone Network), this does not mean that SIP is a replacement of the regular PSTN. What differentiates SIP from the PSTN is the flexibility, robustness and extensible capabilities of the protocol itself. It can be customized to integrated video, IM (Instant Messaging), presence and other features as they emerge. Also SIP networks can be more customizable and scalable in adapting and implementing new features and services that are almost impossible to implement in traditional PSTN networks. [31]

SIP was designed in a way that it relies on other protocols for interactive communications to take place. SIP protocol paves the path for communication to take

place but the interactions between the end parties may be the responsibility of other protocols. Some of these protocols (RTP/RTCP, RSVP) also work with H.323 [26] and are briefly defined in Table 1. [30]

Table 3-1: SIP Supporting Protocols.

RTP/RTCP:	Real-time Transport Protocol/Real-time Transport Control Protocol is responsible for carrying the real time data, i.e., voice and/or video also the text. [31]
RTSP	Real-time Streaming Protocol, controls the streams of the media between the end parties involved in session [30]
RSVP	Resource Reservation Protocol reserves the resources (media) between the end parties. [26]
SDP	Session Description Protocol carries the parameter information (e.g., encoding and the transport protocol involved) between the end parties so that they can negotiate the term for the establishment of the session. [31]
MGCP/MEGACO	Media Gateway Control Protocol is used to interconnect two different media systems such as SIP with H.323 or with PSTN. [26]

In order to establish or terminate a session SIP supports five features, mentioned in RFC 3261: [30]

- **User location:** determines the location of the other end party for communication to take place.
- **User availability:** determines if the other party is available to establish the session.
- **User capabilities:** determines the media and its parameters for communication between the end parties.
- **Session setup:** Once the parameters for the session have been agreed to, both the end parties are informed of the progress of session, which includes ringing at the callee and ring-back-tone at caller ends.
- **Session management:** Management handles the transfer, termination and modification of the parameters during the session. If some services are invoked during the session managing those also falls into session management.

3.1.2 SIP Network Entities

A SIP network contains different entities or components that are responsible for communication between the two end points. In this section, such basic components are briefly discussed. These basic components are user agents, proxy servers, registrar servers and redirect servers as shown in Figure 3-4 [32]. Also, direct exchange of messages is possible between two SIP enabled end devices [31].

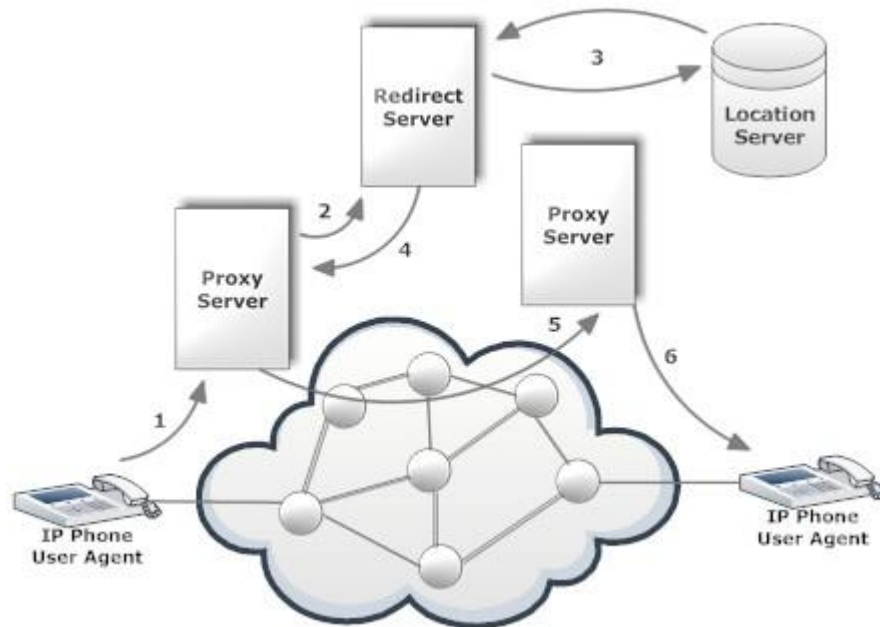


Figure 3-4: SIP Network Entities

User Agents: In a SIP network, end devices have an application installed in them and is known as User Agent (UA). This application acts as both the user agent client (UAC) and also as user agent server (UAS). Purpose of this application is to act as the middleman, hence the word ‘agent’ on behalf of the user to initiate and terminate the session established between other user agents. [33]. Thus, in simplest terms, the SIP end device which initiates the request is acting as UAC whereas, the other end which gives response to that request is acting as UAS.

Proxy Servers: In a SIP network, proxy servers are an important part of the infrastructure. UACs send requests to the proxy server, which in turn processes and passes on the requests to the next hop (other end client device or another server in the line). At the same time UAS sends the response generated for that request to the client device. Proxy server is not to be confused with the UAS, instead the proxy server serves

as the router in IP networks, i.e., forwarding the packets to the correct destination. Proxy servers take part in establishing or tearing down a session and the role of the proxy server fades away as the user agents establish session. Though main function of the proxy server is to route the SIP requests to destination, it can also be used for enforcing a policy, e.g., if a user is allowed or denied of making calls. Also, if necessary the proxy server can alter specific parts of the SIP request before forwarding it. [30][34]

Registrar Servers: Registrar server is the entity in SIP network that is responsible to authenticate and register the users as they become available. The purpose of the registrar server is not only to register the users but also to extract and update that information so that other entities like proxy servers can use that information for contacting that user. An example of such information could be IP address or port number to which they respond to.

Redirect Servers: Redirect servers in SIP network infrastructure have the unique functionality of responding to a SIP request by providing the callee's location information in the network. If the called party is not present on its last updated location or has moved to another location, upon receiving the request, redirect server locates the user with the help of location database and responds to the caller with the current location of the user with a 3xx class response (response messages are categorized into classes and are explained in section 3.1.4).

3.1.3 SIP Request Messages

Signaling is a way of communication between the entities in the SIP network and is based on a series of messages. These SIP messages are categorized into either requests or responses to the request generated. Request messages are initiated when some action is required and are initiated by the SIP clients. Response messages are replies to those SIP request messages and are sent by the SIP server. SIP messages are transported individually over the network in UDP datagrams [31]. SIP messages are identified by their first line, i.e., what type of message it is followed by the message header and then the message body.

A request message is meant to invoke a function on the server and is called Method [30]. In order to distinguish the method name from the message header, they are usually written in uppercase letters. Whereas the header field in a message makes use of both

lower and upper case letters [33]. According to RFC 3261 there are six main methods but separate RFCs define more of such methods. Six methods defined in RFC 3261 are briefly described here.

INVITE: INVITE method is used when a user agent client (UAC) wants to establish a session (multimedia session) with another client and it sends the request to the server. Invite message is sent to the other client (acting as UAS) and if available a proper response (ACK) is received at the client end who initiated the session.

Since INVITE is used to join user for the session, first line in INVITE represents the destination, i.e., the address of the host requested to join. Body of the INVITE request message contains the description of the session. If during the session one of the parties wants to modify the parameters of the session, a re-INVITE has to be sent with the modified parameters. [35]

ACK: ACK for acknowledgement is the message sent to the called party after receiving the final response to the INVITE request. ACK makes use of the 3-way-handshake. ACK method makes sure the server knows of the successful session establishment.

BYE: As the name suggest, BYE messages are used to terminate an ongoing session. BYE is sent from either of the parties (UAs) involved in the session who wishes to hang-up.

CANCEL: CANCEL method is to cancel or abort the session before it is established, i.e., before receiving the final response, the calling party has the option to cancel the session. In most of the cases it is used when the called party has not answered for some time. [33]

REGISTER: Client sends the REGISTER request to let the registrar server know of the user's current location. REGISTER message has the information, such as IP address and the port number on which the UAC is reachable. This information is extracted by the registrar and is updated in the location database for the other SIP entities, e.g., proxy server.

OPTIONS: OPTIONS are the request messages that a UA sends to another UA or a server in order inquire about their capabilities, e.g., what methods, extensions, encoding

etc. are they supporting. It is generated only by the UA and never by the proxy server. [30] [33]

Figure 3-5 [36], shows some of the request and response messages (described in next section).

3.1.4 SIP Responses Messages

SIP response messages are generated by the UAS in response to the request made by the UAC. SIP response messages are similar the response messages used in the HTTP protocol. SIP responses are classified into six reply classes.

Response message contains the reply code and reason phrase. Reply codes are integer numbers from 100 to 699 indicating the type of the response. If any UAC fails to understand the reply code in the message, that response message falls into that response category class [33].

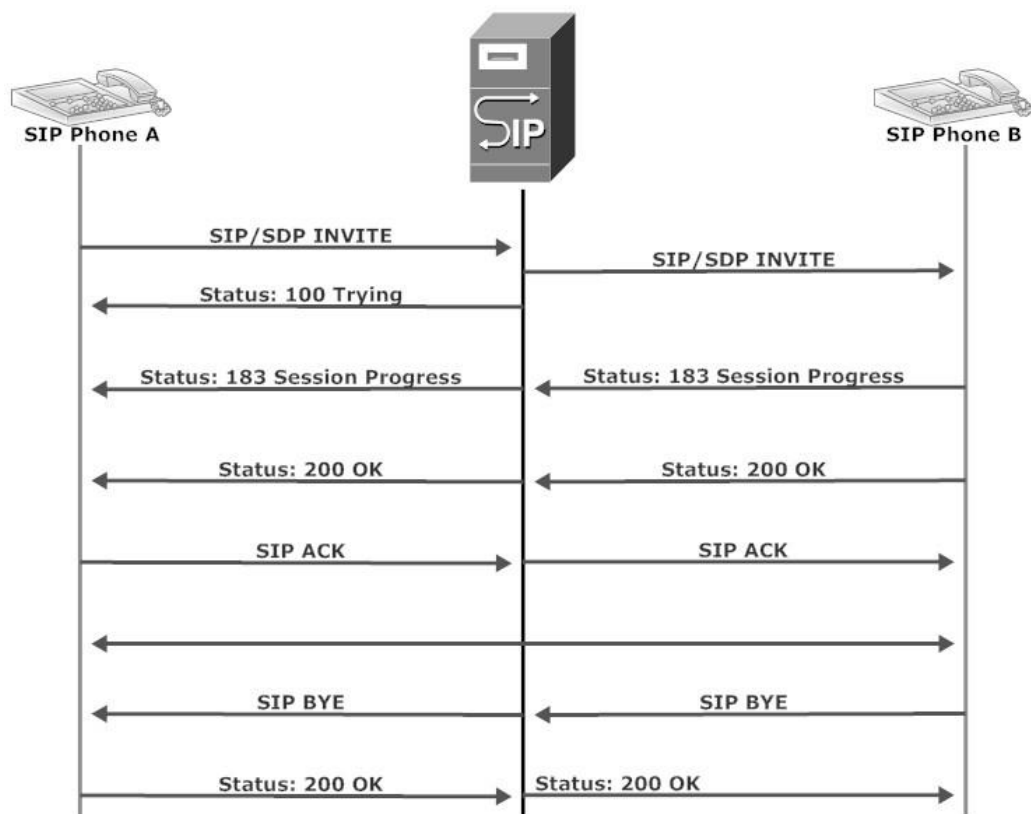


Figure 3-5: SIP Request and Response Messages During Call Flow

1xx: 1xx are the informal responses. Informal responses are sent when there is a delay in processing the request. Purpose of the informal response is to stop the sender from

retransmitting the INVITE request again and to let the recipient know that the server is processing the request but the end result is still not known. They are end-to-end responses and may contain message bodies.

2xx: 2xx response messages show successful transactions. They are positive responses indicating that either the request was accepted or was processed successfully. 2xx response is the final response that a UAC can expect for the processed request. These responses range from 200 to 299.

3xx: 3xx are the redirect responses. When the destination user agent (UAS) is not reachable for any reason 3xx responses are generated with a possible list of destinations where the UAS may be reachable. UAC is then expected to resend the INVITE request based on the new destination address received. 3xx responses are typically generated by the proxy server.

4xx: 4xx depict the errors in the requests sent by the sender. They are the negative responses. These responses depict that either the request sent had errors in it or could not be processed at the server.

5xx: These responses indicate that the server was not able to process the request due to its own fault (e.g., traffic congestion) and the user should retry to send the request, as there was nothing wrong with the request.

6xx: These responses represent a global failure, i.e., the request was not entertained at any server including the server which had the information to entertain that request.

References [31] [33] and [35] were used for this section.

3.1.5 SIP Proxy Servers and VoIP Clients

Section 3.1.2 explained the role of SIP proxy servers. Explained below are two such examples, i.e., a SIP proxy server and a media server that handle the SIP communications in a network. Additionally, an implementation of a VoIP client software is described.

OpenSIPS

OpenSIPS is an open-source SIP proxy server among many others but what makes it stand out are its features like modular design, speed, flexibility, portability and

reliability. Developed in Linux with support for most Unix-like platforms, it runs as daemon (in background) and performs the functions of a proxy server, i.e., altering/manipulating of the headers, forwarding of the requests and policy implementation at very high speeds, i.e., it is capable of handling thousands of calls per second. Implementation of OpenSIPS follows closely the standards that are set by the IETF and ITU for SIP protocol.

OpenSIPS started as SER (SIP Express Router) project in 2001 but in 2005 was bought by TEKELEC and was named the 'iptel SER' project. But due to some differences, a team from SER started the OpenSER, i.e., picked up from where the SER was left off. Later in 2008 again due to differences in the development team, it split into what is now known as the OpenSIPS and Kamailio.

OpenSIPS can also take the roles of router, switch, registrar, application server, redirect server, gateway, and load-balancer among others due its flexible design. This role is determined by the script from "opensips.cfg" file. The script closely relates in syntax to the C programming language. Modules that are part of the script initialize and determine the role that OpenSIPS will play. Modules were designed to give additional functionality of remote authentication and authorization through RADIUS (Remote Authentication Dial In User Service) protocol, number mapping service through ENUM (E.164 Number Mapping), Presence and instant messaging for OpenSIPS, and many other are being designed as OpenSIPS gets its share of popularity. It is due to this flexible nature and high performance and reliability that OpenSIPS has found its application in enterprise class networks and VoIP service providers. [37][38][39]

Asterisk

Asterisk, what was developed as Linux based IP PBX (Private Branch Exchange) by Mark Spencer in 1999 for small businesses, now supports multiple operating systems such as, BSD, MacOS X, Solaris and many others. Also from just being an IP PBX, Asterisk now incorporates the ability to support VoIP gateways, call center systems, and voice mail servers along with all other real-time communication applications that can be designed for Asterisk.

Asterisk, like OpenSIPS, is also licensed under GPL (General Public License), i.e., it is also open-source, but there is a difference between the two: OpenSIPS is a VoIP proxy

server whereas, other is VoIP PBX. Next section discusses in detail the differences between the two. Asterisk supports all the traditional telephony as well as VoIP protocols and can also act as the gateway between the two.

Asterisk acts as the back-to-back user agent (B2BUA), i.e., unlike a proxy server the call terminates at the Asterisk server and the server makes another call to the destination. When answered, the Asterisk server connects the call through. On the other hand, the proxy server just forwards/routes the call to the destination or to the next hop.

Asterisk's features (among many others) include IVR (Interactive Voice Response), conference calls (both audio and video), voice mail facility, call recording, fax service and CDR (Call Detail Records). Like OpenSIPS, Asterisk server also has the configurations and scripts that determine and initialize the functionality as required. [40][41]

Comparison between OpenSIPS and Asterisk

- Difference in architecture: Asterisk, functions as a B2BUA whereas OpenSIPS is a SIP proxy that relays the call towards the destination or the next hop and it has a more simple architecture than that of Asterisks.
- Ability to connect to the PSTN network: OpenSIPS does require a gateway whereas the Asterisk has telephony interfaces that make it easy for it to connect to the PSTN network.
- Capability to deal with NAT Traversal: OpenSIPS does a better job in dealing with NAT than Asterisk, even in cases when both the parties are behind NAT.
- OpenSIPS with its simple architecture can be configured to effectively deal with load balancing based on different parameters configured. Asterisk also can perform load balancing based on parameters configured.
- Another basic difference is in supporting media capabilities. Asterisk being an IP PBX supports many features that a proprietary PBX supports, such as IVR, CDR, conference calling, call recording and many others. On the other hand, OpenSIPS does not support all these media features as it was designed as a Proxy server.

There are many other differences between the two, such as number of handling the calls per second, robustness, scalability and security features. But their selection depends on

the scenario in which they are being utilized. However, combination of both can be used in a network with OpenSIPS on back-end and Asterisk on the front-end. [42][43]

Ekiga – VoIP Client

In order to make a VoIP call, a user will need some kind of software that can carry user data (voice) to and from other user. This software is called Soft phone and is installed in a computer (PC) and/or can be installed in a smart phone. There are different client softwares available, some use company specific proprietary protocols to make calls and other follow the open standards and are free to use. One such software is Ekiga, available for different platforms like Linux, UNIX and Windows.

Ekiga, previously known as GnomeMeeting, is open-source VoIP, IP Telephony and video conferencing client software that allows to make free calls as well as to start IM conversations with another Ekiga client user [44]. It is not only a PC-to-PC VoIP software, but also supports calling to the traditional PSTN network with the help of VoIP service providers.

Ekiga has been around in the VoIP service now for over a decade (GnomeMeeting was introduced in 2001) [45]. Ekiga's code has evolved over the time making Ekiga is one of the free softphones available for users. It supports both major protocols in VoIP, i.e., SIP and H.323. It makes use of state of the art free codecs to support features like echo cancelation, video calling, call holding, call transfer along with IM. [44]

Currently Ekiga version has been released as a major update from its previous version, some of the main features are as described below (details of these features can be found of Ekiga's official website): [46]

- Ekiga comes with a user friendly and modern GUI (Graphic User Interface) look.
- Support for presence, i.e., users are able to see other users' current status.
- Use of codecs to support High Definition (HD) quality sound and DVD quality video.
- Ekiga does not restrict users to one service provider, but gives them freedom to let them run Ekiga with their own choice of the service provider.
- Depending on the service provider, Ekiga also supports text messages to cell phone.
- Support for Address Book (saving contacts) both local and remote.

- Ekiga is inter-compatible with other soft-phones, traditional phones, IP-PBXs (Private Branch Exchange).
- It supports DirectX 9 Video Capture/Output on Windows platform.
- Understands the needs of administrators, some of the settings can be blocked as per administrator's requirement.
- Ekiga also supports IPv6 on experimental bases.
- Assisted NAT support, i.e., it detects the NAT settings by itself so that the user does not have to worry about them.

3.2 Virtualization

Virtualization is a technology which allows the running of multiple OS instances at the same time on a single machine. It can also be defined as the running of different software programs on hardware and/or software in a simulated environment and that environment is called Virtual Machine (VM) [47]. Physical hardware of the computer can be more effectively utilized with the help of virtualization. Virtualization is mainly implemented as software, but with its increased popularity vendors have started to support virtualization by making special processors that have features to support virtualization processes explicitly. [48]

A physical machine can have single or multiple virtual machines (VMs) running on it. Each VM can run an OS entirely different from the OS of physical machine or from that of other VMs. OS and its instances running on a VM is called 'guest operating system' and the one running on the physical machine is called 'host operating system'.

Virtual Machine Monitor (VMM) or Hypervisor is responsible for interactions between the physical and virtual machine; it controls the allocations and access of the physical machine's resources, such as CPU, Memory, HDD, NIC etc. to the VM(s). With Hypervisor, particular set of resources of the physical machine can be allocated and given access to the VM and thereby isolating the VM from consuming an excessive proportion of the resources of the physical machine. Sharing of the resources, e.g., files on the host machine is also possible. [47]

Virtualization can be implemented in different forms but, in particular, the one which has gained popularity over the others is 'full virtualization'. In this type of virtualization, most of the physical machine's hardware interfaces are replicated by the

hypervisor for the virtual machine's OS and the applications running on it. Thus, the guest OS sees it as its own hardware, which includes, but are not limited to CPU, HDD, memory, NIC cards, I/O peripherals.

Full virtualization can be implemented in two ways, which are bare-metal (native) virtualization and hosted virtualization. Figure 3-6 [47], shows their differences.

Bare-metal or native Virtualization: In this architecture of virtualization, there is no host OS present and the hypervisor is made to run directly on the top of the hardware and can even be integrated with the firmware of the physical machine.

Server Virtualization is an example of bare-metal or native virtualization. Server virtualization can be implemented in organizations running multiple servers to provide service or process information. As the servers are not utilized to their full capacity and most of the times are underutilized, running multiple instances of servers side by side can prove to be an effective solution, as can be seen in Figure 3-6 (left hand side) where the hypervisor runs multiple guest operating systems side by side without any host OS.

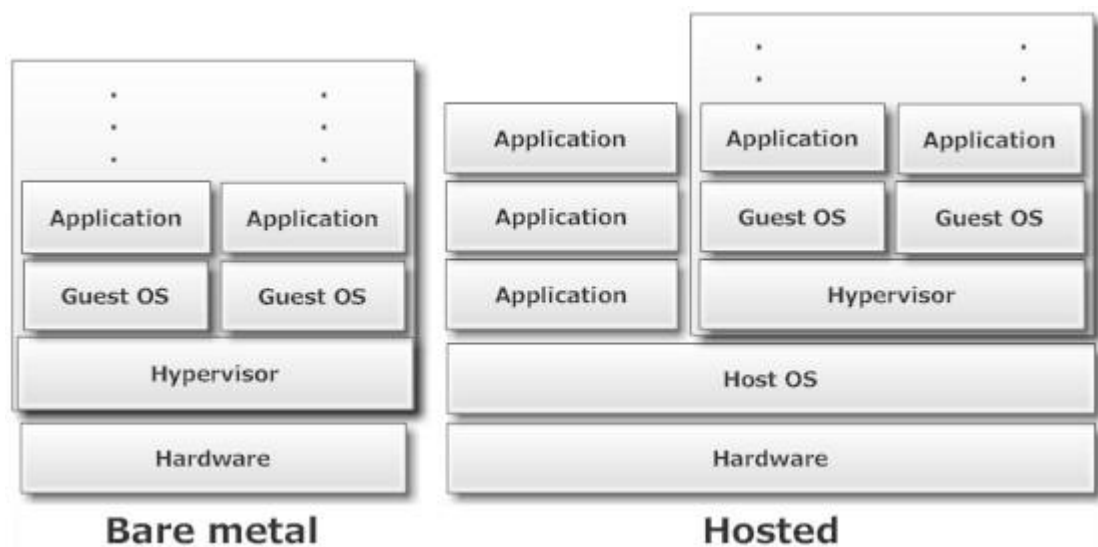


Figure 3-6: Bare metal and Hosted Virtualization Technologies

Hosted Virtualization: In hosted virtualization, hypervisor runs on the top of host operating system. This architecture of virtualization facilitates the users to run applications along the side on both the physical machine OS as well as on the VM's OS.

Desktop Virtualization is an example of hosted virtualization. Such virtualization technique can find its best application among the end users, who wish to have two OSs

without spending much on separate hardware for the OS. Another common reason to go for desktop virtualization is to have an OS that supports applications that run on that particular OS. Handling of the multiple OS on a single physical machine should not be an issue, as the technology has advanced so fast that, computers have enough processing and memory power to run multiple VMs on a single desktop, right hand side of Figure 3-6 shows the hosted virtualization technology concept implementation where host OS runs hypervisor which in turn runs multiple instances of the VMs or the guest OSs.

3.2.1 Benefits of virtualization

As mentioned in the previous section, how organizations/companies and users can benefit from virtualization, its benefits are not limited to these only. Other benefits or applications of virtualization are: [49]

- Companies/service providers that run multiple dedicated servers can benefit from the virtualization most. As the dedicated servers in most of the cases are underutilized, running two or more servers on the same physical machine can save the resources of that organization along with manageability and scalability.
- It is easier to scale the network with virtualization, as a single physical machine hardware can support the running of multiple VMs on it, rather than implementing dedicated machines (servers) in the network which are in most of the cases are underutilized.
- VMs are easy to manage, i.e., they can be easily installed (created) in an already up and running working environment, they can also be removed if there is a fault (bug/Trojan/virus) in the VM without affecting the whole network or even other VMs running on the same physical machine. Since the VM (guest OS) is a file on the physical machine, it is easy to backup and restore the entire VM.
- VMs do not care about the hardware on which they are running, as long as that hardware supports VMs. This feature of virtualization can be exploited to run legacy applications that require a specific OS, and new hardware does not support running those operating systems.
- VMs provide a safer and cost effective testing environment. An isolated virtual environment can be created to run/test un-trusted applications that have the potential to harm the OS file system or other network devices.

3.2.2 Virtual Machine Softwares

VirtualBox: Virtual box was designed by the German company ‘Innotek’ but later on was bought by ‘Sun’ and now it is under the ownership of Oracle and is known as ‘Oracle VM VirtualBox’.

VirtualBox is a "type 2" hypervisor, i.e., it works on an existing OS, designed for the x86 based systems and supports multiple operating system platforms and their flavors such as, Windows, Linux, Mac OS X and Solaris. VirtualBox supports multiple instances of virtual machines on to which the guest OS can be installed and run on the host OS. VirtualBox offers features such as [50]:

- Portability - It is easy to transfer a VM from one host to another.
- Software virtualization – Can make VirtualBox run on old desktop machines that do not have latest processors like Intel VT or AMD-V.
- Guest additions – Additional packages that make guest OSs run smoother and provide features such as shared folders, 3D virtualization, seamless windows.
- Hardware support and remote machine display are among many others.

VMWare: VMWare is another software package that exploits the virtualization techniques to run virtual machine instances in bare-metal or hosted OS modes. Founded in 1998, their first product VMWare Workstation is a ‘type 2’ hypervisor, launched in 1999, became a success. It ran on x86 platform and supported installation of guest OSs such as Windows, Linux, BSD and Solaris. VMware also has the US patent for the techniques that they explored in the virtualization world.

VMWare also introduced their server version, which required a host OS, but later they improved their product and launched the first bare-metal technology supported server with the ‘ESX’ series. VMWare Company also acquired many other small startup companies and now they have a product line that can deal from desktop virtualization to full data center server virtualization. [49] [51]

KVM (Kernel-based Virtual Machine), QEMU (Quick Emulator), Microsoft Virtual PC, Microsoft Hyper-V are among other solutions/software and emulators that exploit virtualization technologies to install and operate other OSs on host machines.

3.3 IP Addressing Scheme

This section deals with the concept of the IP addressing schemes. Internet Protocol (IP) is a protocol that works in conjunction with a suite of protocols for communication over the Internet or over a network. This is referred as TCP/IP for Transmission Control Protocol and Internet Protocol stack. IANA (Internet Assigned Numbers Authority) holds the authority to assign and manage IP addresses over the globe.

IP address is the address that depicts the location of a user on the network or Internet, where that user can be reached. IP address format has evolved over a period of time and current versions of IP addresses being used over the Internet are IPv4 and IPv6. IPv6 is a new concept and was designed and developed as backup of IPv4, i.e., once they run out of the address space devices over the Internet will use IPv6. This section does not deal with IPv6.

IPv4 address is a 32-bit long binary number represented in dotted decimal format. This 32-bit number is divided into four sets of 8-bit binary (a byte or an octet) numbers. These binary 8-bits are represented in decimal form (i.e., from 0-255) and thus the name 'dotted decimal'. Network Interface Card (NIC) is the one who is assigned this address. It is worth mentioning that a NIC has two sets of addresses, one being the software address, i.e., IP address and other is hardware address, which is hard coded into NIC cards and is used over the Local Area Network (LAN) for finding hosts [52].

An Example would be, in binary "01010010.10000010. 00010111.01001101" and the same IP address in dotted decimal is written as, "82.130.23.77".

An IP address consists of two parts: the network address part and the host address part. Network address is the address that a router uses to route packets to a remote destination. This part of the address is same for all the devices in that particular organization, area or region. However the host address part is the one which identifies a particular device on a network. It is a unique address, i.e., no two devices share the same host address on a network.

Based on the network address part, IP address space is classified into 5 classes, i.e., A, B, C, D and E. Class D is reserved for the Multicast addresses and E is reserved for the scientific research purposes. Purpose of this classification is to allocate the IP address

space as per one's need of IP addresses, e.g., large organization like Google or IBM may need Class A address; one class A address yields 16,774,215 unique host addresses.

In class 'A' networks, the first bit of the network address is always '0' and the network address is 8-bits long. In class B, the first two bits of the network address are '1 0' and network address expands to first 16-bits. For class C, the first 3-bits of network address are '1 1 0' and network address further expands to first 3 octets. For class D, the network address starts with bits '1 1 1 0'.

IP address class separation is summarized in Table 3-2 [52] [53]. Here 'N' represents the network part of the address, whereas 'H' represents the host part of the addresses in an IP address. Table 3-2 also represents network numbers in dotted decimal format. IP addresses having the network address starting from 1-126, fall in class A of IP addressing scheme. Similarly, from 128-191 in class B and 192-223 in class C.

Table 3-2: IP Address Classes

IP class	Address Classification	Network Address	No. of hosts per Network
	Criteria	Number	Address Number
Class A	N.H.H.H	1-126	16M
Class B	N.N.H.H	128-191	64K
Class C	N.N.N.H	192-223	254

Over the last 2 decades, the use of Internet has grown very rapidly making the IPv4 address space almost exhaust. To overcome this problem IANA introduced reserved block of IP addresses from class A, B and C known as Private IP addresses. Those could be used within a network. Such addresses are required and utilized by the network devices that do not need Internet connectivity but they need to communicate with other devices on the same network, e.g., internal network of a large enterprise or a LAN network. Those address blocks are summarized in Table 3-3. [54]

Table 3-3: IP Address Blocks For IP Classes

Class A	10.0.0.0 - 10.255.255.255
Class B	172.16.0.0 - 172.31.255.255
Class C	192.168.0.0- 192.168.255.255

3.3.1 Classless Inter-Domain Routing (CIDR)

CIDR [55] was introduced to overcome the problem of class B IP address exhaustion and route aggregation, i.e., to facilitate service providers to summarize their routes in a single entry for continuous blocks of assigned IP addresses.

CIDR introduces the new concept of IP prefix, written as a forward slash sign ‘/’ in the end of the IP address instead of the mask. The prefix number (defined in RFC 1878 [56]) represents the length (numbers) of bits that belong to network portion of the address. Table 3-4 shows the prefix notations for private IP addresses block.

Table 3-4: IP Prefix Notation for Private IP Addresses Block

Class A	10.0.0.0 - 10.255.255.255	(10/8 prefix)
Class B	172.16.0.0 - 172.31.255.255	(172.16/12 prefix)
Class C	192.168.0.0- 192.168.255.255	(192.168/16 prefix)

CIDR in simple terms can be explained as use of the mask bits to identify the destination network rather the class of IP address, and thus the name classless. With this new concept, changes were required in the routing protocols, and in the firmware of routers, i.e., the interpretation was required for the subnet mask bits representing a reachable network. [55]

3.3.2 Subnetting

Private addresses solved the IPv4 exhaustion issue, but only for the time being. With the speed network and organizations were using growing they needed another solution. IETF came up with another solution known as subnetting. Subnetting is a way of further dividing one’s network into a bunch of smaller networks [52]. This is done by taking the bits from the host part of the address and including then in network part. After subnetting, network devices should be able to understand the address, i.e., which address is a subnetted address. This is done by assigning a subnet mask. Subnet mask makes the machine to distinguish the network part and the host part of an address. Default subnet masks for class A is 255.0.0.0, for class B is 255.255.0.0 and for class C is 255.255.255.0.

Other reasons for subnetting a network could be simplified management of networks, reduced traffic and traffic optimized routing. Subnetting can be either classfull or classless known as VLSM short for Variable Length Subnet Masks [56].

A classful subnetted network has same subnet mask for all the hosts in the network. Whereas in classless subnetting hosts can use a different subnet mask. An example of class C subnetting (classfull and classless) is as under with network address 192.168.1.0/24 or mask 255.255.255.0

If we want to subnet (further divide) it to have 8 networks and each having 30 hosts) we could use classless subnetting. Network addresses would be like this, 192.168.1.0/27, 192.168.1.32/27, 192.168.1.64/27, 192.168.1.96/27 ... so on till 192.168.1.224/27.

But if the requirement of users is different per subnet, i.e., variable number of hosts per network, then using classful subnetting would result in wastage of IP addresses. This problem could be resolved with classless or VLSM subnetting.

192.168.1.0/27 would yield 30 hosts. Next 192.168.1.32/28 would yield 14 hosts for this subnet. 192.168.1.48/30 would yield 2 useable IP addresses, suitable on links where there are no hosts to be connected, such as serial links between routers.

3.4 Virtual LAN

VLAN for virtual Local Area Network is the IEEE 802.1Q [57] standard, developed by the IEEE 802.1 working group. This section follows the reference [58]. As the name suggests Virtual LAN, the concept is same as that of a Physical LAN except that the groups of devices (computers) are logically grouped together, so that they appear to be in the same LAN. Manageable layer-2 device, i.e., switch is required to create VLANs.

A layer-2 network is a large broadcast domain for a given network and switches breakup only the collision domain. So, if someone wanted to reduce that broadcast domain, routers had to be introduced in that LAN. But, the issue can be solved with the help of VLAN. Same broadcast domain will be reduced to the number of devices that are expected to be in that particular LAN.

A VLAN is created by administratively defining switch ports into a logical group. Devices (nodes) that connect to these ports are able to communicate with each other, as

they are logically grouped in one broadcast domain. VLAN makes use of the tags (VLAN ID) to achieve this. These tags are sent along with the frame and are removed by the switch before sending to the destination station. VLAN is not restricted by the geographical limitations, i.e., set of computers or network resources belonging to different geographical regions can be tied into same logical group.

Since VLAN limits the broadcast domain to the devices that are connected logically, those devices cannot send packets to other VLANs. In order for the VLANs to communicate with each other a layer-3 device, i.e., a router or an L-3 Switch should be present.

A VLAN should not be confused with an IP subnet. An IP subnet which is also a group of network devices connected together can be further fabricated into VLANs as per need and requirement. However, the reverse is not possible. [58]

Following Figure 3-7 [59], depicts the configuration of the two VLANs in a single switch.

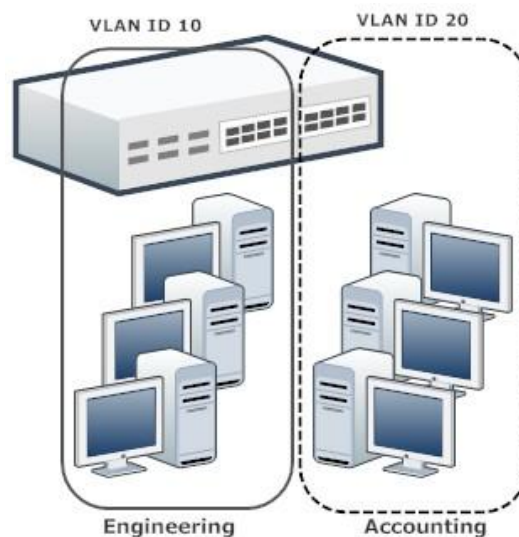


Figure 3-7: Ports Separated Into VLANs In Switch

3.5 IP Address Allocation

In an IP based network, assigning the IP address to the hosts can be done in one of the two ways, statically or dynamically.

Static IP allocation: In this method of allocation, IP addresses are configured statically on the hosts in a given network. Assigning static IPs also means IPs are allocated

permanently and they do not need to be changed unless there is a change in the network topology. This works well when there are only few hosts to manage for a network administrator or assigning the IP addresses to network devices like routers and switches who's IPs should remain static/permanent.

Dynamic IP allocation: Dynamic allocation of the IP addresses is opposite to static allocation, i.e., IP address is allocated out of a pool/block of addresses in a random fashion. When a host becomes online, i.e., connects to the network, it sends a request to receive the IP address. Upon seeing that request, server responsible for allocating IP addresses, sends out a reply with the IP address assigned to that host. That IP address is valid for a limited period of time. In other words, the server leases the IP address to the requesting host. It must be kept in mind that host setting should be configured to receive IPs automatically over the network.

The protocol that makes all this possible is known as Dynamic Host Configuration Protocol or DHCP and the server is known as the DHCP server.

3.5.1 Dynamic Host Configuration Protocol (DHCP)

DHCP, defined in the RFC-2131 [60], is an extension or rather based on the same message format as BOOTP (Bootstrap) protocol, but there are differences between them. In order to get the IP address from BOOTP, the client's physical address must be entered in the BOOTP table first by the network administrator, in contrast DHCP does not require the hardware address, it detects the host dynamically. Also, the host can download an operating system image with BOOTP, to boot from, but DHCP does not have this feature. [52]

DHCP does not only assign IP addresses to the hosts, it can also provide other configuration parameters to the hosts such as, default gateway, DNS (Domain Name Service) IP address and many others parameters that a host may require in a certain work environment.

There are three ways in which a DHCP server could perform its operations depending on what the network administrator is interested in. These are the following:

- Automatic Allocation: A permanent IP address is allocated to the host.

- **Dynamic Allocation:** This type of allocation has been explained earlier, i.e., IP address is allocated or leased for a certain period of time from pool of IP addresses. IP address tenure can be renewed after 50% of the time has elapsed by the same host. But, if not in use by the host, IP can be released and is allocated to another host.
- **Manual Allocation:** In this allocation the network administrator allocates the IP address to the client and DHCP is used to assign that IP address, once that host becomes online or connects to the network.

Working Principle: In order to get an IP address from the DHCP Server, the host (client) broadcasts a DHCPDISCOVER message to reach the DHCP Server. DHCP Server in return, sends a unicast message with DHCPOFFER. This message contains all the necessary parameters along with the IP address for the client to operate. If the client accepts the offer, it will broadcast out a DHCPREQUEST message to formally request all the parameters assigned to it. DHCP confirms that request with a DHCPACK unicast message to let the client know, it has been assigned an IP address along with all the parameters. Figure 3-8 [61], shows the transaction of messages between a DHCP server and client.

However, if there are multiple DHCP servers on the segment (network), the client might accept the first offer that it receives. Also while making the DHCPOFFER, the server reserves the IP address for the client, even though it is not sure, if the client will accept the offer or not. IP is reserved until it receives the reply from client. [60][62]

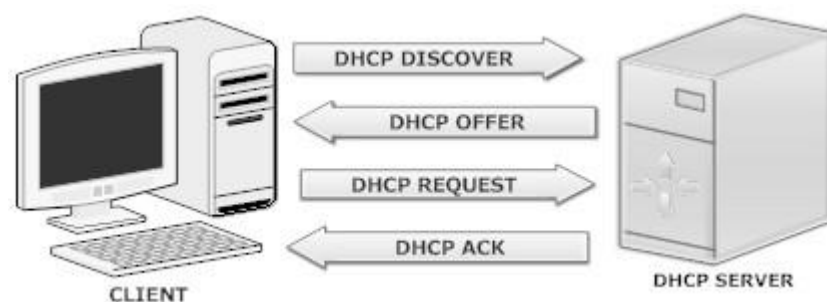


Figure 3-8: DHCP Messages Sent Between Server and Client

3.6 Domain Name System

This section is a brief description of the DNS for Domain Name System and follows the reference [63]. There are many RFCs that describe in detail the working of DNS.

As discussed in the previous section about IP addresses that every host on the Internet has an IP address to which it responds to, when someone else from Internet want to connect or communicate with him. Remembering the IP addresses of every host with one should communicate becomes very difficult and the solution for this problem came in the form of name servers.

A name server in its simplest definition is the server that resolves (maps) the names into IP addresses and vice versa. Thus, a host on the network queries a name server to resolve a particular name into its IP address on which it can be reached. This solution works as long as the names in the database are manageable but unfortunately with the speed the networks are growing name servers faced the problems, such as organization of entries in the database, scalability, and management of the servers [63]. A more feasible solution was required and was presented in the form of Domain Name System (DNS).

DNS uses the hierarchical name approach, i.e., like a tree to address the problems mentioned above. Root node is places at the top of the tree and is followed by the TLD (Top level domain) then SLD (Second level domain) and so. Each level is separated by a dot. It is worth to mention here that, a silent dot is used to represent the root for most of the time but it might be very important in some of the cases [63]. As shown in Figure 3-9 [64] TLDs can be categorized into following categories:

- Generic Top-Level Domains (gTLD): For example, .com, .edu, .net, .org, etc.
- Country Code Top-Level Domains (ccTLD): For example, .fi, .ca, .pk, etc.

A domain name, e.g., oursips.lab.com is basically a combination of TLD and SLD names and is written from left to right, i.e., highest level goes in the left and lowest to the right, separated by a dot. In this example, '.com' is the TLD and 'lab' is the sub-domain of 'com' and 'oursips' is the hostname (in this case a SIP server). 'Oursips.lab' belongs to the TLD 'com'.

When a client request to a name comes and the name server is not able to resolve it, it sends the query to root server which replies with a referral to the TLD who could resolve the query. TLD upon query also returns the referral to the name server who has the final information about the host or the server.

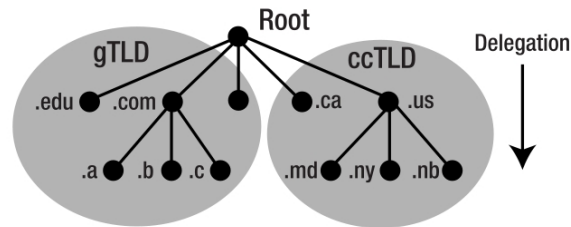


Figure 3-9: DNS Hierarchical Structure

A name server may have a single or multiple entries for domain(s). Network entities in that domain or zone, i.e., hosts, servers running services such as SIP or mail server etc. along with the global properties of that domain are saved on the name server in files called the Zone files [63]. Zone file data is defined in the form of Resource Records or RRs and some of the common RRs like SOA, NS, MX and SRV will be briefly defined later in this section. Thus, the zone file translates/maps such services and/or hosts to a particular IP address so that the DNS software can reply when inquired by a host on the network.

Common RRs in a zone file are: [65]

- **SOA** - Start of Authority (SOA) is the data that defines the zone authority and is required in all zone files.
- **A** – These records in zone files are used to identify the host IP addresses within a zone.
- **NS** – NS defines the authoritative name server for the domain.
- **PTR** – PTR for pointer is used to refer to another part of the domain name space.
- **SRV** – Records information about well known network services of domain.
- **MX** - These records are used to control email servers in the domain.
- **NAPTR** – NAPTR (Name Authority Pointer) records contain information on what kind of transport protocol will be used for the service mentioned in SRV records.

In environments where redundancy and efficiency is required, i.e., to reduce the query times and load balancing, zone file(s) are kept on multiple DNS servers. The original zone file is kept on the primary server who has the authority to change/modify the zone file. Secondary server holds the copy of the file and is regularly updated via the zone transfer process.

3.7 Routing in IP networks

This section deals with IP routing, i.e., how the packets are moved across the network to another network between the hosts and what routing protocols make this happen.

IP routing or just routing is the process of transferring IP packets from one host across the internetwork to another host on a different network. To an untrained eye routing may seem to be same as the switching but, it is not. Switching or bridging is a layer-2 process and it deals with transfer of the frames within the network. Whereas, routing is a layer 3 process and involves a router (layer-3 device) or series of router to deliver the packet from source to destination.

A router is always aware of its directly connected networks, i.e., it knows how to route packets to that network. However, for other networks the router makes its forwarding decisions based on the routing table. A routing table is a list of entries (routes) that a router learns either from neighboring routers or the routes are entered directly by the administrator. It is the job of a router to calculate the best route for the packets through the network to the destination unless the routes are static.

When a router receives a packet for routing, it sees its destination address. If the network is directly connected then it routes the packet through that interface. If the destination network is not directly connected then it looks into its routing table, and if the network can be reached through neighboring routers then it forwards the packet to them for delivery. Otherwise the packet is dropped.

Routes in a routing table can be categorized into two, static routes and dynamic routes [52].

Static routes: Routes which are defined into the routing table by the network administrator himself are static routes. Like static IP addresses, this solution works well when there are a few networks to administrate and the network links are stable. If network links are not stable, i.e., if they fluctuate, then using static routes is not feasible.

Dynamic routes: Dynamic routes are the result of using a routing protocol which is responsible for the routes in the routing table. The protocol running on a router communicates with other routers running the same protocol and they exchange their information regarding the networks known to them. Once information is exchanged

they update their routing tables. If there is any change in topology, e.g., addition of a new route or route failure, information is passed on through the protocol responsible on and all routers so they can update their routing tables. The most common dynamic routing protocols used in networks are RIP and OSPF. It should be noted that RIP and OSPF both are intra-domain routing protocols or Interior Gateway Protocols, i.e., they are used within an autonomous system. An autonomous system (AS) is controlled by single administrative entity, comprising a network or set of networks. AS region could be a country, large organization or an ISP (Internet Service Provider). Inter domain routing protocols or Exterior Gateway Protocols that work between ASs are outside the scope of our work.

3.7.1 Routing Information Protocol (RIP)

Dynamic routing protocols are categorized by the algorithm they are using for the exchange of the routes. These algorithms are known as Distance Vector algorithm and Link State algorithm. RIP protocol is a distance vector protocol. In the distance vector algorithm the distance is sole criteria of choosing the best (shortest) path to the remote network. Here, distance is counted in terms of the hops, i.e., after how many hops (routers) a network can be reached. However, if there are multiple links to the same remote network with the same number of hops, traffic is load-balanced between those links. [66]

RIP protocol makes the routers exchange their complete routing table entries with their directly connected neighbor routers, i.e., the routers know the network only from their neighbor's point of view and do not have the complete topology of the network. This is also called 'routing by rumor' as whatever information (routing table) it receives from other routers, it just believes on it. This can also cause routing loops. Once the routing tables from the neighbors are exchanged, they combine the information with their own. This process is called converging and during this process it is possible that no data may be exchanged [52]. Routers using the RIP routing protocol exchange their tables every 30 seconds. RIP uses different timers to update, validate and remove entries from the routing table. [66]

RIP can cause use routing loops in the network. There are different ways in which it can be avoided, such as maximum hop count is limited to 15 and after that the route is declared unreachable. Others are split horizon, route poisoning and hold-downs. [66]

Detailed information of the RIP versions 1 and 2 can be found from RFCs 1058 and 2453, respectively.

3.7.2 Open Shortest Path First (OSPF)

OSPF [67] in itself is a vast topic and cannot be covered fully here. A brief description of OSPF version 2 will be presented here. OSPF version 3 is used with IPv6 Protocol. This section follows the reference [67].

OSPF is a widely accepted and adopted dynamic routing protocol running in medium to large size networks. As the name is, shortest path first (SPF), the protocol finds the best shortest path to get to the destination. The word ‘open’ implies that it is an open standard and is defined in RFC 2328. OSPF is a link state dynamic routing protocol based on Dijkstra’s algorithm.

In the link state algorithm, routers share the state information (i.e., IP addresses, mask, status etc.) of their directly connected links. This information is spread with the help of flooding protocol (Link State Advertisements or LSAs) with every router in the network such that, every router in the network has the exact same topology of the network [68]. Thus routers executing OSPF receive updates in the form of link state updates and build their own routing table based on the information received. This feature of OSPF (along with others) helps the network to converge faster than the DV (distance vector) routing protocols.

OSPF maintains the topology database in a hierarchical manner. This is done by using the LSA packets. Different types of LSAs are originated by routers containing different topology information. Upon receiving these LSAs each router creates databases to get an accurate topology of the network.

On a multi-access network routers running OSPF elect a designated router (DR) and a backup of DR, i.e., the so called BDR. All routers on that multi-access create an adjacency with DR and BDR and forward their link information to DR and BDR. In return, it is the function of the DR to forward that information through LSAs to all other routers in that segment. BDR takes over when DR becomes unavailable.

Unlike RIP, OSPF takes the hierarchical approach, i.e., it divides the network into areas. Though for smaller networks it not necessary to have multiple areas other than Area-0.

In this hierarchy, Area-0 is the backbone area and all other areas must connect to this Area 0. Routers which connect other areas to Area-0 are called Area Border Routers (ABRs). Different areas communicate with each other through the backbone area. Each area sums up their routes and sends to Area0 which also summarizes the routes to all connected areas [68].

OSPF uses the concept of Router ID, which serves as the name or label for that router in the network. Router ID is chosen as the highest loopback IP address among the others configured on the router. If no loopback addresses are configured then the highest active physical IP gets to be Router ID. However, if nothing is configured, OSPF process remains non functional.

3.8 Summary

This chapter discussed the concepts, protocols and technologies such as, VoIP, SIP and virtualization that can be utilized in any given network to facilitate both the users and the network. Implementations of private IP addressing schemes with CIDR approach and services like, DHCP and DNS are not new to networks. However, their integration in the network that is based on virtual machines and destined to provide VoIP service for its clients in different VLANs can prove to be a challenge. This chapter provided the theoretical basis and an initial practical approach in solving the problem like, which VoIP protocol can fulfill the network requirement, which SIP server should be selected in terms of performance, features and user accommodation.

4 Test Network Planning and Configuration

In any network, planning is the crucial step, it is the sketch that tells us the basic information of the network, such as what kind of topology is going to be used to connect devices, network's security features, what kind of and how many devices (servers, routers, switches) are there and lastly how many clients (hosts) can this network support.

Test networks are built to have an idea of how and what kind of infrastructure should a real network have, i.e., what kind of network equipment (servers, routers etc) will be required and how should they be configured. This chapter focuses on building such a test network for the 'software defined radio test-bed'.

4.1 Test Network Design

Network design/topology is of significant important in any network before implementation and configuration of the network equipment. It is important to lay out the design of the network, so that one must have the full understanding how the network should look like and how will this network behave once up and running. Also this makes it easy to troubleshoot when and if anything goes wrong.

Networks are designed according to some specifications, i.e., to meet certain criteria specified beforehand. This test network was required to support voice capabilities and to assign dynamic IPs to the clients (hosts) that connect to it. The network should additionally support the name mappings, as well as mobility for the clients. Most importantly, due to the lack of resources it was required to have this network with minimal use of network equipment.

To minimize the use of network resources, technology known as virtualization (explained in previous chapter) will be taken advantage of. There are many software implantations available and with the help of those multiple VMs can be run at the same time, each running their own instances of code.

Figure 4-1 is the block diagram (topology) of the test network. It shows two base stations (BS1 and BS2) supporting different services for the clients and they are connected to a SIP server, which provides the voice capabilities for the network. Since

this represents the test core network, i.e., the wired version of the network, base stations are connected to a manageable switch, to which clients also connect.

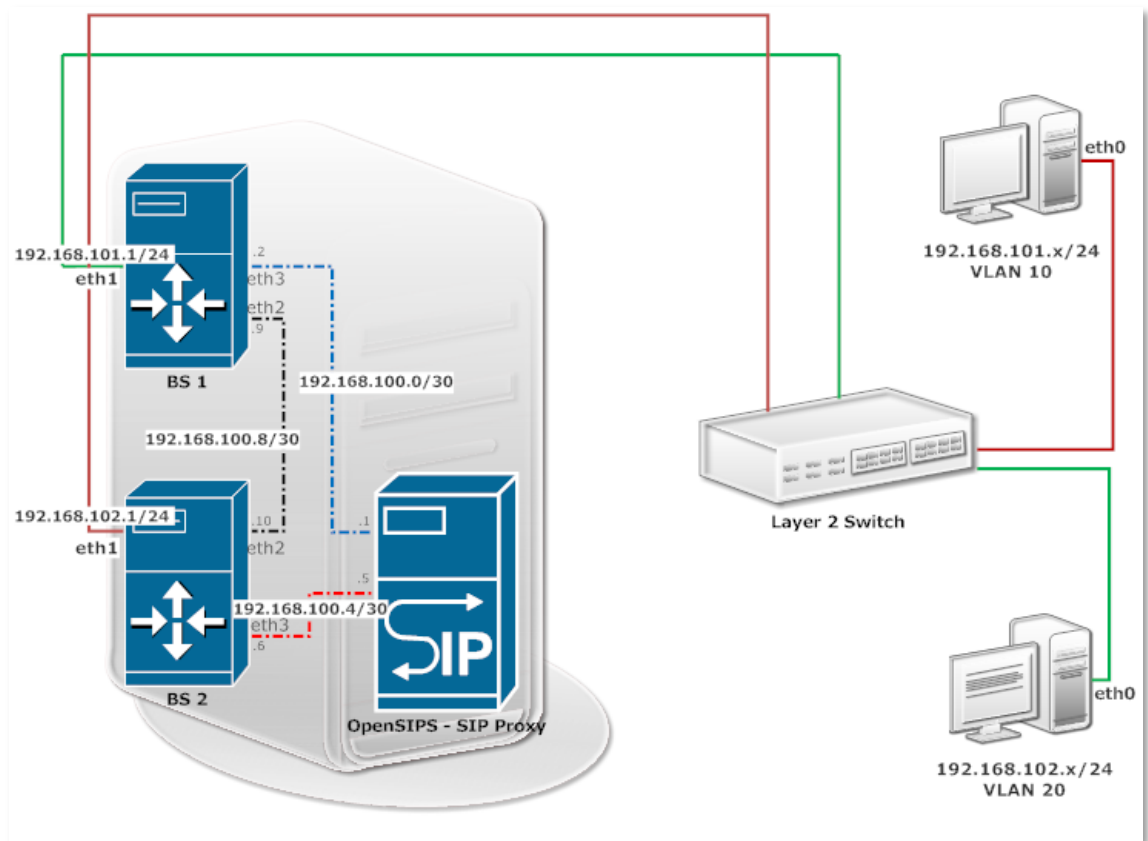


Figure 4-1: Test Network Topology

Setup of this lab test network is such that virtualization software ‘VirtualBox’ is installed on a host machine running Linux OS (Ubuntu 12.04). With the help of VirtualBox, three separate instances of virtual machines namely BS1, BS2 and OpenSIPS are installed and configured on host machine. For the simplicity of the design of the test network and scarcity of the resources, Linux OS Debian was installed on all three VMs.

In order for the client to get connected to the network, i.e., obtain an IP address along with the address to name mapping service, DHCP and DNS servers are installed and configured on the BS1 and BS2 virtual machines. Whereas the SIP server OpensIPS, has its setup configured on the third VM.

Also, as both base-stations (BS1 and BS2) represent two separate networks, their separation is achieved with the use of a layer-2 manageable switch having VLAN capability. A manageable switch works in conjunction with the host machine’s NIC

cards. Physical Linux machine have 3 Physical NIC cards, one connected to the Internet whereas the other two are bridged to the VMs. Users connect to the switch to gain access to the network and its services.

4.2 VirtualBox Settings

As explained in the previous section, the objective was to build this test network with limited amount of resources, thus making use of a well exploited technology, i.e., virtualization was inevitable. Among other softwares mentioned in the chapter 2, Virtualbox was simple to manage VMs and easy to use.

Because of its powerful features and flexibility of use, Debian OS was chosen to support BS1, BS2 and SIP server's features.

After installing the Virtualbox on the Linux machine and Debian OSs for BS1, BS2 and SIP server, each VM should be configured properly, so it can make use of the physical machine's resources, such as the NIC cards.

NIC cards of the VMs should be either in NATed, bridged or internal mode. NATed mode is used so that the virtual machine can access Internet. Note that this interface will be removed in the end, once all the configurations are done. Whereas, bridged mode is used to give access to the physical machine's interface (only for BS1 and BS2). Lastly, internal mode is used to create internal networks between the virtual machines. Figure 4-2 shows one such setting for BS1, whereas complete settings for all the VMs are shown in Table 4-1 after that.

Table 4-1: NIC card configurations on VirtualBox

Virtual Machine	Network Adapter1	Network Adapter2	Network Adapter3	Network Adapter4
BS1	Optional NAT	Bridged to eth0	Internal Network	Internal Network
BS2	Optional NAT	Bridged to eth2	Internal Network	Internal Network
SIP Server	Optional NAT	Internal Network	Internal Network	

4.3 IP Addressing Scheme

IP addressing scheme used in the test network should also replicate the IP addressing scheme that will be used in the real network.

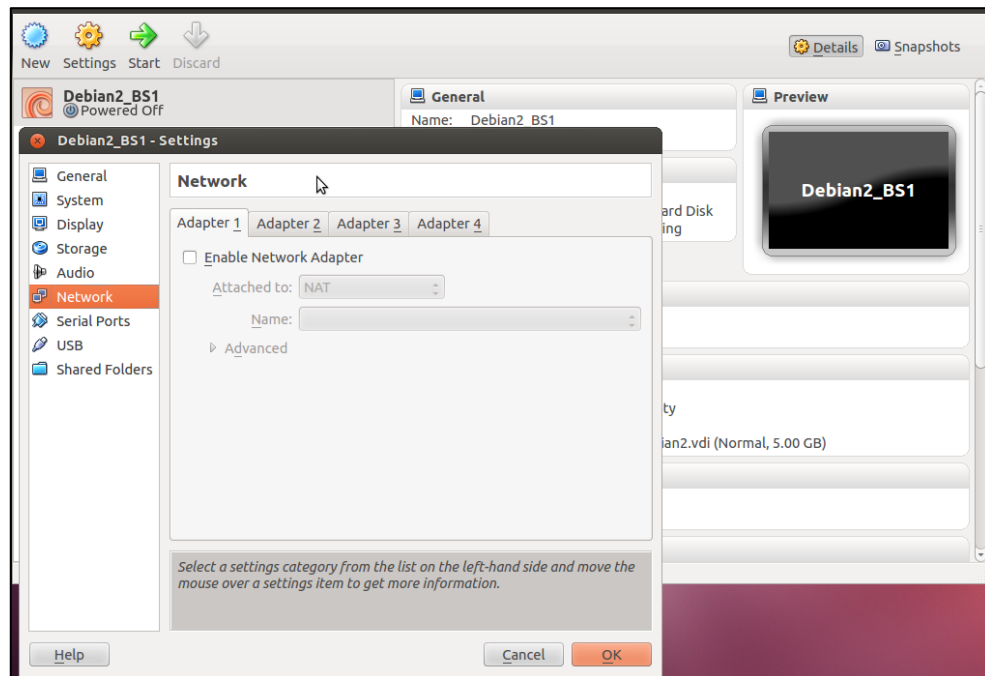


Figure 4-2: Network Adapter Setting For BS1

In the previous chapter, we discussed about the different schemes by which a set of IP addresses can be allocated in a particular fashion. In this test network, it is required that the client station should get an IP address from the base-station to whom they are connected to. Also, the base stations need to be connected to each other and the base stations are connected to the SIP server to provide the VoIP services for the clients connected.

Since this a test work, private addressing scheme is used. Private addresses are the addresses that do not need to be routed on the Internet and are only used in Intranet infrastructure setups. The IP addressing scheme that fitted the network design was to allocate a whole subnet of class C addresses space to one particular base station, so that clients can be recognized as to which base station they are connected. Also, with the use of class C subnets it was also made sure that the number of clients that can connect to a particular base station are well enough. One class C subnet can allocate 254 addresses per network.

As mentioned before, base stations need to be interconnected with each other and all the base stations also need to be connected to the SIP server. Allocating the whole Class C address subnet block would have wasted a lot of IP addresses. CIDR and subnetting technique was also applied to solve the IP address space wastage problem. Since the

requirement is to have only 2 IP addresses from a same subnet to connect two machines so the use of '/30' or '255.255.255.252' was most appropriate.

A 255.255.255.252 subnet yields 4 IP addresses. First IP address (e.g., 00) is always the network address, i.e., is used to identify the network. Network address is followed by two IP addresses (e.g., 01 and 02) that can be allocated to the NIC cards of a host machines. Last IP address (e.g., 03) of a /30 subnet is the broadcast address of the particular subnet.

Thus the IP addresses used were (also shown in Figure 4-1 of test network),

- 192.168.101.0/24 and 192.168.102.0/24 were used for the clients to connect to the base stations. The clients can get IPs in the mentioned range, i.e., from 192.168.101.1 to 192.168.101.254. Likewise for the '102' network.
- 192.168.100.0/30 was used to connect base station 1 to SIP server.
- 192.168.100.4/30 was used to connect base station 2 to SIP server.
- 192.168.100.8/30 was used to connect base station 1 and base station 2.

These IP addresses need to be assigned to on appropriate interfaces. A sample configuration of 'BS 1' is shown in Figure 4-3. It is worth mentioning here, that editor used for this purpose was 'gedit'. However many other editors are available to perform the same function.

```
gedit /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo eth1
iface lo inet loopback

# The primary network interface

iface eth1 inet static
address 192.168.101.1
netmask 255.255.255.0
network 192.168.101.0
broadcast 192.168.101.255
```

Figure 4-3: Sample Interface Configuration of BS-1

4.4 Routing Scheme for Test Network

Routing schemes in networks are chosen to match the design of the network, i.e., how big a network is and what kind of routing protocol will facilitate in achieving a fast convergence, along with minimal routing overhead traffic in the network.

As this is a test network and involves only two base stations that handle traffic, and connect with a SIP server, managing routes can be done manually. Thus, static routing seemed to be the best solution for this kind of setup rather than a dynamic routing.

Advantage of using the static routing over the dynamic routing is that there is no overhead in the network traffic and also it takes a lot of processing power to calculate the paths. Whereas in static routing base-stations will know exactly where to route the packets for a particular destination.

Figure 4-4 is the sample configuration of setting up a static route in BS-1. First route defines the route how the clients from BS-1 can reach the clients on BS-2. Second route defines the route for the BS-1 network clients to reach to the SIP server for voice service.

```
gedit /etc/network/interfaces

up route add -net 192.168.102.0 netmask 255.255.255.0 gw
192.168.100.10
down route del -net 192.168.102.0 netmask 255.255.255.0 gw
192.168.100.10

up route add -net 192.168.100.0 netmask 255.255.255.252 gw
192.168.100.1
down route del -net 192.168.100.0 netmask 255.255.255.252 gw
192.168.100.1
```

Figure 4-4: Sample Static Route Configuration of BS-1

4.5 Configuring DHCP Server

In order to connect to the network, a client (host) requires an IP address so it can communicate with other hosts on the same network or with others (on Internet) if allowed to. Network administrators strategically divide the subnets of IP addresses that clients acquire either dynamically or set the IP address manually.

As discussed earlier Dynamic Host Control Protocol (DHCP) is used to dynamically allocate IP addresses.

Setting up IP addresses manually can be problematic for the clients every time they connect to network even in a test network. Thus, dynamic allocation is implemented for the convenience of both, clients as well as network administrators. This way, the binding of the IP to the user can also be tracked easily (from the security point of view).

In our test network, DHCP server is installed in the base stations, which is also the entry point of the hosts on the network. Every base station runs a separate subnet of the IP address block, so that users can be identified on location basis, i.e., from which area (base station) they are connected to the network and thereby distinguishing them from other base-station users.

Installing a DHCP server on Debian machine is done from the Internet Systems Consortium repository. Configuration files are edited to make sure that DHCP is listening on correct interfaces for DHCP requests and giving replies to. Detailed configuration is provided in Appendix D section.

Sample of the DHCP configuration from BS-1 is shown in Figure 4-5.

```
gedit /etc/dhcp/dhcpd.conf

ddns-update-style none;
option domain-name "oursips.lab.com";
option domain-name-servers 192.168.101.1;

default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

# This is a very basic subnet declaration.

subnet 192.168.101.0 netmask 255.255.255.0 {
range 192.168.101.11 192.168.101.20;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.101.255;
option routers 192.168.101.1;
}
```

Figure 4-5: Sample DHCP Server Configuration of BS-1

4.6 Configuring DNS Server

DNS is the service used in the network to provide a mechanism that translates the name into IP address and vice versa. Also, in a network (domain) there can be multiple services running such as mail service (servers), voice service and many others. Clients that connect to the network need to be pointed to the right server which is also the role of DNS server in a network. Details on DNS can be read in Chapter 3.

The network topology diagram shown in Figure 4-1 reveals that the SIP server is connected with two base stations via two different interfaces in separate subnets without any extra support. If mobility was to be supported in this test network, clients would have needed to change the SIP server's IP address in the VoIP client every time they moved from one BS to another.

In order to facilitate the roaming capabilities and to avoid manual changing of the IP address every time, the user gives the domain name of the server in the VoIP client, thereby solving the roaming issue.

Since virtualization was used to set up the whole network infrastructure, it was difficult to implement one single DNS server from which clients would have inquired about the SIP server's IP address. Thus, two separate DNS servers were installed in both of the base-stations, so that they can separately point to the same SIP server.

For the DNS server software, BIND v9 (Berkeley Internet Name Daemon version 9) was used. There are other solutions available for setting up a DNS server, but BIND allows scalability, flexibility and in the long run more security options.

As stated in previous chapter, DNS consists of a primary master server, and zone files that hold different type of records. For the test network, primary master server for the zone "lab.com" with zone file name "db.oursips" (can be anything) was created. This is done by editing the file "named.conf.local". Figure 4-6 shows the sample DNS configuration for BS-1. Detailed configuration is shown Appendix D.

A zone file created for the test network is shown in Figure 4-7. It is worth mentioning here that the test network has the SIP service in a different network domain. Thus, NAPTR and SRV should also be included in the zone file. So that server, which is

responsible for giving SIP service should be pointed out correctly in a domain, i.e., its SIP server's IP should be resolved properly.

```
gedit /etc/bind/named.conf.local

zone "lab.com" {
    type master;
    file "/etc/bind/db.oursips";
    allow-query {any;};
    notify no;

};

}
```

Figure 4-6: Sample DNS Configuration of BS-1

```
gedit /etc/bind/db.oursips

$TTL 3h
lab.com. IN SOA ns.lab.com. lok.lab.com (
    1          ; Serial
    1d         ; Refresh after 3 hours
    1d         ; Retry after 1 hour
    4w         ; Expire after 1 week
    1h )       ; Negative caching TTL of 1 hour

;
; Name servers
;
lab.com. IN NS  ns.lab.com.

;
; Addresses for the canonical names
;
localhost.lab.com.      IN A      127.0.0.1
ns.lab.com.             IN A      192.168.101.1
oursips.lab.com.        IN A      192.168.100.1

;

;
IN      NAPTR      50 50  "s" "SIP+D2U" "" _sip._udp.ns.lab.com.
;
_sip._udp.ns.lab.com.  86400 IN      SRV      0 1 5060 oursips.lab.com.
```

Figure 4-7: Sample Zone File Configuration of BS-1

4.7 VLAN Configuration

As explained in previous chapter, the VLAN concept corresponds to a LAN except that it is a method of limiting the broadcast domain of a particular set of hosts that may belong to one specific area or department, thereby reducing the one big broadcast domain into multiple smaller domains.

VLANs are created with a manageable Layer-2 Switch. Hosts in newly created VLANs interact with each other with the help of a Layer-3 device, i.e., a router or a L3 switch.

The test network required to implement the mobility of the users, i.e., roaming between two base stations. With the help of VLANs two separate broadcast domains (VLANs) were created in a switch, replicating the idea of roaming (moving) from one base station to another. HP ProCurve Switch was used to create the VLANs, VLAN10 and VLAN20 for base-station 1 and base-station 2, respectively. Ports were assigned to a particular VLAN so that when a client connects to a particular port it receives its IP from the associated base-station.

Figure 4-8 shows the VLAN configuration page in HP's ProCurve switch. Other figures showing detail configurations are included in Appendix C.

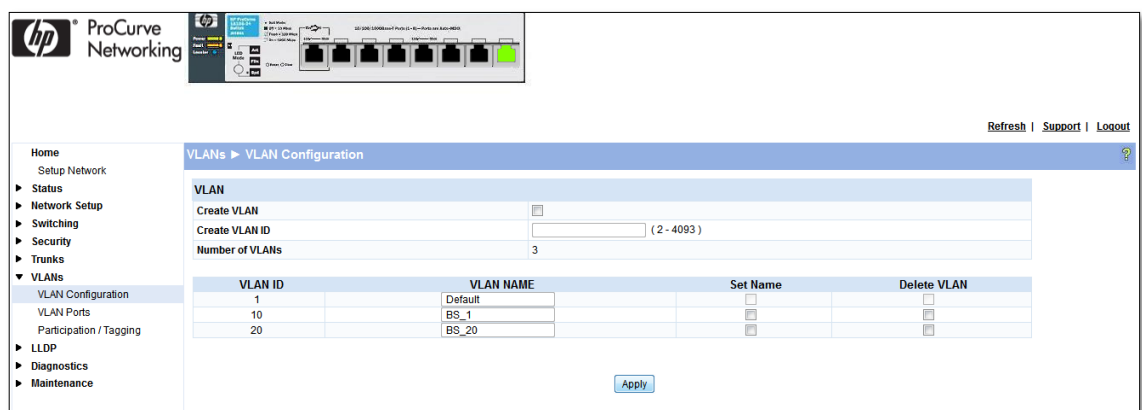


Figure 4-8: HP Switch VLAN Configuration

4.8 Configuring SIP Server

There are different open source SIP servers that can be configured in a network to provide voice (SIP) service to the clients. Some of those were discussed in Chapter 3. Since this is a small scale (test) network, choosing any of the open source SIP servers

would have been sufficient. But based on its wide range of features along with flexibility, scalability and easy configuration for real network infrastructure, OpenSIPS was selected for the test network. Features of OpenSIPS have also been discussed in Chapter 3.

Note that the Step-by-step installing can be found in Appendix D section.

OpenSIPS setup files can be downloaded from their official website and the installation instructions are also mentioned there. However, in order for the OpenSIPS to work, some libraries need to be installed before installing and configuring OpenSIPS. Also, other than the required libraries, ‘mysql’ server and client are also installed on the same machine (in this scenario, it can also be installed on a separate machine) to maintain a database of the user information. Also, during the installation, a secure password should be chosen when asked.

Since OpenSIPS has a modular design, as discussed in previous chapter, it also presents a GUI for installing and uninstalling the modules that can be added or removed during the installation. ‘Menuconfig’ as shown in Figure 4-9 below is the new graphical way to compile the OpenSIPS modules.



Figure 4-9: OpenSIPS Main Configuration Menu

Other than selecting and installing the required modules, some OpenSIPS files should also be edited as required, so that OpenSIPS works properly according to the required network parameters. One such file is ‘opensipsctrl’. Along with this editing, a database is created and a script (also to be modified) is generated. This script is where OpenSIPS gets its instructions for the use of services, such as dial-plan, TCP or TLS support and use of presence.

In order for the OpenSIPS to have its own separate log file, modifications are done in 'syslog' file. Details are mentioned in Appendix D.

What is described above is for the OpenSIPS installation and configuration. However to create and maintain the user database in a more convenient way, a GUI based control panel can also be installed and the step-by-step guide to that is described in Appendix D.

For OpenSIPS-CP, 'Apache2' server needs to be installed along with necessary dependencies and modifications of the files as given in OpenSIPS installation. After the installation of CP, web browser with IP address pointing to the OpenSIPS's server IP address opens up a following screen as shown in Figure 4-10. Username and password, default being, 'admin' and 'admin', respectively, were added during the installation phase of CP.



Figure 4-10: OpenSIPS-CP Login Screen

4.9 Ekiga Installation

VoIP client software Ekiga can be downloaded from their official website for both the Linux and Windows platforms. This section does not show its installation process. A step-by-step guide to install Ekiga is available on their web pages and is easy to follow.

4.10 Testing of Lab-based Core Network

The goal was to layout the design of a network and to configure it, within a virtual environment that can successfully make SIP users register and allow calls between them. The objective was successfully achieved: a call between the registered users was made as shown in Figures 4-11 to 4-14 below.

Before the users get registered to the network, users must be created in the database, as shown in Appendix D, i.e., using the OpenSIPS CP. This resembles to the same scenario in which as username and password are created when registering for an email account.

Figure 4-11, shows the parameters that are supplied while registering a user in Windows platform. Instead of IP address of the SIP server, a domain name 'oursips.lab.com' is used, so that even if the user moves (un-plugs from one VLAN and re-plugs into another VLAN), domain name is still resolved to the IP address of the SIP server. For simplicity, username, password and authentication user fields remain same, i.e., User2. Right side of Figure 4-11 shows that the user has successfully registered with the SIP server, OpenSIPS. Same procedure works for the registration of the 'user3' on Linux platform as shown in Figure 4-12.

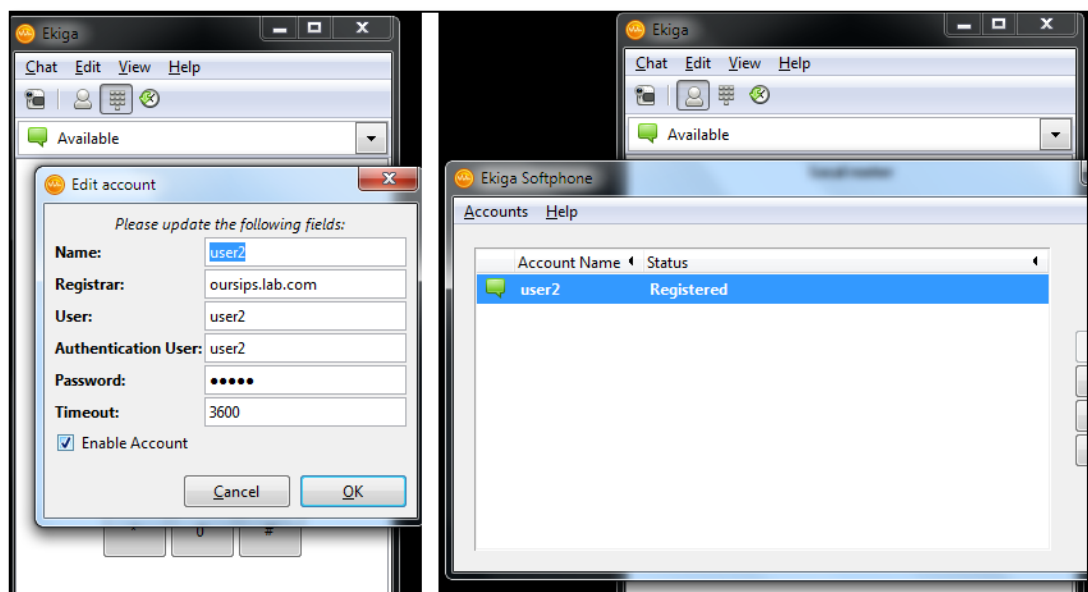


Figure 4-11: Registration of User2 on VoIP Client Ekiga on Windows Platform.

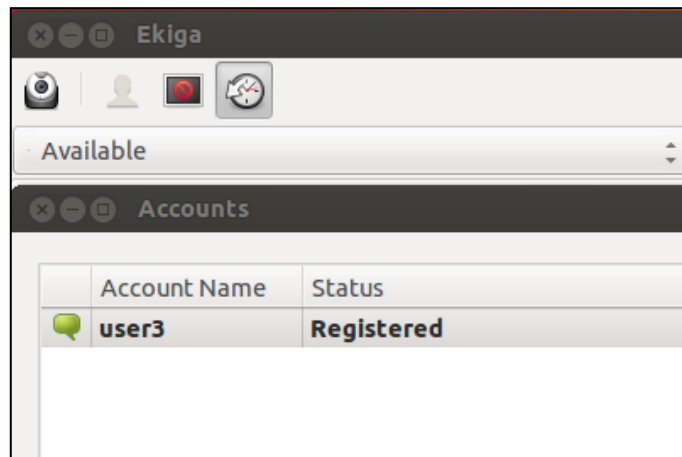


Figure 4-12: Registration of User3 on Linux Platform.

Next two Figures 4-13 and 4-14 show the call progress that was initiated from the user2 towards user3 via OpenSIPS. Left part of Figure 4-8 shows the call initiation, whereas the right shows the call in progress between the two users.

The call that was initiated from user2 can be seen in Figure 4-13 (left), i.e., an incoming call from user2 to user3. Right hand side of Figure 4-14 demonstrates the video capabilities of OpenSIPS with user2 in the picture.

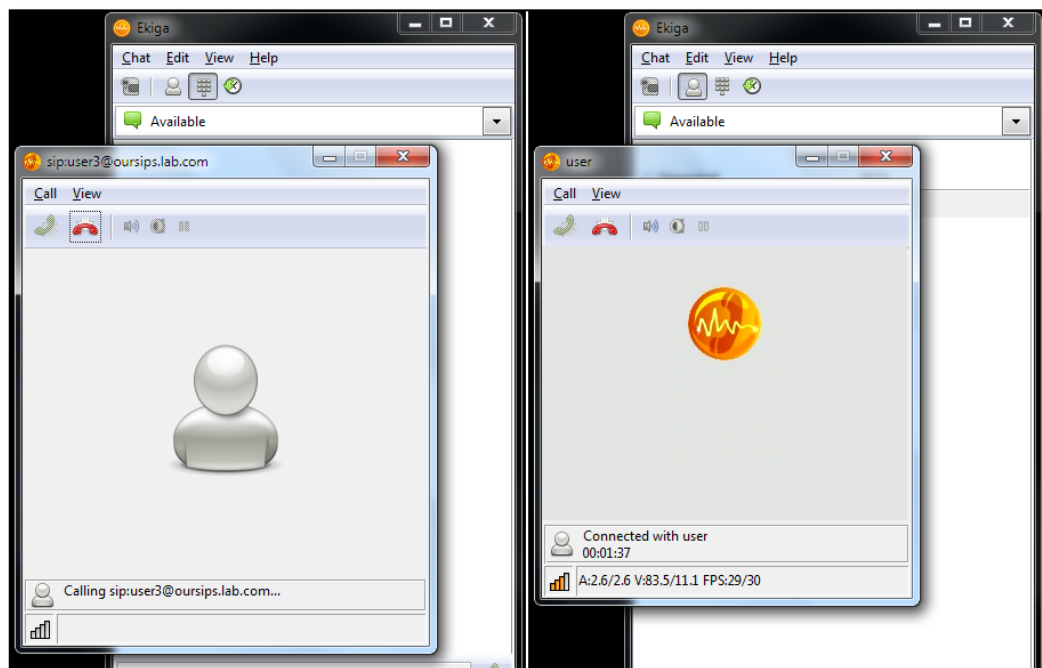


Figure 4-13: Call Initiation and Progress Between User2 and User3

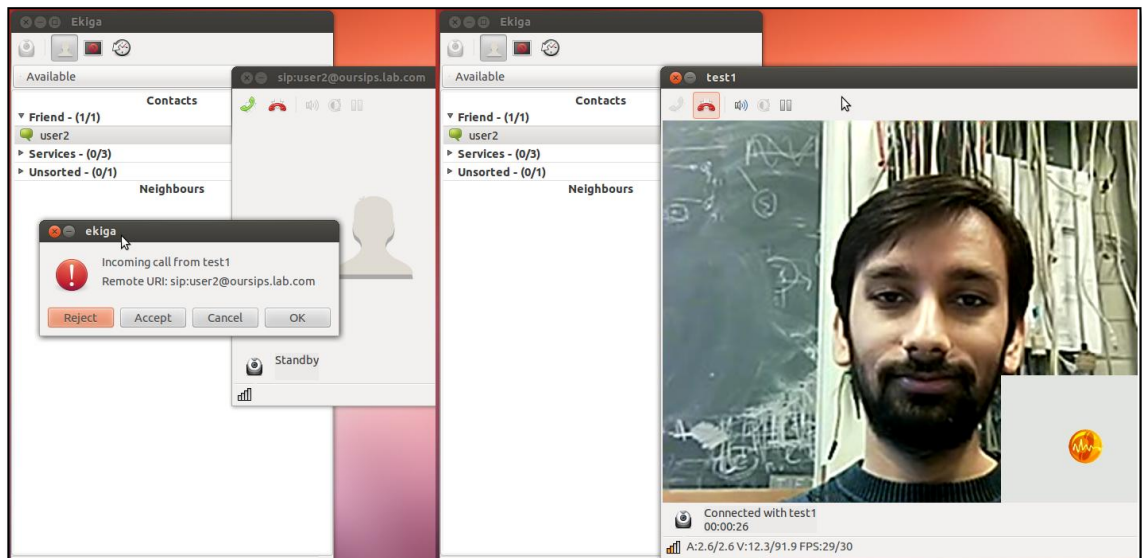


Figure 4-14: A Video Call between User2 and User3

4.11 Test Network Issues

Even though a successful call was made between the two registered users, the test core network presented its own problems, some related to the client software (Ekiga), whereas other related to the virtual machine software operating system support capabilities.

Few glitches and problems that could not be resolved the work are the following:

- During the initial configuration phase 'test1', 'test2' users were created and later on those were changed to user1, user2 and so on. But, as be seen from Figure 4-14, 'test1' appears in calling name instead of the 'user2'. This is due to a cache problem on multiple levels, i.e., virtualized setup and with VoIP client 'Ekiga', which retains the old usernames from the OpenSIPS, even though there was a clean and fresh installation of the SIP server.
- One purpose of installing and configuring DNS server on both base-stations was to facilitate the end user of changing of the IP address of the SIP server, when entering user parameters for user registration. This objective was partially achieved.

When the LAN cable is directly un-plugged from one VLAN and is re-plugged into another VLAN, user registration with SIP server fails. However, if the user

himself first disconnects VoIP client software (Ekiga) and then un-plugs and re-plugs into another VLAN, registration with the SIP server was not a problem.

- Finally, an attempt was made to connect a WLAN USB to the network to transmit voice over wireless using this setup of test core network. But, since this whole setup was built on virtual machines using the VirtualBox software, a limited support for wireless drivers is available. Thus, we did not succeed in objective of connecting the test network with WLAN USB and transmit voice using it.

5 Conclusion

As the Internet has taken over the world with high speed data connections (broadband Internet) so has the concept of voice over IP (VoIP). Traditional voice calls that used the circuit-switched technology are fading away and are being replaced by the packet-switched technology. This holds true for the cellular networks too in particular the 4th generation, i.e., the LTE. The LTE network is an evolved IP based network that relies on the high speed packet core network along with IMS to give its users the voice and the multimedia services. The solution that is being worked on and is considered as the long term solution for transmitting voice over LTE network used VoIP with SIP protocol and is called VoLTE using VoIMS.

The core network domain plays a role of backbone in any network infrastructure as it consists of nodes which process network's signaling and data. These nodes are inter and intra-connected to each other and to the outside networks to route, e.g., circuit-switched calls or packet-switched data to their proper destination.

In this research work, we have tried to design, implement and verify the workings of the voice and data capable core network for LTE type software defined radio (SDR) testbed. The goal here was, first to search and select the tools, programs and technologies that can satisfy the network and user requirements and then verify the working of the core network design.

To summarize the features and implementation, due to the scarcity of the resources the research work explores the option of using virtualization software so that with its help multiple virtual machines can be run on a single physical host machine. Voice and other multimedia services for the network use SIP protocol and are implemented with an open-source SIP proxy server OpenSIPS. In order to facilitate the users of the network facilities such as automatic IP allocation, name to IP address mapping and mobility were implemented.

For any network choosing the right IP addressing scheme is necessary whether it is for test (lab) network or a network in the real world. This concept was reflected in the IP addressing that was chosen for the thesis work, i.e., using the private IP addressing scheme with CIDR approach to avoid wastage of the IP addresses. Also, allocation of the IP address to the clients was realized through DHCP servers, installed and

configured in base stations. IP address to name mapping and vice versa feature was implemented in the network with DNS servers also, installed and configured in each base station. This helped in supporting the mobility feature, i.e., users did not have to manually change (the settings in their VoIP client software) the IP address of the SIP server every time they move between the base stations.

After the installation and configuration of the SIP server (OpnSIPS), user database was created and with the help of GUI based control panel. VoIP client software was installed and configured accordingly. Verification of the network was done by making successful voice and video calls between the users that were located (registered) in different networks (created through VLANs) as well as, between the users in same network or base station.

On the last note, it is neither advisable nor recommended to run the real-time services such as SIP server proxy on virtual machines as they may introduce jitters and other delays to the voice calls and may lack the support for system files and drivers. In the case of our thesis work, support for WLAN USB drivers was lacking in the virtualization software and thereby one could not test the network for transmitting voice over WLAN. Also, the mobility feature worked partially or only after following steps in a certain manner. Other issue that remained unsolved was related to the VoIP client software Ekiga.

References

- [1] T. Casey, H. Hämmäinen, J. Manner, J. Poikonen, K. Ruttik and O. Tirkkonen, "KognitiiviradioPäätelaitteista Arvoverkkoihin," Espoo, 2012.
- [2] I. Mitola, J., "Software radios: Survey, critical evaluation and future directions," IEEE Aerospace and Electronic Systems Magazine, vol. 8, pp. 25 -36, April 1993.
- [3] V. Prasad, "Network Time Synchronization in Time Division - LTE systems," Master's Thesis, 2013.
- [4] Jukka Lempiäinen, Matti Manninen, Radio Interface System Planning for GSM/GPRS/UMTS., Kluwer Academic Publishers, 2002.
- [5] J. Korhonen, Introduction to 3G Mobile Communications. 2nd Edition. Boston: Artech House Mobile Communications Series, 2003.
- [6] H.Holma, A. Toskala., WCDMA for UMTS. Radio Access For Third Generation Mobile Communications. John Wiley & Sons Ltd., 2000.
- [7] P. Biswas, "WCDMA - Technology for 3G cellular system," Indian Institute of Technology, Kharagpur.
- [8] "UMTS Network Architecture (Third Generation Networks)," The-Crankshaft Publishing, [Online]. Available: <http://what-when-how.com/roaming-in-wireless-networks/umts-network-architecture-third-generation-networks/>. [Accessed 30 July 2013].
- [9] Miikka Poikselkä, Harri Holma, Jukka Hongisto, Juha Kallio and Antti Toskala., VOICE OVER LTE (VoLTE)., Chennai: John Wiley & Sons, Ltd., 2012.
- [10] D. A. Ansari, "LTE Architecture: Network Elements and Interfces.," White n Green, 29 May 2013. [Online]. Available: <http://whitengreen.com/blog.php?id=107-lte-architecture-network-elements-and-interfaces>. [Accessed 30 July 2013].
- [11] "Architectural Considerations for Backhaul of 2G/3G and Long Term Evolution Networks," Cisco Inc., 2010. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns973/white_paper_c11-613002.html. [Accessed 30 July 2013].
- [12] 3GPP TS 36.300 V10.0.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN), 06-2010.
- [13] S. Sesia, I. Toufik, M. Baker., LTE - The UMTS Long Term Evolution from Theory to Practice. John Wiley & Sons, Ltd., 2009
- [14] H. Holma and A. Toskala., LTE for UMTS OFDMA and SC-FDMA Based Radio

Access., John Wiley & Sons Ltd., 2009.

- [15] 3GPP TS 23.203 V10.6.0, Technical Specification Group Services and System Aspects; Policy and charging control architecture, 03-2012.
- [16] UMTS Signaling (e)., Award Solutions, Inc. Training Solutions., www.awardsolutions.com
- [17] Anritsu, "Voice Over LTE. VoLTE," Anritsu Whitepaper, 09-2012.
- [18] 3GPP TS 23.272 V10.7.0, Circuit Switched (SC) fallback in Evolved Packet System (EPS)., Release 10, 2012-03
- [19] X. Wu, "Circuit-Switched Fallback, Ultra-Flash CSFB." Huawei Whitepaper., 2013.
- [20] Itsuma Tanaka, Takashi Koshimizu and Katsutoshi Nishida., "CS Fallback Function for Combined LTE and 3G Circuit Switched Services.," NTT DOCOMO Technical Journal., vol. 11, no. 3.
- [21] N. Russell., "IR.92 - IMS Profile for Voice and SMS. Version 7," GSM Association, 2013.
- [22] 3GPP TS 23.216 V8.7.0, Single Radio Voice Call Continuity (SRVCC)., Release 8, 2011-06
- [23] "Evolve to richer voice with VoLTE: Winning the revenue advantage with LTE smartphones," *Liquid Net, NSN Whitepaper*. 2012.
- [24] Itsuma Tanaka, Takashi Koshimizu., "Overview of GSMA VoLTE Profile," *NTT DOCOMO Technical Journal*, vol. 13, no. 4.
- [25] "IP | PBX | Telephony | Installations," Living Intelligent , [Online]. Available: <http://livingintelligent.com/com-voice-ippbx.htm>. [Accessed 30 July 2013].
- [26] R. Arora, "Voice over IP : Protocols and Standards," 1999. [Online]. Available: http://www.cse.wustl.edu/~jain/cis788-99/voip_protocols/. [Accessed 30 July 2013].
- [27] "Voice over Internet Protocol Definition and Overview," Teliqo, 2012. [Online]. Available: <http://www.teliqo.com/voip/>. [Accessed 30 July 2013].
- [28] Saverio Niccolini, Dr. Rosario Giuseppe Garroppo, Dr. Jörg Ott, Stefan Prella, Dr. Jiri Kuthan, Dr. Sven Ubik, Dr. Margit Brandl, Dimitris Daskopoulos, Egon Verharen, Erik Dobbelssteijn., "IP Telephony Cookbook, Chapter 3: Protocols.," [Online]. Available: http://dev.gentoo.org/~solar/Cookbook_D1/ch02s02.html. [Accessed 30 July 2013].
- [29] "A Primer on the H.323 Series Standard," VTEL, [Online]. Available: <http://www.vtel.com/support/galaxy/h323primer.htm>. [Accessed 30 July 2013].

- [30] J. Rosenberg, H. Schulzrinne, H. Schulzrinne, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol. RFC 3261," IETF, June 2002.
- [31] J. Janak, "SIP Introduction," FhG FOKUS, 2003.
- [32] "SIP Registrars Definition and Diagram," Voip Dictionary, 2009. [Online]. Available: <http://www.voipdictionary.com/IPTelephonyDictionary-Redirection-Server-Redirect-Server-Definition.html>. [Accessed 30 July 2013].
- [33] A. B. Johnston, SIP: Understanding the Session Initiation Protocol, Artech House Telecommunications, 2009.
- [34] T. Kelly, VoIP for Dummies, Indianapolis, Indiana.: Wiley Publishing, Inc., 2005.
- [35] R. Kantola, "Session Initiation Protocol," Aalto University. S-38.3115 Course Slides., Espoo., 2013.
- [36] "VoIP SDK - Session initiation protocol," OZEKI Systems Ltd., [Online]. Available: http://www.voip-sip-sdk.com/p_230-session-initiation-protocol-voip.html. [Accessed 30 July 2013].
- [37] F. E. Goncalves, Building Telephony Systems with OpenSIPS 1.6, Birmingham, UK. Packt Publishing, 2010.
- [38] "openSIPS," 2008. [Online]. Available: <http://www.opensips.org/>. [Accessed 03 August 2013].
- [39] "A review of OpenSIPS," The Smartvox Knowledgebase, 31 December 2011. [Online]. Available: <http://kb.smartvox.co.uk/opensips/opensips-explained/>. [Accessed 03 August 2013].
- [40] Malcolm Davenport, Sean Bright, "Asterisk as Swiss Army Knife of Telephony," Asterisk Project, 08 May 2012. [Online]. Available: <https://wiki.asterisk.org/wiki/display/AST/Asterisk+as+a+Swiss+Army+Knife+of+Telephony>. [Accessed 03 August 2013].
- [41] "Asterisk Quick Start Guide," Digium, The Asterisk Company, 2012.
- [42] "Comparison of OpenSIPS and Asterisk," The Smartvox Knowledgebase, 17 01 2012. [Online]. Available: <http://kb.smartvox.co.uk/opensips/opensips-asterisk/>. [Accessed 03 August 2013].
- [43] "Asterisk vs OpenSIPS," VoIP Today, September 2009. [Online]. Available: http://www.voiptoday.org/index.php?option=com_content&view=article&id=231:asterisk-vs-opensips&catid=55:pbx-comparison&Itemid=103. [Accessed 03 August 2013].
- [44] Damien Sandras, Eugen Dedu and Yannick Defais., "Manual - Ekiga," Ekiga, [Online]. Available: <http://wiki.ekiga.org/index.php/Manual>. [Accessed 30 July 2013].

2013].

- [45] D. Sandras, "5 years of Open Source Voice over IP," Ekiga Blog, [Online]. Available: <http://blog.ekiga.net/?p=27>. [Accessed 30 July 2013].
- [46] "Ekiga Softphone Features," Ekiga, [Online]. Available: <http://ekiga.org/ekiga-softphone-features>. [Accessed 30 July 2013].
- [47] Karen Scarfone, Murugiah Souppaya and Paul Hoffman, "Guide to Security for Full Virtualization Technologies," National Institute of Standards and Technology. January, 2011.
- [48] IBM, "Virtualization in Education," IBM Global Education, White Paper. October, 2007.
- [49] J.Ramos, "Security Challenges with Virtualization," Master's Thesis.Universidade De Lisboa., December, 2009.
- [50] "Oracle VM VirtualBox," Oracle Corporation, 2004-2013. [Online]. Available: <http://www.virtualbox.org/manual/>. [Accessed 03 August 2013].
- [51] "About VMWare (VMW)," VMWare, [Online]. Available: <http://www.vmware.com/company/>. [Accessed 03 August 2013].
- [52] T. Lammle, Cisco Certified Network Associate Study Guide, 6th Ed., Sybex - Wiley Publishing, Inc., 2007.
- [53] Information Sciences Institute, University of Southern California, "RFC: 791 - Internet Protocol, DARPA Internet Program, Protocol Specification.," IETF, 1981.
- [54] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear., "Address Allocation for Private Internets," IETF RFC: 1918, February, 1996.
- [55] V. Fuller, T. Li, J. Yu, K. Varadhan, " RFC: 1338 - Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, IETF" 1993.
- [56] T. Pummill, B. Manning, "Variable Length Subnet Table for IPv4," IETF RFC: 1878, December, 1995.
- [57] IEEE Std 802.1QTM, Virtual Bridged Local Area Networks.
- [58] S. Varadarajan, "Virtual Local Area Networks," Washington University in St. Louis., 14 August 1997. [Online]. Available: http://www.cse.wustl.edu/~jain/cis788-97/ftp/virtual_lans/index.htm [Accessed 30 July 2013].
- [59] "VLAN Overview - ICND2 04," Steve's Media Wiki, September 2010. [Online]. Available: http://www.stevebrookes.id.au/mediawiki/index.php/VLAN_Overview_-

_ICND2_04. [Accessed 30 July 2013]

- [60] R. Droms, "Dynamic Host Configuration Protocol," IETF RFC: 2131, March 1997.
- [61] "System Administration: DHCP Server and Client Communication," [Online]. Available: <http://nitindupare.blogspot.fi/2012/08/dhcp-server-and-clients-communication.html>. [Accessed 31 July 2013].
- [62] Cisco IOS IP Configuration Guide, Release 12.2, Cisco Systems, Inc, 2001–2006.
- [63] R. Aitchison, Pro DNS and BIND, New York: Apress, 2005.
- [64] "Domain Name to IP Resolution Process - Root Name Server," Support Sages, 25 May 2010. [Online]. Available: <http://www.supportsages.com/blog/2010/05/the-domain-name-to-ip-resolution-process-part-ii-v-root-name-servers/>. [Accessed 31 July 2013].
- [65] P. Mockapetris, "Domain Names - Concept and Facilities," IETF RFC: 1034, November 1987.
- [66] C. Hedrick, "Routing Information Protocol," IETF RFC: 1058, June 1988.
- [67] J. Moy, "OSPF Version 2," IETF RFC: 2328, April 1998.
- [68] R. Malhotra, "IP Routing", O'Reilly & Associates, Inc., 2002.

Appendix A

Configuration of the BS-1 is shown here.

- VirtualBox interface setting configurations are shown in *Table 4-1*, and IP addresses used for the test network are mentioned in section, 4.3.

```
# gedit /etc/network/interfaces

# This file describes the network interfaces available on your
system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo eth1
iface lo inet loopback

# The primary network interface

iface eth1 inet static
address 192.168.101.1
netmask 255.255.255.0
network 192.168.101.0
broadcast 192.168.101.255

# allow-hotplug eth2
iface eth2 inet static
address 192.168.100.9
netmask 255.255.255.252
network 192.168.100.8
broadcast 192.168.100.11

# allow-hotplug eth3
auto eth3
iface eth3 inet static
address 192.168.100.2
netmask 255.255.255.252
network 192.168.100.0
broadcast 192.168.100.3

# static route
up route add -net 192.168.102.0 netmask 255.255.255.0 gw
192.168.100.10
down route del -net 192.168.102.0 netmask 255.255.255.0 gw
92.168.100.10

up route add -net 192.168.100.0 netmask 255.255.255.252 gw
192.168.100.1
down route del -net 192.168.100.0 netmask 255.255.255.252 gw
192.168.100.1
```

- To enable routing, edit 'sysctl.conf' file and change the 'ip_forward' value to '1' and make the changed permanent by running another command on main command as shown below.

```
# gedit /etc/sysctl.conf
    net.ipv4.ip_forward = 1

# sysctl -p /etc/sysctl.conf
```

- DHCP server configuration. Install the package "isc-dhcp-server" and edit the interface on which the dhcp server should listen for the request.
- After editing the configuration files, DHCP initialization should be restarted along with networking.

```
# apt-get install isc-dhcp-server

# gedit /etc/default/isc-dhcp-server

    #Change the interface parameter ,i.e., interface to which the DHCP
    #server should listen for the requests.
    INTERFACES="eth1"

# gedit /etc/dhcp/dhcpd.conf

    ddns-update-style none;
    option domain-name "oursips.lab.com";
    option domain-name-servers 192.168.101.1;

    default-lease-time 600;
    max-lease-time 7200;

    authoritative;

    log-facility local7;

    # This is a very basic subnet declaration.

    subnet 192.168.101.0 netmask 255.255.255.0 {
    range 192.168.101.11 192.168.101.20;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.101.255;
    option routers 192.168.101.1;
    }

# /etc/init.d/networking restart
# /etc/init.d/isc-dhcp-server restart
```

- DNS server was configured as below. Bind9 was installed for DNS services and required files were edited, i.e., created the primary master server for the zone

“lab.com” with the zone file name “db.oursips”. Then zone file is edited with “named.conf.local”. Also the reverse zone file is created to map the IP address to the name. Lastly, the ‘names.conf.options’ file is edited to give the directory location and in case there are any forwarder IPs.

```
# apt-get install bind9

# gedit /etc/bind/named.conf.local

    zone "lab.com" {
        type master;
        file "/etc/bind/db.oursips";
        allow-query {any;};
        notify no;

    };

    zone "168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.168.192";
        notify no;
        allow-query {any;};
    };
```

```
# gedit /etc/bind/db.oursips

$TTL 3h
lab.com. IN SOA ns.lab.com. lok.lab.com (
    1          ; Serial
    1d         ; Refresh after 3 hours
    1d         ; Retry after 1 hour
    4w         ; Expire after 1 week
    1h )       ; Negative caching TTL of 1 hour
;
; Name servers
;
lab.com. IN NS  ns.lab.com.
;
; Addresses for the canonical names
;
localhost.lab.com.      IN A      127.0.0.1
ns.lab.com.              IN A      192.168.101.1
oursips.lab.com.         IN A      192.168.100.1
;
;
IN      NAPTR    50 50  "s" "SIP+D2U" "" _sip._udp.ns.lab.com.
;
_sip._udp.ns.lab.com.    86400 IN    SRV    0 1 5060 oursips.lab.com.
```

```
# gedit /etc/bind/db.168.192

$TTL 3h
@ IN SOA ns.lab.com. lok.lab.com (
    1          ; Serial
    1d        ; Refresh after 3 hours
    1d        ; Retry after 1 hour
    4w        ; Expire after 1 week
    1h )      ; Negative caching TTL of 1 hour

;
; Name servers
;
168.192.in-addr.arpa.  IN NS  ns.lab.com.
;
; Addresses point to canonical name
;
1.101.168.192.in-addr.arpa.  IN PTR ns.lab.com.
1.100.168.192.in-addr.arpa.  IN PTR oursips.lab.com.
```

```
# gedit /etc/bind/named.conf.options
```

```
OPTIONS{
    directory "/var/cache/bind";
    recursion no;
    auth-nxdomain no;
    listen-on-v6 { any; };
};
```

```
# /etc/init.d/bind9 restart
```

Appendix B

- Configuration of the BS-2 is shown here.
- VirtualBox interface setting configurations are shown in *Table 4-1*, and IP addresses used for the test network are mentioned in section, 4.3.

```
# gedit /etc/network/interfaces

# This file describes the network interfaces available on
# your system
# and how to activate them. For more information, see
# interfaces(5).

# The loopback network interface
auto lo eth1 eth2 eth3
iface lo inet loopback

# The primary network interface

#allow-hotplug eth1
iface eth1 inet static
address 192.168.102.1
netmask 255.255.255.0
network 192.168.102.0
broadcast 192.168.102.255

iface eth2 inet static
address 192.168.100.10
netmask 255.255.255.252
network 192.168.100.8
broadcast 192.168.100.11

# allow-hotplug eth2
iface eth3 inet static
address 192.168.100.6
netmask 255.255.255.252
network 192.168.100.4
broadcast 192.168.100.7

# Static Route
up route add -net 192.168.101.0 netmask 255.255.255.0 gw
192.168.100.9
down route del -net 192.168.101.0 netmask 255.255.255.0 gw
192.168.100.9
```

- Routing is enabled in a similar way as shown in Appendix A.
- DHCP server configuration; almost similar as done in Appendix A.

```
# apt-get install isc-dhcp-server

# gedit /etc/default/isc-dhcp-server

#Change the interface parameter, i.e., interface to which the DHCP
#server should listen for the requests.
INTERFACES="eth1"

# gedit /etc/dhcp/dhcpd.conf

ddns-update-style none;

option domain-name "lab.com";
option domain-name-servers 192.168.102.1;

default-lease-time 600;
max-lease-time 7200;

authoritative;
log-facility local7;

# This is a very basic subnet declaration.

subnet 192.168.102.0 netmask 255.255.255.0 {
range 192.168.102.11 192.168.102.20;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.102.255;
option routers 192.168.102.1;
}

# /etc/init.d/networking restart
# /etc/init.d/isc-dhcp-server restart
```

- DNS server configuration also, similar to what is given in Appendix A.

```
# apt-get install bind9

# gedit /etc/bind/named.conf.local

zone "lab.com" {
type master;
file "/etc/bind/db.oursips";
allow-query {any;};
notify no;

};

zone "168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.168.192";
notify no;
allow-query {any;};
};
```

```
# gedit /etc/bind/db.oursips

$TTL 3h
lab.com. IN SOA ns2.lab.com. lok.lab.com (

        1          ; Serial
        1d         ; Refresh after 3 hours
        1d         ; Retry after 1 hour
        4w         ; Expire after 1 week
        1h )       ; Negative caching TTL of 1 hour

;
; Name servers
;
lab.com. IN NS  ns2.lab.com.

;
; Addresses for the canonical names
;
localhost.lab.com.      IN A      127.0.0.1
ns2.lab.com.            IN A      192.168.102.1
oursips.lab.com.        IN A      192.168.100.5
;
IN      NAPTR 50 50  "s" "SIP+D2U" "" _sip._udp.ns2.lab.com.
;
_sip._udp.ns2.lab.com.  86400 IN    SRV    0 1 5060 oursips.lab.com.
```

```
# gedit /etc/bind/db.168.192

$TTL 3h
@ IN SOA ns2.lab.com. lok.lab.com (

        1          ; Serial
        1d         ; Refresh after 3 hours
        1d         ; Retry after 1 hour
        4w         ; Expire after 1 week
        1h )       ; Negative caching TTL of 1 hour

;
; Name servers
;
168.192.in-addr.arpa.  IN NS  ns2.lab.com.

;
; Addresses point to canonical name
;
1.102.168.192.in-addr.arpa.  IN PTR ns2.lab.com.
5.100.168.192.in-addr.arpa.  IN PTR oursips.lab.com.
```

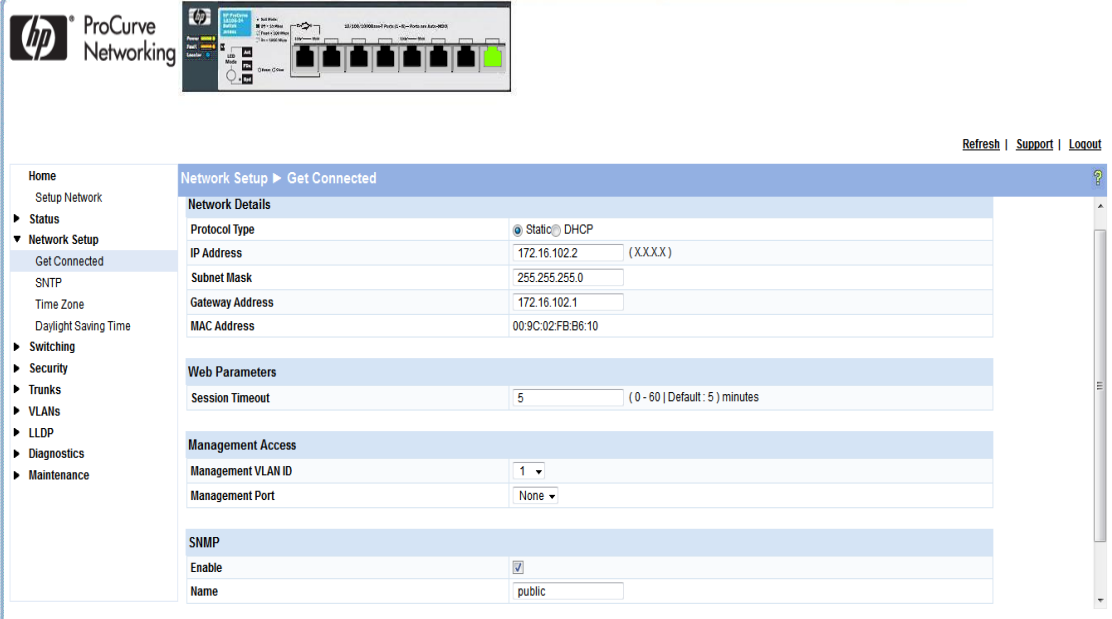
```
# gedit /etc/bind/named.conf.options

OPTIONS{
    directory "/var/cache/bind";
    recursion no;
    auth-nxdomain no;
    listen-on-v6 { any; };
};
```

```
# /etc/init.d/bind9 restart
```


Appendix C

Shown here is the VLAN configuration through an HP ProCurve switch. First figure show the switch configuration, i.e., management IP address.



HP ProCurve Networking

Refresh | Support | Logout

Home
Setup Network
Status
Network Setup
Get Connected
SNTP
Time Zone
Daylight Saving Time
Switching
Security
Trunks
VLANs
LLDP
Diagnostics
Maintenance

Network Setup > Get Connected

Network Details

Protocol Type: ☒ Static ☐ DHCP

IP Address: 172.16.102.2 (XXX)

Subnet Mask: 255.255.255.0

Gateway Address: 172.16.102.1

MAC Address: 00:9C:02:FB:B6:10

Web Parameters

Session Timeout: 5 (0 - 60 | Default: 5) minutes

Management Access

Management VLAN ID: 1

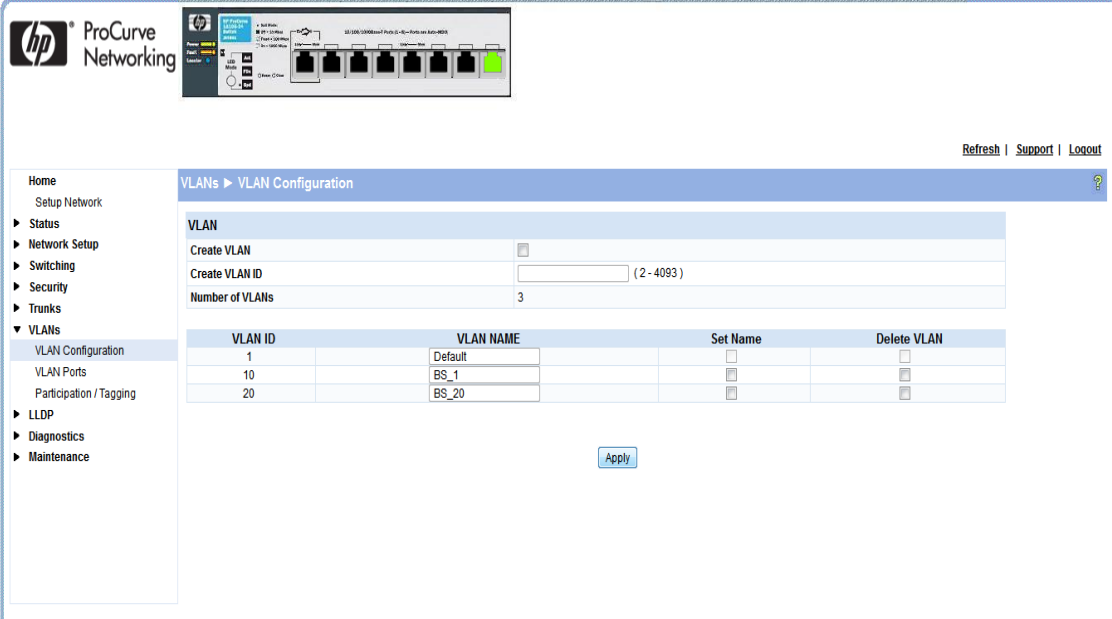
Management Port: None

SNMP

Enable: ☒

Name: public

Here in the second figure below, VLAN IDs and VLAN names are being configured.



HP ProCurve Networking

Refresh | Support | Logout

Home
Setup Network
Status
Network Setup
Switching
Security
Trunks
VLANs
LLDP
Diagnostics
Maintenance

VLANs > VLAN Configuration

VLAN

Create VLAN: ☐

Create VLAN ID: (2 - 4093)

Number of VLANs: 3

VLAN ID	VLAN NAME	Set Name	Delete VLAN
1	Default	<input type="checkbox"/>	<input type="checkbox"/>
10	BS_1	<input type="checkbox"/>	<input type="checkbox"/>
20	BS_20	<input type="checkbox"/>	<input type="checkbox"/>

Apply

Next figure shows the port configurations, i.e., which of the switch are assigned to which VLAN. 'U' is for untagged, i.e., VLAN traffic for that particular VLAN can go

untagged. 'E' is for excluded ports and 'T' for tagged, i.e., it will carry all the tagged traffic between VLANs.

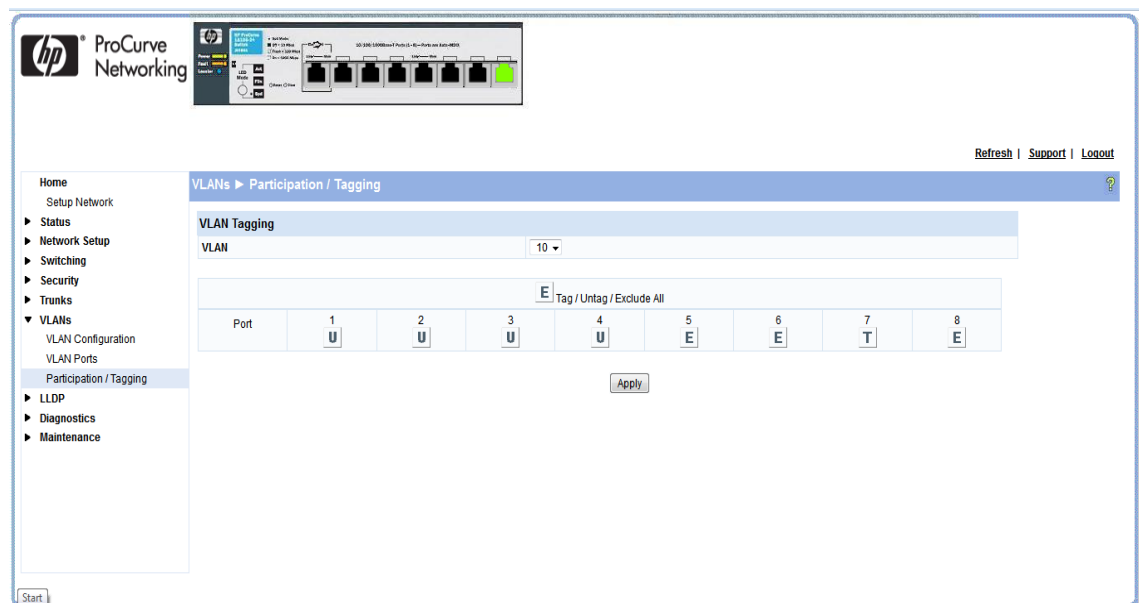
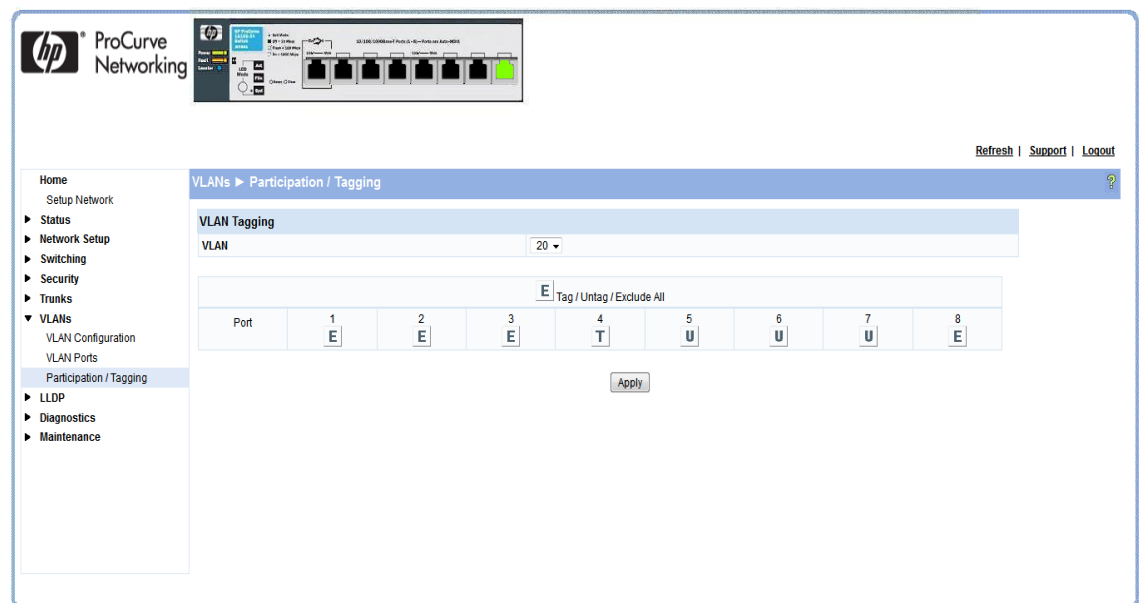


Figure below, shows the port configuration for VLAN 20 (BS-2).



Appendix D

Appendix D shows the configuration of the OpenSIPS server and the interface settings on the machine running the OpenSIPS. But in order to install and configure the OpenSIPS some libraries and dependencies should be installed first as given here.

```
# gedit /etc/network/interfaces

# This file describes the network interfaces available on
your system
# and how to activate them. For more information, see
interfaces(5).

# The loopback network interface
auto lo eth1 eth2
iface lo inet loopback

# The primary network interface

#allow-hotplug eth1
iface eth1 inet static
address 192.168.100.1
netmask 255.255.255.252
network 192.168.100.0
broadcast 192.168.100.3

iface eth2 inet static
address 192.168.100.5
netmask 255.255.255.252
network 192.168.100.4
broadcast 192.168.100.7

# Static Route
up route add -net 192.168.101.0 netmask 255.255.255.0 gw
192.168.100.2
down route del -net 192.168.101.0 netmask 255.255.255.0 gw
192.168.100.2

up route add -net 192.168.102.0 netmask 255.255.255.0 gw
192.168.100.6
down route del -net 192.168.101.0 netmask 255.255.255.0 gw
192.168.100.6
```

```
# apt-get install build-essential
# apt-get install subversion gcc make
# apt-get install bison flex m4
# apt-get install ncurses-dev

# apt-get install perl libdbi-perl libdbd-pg-perl libdbd-mysql-perl
# apt-get install libfrontier-rpc-perl libterm-readline-gnu-perl
libberkeleydb-perl

# apt-get install libsctp1 libxml2-dev libexpat1-dev
# apt-get install libxmlrpc-c3 libxmlrpc-c3-dev
# apt-get install libsnmp-dev libconfuse0 libconfuse-dev
```

Mysql server is installed in order to have the database for the SIP Server. When root password is asked, administrator should carefully choose the password.

```
# apt-get install mysql-server libmysqlclient-dev
```

Next step is to change to the directory where OpenSIPS files should be downloaded. Different ways of downloading are mentioned on their official website.

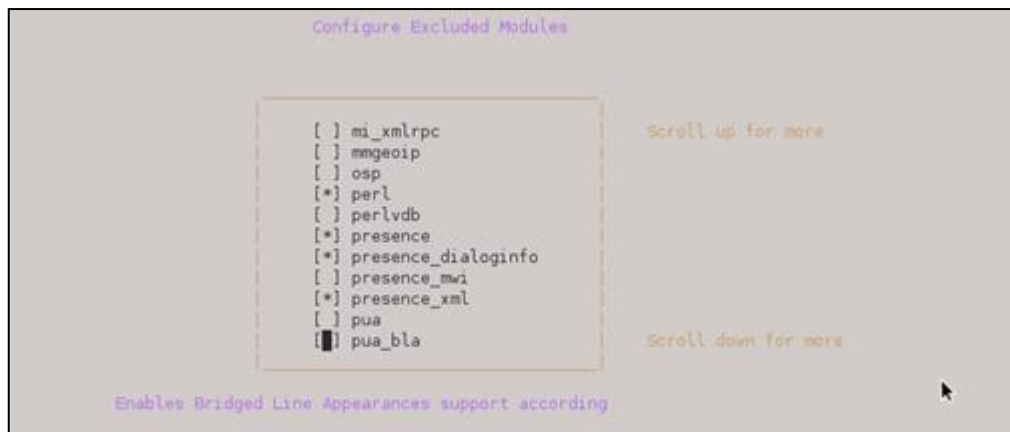
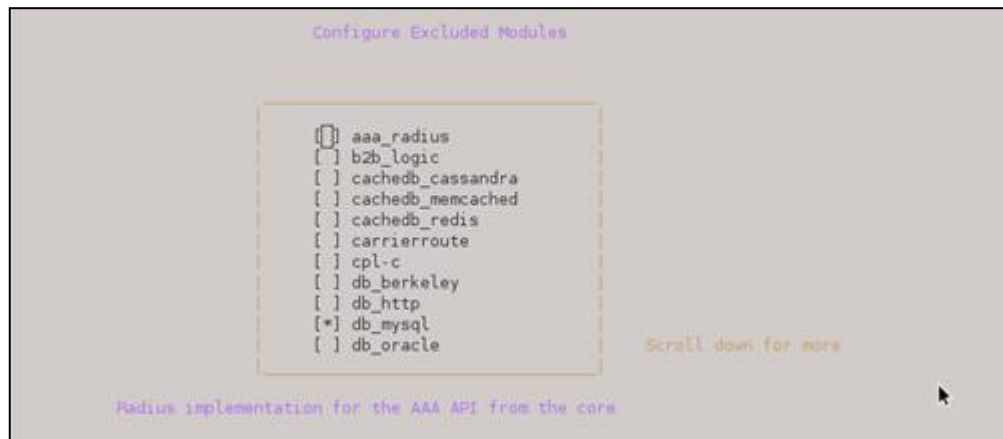
```
#!/usr/local/src

svn co https://opensips.svn.sourceforge.net/svnroot/opensips/branches/1.8
opensips_1_8
```

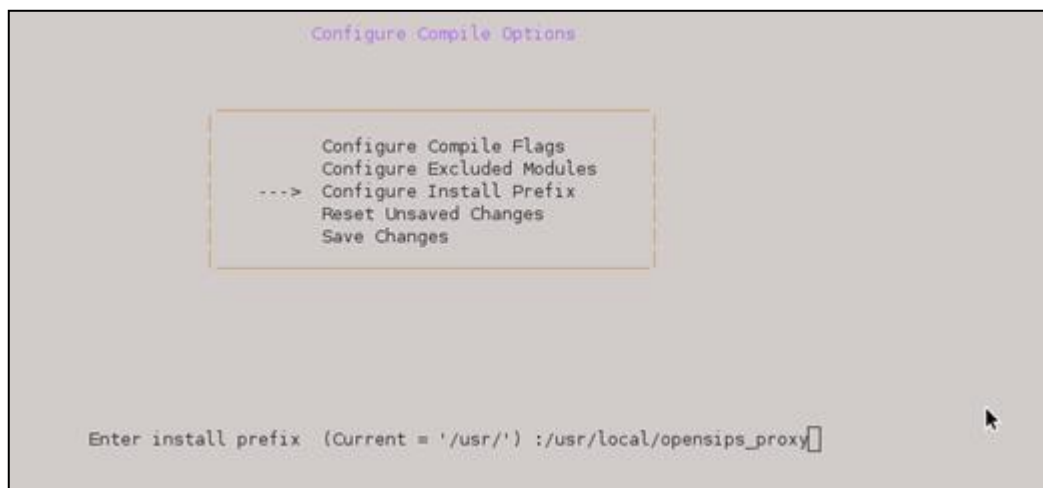
Change into that directory and type “make menuconfig”. Menuconfig is the new and graphical way to install OpenSIPS. Use arrow keys to navigate. In configure and Compile Options, excluded modules need to be selected, i.e., the module that OpenSIPS will support, such as mysql module, presence module and others. Figure below show the configuration steps.



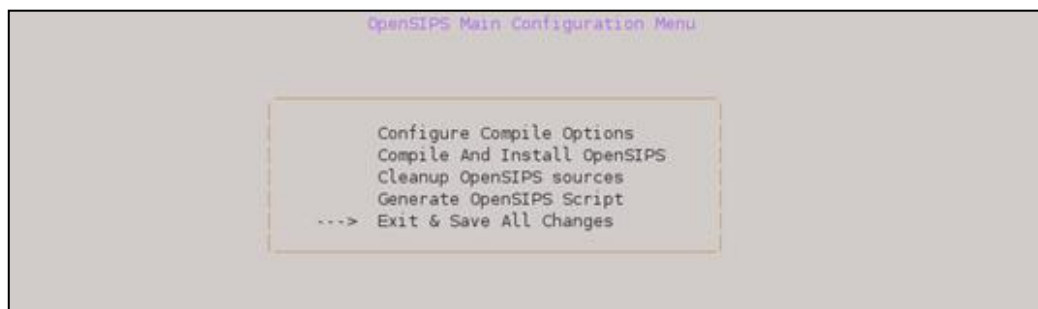
Select the options by pressing spacebar. Modules selected here are, db_mysql, perl, presence, presence_dialoginfo and presence_xml.



After selecting the following options, press 'q' or left arrow to exit from this menu. Go in the 'Configure Install Prefix' menu and type in the destination target directory where OpenSIPS should be installed. Here chosen directory was "/usr/local/opensips_proxy". Press 'Enter' and then 'q' or back arrow to exit from this menu.



After exiting from this menu, go to main menuconfig screen, select the 'Compile and Install OpenSIPS' option. After the installation is done navigate to the 'Exit and Save all Changes' and press 'Enter'.



After installing OpenSIPS next thing to do is to edit the opensipsctlrc file. Uncomment and change the following options as per requirement, e.g., the domain name, database engine name and the user who will access this database along with the database password.

```
root@opensips:/usr/local/opensips_proxy/etc/opensips# gedit
opensipsctlrc

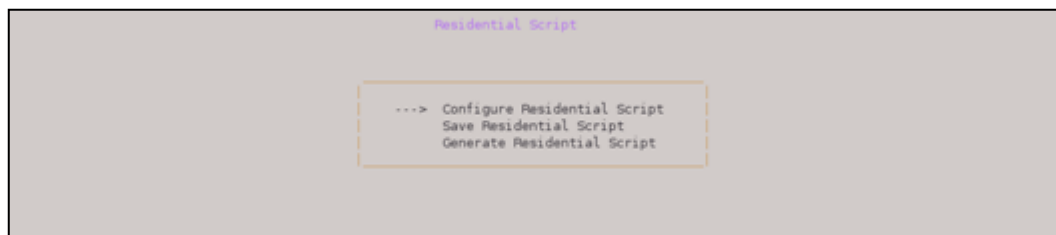
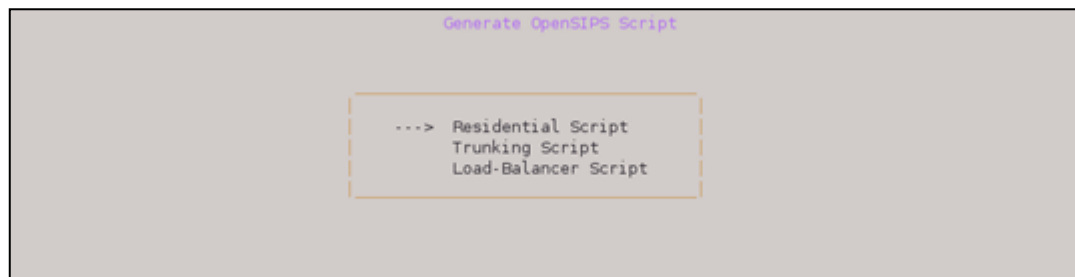
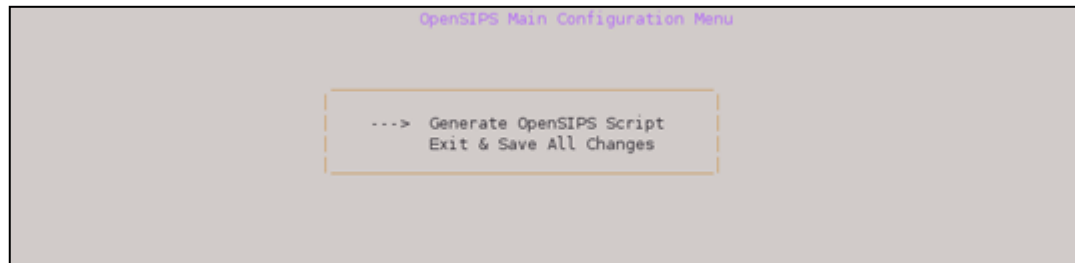
SIP_DOMAIN=lab.com
    DBENGINE=MYSQL
    DBHOST=localhost
    DBNAME=opensips
    DBRWUSER=opensips
    DBWPW="opensipsrw"
    DBROOTUSER="root"
```

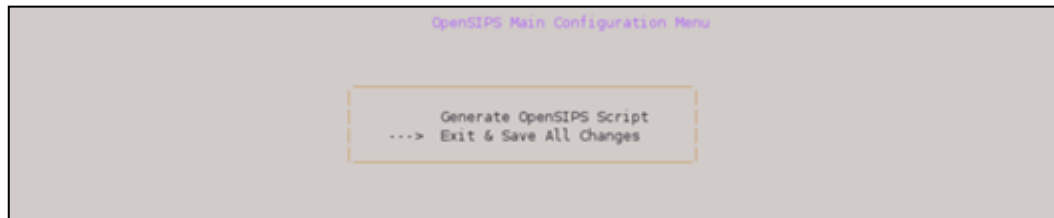
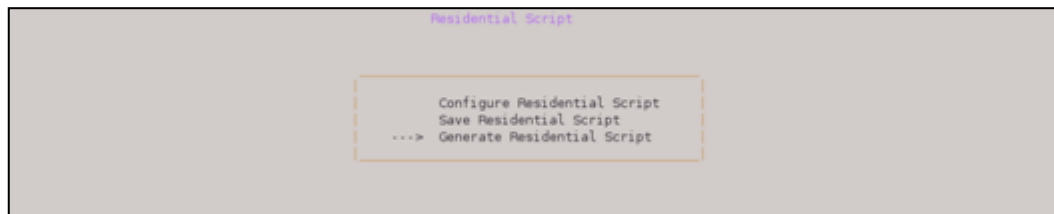
Next step is to create the database for OpenSIPS from where it reads the data. In the 'sbin' directory and execute in the following command.

```
root@opensips:/usr/local/opensips_proxy/sbin# ./opensipsdbctl create
```

After creating the database, next is to generate the OpenSIPS script (residential). OpenSIPS residential script is basically the OpenSIPS's configuration file having the information, such as if TCP or TLS support should be there or not, use dial-plan if any, use of presence should be there or not.

```
root@opensips:/usr/local/opensips_proxy/sbin# ./osipsconfig
```





This will generate the script. Change in to the directory where script is generated and edit the file as required. It is worth mentioning here the following points:

- 'opensipsw' is the default password for mysql database. If given any other password remember to change in all locations.
- Mysql database is hosted on local machine, so no change to '@localhost'. If mysql database is on another machine, give the IP address of that machine.
- Check for the "mpath", i.e., where modules were installed, i.e., the directory path should be correct.

```
root@opensips:/usr/local/opensips_proxy/etc/opensips# gedit
opensips_residential_2012-10-23_13:45:48.cfg
```

```
##### Global Parameters #####

debug=3
log_stderr=no
log_facility=LOG_LOCAL1

fork=yes
children=4

/* comment the next line to enable the auto discovery of local aliases
   based on revers DNS on IPs */
auto_aliases=no
alias=oursips.lab.com:5060

listen=udp:172.16.100.1:5060    # CUSTOMIZE ME
listen=udp:172.16.100.5:5060  # CUSTOMIZE ME

disable_tcp=no
listen=tcp:172.16.100.1:5060  # CUSTOMIZE ME
listen=tcp:172.16.100.5:5060  # CUSTOMIZE ME

disable_tls=yes
```


With “# CUSTOMISE ME”, it is easier to spot where the changes are required in the configuration file.

After making these changes, it is time to run OpenSIPS, and for that copy files from the source folder to the system. Change to the directory where OpenSIPS files were downloaded and copy this file, change it to be executable and edit that file.

```
root@opensips:/usr/local/src/1.8/packaging/debian# cp opensips.init
/etc/init.d/opensips

root@opensips:/usr/local/src/1.8/packaging/debian# chmod +x
/etc/init.d/opensips

root@opensips:/usr/local/src/1.8/packaging/debian# gedit
/etc/init.d/opensips
```

Following editing should be done.

- Give path where the executable OpenSIPS daemon exists and also change RUN_OPENSIPS to 'YES'
- Find the ‘Options’ in this file and include the path of the residential script generated as shown without any ‘\’

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/opensips_proxy/sbin/opensips
NAME=opensips
DESC=opensips
HOMEDIR=/var/run/opensips
PIDFILE=$HOMEDIR/$NAME.pid
DEFAULTS=/etc/default/opensips
RUN_OPENSIPS=yest

OPTIONS="-P $PIDFILE -m $$_MEMORY -M $_MEMORY -u $USER -g $GROUP -
f /usr/local/opensips_proxy/etc/opensips/opensips_residential_2012-
11-1_10:41:40.cfg"
```

Next is to copy this other file and edit it too.

```
root@opensips:/usr/local/src/1.8/packaging/debian# cp
opensips.default /etc/default/opensips

root@opensips:/usr/local/src/1.8/packaging/debian# gedit
/etc/default/opensips
```

Following editing should be done.

- Change RUN_OPENSIPS to yes and Change the memory size to 128
- Username and Group to which user is added should also be specified who should have permission to run OpenSIPS.

```
# OpenSIPS startup options

# Set to yes to enable opensips, once configured properly.
RUN_OPENSIPS=yes

# User to run as
USER=opensips

# Group to run as
GROUP=opensips

# Amount of shared memory to allocate for the running OpenSIPS
server (in Mb)
S_MEMORY=128

# Amount of pkg memory to allocate for the running OpenSIPS server
(in Mb)
P_MEMORY=4
```

One last thing to do is to make changes in the syslog file, so that the opensips could have its own log file. Add a line to the 'rsyslog.conf' file in the end and the restart the service along with opensips.

```
root@opensips:~# gedit /etc/rsyslog.conf

local1.*                                -/var/log/opensips.log
```

```
root@opensips:~# /etc/init.d/rsyslog restart
```

Finally, stop and then start the opensips service. If everything is configured properly, it should start.

```
root@opensips:~# /etc/init.d/opensips stop
root@opensips:~# /etc/init.d/opensips start
```

Next is to install the control panel for OpenSIPS in order to manage the users. OpenSIPS-CP is graphical ways to handling the OpenSIPS users. For OpenSIPS-CP we need to install and configure the 'Apache' server. We need following dependencies to be installed.

```
apt-get install apache2 libapache2-mod-php5 php5 php5-cli php5-gd
php5-mysql php-pear m4 libperl-dev

# For accessing mysql database

pear install mdb2
pear install MDB2#mysql
```

Next we change to the “/var/www” directory and download the opensips control panel files.

```
svn co https://opensips-cp.svn.sourceforge.net/svnroot/opensips-
cp/trunk opensips-cp
```

Next we change into the ‘Config’ directory in the downloaded folder and edit the file ‘db.inc.php’. Hostname and the mysql database password should be the same as we mentioned in the OpenSIPS’s configuration. Edited output is also shown below.

```
root@opensips:/var/www/opensips-cp/config# gedit db.inc.php
```

```
//database driver mysql or pgsql
$config->db_driver = "mysql";

//database host
$config->db_host = "localhost";

//database port - leave empty for default
$config->db_port = "";

//database connection user
$config->db_user = "root";

//database connection password
$config->db_pass = "opensipsrw";
//database name
$config->db_name = "opensips";
```

Next file to edit is ‘boxes.global.inc.php’. In this file the path of the FIFO management Interface should taken care of. This path should be the same as in the residential script of the OpenSIPS configuration file. Also, we are not using the ‘monit’ method, therefore those lines in the file be commented. Output of the edited file is also shown below.

```
root@opensips:/var/www/opensips-cp/config# gedit boxes.global.inc.php
```

```
// mi host:port pair || fifo_file
// $boxes[$box_id]['mi']['conn']="127.0.0.1:8000";
$boxes[$box_id]['mi']['conn']="/tmp/opensips_fifo";

// monit host:port
// $boxes[$box_id]['monit']['conn']="192.168.0.1:2812";
// $boxes[$box_id]['monit']['user']="admin";
// $boxes[$box_id]['monit']['pass']="pass";
// dd$boxes[$box_id]['monit']['has_ssl']=1;
```

Next file to edit is ‘INSTALL’ in the ‘opensips-cp’ directory.

```
root@opensips:/var/www/opensips-cp# gedit INSTALL
```

From this file copy the text,

“config/tools/admin/add_admin/ocp_admin_privileges.mysql” and then execute the following command. This is done so to give the ‘admin’ the root privileges.

```
root@opensips:/var/www/opensips-cp/config# mysql -uroot -p opensips
< /var/www/opensips-
cp/config/tools/admin/add_admin/ocp_admin_privileges.mysql
```

Also, in this file under the heading ‘additional step’; for being able to login create an admin account with the following username and password: admin/admin, copy and add this line “INSERT INTO ...” it into the mysql database.

```
root@opensips:/var/www/opensips-cp/config# mysql -u root -p

mysql> use opensips;

INSERT INTO ocp_admin_privileges
(username,password,hal,available_tools,permissions) values
('admin','admin',md5 ('admin:admin'),'all','all');
```

Lastly, make sure if correct path is mentioned in the following file.

```
root@opensips:/var/www/opensips-cp/config/tools/system/dialog#
gedit local.inc.php
```

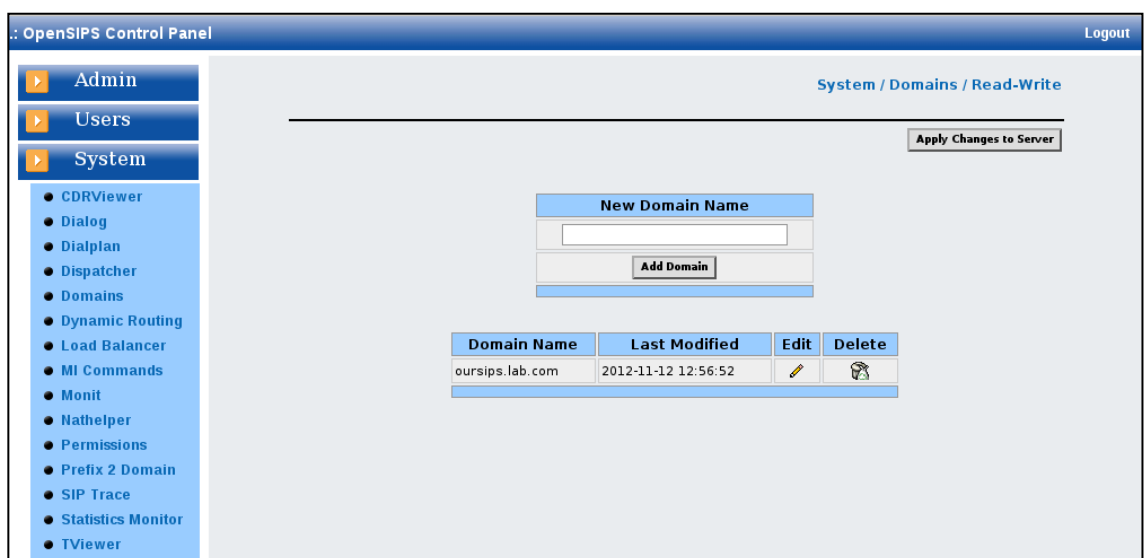
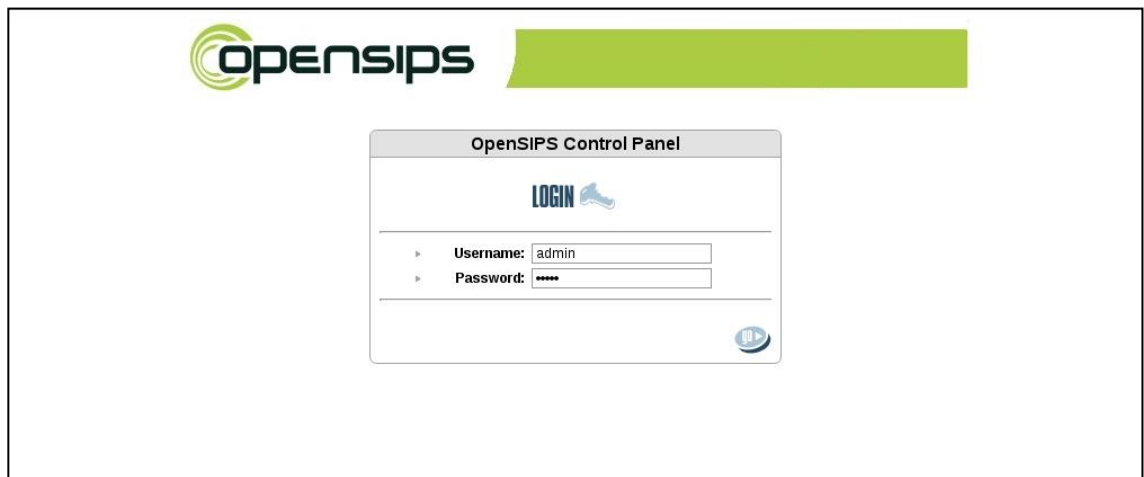
```
$box[1]['mi']['conn']="/tmp/opensips_fifo";;
```

Finally, restart the apache service with the following command.

```
root@opensips:~# /etc/init.d/apache2 restart
```

To configure OpenSIPS CP to the browser and type in the following command with local IP addresses to, e.g., 192.168.100.1 in test network's case and clicking the 'web' link should open the OpenSIPS CP. Username and password that we added was 'admin' and 'admin'. For we can manage (add/delete) users. We also need to specify the domain name as shown in the figures below.

```
http://192.168.100.1/opensips-cp/
```



OpenSIPS Control Panel

Logout

Admin

Users

ACL Management

Alias Management

User Management

System

Users / User Management / Read-Write

Add New User

Username:

user1

Domain:

oursips.lab.com

Email:

user1@aalto.fi

Alias Username:

123

Alias Type:

dbaliases

Password:

.....

Confirm Password:

.....

Register

Go Main

OpenSIPS Control Panel

Logout

Admin

Users

ACL Management

Alias Management

User Management

System

Users / User Management / Read-Write

New User added!

Username

Domain

ANY

Email:

☒ All Users:

☐ Online Users:

☐ Offline Users:

Search

Show All

Username	Email Address	Contacts	Alias	Group	Edit	Delete
user1@oursips.lab.com	user1@aalto.fi					
user2@oursips.lab.com	user2@aalto.fi					
user3@oursips.lab.com	user3@aalto.fi					
user4@oursips.lab.com	user4@aalto.fi					

Page: 1

Total Records: 4

Add New

Copyright © 2006-2011 OpenSIPS Project