Development and Validation of VoIP features in PacketCable Technology

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(Communication Engineering)

By

Ankit Bharatbhai Solanki

(10MECC14)

Department of Electronics & Communication Engineering

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2012

Development and Validation of VoIP features in PacketCable Technology

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(Communication Engineering)

By

Ankit Bharatbhai Solanki

(10MECC14)

Under the Internal guidance of

Prof. Shailesh Pandey

and

External guidance of

Mr. Shard Gupta

Department of Electronics & Communication Engineering

Institute of Technology

Nirma University

Ahmedabad-382 481

Declaration

This is to certify that

1. The thesis comprises of my original work towards the degree of Master of Technology in Communication Engineering at Nirma University and has not been submitted elsewhere for a degree.

2. Due acknowledgement has been made in the text to all other material used.

Ankit Bharatbhai Solanki

Certificate

This is to certify that the Major Project entitled **"Development and Validation of VoIP features in PacketCable Technology"** submitted by **Ankit Bharatbhai Solanki (10MECC14)**, towards the partial fulfilment of the requirements for the degree of Master of Technology in Communication Engineering of Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Internal Guide

Prof. Shailesh Pandey
(Assistant Professor, EC)

Programm Co-ordinator

Dr. D. K. Kothari
(Professor, EC)

HOD

Prof. A. S. Ranade
(Professor, EC)

Director

Dr K. Kotecha
(Director, IT, NU)

Acknowledgement

It is my immense pleasure to acknowledge the persons without whom this dissertation would be incomplete or even impossible.

I owe my deepest thanks to my internal guide Prof. Shailesh Pandey (Assistant Professor, Nirma University, Ahmedabad)and external guide Mr. Shard Gupta (Team Leader, ISS-HED, ST Microelectronics) to provide me the guidance throughout the dissertation. They are always there to support me whenever I need them.

I wish to thank first and foremost, Prof. A. S. Ranade, Head of Department, Electrical Engineering and Dr. D. K. Kothari, Program Coordinator, M.tech - Electronics & Communication(Communication Engineering), for allowing me to pursue my dissertation from ST Microelectronics, Greater Noida. I am thankful to ST Microelectronics for providing me the exposure to industry.

I consider it an honor to work with such a team headed by Mr. Mazhar Umer, Section Manager, ISS-HED. I would like to thank my team members Mr. Amandeep Singh (Technical leader), Mr. Sandeep Pandey (Technical Leader) and Mr. Vaibhav Bhadade (Senior Software Engineer) for providing me the space to work with them. I thank all my team members for sharing their experience with me and helping me in situation where I need them whether it is technically or personally.

I am deeply obliged to all the faculties, Support staff, Lab assistants of Nirma University, Ahmedabad for sharing their knowledge and experience with me.

I cannot find words to express my gratitude to my dearest and nearest friends whom I missed a lot during the period, for their unconditional love and support. I am indebted to all of them for the golden time of my life which I had spent with them. Raghav, Vyoma, Tushar, Preety, Vishal and Nidhi thanks a lot for being with me and making my life enjoyable.

Last but not the least I would like to acknowledge my parents without them I may not be able to acknowledge the above all personalities. It is their love, support and sacrifice because of which I am here at this stage of my life. I would also like to express my love to my younger brother Divyang for giving me the reason to live life.

- Ankit Bharatbhai Solanki
10MECC14

Abstract

PacketCable is a CableLabs specifications in effort to design and support convergence of voice, video and other IP based applications over the existing cable networks. PacketCable 2.0, the latest version of the PacketCable is based on the IP Multimedia Services developed by the 3^{rd} Generation Partnership Protocol. PacketCable 2.0 is a SIP based architecture for providing IP multimedia services. Present work is focused on providing analog telephony features on the managed IP network. SLIC-SLAC chip has been interfaced with the Set Top Box to avail with the basic analog telephony functions which are provided by the PSTN. An individual hardware component as Cable Modem is provided with the set top box to obtain the IP connectivity. Connection between user and service provider secure between the user and service provider by means of provisioning of the User Equipment. Compression of voice frames with the help of the CODEC available and converting them in the RTP packets will help in transmission on the IP network. Controlling of packets flow between the sender and the receiver is done with the help of the Real Time Control Protocol (RTCP). Extended report of RTCP which contains the different parameters related to source and destination. These parameters can be used for different purposes which are helpful in improving the voice quality. Voice quality in the PacketCable can be affected by signal level, echo, delay, codec and other packet impediments. Effect of this parameter and their possible solutions are discussed. Measuring delay for the different case is also shown. Jitter buffer which is used at the receiver side to compensate for the jitter. Adaptive Jitter Buffer which is used in the PacketCable has the range of 20-420msec theoretically. Adaptive jitter buffer with its mechanism of how the jitter will be controlled for the PacketCable has been explained in detail. Algorithm to calculate the packet loss and packet discard is also discussed. Adaptive Jitter Buffer used in the PacketCable behaves according to the theoretical limit. PacketCable in all is the new technology developed and validated for the new era of communication.

Contents

List of Figures

List of Tables

List of Abbreviation

3GPP	3rd Generation Partnership Protocol
ADPCM	Adaptive Differential Pulse Coded Modulation
ADSC	Audio Stream Control
API	Application Programming Interface
BPI	Baseline Privacy Interface
CELP	Code Excited Linear Prediction
CM	Cable Modem
CMTS	Cable Modem Termination System
CODEC	Coder Decoder
CSCF	Call Session Control Function
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DOCSIS	Data Over Cable System Interface Specification
DQoS	Dynamic Quality of Service
DTMF	Dual Tone Multi Frequency
DVB	Digital Video Broadcast
E-CSCF	Emergency Call Session Control Function
E-DVA	Embedded Digital Voice Adaptor
ENUM	E.164 Numbering
FQDN	Fully Qualified Domain Name
HD	High Definition
HFC	Hybrid Fiber Coaxial
HSS	Home Subscriber Server
HTTP	Hyper Text Transfer Protocol

I-CSCF	Interrogating Call Session Control Function
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPV4	Internet Protocol Version 4
IPV6	Internet Protocol Version 6
ISDN	Integrated Service Digital Network
ISUP	ISDN User Part
JTAG	Joint Test Action Group
KDC	Key Distribution Server
LAN	Local Area Network
MAC	Media Access Control
MG	Media Gateway
MGC	Media Gateway Controller
MIB	Management Information Base
Modem	Modulator Demodulator
MSO	Multimedia System Operator
MTA	Multimedia Terminal Adaptor
NAT	Network Address Translator
NCS	Network based Call Signaling
OSS	Operational System Support
PACT	Provisioning Activation Configuration Tool
PAM	Pulse Amplitude Modulation
PBX	Private Branch Exchange
PCM	Pulse Coded Modulation
P-CSCF	Proxy Call Session Control Function
POTS	Plain Old Telephone System
PSTN	Public Switched Telephone Network
PUI	Public User Identity

QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RFI	Radio Frequency Interface
RJ	Registered Jack
RKS	Record Keeping Server
RTP	Real-time Transport Protocol
SCDMA	Synchronous Code Division Multiple Access
S-CSCF	Session Call Session Control Function
SG	Signaling Gateway
SIP	Session Initiation Protocol
SLAC	Subscriber Line Audio processing Circuit
SLIC	Subscriber Line Identification Circuit
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
SS7	Signaling System 7
STB	Set Top Box
STUN	Simple Traversal of UDP through NAT
TDMA	Time Division Multiple Access
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identiter
USB	Universal Serial Bus
VoIP	Voice over Internet Protocol

Chapter 1

Introduction

1.1 Voice over IP

VoIP is simply the transport of voice traffic using the Internet Protocol (IP), hardly a surprising definition. However, it is important to note that VoIP does not automatically imply voice over the Internet. The Internet is a collection of interconnected networks, all using IP. The connections between these networks are used by anyone and everyone for a wide range of applications, from e-mail to file transfer to electronic commerce (e-commerce). One of the greatest challenges to VoIP is voice quality and one of the keys to acceptable voice quality is bandwidth.

If we are to ensure that sufficient bandwidth is available to enable high-quality voice, then we need to control and prioritize access to the available bandwidth. One person transferring a huge file may cause other users' transactions to proceed more slowly. As a result, voice quality over the Internet today may vary from acceptable to atrocious.

As technology changes and as more and more bandwidth is made available, it is possible that high-quality voice over the Internet may become the norm rather than the exception. However, that day is still not at hand. Internet telephony may be considered a subset of, or a special case of, VoIP.

1.2 PacketCable Motivation

PacketCable is a CableLabs-led project initiated by cable operators with the Objective of defining a QoS-enabled, IP-based services delivery platform which utilizes the capabilities of the DOCSIS access network[12].

Providing voice over IP infrastructure is not the only way to provide the services. The same service can also be provided on the cable MSOs network using a portion of RF spectrum. One way to do provide RF channel for the user is to have Time Division Multiplexing and provide a circuit switched network. But the main disadvantage of this system would be inefficient use of the available bandwidth. The reason for this is when the user is idle the allocated bandwidth to that user cannot be used by other user and is waste of bandwidth. So to utilize the bandwidth available, packet switched network is the best solution. PacketCable provides the same concept of packet switched network to provide voice communication over the existing cable MSO network. Cable providers or Multiple Systems Operators (MSOs) are expanding their competitive offerings in an effort to better serve their customers and increase revenue. High-speed data, video-on-demand, interactive gaming, teleconferencing, and telephony are some examples of these value-added services. PacketCable provides a framework MSOs can use to ensure the successful deployment of these services over a broadband cable network. Although many of the concepts and technologies presented in this book could be applied to implementing any one of these services, the primary focus is on telephony. As you will soon learn, telephony is the scope of PacketCable 1.x. PacketCable 2.0 and PacketCable Multimedia address a broader scope-the delivery of any type of voice, video, and/or data multimedia service over a broadband infrastructure.

In recent years, MSOs have made a substantial investment in upgrading their existing infrastructure into a bi-directional Hybrid Fiber Coaxial (HFC) system capable of providing reliable high-speed Internet access and digital cable television. The bi-directionality component permits the transmission and receipt of IP packets, and it is this IP network that provides high-speed data (HSD) services. Additionally, this same IP network can also be used to provide VoIP voice services at a much smaller cost than that of the Incumbent Local Exchange Carriers (ILEC's) legacy circuit switched networks.

The PacketCable project defines specifications for the various components in the architecture. This architecture constitutes a baseline of what is needed to support broadband services over a DOCSIS/IP infrastructure. The creation of specifications enables you to have different vendors for each component, and as long as components are PacketCable certified or qualified, they can work together without issue.

The PacketCable architecture supports several end-to-end functions, such as signaling for services, quality of service (QoS)-enabled media transport, security, provisioning of the client device, billing, and other network management functions. The initial PacketCable charter was to design a network that provides telephony services with voice quality as good as the existing public switched telephone network (PSTN) service and with the same feature suite. By using an IP backbone, new features previously not possible with a circuit-switched architecture are now possible as well. A majority of this deals with issues associated with implementing voice; however, you will also get a glimpse into the more generic case of implementing any multimedia solution

1.3 Different versions of PacketCable

PacketCable is a CableLabs specification effort designed to support the convergence of voice, video, data, and mobility technologies. PacketCable defines architecture and a set of open interfaces that leverage emerging communications technologies, such as the IETF Session Initiation Protocol (SIP), to support the rapid introduction of new IP-based services onto the cable network. There are different versions of PacketCable available from its evaluation. First version of PacketCable is known as the PacketCable 1.0 and the latest version of PacketCable is PacketCable 2.0

1.3.1 PacketCable 1.0

PacketCable 1.0 defines an end-to-end architecture for the delivery of residential voice telephony over IP, commonly referred to as digital voice. The end-to-end architecture describes a complete system that includes: device provisioning, call signaling, event messaging (accounting), configuration management, QoS, PSTN interconnection, and security.

Several factors differentiate PacketCable residential voice telephony services from traditional “IP telephony” services.

For example: PacketCable is a phone-to-phone service rather than a personal computer-based telephony service. PacketCable services are guaranteed priority delivery on the DOCSIS access network ensuring a consistent, high-quality service.

PacketCable services are not delivered over the public Internet. Cable operators deploying PacketCable use managed IP backbones that provide service delivery consistent with that of the DOCSIS access network.

1.3.2 PacketCable 1.5

At a very high level, the PacketCable 1.0 architecture contains three networks: the “DOCSIS HFC Access Network”, the “Managed IP Network” and the PSTN. The Cable Modem Termination System (CMTS) provides connectivity between the “DOCSIS HFC Access Network” and the “Managed IP Network.” Both the Signaling Gateway (SG) and the Media Gateway (MG) provide connectivity between the “Managed IP Network” and the PSTN.

In addition to containing all the functionality of PacketCable 1.0, PacketCable 1.5 extends the PacketCable residential voice capability with new capabilities such as Fax and Modem support, analog trunking for PBXs.

1.3.3 PacketCable 2.0

PacketCable 2.0 is a set of new specifications being developed by CableLabs, its member companies, and leading communications equipment vendors. The objective of this effort is to create an architecture that extends cable's existing Internet Protocol service environment to accelerate the convergence of voice, video, data, and mobility technologies. PacketCable 1.0 and 1.5 supports NCS-based clients for telephony services. PacketCable 2.0 adds support for SIP based clients with a variety of capabilities.

The PacketCable 2.0 core service architecture is based on common standards technologies such as the Session Initiation Protocol (SIP) and the IP Multimedia System (IMS). By utilizing a common core, application servers in the network are able to deliver services to a range of clients independent of the topology of the access network. Intentionally non-service specific, PacketCable 2.0 takes a modular approach and defines the functional elements and communication interfaces necessary to support a wide array of service types.

CableLabs has defined a number of applications that utilize the PacketCable 2.0 core network such as residential SIP telephony, cellular integration, business services, and HD voice.

1.4 Problem Statement

The primary objective of the project is to understand and explore this new technology named PacketCable. Getting good knowledge of the PacketCable along with its architecture and different components used. Understand and getting used to the Hardware and software going to be used for the PacketCable. Establishment of set up which is required for the purpose of validating PacketCable on DOCSIS environment. Study different Codec used in the PacketCable for the voice communication. Go through different parameters which affect the voice quality in PacketCable. Measure these parameters and try to optimize them. Proper Understanding of jitter buffer used in general for VoIP and try to visualize the jitter buffer used for the PacketCable and if possible try to improve the performance.

1.5 Thesis Organization

Rest of the thesis is organized as follows.

Chapter 2 describes the basic difference between PacketCable and 3GPP IMS. The next section of the chapter describes architecture of the PacketCable 2.0 with the proper explanation of each and every part of the architecture. Physical and MAC layer protocol used in the case of the PacketCable is the DOCSIS (Data over Cable System Interface Specification) with its different version is explained.

Basic functioning of the Set Top Box which is used as an interface to connect to the existing cable network is explained in the **chapter 3**. A Subscriber Line Interface Circuit (SLIC) - Subscriber line Audio processing Circuitry (SLAC) which is provided to accomplish the services provided by the PacketCable. Interfacing and configuration of this SLIC-SLAC device and setting up different parameters are explained in the next section. Basic hardware block diagram with their functioning is followed in the next section. Provisioning and SIP signaling is explained which is an integral part of the PacketCable to provide the security in the PacketCable. In the last section of the chapter a basic call procedure with its flow of SIP signaling in call establishment and call termination is explained.

Chapter 4 discussed about how the analog voice can be converted to the digital voice using the codec. Codec used in the case of them PacketCable is G711 and G729. After the digitization of the voice comes the part of making these digitized voice frames into the form of the IP packets so that they can be sent over the IP network. To convert a voice frames in voice frames and make the communication in a real time communication RTP protocol has been used. RTCP is the control protocol which is used to convey the end to end control information between the sender and the receiver. RTCP-XR which is the extended version of the RTCP is also explained in the later section.

Chapter 5 focuses on the voice quality in the PacketCable. Different parameters which can affect the voice quality are explained in detail in this chapter.Measurement and result of some of the parameters are there in this chapter.

PacketCable uses adaptive jitter buffer to avoid the jitter. Jitter buffer with its mechanism of how it handles the jitter has been explained in **Chapter 6**.

Chapter 7 Conclusion and Future Work comments on the results obtained. It also includes the future work which is going to be performed for the further enhancements of the PacketCable technology.

Chapter 2

PacketCable Architecture

2.1 PacketCable and 3GPP IMS

PacketCable is based on the IP Multimedia Subsystem (IMS) as defined by the 3rd Generation Partnership Project (3GPP). 3GPP produces technical reports and technical specifications for 3rd mobile system networks and GSM[12]. The scope of the 3GPP includes development of the SIP based IP communication for the mobile networks. The main function of the PacketCable is to align with the existing IMS architecture and specification and try to enhance the capabilities. To achieve the same purpose PacketCable uses the same functional entities and reference points which are used in the IMS. The reason behind using the same component is to minimize the effort and cost required to deploy PacketCable architecture.

2.2 Enhancement in the IMS using PacketCable

- Support for Quality of Service (QoS) for IMS-based applications on DOCSIS access networks, leveraging the PacketCable Multimedia architecture.

- Support for additional access signaling security and UE authentication mechanisms

- Support for provisioning, activation, configuration, and management of UEs
- Support for regulatory requirements such as number portability, preferred carrier, and PacketCable lawful interception

2.3 PacketCable 2.0 Reference Architecture

The Figure 2.1 below shows the reference architecture of the PacketCable 2.0. From the figure we can see that the whole of the PacketCable 2.0 architecture is being divided into the logical or functional groupings. In the figure below UE (User Equipment) or MTA (Multimedia Terminal Adapter) stands for the one who is receiving the services of the PacketCable. A UE may be an IPV4 enabled node or IPV6 enabled node according to that it will get IP address [12].

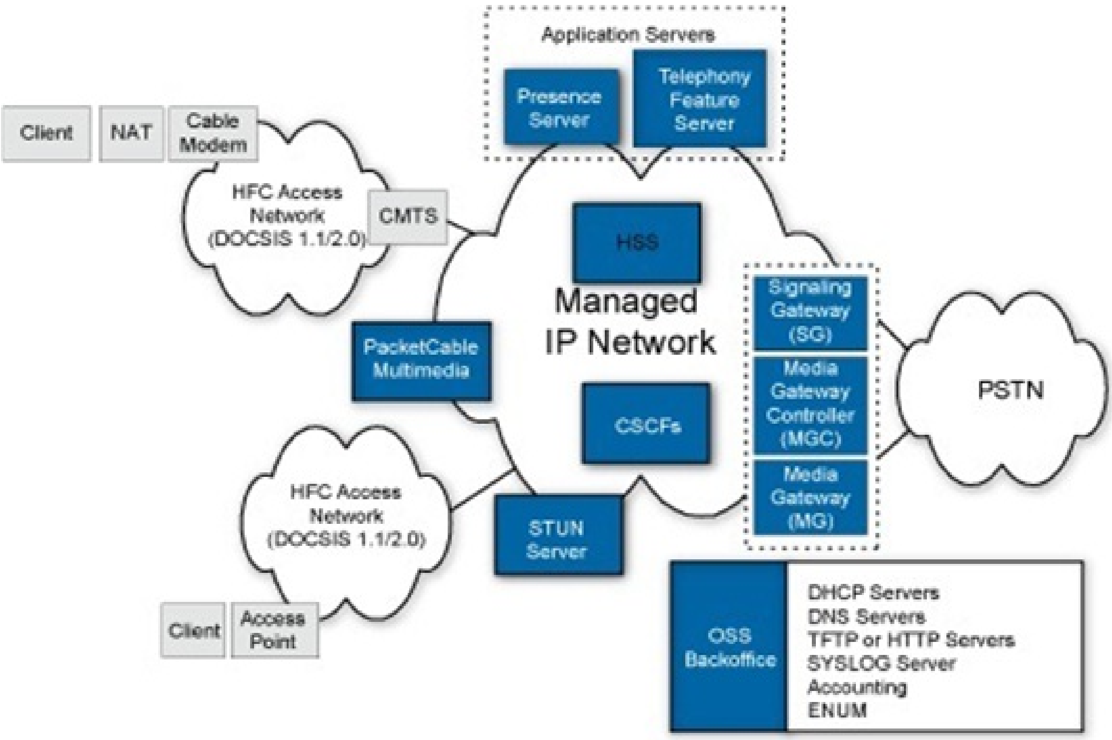

Figure 2.1: PacketCable 2.0 reference architecture[12]

PacketCable assumes a model composed of users, UEs and Public User Identity (PUI). PUI is a unique identification given to each of the UEs. This PUI may be an E.164 number or an alphanumeric identifier. For example a user may have the multiple numbers of UEs and each of them must have their own unique PUI.

2.3.1 Client

- The user who is using the PacketCable services with the help of the UE is known as the client.
- Sometime it may happen that NAT (Network Address Translator) or Firewall may be present on the UE side.
- As NAT may change the IP address and Ports of the packets and Firewall may restrict the connectivity to the network.
- In this scenario to avail user with PacketCable functionality, necessary care has to be taken.
- Client along with the Cable Modem embedded in PacketCable architecture is known as either e-MTA or e-UE.

2.3.2 Access Point

To provide the IP connectivity to the UEs through the other kind of the access network rather than DOCSIS. e.g. it may be Wi-Fi, LAN, Bluetooth etc.

2.3.3 Cable Modem

- Cable Modem is used to connect UE to the DOCSIS network.
- The cable modem is defined in the DOCSIS specifications and provides a means for sending IP packets across the RF or HFC network.

2.3.4 Cable Modem Termination System (CMTS)

- Same As the CM, CMTS is also defined in the DOCSIS specification.
- CMTS is basically located at the MSOs headend and in conjunction with the CM it is used to provide broadband data transport service over the cable HFC network.
- It is the first component inside the trust boundary.

2.3.5 PacketCable Multimedia

- PacketCable Multimedia defines an IP-based platform for delivering QoS-enhanced multimedia services over DOCSIS access networks.
- The PacketCable Multimedia architecture defines an Application Manager and Policy Server.
- The Application Manager translates application specific resource requests to PacketCable Multimedia requests and forwards these requests to the Policy Server.
- The Policy server enforces network policy and installs the resulting QoS policy on the CMTS for enforcement.
- This platform allows the core capabilities of PacketCable to support a wide range of IP based service beyond telephony. These are QoS, authorization and admission control, accounting, security etc.

2.3.6 Call Session Control Functions (CSCFs)

There are four types of CSCFs in the case of the PacketCable architecture. Each of them having their own functionality to perform.

Proxy Call Session Control Function (P-CSCF)

- A UE enters into the SIP infrastructure of the PacketCable through the P-CSCF.

- It is the trust boundary between the trusted and entrusted parts of the network. Entrusted components in the PacketCable network are UE and its embedded Cable Modem. And the components in the SIP infrastructure are the trusted.

- It performs routing of SIP message from UE to the I-CSCF or S-CSCF and vice versa.

- Maintains security between itself and UE. And also authenticate the Personal User Identity (PUI).

- It is a solution to the UEs which do contain NAT with them. P-CSCF provides NAT bindings for SIP signaling.

- Interaction with the PacketCable Application Manager and QoS management.

Serving Call Session Control Function (S-CSCF)

- S-CSCF performs the function of the registrar for a PUI by giving them a Globally Routed User Agent URIs.

- S-CSCF also performs SIP user authentication and authorization.

- Routes the messages to the P-CSCF of UE which are served by the S-CSCF

- Routes the messages to the I-CSCF of UE which are not served by the S-CSCF

- Subscriptions are associated with the S-CSCF and subscription data are stored in the Home Subscriber Servers.

Interrogating Call Session Control Function (I-CSCF)

- I-CSCF interacts with the HSS to determine the binding between the subscription and S-CSCF.

- Querying the HSS to obtain the S-CSCF and then routing SIP requests from another network operator to the correct S-CSCF.

Emergency Call Session Control Function (E-CSCF)

- It is used for routing only the emergency sessions.

2.3.7 Home Subscriber Server (HSS)

- HSS is responsible for the association between a subscription and S-CSCF, subscription profile information, subscription security and transparent data for usage by Application Servers.

- HSS supports session establishment for terminating traffic, it provides information on which S-CSCF is assigned to handle a Public User Identity

- It also supports security by various authentication schemes and also provides service provisioning by providing access to the service profile data for use by the S-CSCF.

- The HSS may also store application related data for the application server.

2.3.8 STUN Server

- A STUN Server is an entity that receives STUN requests, and sends STUN responses. STUN requests are typically binding requests which are used to determine the bindings allocated by NATs.

- The UE sends a Binding Request to the server, over UDP. The server examines the source IP address and port of the request, and copies them into a response that is sent back to the UE.
- The STUN server is a functional component within the P-CSCF is used by SIP UEs in order to maintain the NAT bindings for signaling. These STUN messages also act as keepalives, allowing the UE to determine PCSCF availability and detect NAT reboots.

2.3.9 Application Servers

Application server provides application specific services.

Presence server

- The Presence Server Functions is a group consisting of specialized Application Servers that support exchange of Presence data. They act as the focal point for connecting sources of presence information and interested parties.

Telephony Feature Server

- It is the application server specifically dedicated for the purpose of the telephony features.

2.3.10 PSTN Gateway

PSTN gateway consists of three basic Gateways

Signaling Gateway (SG)

- It performs signaling at transport layer between SS7 based transport and IP based transport used in all the releases of PacketCable.

- The SG does not interpret the application layer, but does interpret the layers needed for routing signaling messages.

Media Gateway Controller (MGC)

- MGC performs protocol conversion between SS7 ISUP messages and PacketCable call control protocols.

- It also provides connection control of the media channels in the Media Gateway (MG).

Media Gateway (MG)

- The MG provides bearer channel conversion between the circuit switch network and the IP RTP media streams in the PacketCable network.

- The MG may introduce codecs and echo cancellers, etc., as needed to provide the bearer channel conversions.

2.3.11 Operational Support System BackOffice (OSS)

- Operational Support Systems provide various functions like accounting and UE provisioning. It also supports accounting information, provisioning of UE, distribution of the IP addresses to the UEs. And for this purpose different servers are available in the Operational Support System

Dynamic Host Configuration Protocol (DHCP) Server

- A DHCP server is used when the local network is in control of the service provider.

- It provides UE the information to connect to the network such as IP address and DNS servers.

- It is an integral part of the PacketCable for the purpose of the provisioning framework.

Domain Name System (DNS) Server

- A DNS server is used to resolve DNS entities (e.g., FQDNs, SRV records) into network addresses and vice-versa.
- A DNS server is used by E-UE and UE Provisioning Frameworks and for signaling address resolution.

ENUM Server

- An ENUM server is used to store and translate E.164 numbers to SIP URIs.

Provisioning Server

- It is the most important component in the PacketCable architecture deployed for the purpose of the provisioning, configuring and management of the eUEs.
- Provisioning can be of three types Basic, Hybrid and secure.

Key Distribution Centre (KDC) Protocol

- It is another protocol which is associated with the process of provisioning of E-DVA.
- In the context of the PacketCable KDC protocol is for the security purpose.
- It is used for key management on the interfaces between an embedded UE and the Provisioning Server as part of the E-UE Provisioning Framework.
- It authenticates the DVA and provides them with the tickets to permit secure access of the rest of the provisioning process.

Configuration Server

- The Configuration server is responsible for providing the configuration to an embedded UE as part of the E-UE Provisioning Framework.
- For embedded UEs this is accomplished using TFTP, and optionally HTTP protocols.

Syslog Server

- Within the E-UE framework, embedded UEs can transmit management event notifications using the syslog protocol. The network entity receiving these notifications is the syslog server.

Record keeping Server (RKS)

- This server is used to keep the accounting messages from the different components of the PacketCable network.
- And this information can then be further provided to other system such as billing server.

2.4 DOCSIS

DOCSIS stands for Data over Cable System Interface Specifications. DOCSIS is an international telecommunication standard which provides specification for transmission of high speed data on the existing cable television system. It defines physical and at data-link layer DOCSIS MAC layer protocol for IP connectivity over HFC cable plant. Adoption of DOCSIS accelerates the deployment of data-over-cable services and ensures interoperability of equipment throughout system operators infrastructures. DOCSIS specification has different releases containing DOCSIS 1.0, DOCSIS 1.1, DOCSIS 2.0 and DOCSIS 3.0. Latest version of the DOCSIS should also be able to provide backward compatibility i.e. DOCSIS 3.0 should also be able to support the DOCSIS 1.0 network.

Separate parts of RF spectrum are used for the communication in each direction. Upstream direction is for the communication between Cable Modem (CM) and Cable Modem Termination System (CMTS). Upstream communication supports the channel width of 200 kHz, 400 kHz, 800 kHz, 1.6 kHz or 3.2 kHz. Latest version of DOCSIS 2.0 onwards also supports upstream channel width of 6.4 kHz. Modulation used in the upstream direction is generally either QPSK or 16-QAM.

Downstream communication is the reverse case of upstream i.e. from CMTS to CM. channel width of downstream direction is 6MHz.Downstream signals are modulated using 64-QAM or 256-QAM.

2.4.1 DOCSIS 1.0

DOCSIS 1.0 defines Radio Frequency Interface (RFI) the specifications for the bidirectional transfer of IP packets over HFC network between Cable Modem (CM) at subscriber and Cable Modem Termination System (CMTS) at MSOs headend. This version of DOCSIS is not useful to establish the PacketCable network.

At the MAC layer DOCSIS 1.0 uses Time Division Multiple Access (TDMA) for upstream transmission. Security of the data transfer between CM and CMTS or vice versa is provided with the help of the Baseline Privacy Interface (BPI). BPI provides 56-bit DES encryption of data transfer between CM and CMTS.

2.4.2 DOCSIS 1.1

DOCSIS 1.1 builds upon DOCSIS 1.0 by incorporating quality of service (QoS) functionality into the DOCSIS MAC protocol. Because of the introduction of the QoS, DOCSIS 1.1 can be used for the real time application such as voice.

Along with QoS DOCSIS 1.1 provides other features like security using BPI+, fragmentation, support of Simple Network Management Protocol (SNMP) Management Information Bases (MIB). At the MAC layer DOCSIS 1.1 is supported with TDMA for upstream transmission.

2.4.3 DOCSIS 2.0

DOCSIS 2.0 provides Advanced-TDMA and Synchronous-CDMA at MAC layer for the upstream direction. Earlier version of the DOCSIS only supports either QPSK or 16-QAM for the upstream direction. But DOCSIS 2.0 supports QPSK and 16-QAM, 8-QAM, 32-QAM, and 64-QAM modulation encodings. DOCSIS 2.0 also uses BPI+ for the security purpose.

2.4.4 DOCSIS 3.0

DOCSIS 3.0 decouples the physical layer of DOCSIS from the MAC layer. This allows for channel bonding where multiple physical upstream and downstream channels can be combined to form a larger logical channel. DOCSIS 3.0 also supports the use of IP version 6 and the migration from IPv4. For security it uses upgraded version of the BPI renamed as security (SEC).

Chapter 3

PacketCable Components and Provisioning

PacketCable uses existing cable network to provide the services. Set Top Box is the device which is used to connect to the existing cable network. A small box sits on top of a standard TV set is called Set-top Box STB is central to this migration from analog-to- digital broadcasting. One of the latest devices for interactive home entertainment is the set-top box - a device that represents the ideal marriage of information and entertainment. The set-top box promises to enhance the home entertainment experience.

Set-top Boxe (STB) act as a gateway between your PC or telephone, satellite or cable feed (incoming signal.) In terms of ITV, the STB receives encoded and/or compressed digital signals from the signal source (satellite, TV station, cable network, etc.) and decodes (and/or decompresses) those signals, converting them into analog signals displayable on your television. The STB also accepts commands from the user and transmits these commands back to the network, often through a back channel.

Interactive television STBs can have many functions such as television receiver, modem, game console, Web browser, a way of sending e-mail, Electronic Program

Guide (EPG), even CD ROM, DVD player, video-conferencing, cable telephony etc.To provide interactive services, the Set-top Box Figure 3.1, from the standpoint of its hardware, needs four important components: a network, an interface, a buffer, as well as decoder/synchronization Hardware.

Figure 3.1: Set Top Box

- **Components**

  1. **The network interface:** Allows the user to receive data from the server and send data back to the server, in a manner that the server can understand it.
  2. **The decoder:** In order to save storage space, disk bandwidth, and network bandwidth, video or sound are compressed before sending over network. Thus, the end-users need a decoder to decode the incoming stream's data before it's viewable. This is part of what a modem does. The decoding process is sometimes known as Demodulation or Heavy Lifting.
  3. **The buffer:** Due to delay jitters in the network, the arrival time of a video stream may vary. For continuous consistent playback for the viewer

the stream is often received one or even a few seconds before it's actually seen by the end-user. This way the viewer dont realize any fluctuations in signal received.

4. **Synchronization hardware:** In simple words it is a kind of a computer that translates digital signals into a format that can be viewed on a television screen. Today digital TV usually requires a set-top box (STB), which is used to decode and tune digital signals and converts them to a format that is understood by analog TV.

In a STB, the tuner receives a digital signal from a cable, satellite or terrestrial network and isolates a particular channel.

- **Characteristics of a Set-Top Box**

  1. Decodes the incoming digital signal
  2. Verifies access rights and security levels
  3. Displays studio-quality pictures on TV set

- **Working Of a Set-Top Box**

  1. Tuner receives a digital signal from a cable/satellite or terrestrial network.
  2. The Demodulator converts it into the binary format, checks for errors and forwards it to the De-multiplexer
  3. De-multiplexer extracts the audio, video and data from the binary stream and sends it to the appropriate component

3.1 SLIC-SLAC Chip

To provide the telephony functionality to the user along with the television using the same set top box an external voiceport chip is provided. This chip provides the functionality of the PSTN. SLIC is the circuitry which is present in the telecommunication network. It is an interface between the local loop of the POTS and the PSTN. The function of this line card is to perform the BORSCHT functions. BORSCHT is an acronym used for the functionality checked by the line card. B (Battery feed), O (Overvoltage detection), R (Ringing), S (line Supervision), C (Codec), H (Hybrid), T (Testing).

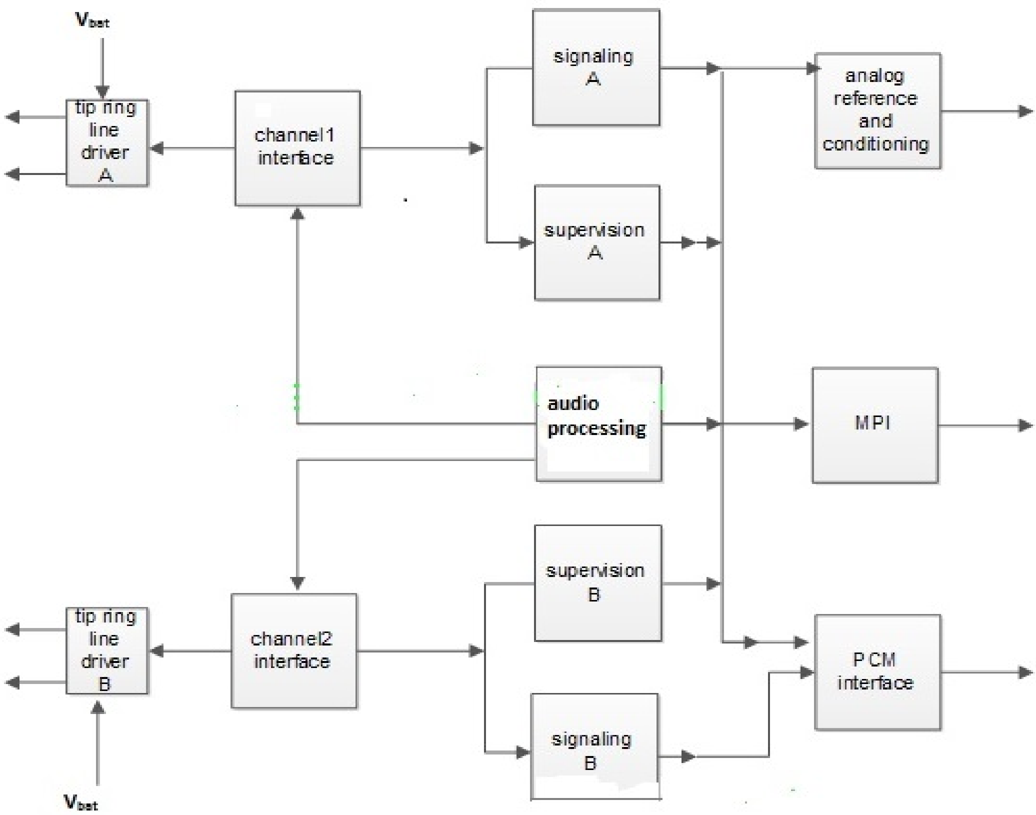

Figure 3.2: VoicePort chip block diagram

The VoicePort chipset Figure 3.2 shown above is a dual channel telephone line interface targeted for the voice application. The chipset performs all the necessary functions required for the purpose of the telephony. It works on input voltage as low as +3.3V and can handle voltage maximum up to +35V.Chipset generates - 48V required to drive the ring terminal of both the channels internally with the help of the switching regulators provided. It has high efficiency and corresponding low power dissipation. All ac, dc and switching regulator parameters can be programmed with the help of the Microprocessor interface.

The chipset provides the functionalities as shown below:

- It performs Battery feed, Overvoltage protection, integrated Ringing, line Supervision, CODEC, Hybrid (2wire to 4wire), and line Test (BORSCHT) functions for both the channels.
- Supervision circuitry performs loop supervision, off hook detection, on hook detection, line testing for fault, impedance matching of network, 2wire to 4wire impedance balance, ring trip detection.
- Signaling circuitry different tone generation and control like ringing ring back tone, dial tone, DTMF tone, caller ID.
- Audio processing performs the conversion of the voice from analog to digital. This digital signal is then applied to the user programmable filters which apply the gain to the samples.

So from this we can say that whole of the Plain Old Telephony System has been converted into the small chip. These functions are also software programmable.

As already mentioned above that all the functionalities are software programmable, filters present in the chipset are also programmable. To program these filters one has to change the values of set of the coefficient. To calculate the coefficients filter coefficient generator tool has been used. With the necessary data given as the input it will calculate the respected coefficients set for the filters. Along with the filters coefficient set it will calculate the values of the return loss for the purpose of the impedance matching with the rest of the network.

The generated values of these filter coefficients are than applied to the tool known as VoicePort chipset configuration which generate the different profiles for the proper configuration of the VoicePort chipset. This is a tool in which we are generating different parameters such as ringing profile, ac profile, dc profile, cadence profile, metering profile. User will enter the values according to the need of the application otherwise they all have their own by default values. And the result of the profile wizard would be an API to allow easy integration with system software and quickly enable implementation of the required product features.

3.2 Basic Hardware Block Diagram

Figure 3.3 shows the interfacing of the SLIC-SLAC with the Set Top Box to obtain the PacketCable technology.

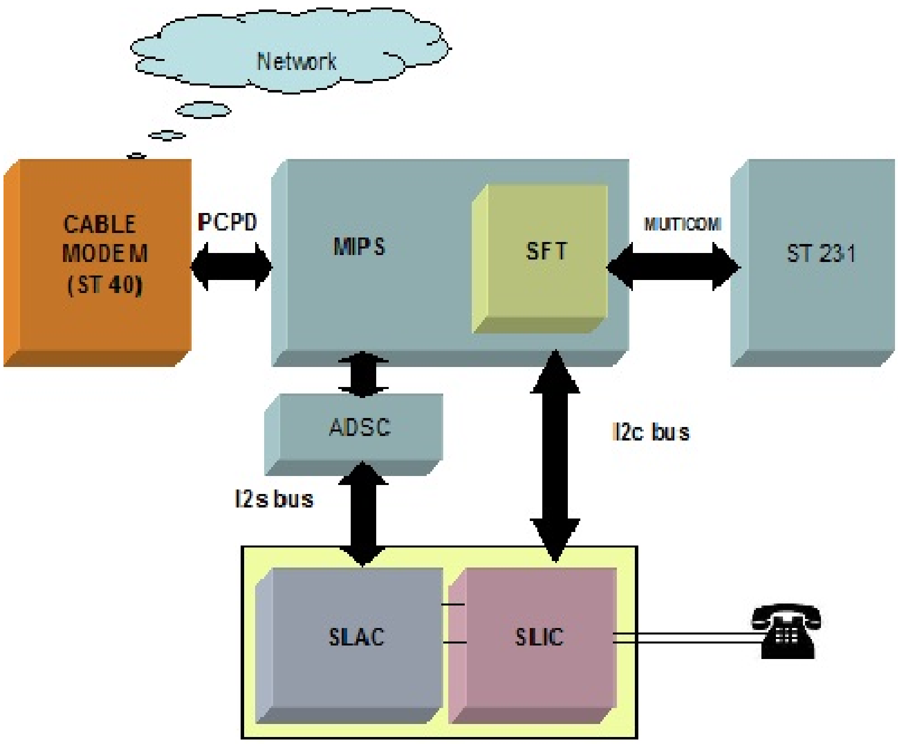

Figure 3.3: Hardware block diagram

3.2.1 Audio Stream Controller (ADSC)

Audio stream controller (ADSC) is a processor which acts a DMA.It takes PCM data from ATA and provides it to MIPS and similarly takes the PCM from MIPS processor and writes it to ATA.

3.2.2 Embedded Cable Modem (eCM-ST 40)

A cable modem is a type of network bridge and modem that provides bi-directional data communication via radio frequency channels on a HFC and RFC infrastructure. Cable modems are primarily used to deliver PacketCable Software Architecture consists of building blocks which are required to provide voice over IP (VOIP) telephony services over a cable network. Cable modem St 40 runs on OS 21.

3.2.3 Audio CODEC (ST231)

Audio codec is a DSP processor which performs CPU intensive MIPS processes related to voice. It also contains fax data pump. It also performs echo cancellation process.

3.3 Board (STB) Flashing

Board flashing is the procedure in which we are bringing the board up for the PacketCable 2.0 using flash memory. In this procedure we are flashing board with the boot loader file and kernel image in the flash memory of the board. The flash contents on the board will be non volatile. Kernel image will contain Linux kernel, directory structure, libraries, eCM binary, PacketCable2.0 binary, SLIC driver.

The basic component required for the board flashing is shown in Figure 3.4:

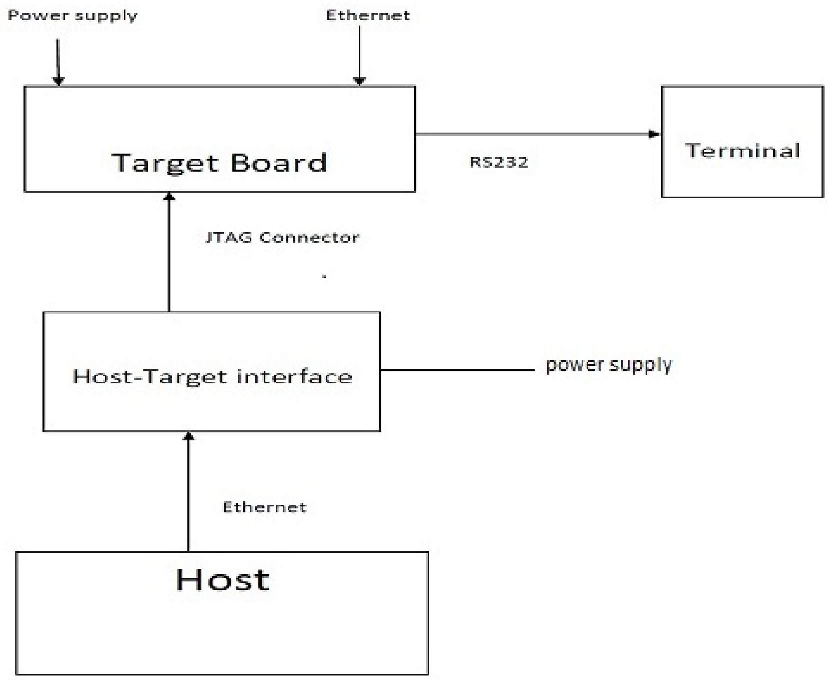

Figure 3.4: Board flashing arrangement

Host is the one which contains the all required files to be flashed which should be a LINUX machine having STAPI tool chain. Target is the board which has to be flashed. Serial connection with the terminal has been provided to observe and perform the flashing procedure. To connect host to the target a host target interface has been provided. This interface use JTAG connector to connect to the host. In the board flashing procedure major steps includes are erasing flash memory, flashing boot loader, flashing kernel image. After flashing procedure, the board will get its ETH0 IP address, all the require eCM binary, loadable modules permanently and then after there is no need to connect to the host again.

3.4 Multimedia Terminal Adapter (MTA)

MTA Figure 3.5 is a device customer will direct interface with in order to receive the services of telephony. Thus MTA is where the users analog phone will be connected.

MTA provides FXS interface port for phone connectivity using RJ-11 cables. In general we can say that the MTA is a device which collects the analog voice from the telephony interface and digitize it and send it to the network.

MTA are basically of two types standalone MTA and embedded MTA. Embedded MTA is a device which has an FXS port along with the integrated DOCSIS cable modem on the same physical device. A standalone MTA is a device which is physically a separate device from the cable modem.

MTA along with the analog to digital conversion also provides all the functionality provided by the PSTN. These functionality includes dial tone, ring tone, call waiting tone etc.

MTA reside on the user side so MTA is an untrusty component. Also as all the MTA traffic travels through the HFC network which is a shared media. So there are chances of theft of services. To keep the MTA functionality controlled and portable it must be provisioned remotely.

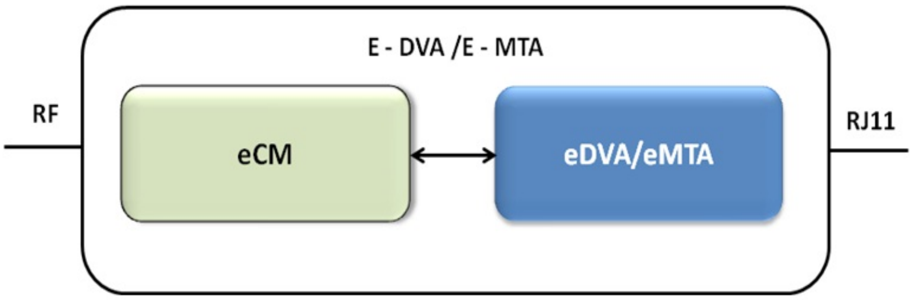

Figure 3.5: E-MTA Device

3.5 Cable Modem

Cable modem is defined in DOCSIS specifications and provides means for sending IP packets over the RF or HFC cable network. MSOs cable feed is connected to the Cable modem. This feed is typically split using a simple two-way splitter with the MSO feed as the input to the splitter and one of the outputs to the cable modem and the other to the television or cable set-top box (STB) for video service. Cable modem also needs to be provision as the MTA but they both get provisioned separately with their own configuration file. Both the cable modem and MTA has their own MAC address, IP address and FQDN (Fully Qualified Domain Name)[10].

The customer connects his network to the cable modem via an Ethernet, USB, or wireless interface. This connection can be direct or through some networking device such as a router or switch. The modem takes packets from the LAN interface (Ethernet, USB, or wireless) and transfers them out the cable interface to the MSO. These IP packets are encapsulated in DOCSIS frames as they are sent across the HFC network. If a standalone MTA is connected via the LAN interface, these packets are treated just as any other customer premises equipment (CPE) device. If the CM is also an embedded MTA, packets from the telephony devices are also encapsulated in DOCSIS frames and sent out the cable interface. Typically, the modem functions as a simple bridge between these interfaces. In some vendor implementations the cable modem can function as a router as well, although the bridging functionality is all that is required by DOCSIS. Keep in mind that for packets from the LAN interface to be forwarded, the cable modem must issue bandwidth requests. The CMTS controls how upstream bandwidth is allocated, so the CM is dependent upon the CMTS to provide the bandwidth it requires.

3.6 Cable Modem Termination System (CMTS)

Same as the cable modem, cable modem termination system also has been specified in the DOCSIS specification. It is typically located on the MSOs headend. CMTS provides the DOCSIS connectivity to the cable modems [10]. CMTS has the following functionality to perform:

- Allocating the upstream bandwidth
- Providing QoS to the CMs

- Classifying packets from the network and assign them appropriate QoS
- Recording usage of the resources via event messaging

3.7 Provisioning

Provisioning is the integral part of the configuration of the E-MTA in the case of the PacketCable. Provisioning of the E-MTA includes functions such as resource initialization, registration and attribute configuration. PacketCable supports two provisioning framework: E-MTA and MTA.

PacketCable supports three types of provisioning:

- Secure
- Hybrid
- Basic

The secure method is the recommended because it enables the security in the provisioning process. So that only authorized users only can access the PacketCable network and chances of theft of services are very negligible. Other two types of services were introduced because some operators are ready to take risk with their security for a simpler solution.

- **E-MTA:** A MTA which has embedded cable modem with it is known as the E-MTA. Because of the fact that CM is embedded with the MTA there are no chances of residing NAT and Firewall between CM and MTA. It should always be kept in mind that provisioning of the E-MTA is defined in two parts: DOCSIS cable modem provisioning part and E-MTA provisioning.
- **MTA:** MTA provisioning applies to the MTAs which resides behind the NAT and Firewall. So we can say that MTA is a standalone device. The MTA portion of the E-MTA has its own MAC address, IP address, and FQDN (Fully

Qualified Domain Name) than the Cable Modem. It should be noted that E-MTA and DOCSIS Cable Modem goes through the individual provisioning process. MTA has its own configuration file apart from the CM. Provisioning can be in three manners: Basic, Hybrid and Secure.

3.8 Provisioning Protocols

PacketCable uses existing protocol for the purpose of provisioning whenever required. These are Dynamic Host Configuration Protocol (DHCP), Trivial File Transfer Protocol (TFTP), Domain Name System (DNS) and Simple Network Management (SNMP). Other than these PacketCable uses protocol such as Kerberos and System Logging (Syslog).

- **Kerberos:** It is a protocol used for the purpose of the security during the provisioning process to the PacketCable client and to the services. With the use of this protocol PacketCable client and services would be able to identify each other securely.

- **System Logging (Syslog):** It is a protocol which is used to notify the logging server. Each of the devices in the network in network is configured in such a manner that they should be able to send the logging message to the logging server. And this protocol do not required any kind of acknowledgement. Different devices can send log messages to the same server. These messages are generally passed over the UDP.

3.9 E-MTA provisioning framework

The Figure 3.6 shows the reference points for the provisioning of the E-MTA. Explanation of each of each of the reference point is given as below

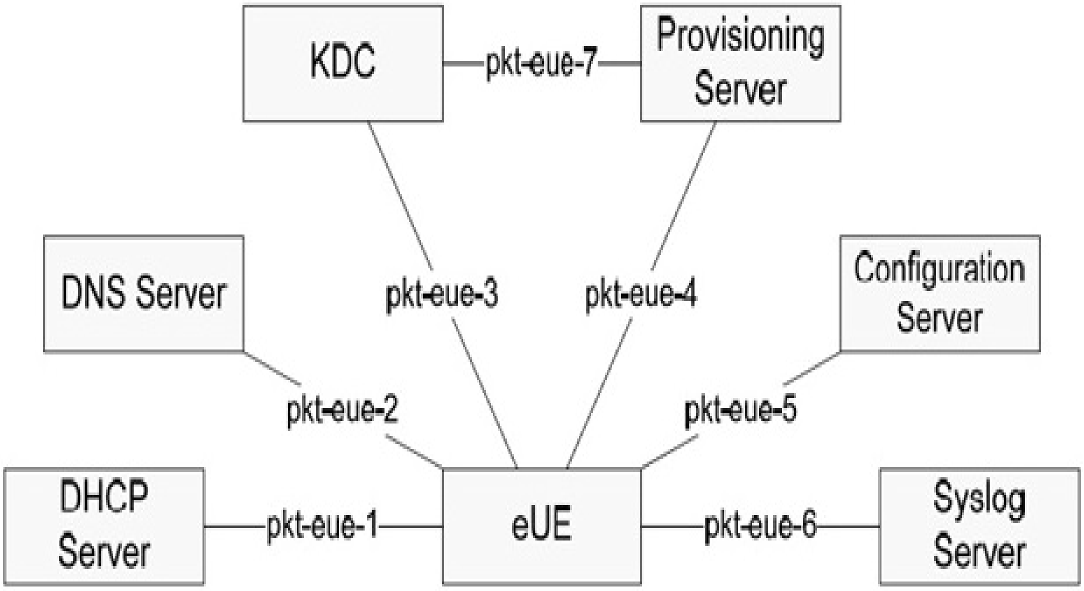

Figure 3.6: E-MTA provisioning reference points [12]

- **Pkt-eue-1:** It is a reference point which deals with the DHCP sever of the PacketCable network. In this reference point eMTA obtains IP network information i.e. IP addresses and DNS sever address.

- **Pkt-eue-2:** It is a reference point which deals with the DNS to get the location of the network elements or routing of the messages.

- **Pkt-eue-3:** This reference point let the eMTA authenticate itself to Key Distribution Center (KDC) using Kerberos protocol.

- **Pkt-eue-4:** This reference point Allows the eMTA to authenticate and exchange device capabilities with the Provisioning Server. The eMTA also uses this interface to obtain configuration information and to notify the provisioning server of the configuration retrieval status. The protocol used for authentication is Kerberos. The protocol for notification is SNMP.

- **Pkt-eue-5:** It allows the eMTA to obtain the Configuration File using TFTP or, optionally, HTTP

- **Pkt-eue-6:** Allows the eMTA to report management events via Syslog
- **Pkt-eue-7:** Allows the KDC to obtain information pertaining to the eMTA, such as the provisioned IP address and FQDN (associated with the eMTA's Mac Address)

Flow in which E-MTA gets Provisioning and configuration is shown with the help of the diagram as shown in Figure 3.7:

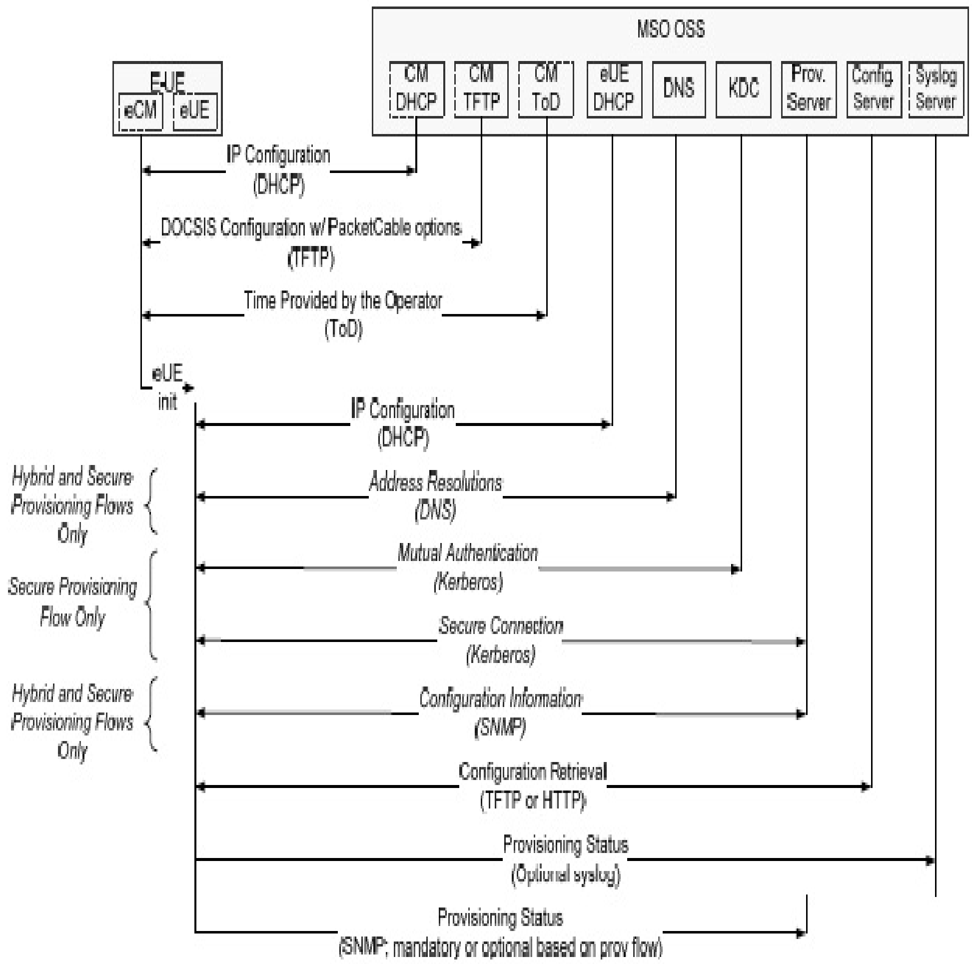

Figure 3.7: PacketCable E-MTA provisioning [12]

For the purpose of provisioning of the Cable Modem and MTA a separate tool has been provided known as Provisioning and Configuration Tool (PACT).

3.10 Provisioning Activation Configuration Tool (PACT)

PACT is the tool which is used for the purpose of the provisioning and configuration of the CMs and MTAs present in the PacketCable provided by the CableLabs. To enable the device for the purpose of the provisioning and configuration entry of the device has to be there on the PACT. A device can be added or remove from the PACT. Each of the CMs and MTAs present on the PACT has been given an identity as their hardware address and a host name. As already mentioned in the previous sections that CM and MTA gets provisioned differently. So both MTA and CM entry has to be there for provisioning. The entries which are required to enter in the PACT for CM or MTA are as below:

- **MAC address:** MAC address of the CM or MTA
- **Type of device:** CM/MTA
- **Host name:** Host Name must to start with device type string. For e.g. CMs Host Name should be begin with CM. This entry is unique. Duplicates may fail the creation/modification
- **Configuration file:** It is a binary file used for the configuration of the device. User can create his own bin file or can select the file from the existing bin file.
- **Static IP:** IP address that is to be given to the CM/UE at the provisioning time.
- **Policy ID:** chose a policy form the existing list. These policies are whether device will be provisioned in the secure, hybrid or basic mode.

3.10.1 E-MTA Configuration file

To configure the device after proper provisioning configuration file has been provided for the each of the E-MTA. This configuration file is known as the TLV (Type/Length/value). This file is in the binary format. Each of the TLV contains some entries these entries are known as the MIB. TLV of an MTA contains the information regarding the users Personal identification, public identity, DNS address, etc. A TLV file once created has to be put on the TFTP server.

3.11 IP Multimedia Subsystem (IMS) Servers

IP multimedia Subsystem is an architectural framework for delivering IP multimedia services over the internet. IMS protocol uses SIP protocol for the purpose of the signaling. To establish the PacketCable 2.0 network setup an Open source IMS server has been established. If a user who wants to use the PacketCable 2.0 service needs to register its UE to this Open IMS server. The server will provide the user a unique identity. This identity may be a number, an email address or an alphanumeric number. But in general it is proffered to go for number. The identity of a user will help in authorization and authentication process. In a similar manner if a user does not want to use the PacketCable 2.0 services entry of the user will be deregister from the Open IMS server. When a user want to use the PacketCable service first of all user will send the SIP message REGISTER, which will be processed by the IMS server. IMS server will look forward for the fact that whether the number has been registered or not. If registered IMS server will send a SIP signal OK telling the user that the number has an entry in the IMS server. And services will be made available to the user. Open IMS server can be established for both the versions of the Internet Protocol, IPV4 and IPV6.

3.12 Session Initiation Protocol (SIP) signaling

PacketCable 2.0 uses SIP for the purpose of the signaling. SIP is a peer - to - peer protocol and follows the client server architecture. SIP is a text - based protocol. It can use UDP as well as the transmission control protocol (TCP) as the transport protocol. In SIP, user agents UAs and network servers will be working in coordination. Proxy, registrar, and redirect are the three types of network servers used in SIP deployment [11].

- **User Agent:** The peers in a session are called UAs. The SIP user agent acts as both server and client. The UA that initiates a request is called a client or user agent client (UAC). The agent to which the request is destined and that which returns a response on behalf of a user is called a server or user agent server (UAS). The response accepts, rejects, or redirects the request. Two SIP user agents can communicate directly in a simple SIP - based voice call.

- **SIP Registrar:** A SIP registrar is an entity where SIP users can be registered. A registrar imparts mobility to the SIP users. A SIP user can register with a registrar. If the user changes the location, the user agent has to register again with the registrar stating the latest contact information. Whenever a call has to be delivered to the user, the registrar can provide the information about the location where the user was active recently. A registrar is typically coexists with a proxy or redirect server and may offer location services.

- **Proxy Server:** A SIP proxy receives a request, makes a determination about the next server to send it to, and forwards the request, possibly after modifying some header fields. As such, SIP requests can traverse through many servers on their way from UAC to UAS. Responses to a request travel along the same set of server but in reverse order.

- **Redirect Server:** The redirect server does not forward requests to the next server. Instead, it sends a redirect response back to the client containing the address of the next server to contact. A redirect server is a server that accepts a SIP request, maps the address into zero or more new addresses, and returns these addresses to the client. Unlike a proxy server, it does not initiate its own SIP request.

- **SIP messages:** SIP messages are basically of two type requests and response. Client issues the request message and server answers them with the help of the respond messages. Each of the users in the PacketCable network is available with its unique identification known as the Uniform Resource Identifier. A user has to be register in order to obtain the services provided by the PacketCable.

- **Request messages:**

  1. **INVITE :** Initiates a call and changes call parameters. The INVITE method indicates that the user or service is being invited to participate in a session. For a two - party call, the caller indicates the type of media it can receive as well as their parameters such as a network destination.
  2. **ACK :** ACK confirms a final response for INVITE. It may contain a message body with the final session description to be used by the callee. If the message body is empty, the callee uses the session description in the INVITE request. This method is only used with the INVITE request.
  3. **BYE :** The BYE request terminates a call.
  4. **CANCEL :** The CANCEL request cancels a pending request, but it does not terminate a call that has already been accepted. A request is considered completed if the server has returned a final response.
  5. **OPTIONS :** Queries the capabilities of servers.
  6. **REGISTER :** Registers the address listed in the to header field with a SIP server.
  7. **INFO :** Sends mid - session information that does not modify the session state.

8. **NOTIFY :** Return current state information.
9. **SUBSCRIBE :** used to request current state and state updates from other end points.

- **Respond Messages:**

1. A 1xx provisional or informational request is for continuing to process the request. Example: 100 Trying and 180 Ringing.
2. 2xx is for indicating the success of an action as the action is successfully received, understood, and accepted. Example: 200 OK
3. 3xx is redirection for additional action needs to be taken to complete the request. Example: 300 is multiple choice, and 301 is moved permanently.
4. 4xx is client error. The request contains bad syntax or cannot be fulfilled at this server. Example: 400 is a bad request, and 408 is a request time out.
5. 5xx is a server error that conveys that a server failed to fulfill an apparently valid request. Example: 500 is a server error.
6. 6xx is a global failure to convey that the request cannot be fulfilled at any server. Example: 600 is busy everywhere, 603 is declined, and 604 does not exist anywhere.

To observe the SIP signaling of a session is going correctly or not Wireshark protocol analyzer has been used. Wireshark is open source software. With the use of the Wireshark we can examine the packets which are flowing in the network. Wireshark displays packets with detailed protocol information. Proper analysis of this Wireshark packet capture can tell us whether the signaling in the PacketCable network is proper or not. If not then we can actually find out where the problem is and according to it take the appropriate action. Wireshark also provide the facility of filtering the packets captured. In the There is a command named as TCPDUMP which is used to capture the live packet while the session is ongoing. And we can see the packet capture in the Wireshark. With the help of the Wireshark we can decode the packet capture in one protocol to other respective protocol.

3.13 Basic call flow in the case of the SIP

Basic call flow in the PacketCable with the SIP signaling is divided into the three main parts. First part is when user initially boot up the board registration of the board with their SIP URIs will occur. And rest two are the call establishment and call termination. To understand the SIP signaling let us take the example of two PacketCable users want to communicate with each other. At that time how the signaling will occur is shown in the Figure 3.8.

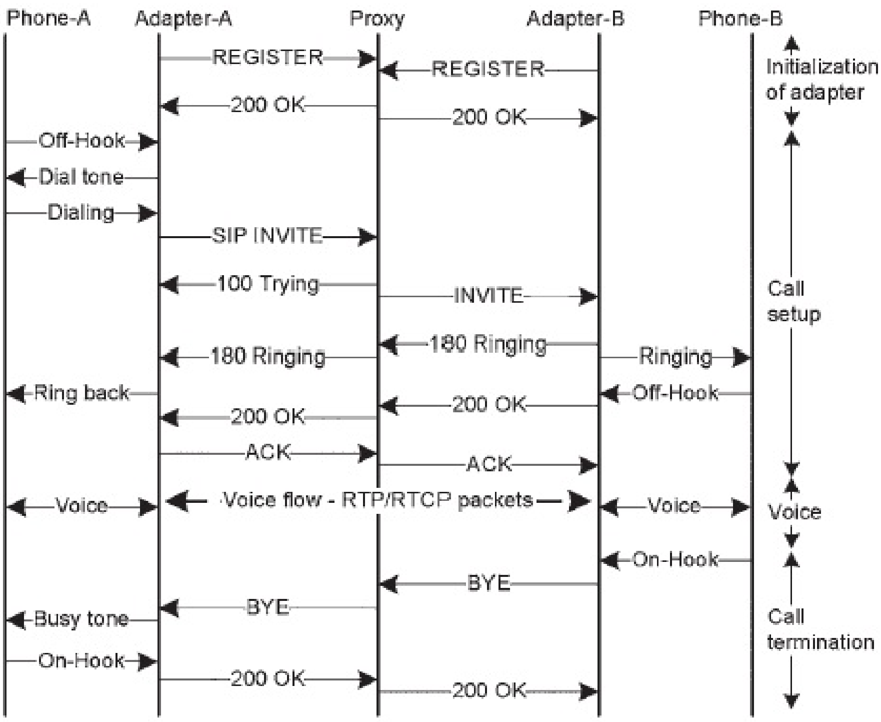

Figure 3.8: Call setup flow between two parties using the proxy server in PacketCable

3.13.1 Call Establishment

- Phone - A connected to adapter - A goes off - hook, and adapter - A sends a dial tone to the phone - A.

- After dialing all the digits of the destination number of adapter - B, adapter A composes the INVITE request with SDP information in its message body to the proxy with the "To header field containing the address of adapter - B.

- The proxy server immediately responds with "100 Trying provisional responses to adapter - A, which indicates that the message is received but not processed.

- Proxy searches the registration database in the location server to find the contact address corresponding to VoIP adapter - B. If it is not found, the error "404 Not Found is returned to the VoIP adapter - A.

- If the contact address is found, then the proxy server decides to proxy the call and creates a new INVITE transaction based on the original INVITE request received from Adapter - A and forwards the INVITE request to the adapter - B.

- The moment the adapter - B receives the INVITE, it looks for the availability of the resources. If the resources are available and the VoIP adapter - B is ready to receive an incoming call, adapter - B responds with a "180 Ringing response to the proxy. Adapter - B generates the ring to the phone connected at adapter - B. Otherwise, it returns an error with an appropriate error code (i.e., 486 "Busy Here)

- The proxy server forwards the 180 Ringing response back to the adapter - A. The response contains same To, From, C_{seq}, Call ID, and branch parameter in Via as in initial INVITE. In general, before forwarding any request, the proxy adds its own address in the Via Header. Changing characteristics of the active session can be done by sending re - INVITE. The message re - INVITE can be used to hold the call or for voice codec switching

- The moment the adapter - A receives a 180 Ringing Response; it generates the ring back tone to the phone connected at adapter - A

- When the phone at adapter - B goes off - hook, adapter - B sends 200 OK Response with SDP information that contains negotiated codecs, Packetization periods, and RTP/RTCP transport addresses

- The proxy forwards the 200 OK responses to calling adapter - A. Additional SIP communications (ACK) may be sent directly to the peer adapter or via the proxy server depending on the record route based on the contact header information found in the 200 OK responses

- Adapter - A generates ACK to the 200 OK response and is sent to adapter - B via the proxy server

- Media flow starts between adapter - A and adapter - B. Media flow is routed depending on the behavior of the proxy while forwarding the SDP information

3.13.2 Call termination

Figure 3.8 also shows call termination flow between two VoIP adapters with the assistance of an intermediate proxy server. The session termination proceeds when phone at adapter - A or adapter - B goes on - hook. When the phone at adapter - B goes on - hook, adapter - B sends a BYE request to adapter - A via proxy. After receiving the BYE request from the peer gateway, adapter - A generates the busy tone on phone - A. When phone - A goes on - hook, adapter - A sends 200 OK in response to the BYE request to adapter - B and no ACK is sent in response to 200 OK message in call termination.

Chapter 4

Voice Compression and Packetization

Voice compression is an important phenomenon which is must to be used in the case of the PacketCable. The reason for the must use is that, as PacketCable is Voice communication over existing cable network with IP packets. Also we are using a very small portion available RF spectrum for the PacketCable telephony. Thus to utilize the bandwidth effectively and accommodate more number of channels and in turn number of users voice compression is being used.

4.1 Sampling

Voice available from the speaker of the phone is in the form of analog signal. Voice signal generally varies in the range of 30Hz to 3400Hz. To digitize the analog signal first step which should be done is to sample the analog signal. According to Nyquist's criteria a signal can be sampled with the frequency twice the maximum signal frequency in order to reproduce the signal successfully at the other side. Now as voice signal has the maximum voice frequency of 3400Hz let us take it as 4000Hz, to sample the signal we need to have the minimum sampling rate of 8000Hz. This sampling process is also known as the Pulse Amplitude Modulation[10].

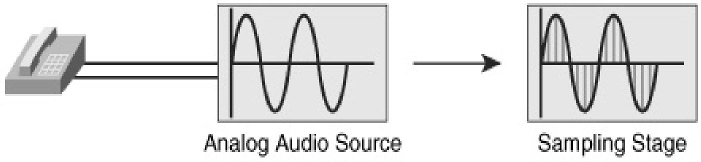

Figure 4.1: PAM signal [10]

The figure 4.1 shows the sample Pulse Amplitude Modulated (PAM) signal. So after this step the analog samples will be available which is not perfectly digitized yet. To make it a completely digitize the next step is to perform the quantization. Quantization is the process in which the voice maximum and minimum levels are divided into the number of levels and each of the level is assigned a value. Generally these numbers of levels are of power of 2 and thus each of the level can be represented using the binary value. These discrete levels are decided differently according to the different algorithm. If the distance between the two levels is equal throughout the range of the maximum and minimum range of the amplitude this is known as the Linear Quantization. Value of the Pulse Amplitude Modulated analog sample is rounded off to the nearest discrete binary value and thus the analog sample is converted into the complete digitized form. Difference between the analog sample value and the value after digitizing it is referred to as the quantization noise. Ideally quantization noise should be as low as possible. To reduce the quantization noise number of levels present in the signal has to be high which in turn will require high number of bits to represent a sample. High number of bits to represent a sample will require bandwidth usage high as compared to the small number of levels. Thus there is always a trade-off between numbers of levels chosen and bandwidth required. To sample a voice signal using the linear quantization 12 bits are enough to represent a sample thus we can say that there are 4096 equidistant levels defined for voice signal. Sampling rate of voice is also 8000Hz i.e. 8000samples per second already defined. Bit rate required for this case is 12 * 8000 = 96kbps. As it turns out, linear quantization levels are not very optimal. In this case, the possible distortion or quantization noise is constant throughout the entire signal range. Thus, in areas where the audio or signal level is lower, the SNR is smaller. This problem is compounded by the fact that a majority of human speech is in the lower signal levels. A better approach is to try to keep the signal-to-ratio level constant. This deficiency will be recovered by some other form of quantization process.

4.2 Compression Codecs

Voice compression codecs are broadly classifies in three categories: **waveform codec, vocoders and hybrid codec**[10].

Waveform - based codecs of G.711, and G.726 encode - decode voice on an actual signal without making an assumption on speech models. They work on any input that is supported by sampling without any significant distortions.

A vocoder is a voice coder that makes use of a vocal tract voice production model. Vocal tract - based compression achieves a better compression ratio than waveform - based codecs. Vocoder - based codecs generate a set of parameters that represent the speech production models. On the receive side, voice is synthetically reproduced based on the parameters. These codecs take very low bandwidth (very high compression) and deliver lower quality.

Hybrid codecs achieve acceptable compression and quality. Compared with vocoders, hybrid codecs deliver better quality, and a wideband version of hybrid codecs can exceed waveform - based codec quality. Several extended techniques in hybrid codecs are beyond the techniques used in vocoders and waveform - based codecs. G729 is the example of the hybrid codec.

The waveform codecs of G.711 and G.726 and the hybrid codecs of the G.729 family are the most popular for VoIP applications. CODECS used in the Packet-Cable are G711 and G729.

4.3 G711 Codec

Generally voice signal levels are of very low level. If we use linear quantization then to reduce the quantization noise number of levels has to increase. But this will require more bandwidth. G.711 also known as the PCM (Pulse Coded Modulation) is one of the technique to mitigate the effect. It uses 8 bits to represent a level. The only difference is the technique used for quantization known as the companding. Companding uses a logarithmic scale for quantization levels that results in keeping signal-to-quantization noise levels or Signal to Noise Ratio (SNR) constant. Companding is the technique in which the quantizing levels are not defined equidistant instead in the region where the signal levels are more and for the high values of signal levels number of quantization levels are less.

Two variants are available in the G.711 are available: -law and A-law. -law is generally used in the North America and Japan. Hence it is known as G.711 or PCMU. Whereas A-law companding technique is used in the Europe and Asia hence the standard is referred to as the G.711a or PCMA. Both of these variations are required in the PacketCable. Though they both are logarithmic companding techniques enough difference do exist between them. Voice quality may get suffer.

To calculate how much bit rate or bandwidth will be required,

Bit rate of the G711 codec:

= (Number of bits to represent a sample) * (Number of samples per second)

= (8 bits / sample) * (8000 samples/sec)

= 64kbps

Value of this required bandwidth is only on the one side. Two make a two way call possible we will require twice the bandwidth i.e. to make a complete full duplex communication 128 kbps.

After quantization we are available with the digital representation of the analog signal. Now the case is to encapsulate this digital signal into the IP packets and send them over the HFC network. The length of time samples which are collected

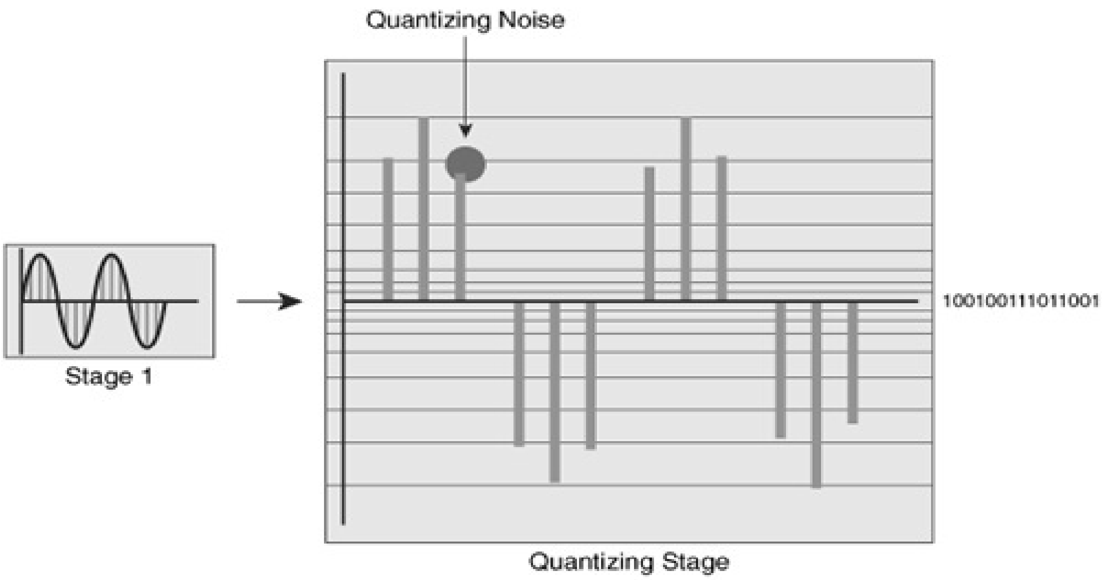

Figure 4.2: PCM Quantization [10]

before encapsulating them into the IP packet determines the voice payload in each packet. The duration of time for which the packets are captured for encapsulation is known as the Packetization Period because this is a periodic process.

PacketCable has three Packetization period 10ms, 20ms and 30ms. Small value of the Packetization period will lower the voice delay because of the fact that the packets will be send more frequently. Also if a packet gets drop in the network it will be less harmful. But at the same time number of packets on the network and overhead in each of the packet will increase, which will be waste of the bandwidth. For PacketCable deployment 20ms of the Packetization period is used.

4.4 G729 Codec

The G.729 codec consists of a separate encoder and decoder. The codec compresses speech samples as frames using a CS - ACELP analysis-by-synthesis procedure. The codec operates with 10 - ms frames with a look ahead of 5 ms, which results in total algorithmic delay of 15 ms. An overview on the encoder and decoder is given in figure 4.3in below [11].

4.4.1 G729 Encoder

The G.729 coder is based on the code - excited linear - prediction (CELP) coding model. For every 10 - ms frame, the speech signal is analyzed to extract the parameters of the CELP model. The parameters are linear - prediction filter coefficients coded as line spectral pairs, adaptive and fixed - codebook indices, and gains. These parameters are encoded and transmitted as payload to VoIP application. Preprocessed speech is analyzed for LP filter coefficients. These coefficients are converted to line spectrum pairs (LSP) and are quantized using predictive two - stage vector quantization (VQ). The open - loop pitch estimation is computed for every 10 - ms frame based on a perceptually weighted speech signal. Closed - loop pitch analysis is performed using the target signal and impulse response by searching around the value of the open - loop pitch delay. The new target signal is computed and used in the fixed codebook search to arrive at optimum excitation. The gains of adaptive and fixed codebook contributions are vector quantized. Finally, the filter memories are updated using the determined excitation signal. For every 80 samples of input, the encoder gives 10 bytes of compressed output making the total bit rate 8 kbps. These 10 bytes consist of several parameters. From the table, it can be observed that G.729 coding splits parameters into several classes with each of them having a few bits. It is entirely different from G.711 compression

4.4.2 G729 Decoder

The decoder generates 80 samples of 16 - bit linear PCM values for every 80 bits (10 bytes) of data. The input parameters for the decoder are LSP coefficients, the two fractional pitch delays, two fixed - codebook vectors, and the two sets of adaptive and fixed codebook gains. Initially, the LSP coefficients are interpolated and converted

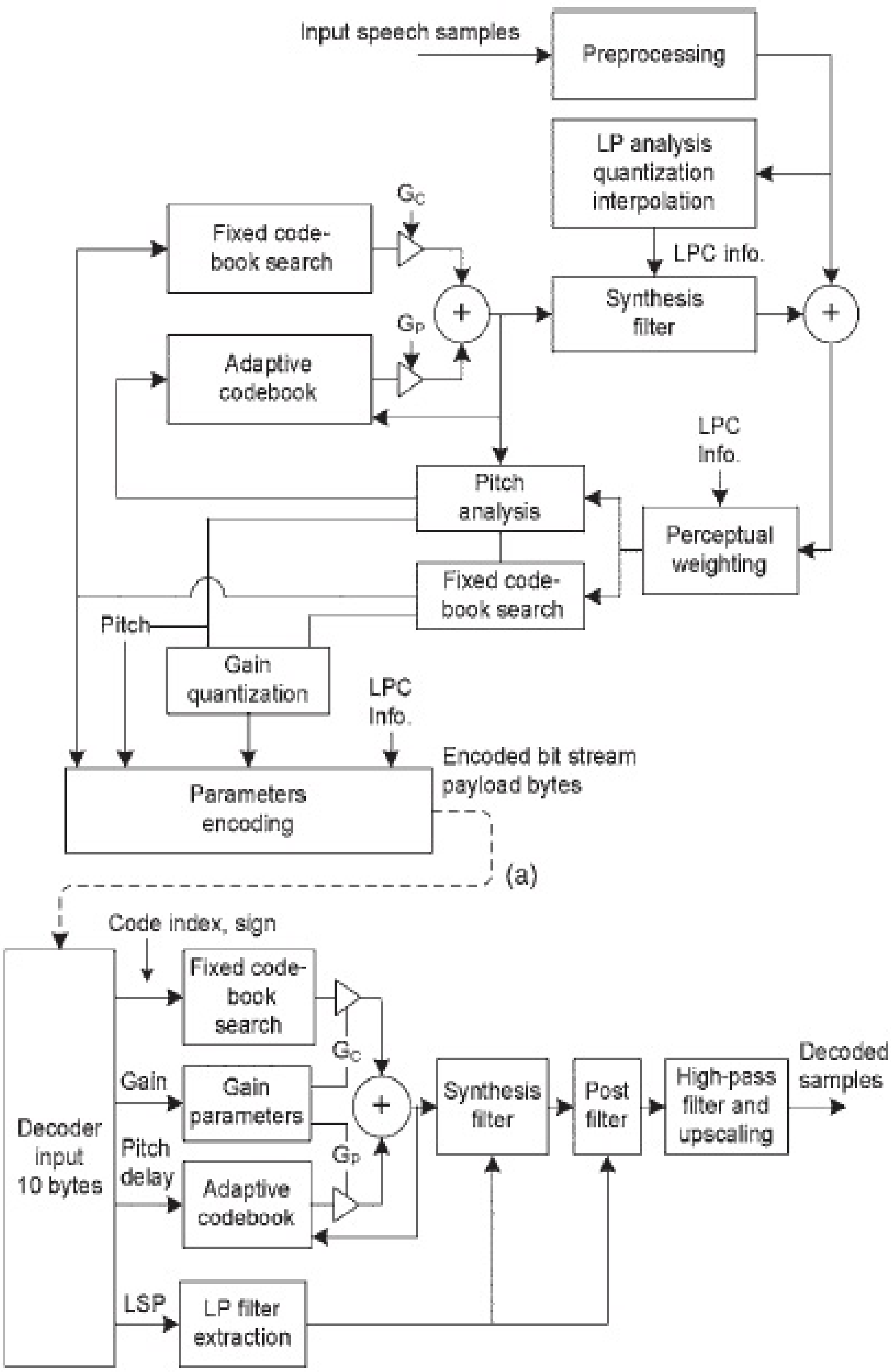

Figure 4.3: G729 CODEC [11]

to LP filter coefficients for each sub frame. For each 5 - ms sub frame, the excitation is constructed by adding the adaptive and fixed codebook vectors.

4.4.3 Speech redundancies and compression

G.711 provides pure quantized logarithmic compression on a sample basis. The basics on speech redundancies are given in. Speech signals have several low - level signals and pauses, which allow for quantization of low - level signals with a few bits, whereas operations like VAD make use of the pause periods to improve compression by about 40

4.5 RTP Packetization

Voice samples from telephone interfaces are compressed using compression codecs such as G.711, G.729 and are framed as payload. Voice payload size varies with the compression codec, compression rate options, and payload duration. Now this voice payload has to travel through the IP network to reach to the desired destination. Also voice communication is a kind of a real time application voice payload has to reach the destination in a given time bound. If it fails to do so then the voice communication gets degraded and effective communication cannot happen. So to make communication real time and reach the voice payload to the destination Real Time Protocol (RTP) is used. RTP is used in general for audio, video, and some other media specific applications. RTP carries media payload and real time parameters for extracting the timing details. RTP uses voice frames available after the compression as the payload and attached a header to this voice payload. These RTP packets are than encapsulated in either UDP or TCP protocol. But as the voice communication must be in the real time and TCP uses Retransmission on not receiving the frames, so TCP cannot be used for the purpose of real time voice communication. RTP packet along with the UDP header will be given to the network which will attach an IP header to it and thus a voice packet with RTP/UDP/IP will be converted into

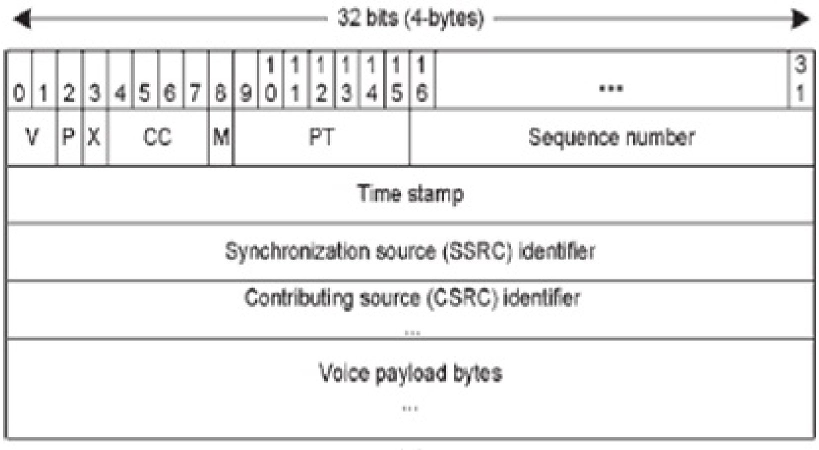

Figure 4.4: RTP packet header [11]

an IP packet and this IP packet can traverse through the IP network to reach to its desired destination. RTP header format is as shown in figure above.

The first 12 octets up to synchronization source (SSRC) are present in every RTP packet, whereas the list of the contributing source (CSRC) identifiers is present only when inserted by a mixer. RTP header parameter details are given below.

- **V** : RTP version (2-bits)
- **P** : Padding bit (1 bit), setting of this bit indicates more padding bytes at the end of the RTP packet.
- **X** : Header extension bit (1 bit) if set, the fixed header will be followed by one extension header. This is for the redundancy scheme.
- **CC** : Number of CSRC fields in the header (4 bits) that follows the fixed header. This caters to 0 to 15 CSRC identifiers.
- **M** : Marker bit (1 bit), It is intended to allow significant events such as frame boundaries to be marked in the packet stream.
- **PT** : Payload Type (7 bits) indicates several payload types including PT for voice codecs of G.711 (PT for PCMU is 0, PT for PCMA is 8), G.729

(PT18), G.723.1 (PT4), G.722 (PT 9), voice activity detector (VAD) packet types and RFC2833 and Fax. Fax and RFC 2833 RTP payload for dual - tone multifrequency (DTMF) events and telephony tones uses dynamic RTP payload types from 96 to 127.

- **A Sequence Number (16 bits)** starts with a random value selected at the time of conversation that helps to recover sequence order of packets at the destination. Adaptive jitter buffer (AJB) and packet loss concealment (PLC) algorithms also use sequence number information to arrange the packets in the right order and to send them to the decoder.

- **A Timestamp (32 bits)** is the sampling instant of the first octet of the payload. The time stamp is incremented by the payload duration. This information is useful at the destination to derive frames from long packets and at AJB for jitter calculation.

- **Synchronization Source (SSRC) (32 bits)**, the SSRC field identifies the synchronization source. This identifier is selected randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier.

- **Contributing Source (CSRC) (32 bits)**, list is 0 to 15 items with 32 bits for each source. The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 can be identified. Mixers using the SSRC identifiers of contributing sources insert CSRC identifiers. In a normal end - to - end voice call, CSRC is not present.

RTP is mandatory with PacketCable voice packets. RTP parameters are useful in extracting several real - time parameters and in ensuring proper playout of packets through jitter buffer.

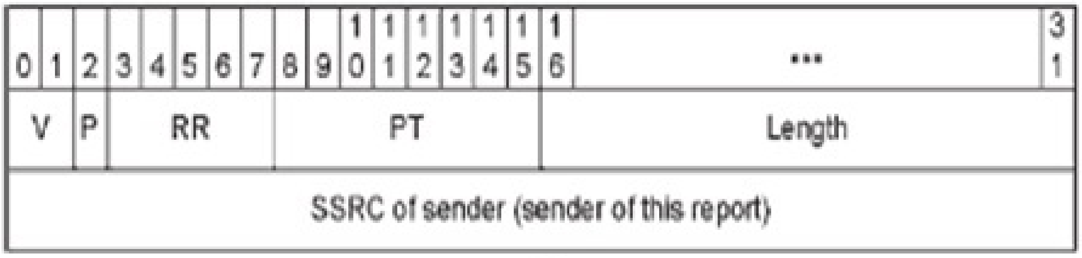

Figure 4.5: RTCP packet header [11]

4.6 Real Time Control Protocol (RTCP)

RTCP is used to convey the end to end quality of the data stream in an RTP session. It provides the information regarding delay, jitter, number of packet loss rate, and number of packets sent. These are the parameters which are used to monitor the health of the connection between the source and the destination. It also provides the timing relation between the in the form of the timestamp. RTCP packets are transmitted at rate of every 5sec. RTCP packets are transmitted on different port than that of RTP[11]. RTCP consists of five types of packets:

- **Sender Report** : Conveys the statistics of active RTP sender.
- **Receiver Report** : Conveys the statistics of active RTP receiver.
- **Source Description** : Description of the source regardless of whether it is a sender or a receiver
- **BYE** : Indicates of the session
- **App** : Application specific packet for experimental use.

Figure 4.5 above shows the RTCP header format:

- **V (2 bits)** : RTCP version
- **P (1 bit)** : Padding bit

- **RR (5 bits)** : Reception report count which shows the number of report in the packet
- **PT (8 bits)** : RTCP packet type
- **Length (16 bits)** of the RTCP packet in 32 - bit words minus one including the header and any padding and the synchronization source identifier (SSRC, 32bits) for the originator of this SR packet.

4.7 RTCP-XR parameters

RTCP - XR [7]is the extension of RTCP as per RFC3611 for monitoring voice quality parameters. It supports real - time monitoring and parameters for post analysis. Several parameters for RTCP - XR are derived from the E - model algorithm (ITU - T G.107 - 2005) and, voice processing modules. RTCP - XR makes use of parameters from packet transmission characteristics that include packet impediments on the network, end - to - end delays, jitter buffer dynamics, and signal transmission characteristics that include signal level, noise level, gain, echo rejections, R - factor, and mean opinion score (MOS) derived from the R - model. These parameters are updated typically once in every 256 voice packets. The use of these parameters and applying feedback for improving the voice quality makes the VoIP system to deliver the highest quality under severe conditions. Jitter buffer can make use of these parameters in optimizing the performance.

Chapter 5

Voice Quality in PacketCable

For several decades, telephone users have been experiencing public switched telephone network (PSTN) based voice communication quality. Subjectively, this experience is used as the main reference for comparing the voice quality from other voice communication systems. PacketCable is no exception. PacketCable has to achieve the voice quality which is better than the PSTN services. The main reason behind the PSTN voice quality is that it is a circuit switched network and also there is a dedicated network for the voice only. Codecs in PSTN are G711 A law and law. Whereas PacketCable is packet switched network where voice is converted to the digital bit stream and sent over IP network on which traffic other than voice is present. Packets from the source to destination can reach via different paths. On destination voice Regeneration of voice from the IP packets may not be as correct to the original transmitted voice. This can also lead to the degradation of voice quality. So to achieve the good voice quality comparable to that of the PSTN is one of the main concerns in the PacketCable.

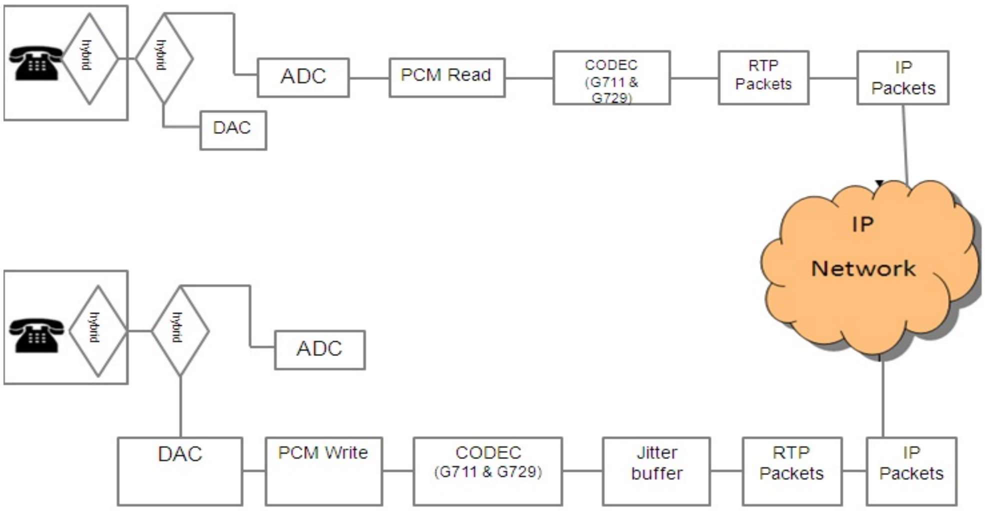

Figure 5.1: Basic voice flow block diagram

5.1 Basic Voice flow in PacketCable

Figure 5.1 shows basic block diagram of how the voice will pass through different processes in order to reach from desired source to the destination[11]. Starting from the voice originator, analog voice from telephone hybrid will be given to the other hybrid. The function of hybrid is generally to convert from two wire to four wire transmission or vice versa. Now this analog voice will be given to the Analog to Digital Converter (ADC) which will sample the voice signal. These samples are than read by the PCM reader and will be given to the voice compression codec algorithm. Compression codec algorithm can be either G711 or G729. According to algorithm the output will be available in the form of the digitized speech. These voice frames will now be packetized in the form of frames as voice payload in the RTP packet. RTP packet along with the voice payload attaches its own RTP header. RTP packet generally placed a fixed size of voice frames as its payload. Different Packetization periods are available to form the RTP packets like 10ms, 20ms and 30ms. But in general Packetization period of 20ms is chosen. After RTP header UDP and IP header will be attached to the RTP header. Finally available IP packets will be placed on the IP network and it will travel through the network and reach to its intended destination. On the receiver side IP packets will be received and from these IP packets, RTP packets can be extracted. As the packets will travel different paths to reach to its destination, packets arriving at the destination are not always in the same sequence as they were transmitted from the source. Receiver will decide this from the header of RTP. And available RTP packets will be placed in the Jitter Buffer. Function of the jitter buffer is to buffer arrived packet before processing it further. So that the output will be a smooth voice and degradation in voice quality can be reduced. Form jitter buffer at periodic time interval voice frames will be read by the decoder. Decoder will decode the voice frames. The decoded voice frames will be given to the Digital to Analog Converter (DAC) which will convert samples into the analog form and tries to reproduce the original speech which was spoken on the transmitter side.

From Above discussion we can see that there are many parameters which can affect the voice quality in the PacketCable. These are listed as below:

- Signal level
- Echo
- Delay
- Codec
- Packet loss, packet delay and packet reordering

5.2 Signal level

When the speech is transmitted from the sender it does contain some power level. Now as this signal travels from the source to destination its level gets decreased because of the attenuation. Now if at the receiver side the received signal power level is low than it may be very difficult to reproduce the speech. In this manner the established communication will not be an effective one. So before transmitting the voice some amplification of voice is done. At the same time if the amplification done is quite high than there are chances of clipping of the speech and listener will hear a noise along with the speech.

5.3 Echo

In telephonic voice conversation, echo is the return of a persons speech with delay, with reduced or modified sound level, and with a certain amount of distortions. Echoes are generally of two types: acoustic echo and electrical echo. Acoustic echo is the echo because of the speakerphone or hands free functionality. Electrical echo is also known as the line echo.

Fundamentally speaking echo is basically due to the fact that the delayed version of the speech is arriving at the receiver is quite late and thus it becomes noticeable. In case of the PSTN the delay in delayed version of the speech is quite small and is not noticeable. Lower delay helps echo to treat it as the sidetone. Sidetone is an attenuated version of the microphone signal fed back to the speaker. Sidetone returns immediately within 1 - ms In VoIP, end - to - end delay is caused by packet - based transmission. End - to - end delays also increase with higher packetization intervals and IP impediments to voice packets. For the long distance call in the PSTN we do require to have the echo canceller in order to cancel the echo because at that time the return time of the echo is of order of 100-300ms. But in the PacketCable because of the packet based transmission nature of the network delay of arrival of the packet is of order of tens of ms for shorter distance too. So we do require a echo canceller in this case also. The Figure 5.2 shows Echo representation.

Echo can be of two types talker echo and listener echo. As the name suggests talker echo in which talker listen to its own voice. Talker echo is explained with an example call between A and B. In the example shown in figure the person at phone - A is speaking and is called the talker. The person at phone - B is the listener. Voice from

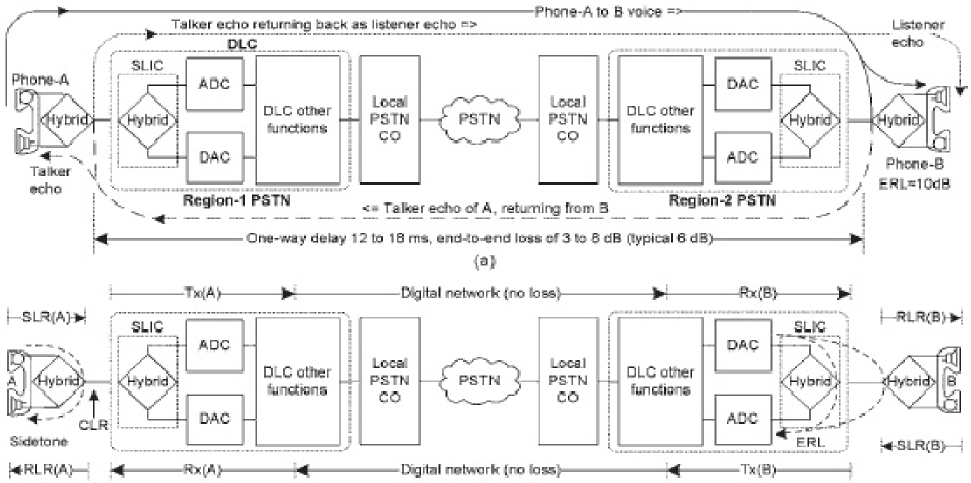

Figure 5.2: Echo representation [11]

A goes up to B, and part of A comes back to A along with voice originating from B. Person at phone - A will be hearing his own voice called talker echo with round trip delay. The talker listening to his own voice after delay is the talker echo. Round - trip delay is the transmission time for voice to travel from A to B and returning from B to A as echo. The delay from either A - to - B or B - to - A is called one - way delay. Average one - way delay is calculated as half of round trip delay. While A is speaking, the presence of echo with increased round trip delay limits the conversation comfort for person - A. The person at A slows down conversation with long pause periods and lowers the speaking sound level to minimize the echo annoyance. The same interpretation is applicable to B. When B is speaking, B will also be getting his own voice (talker echo) with round - trip delay. Listener echo is mainly created when both A and B are not keeping any echo cancellers or minimum losses in the system.

5.4 Delay

Delay is the main component which plays main part in the degradation of the voice quality. Delay is something which is unavoidable. It is always present their in the system. But care should be taken to make it as small as possible. In general end to end delay of less than 100msec is not perceptible, and delay more than 300msec make voice quality poor. Delay of less than 150msec is general value taken as reference. End-to-end delay consists of all components in between the speaker's mouth and the listener's ear. Therefore, this delay is sometimes referred to as mouth-to-ear delay. Delay components can also be categorized as being static (the same for all packets) or dynamic (different for each packet).

It is divided into three components:

- Delay from the source of the speech until it is turned into IP packets
- Delay transmitting these packets across the backbone network
- Delay from the terminating edge of the backbone until these packets are turned back into the speech heard by the receiver.

Delay is composed of fixed and variable delay. This composed of PCM delay, codec delay, packetization delay, network delay, jitter delay.

5.4.1 Delay measurement in PacketCable

There are different components which issues delay in PacketCable as shown above. To measure this delay in PacketCable different techniques are used. In this section we will measure the delay for one way delay, PCM delay which are the most variable delay amongst the available delay producing components. Techniques used to calculate different delays:

- **One way delay measurement** One way delay as shown in Figure 5.3 is the delay from the speakers mouth to the listeners delay. Value of this delay has to be less than 150msec. to measure the one way delay SAGE instrument is used.

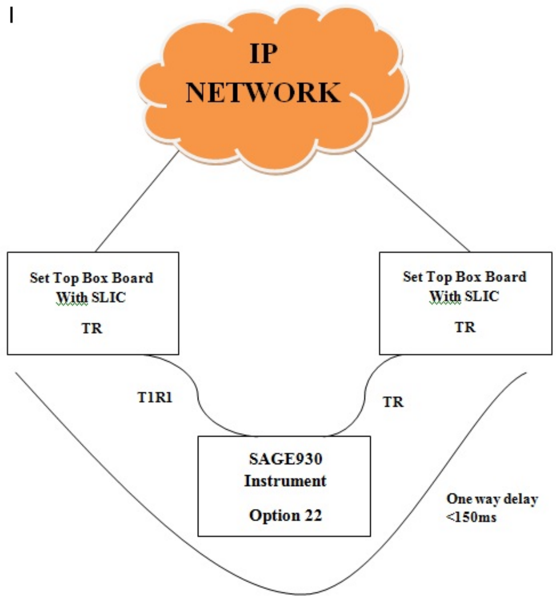

Figure 5.3: One way delay measurement

SAGE is an instrument which is used to measure the different parameters. One way delay is measured by setting the setup above. Now running the board in the different scenario. First scenario is the running PacketCable for the SFT test application in the range of the 110-120msec. And for the PacketCable running on Ethernet the same one way delay is coming in the range of the 128-150msec.

The Table 5.1 is the experiment performed to measure the one way delay. Delay shown in the table is round trip time. To calculate the one way delay from the overall delays just half the value of the round trip delay.

- **PCM driver Path delay:** To measure the PCM driver path delay run the SFT test application with a PCM file as input and store the output in some other file. Now to measure the delay, compare the input file with the output file. The first instance where the first output is obtained is the actual delay in the PCM driver path. Value of the PCM driver path delay is coming in the range of 105-115msec for the PacketCable. So from the above observation we can say that the PCM driver path delay is the major contributor in the overall delay.

Table 5.1: Delay values for the different experiment

NF _CODEC	FN _CODEC	Sample test app delay(msec)
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5
PCM	PCM	228.5

5.5 CODEC role in voice quality

Each of the codec available for the purpose of the PacketCable has its own processing delay. As we all know that the voice sampling frequency 8000Hz fixed. Codec for the PacketCable are G711 and G729. For the case of G711 no processing time is required. So a frame is available at every 0.125msec. So we can say that the encoding time for the G711 is 0.125msec. G729 processes block of the 10msec data at the same time. And it do require the 5msec look- ahead time. So total encoding time require for the G729 codec is 15msec. As the processing time for both the codec is different from each other they both require the different amount of bandwidth to carry the same amount of voice data. And in order to accommodate more number of users required bandwidth for each of the codec should be as minimum as possible. But for the lower bandwidth codec decoder operation does not produce the same voice which is close to the original voice.

Now in the following section we will calculate the bandwidth required for the one way for a call including all the headers. Formula to calculate the bandwidth required per call is as given below:

- **Bandwidth/call = (total packet size/original voice payload) * data rate**

5.5.1 G711

G711 represents data at 8-bits per sample. And as sampling frequency is 8000 samples per second. A sample is available at every 0.125msec. Assuming packetization rate of 20msec. total payload available will be of 160 bytes. After RTP packet is available the same packet will be attached with the UDP and IP header. UDP header generally consists of 8 bytes. IP header of 20 bytes and Ethernet header of 18 bytes. RTP itself consist of the 12 bytes of header.

5.5.2 G729

G729 processes the voice data of 10msec with the look ahead delay of 5msec. so total encoding delay is of 15msec. G729 represents 10msec of data in 10bytes. Total number of frames available for the packetization of 20msec will be of 20 bytes.

Original voice payload = 160/20 bytes (G.711/G.729)

RTP header = 12 bytes

UDP header = 8 bytes

IP header = 20 bytes

Ethernet header = 18 bytes

Total packet size = 218/78 bytes (G.711/G.729)

So the bandwidth required per call for both the codec can be calculated as shown in below.

Bandwidth required per call for G.711 = (218/160) * 64 kbps =87.2 kbps

Bandwidth required per call for G.729 = (78/20) * 8 kbps =31.2 kbps

5.6 Measurement of Voice Quality using E-model

This model examines signal and packet transmission characteristics to predict voice quality on a linear scale. The objective of the E model is to determine a transmission quality rating (R) R - factor or R value that incorporates the mouth to ear characteristics of an end - to - end speech path This model helps in analyzing and identifying the root causes of voice quality degradation. The R - factor is mapped to subjective MOS and to many other voice quality parameters. The typical useful range for the R factor is 50 to 94 for narrowband telephony. An R - value below 50 is not suitable for continuing call conversation. It is widely used for VoIP service quality measurements. In narrowband telephony, the R - factor ranges from 0 to 100, with 100 being the MOS equivalent of 4.5 that is achieved only with direct linear 16 - bit samples. Voice compression reduces the R - factor. In an end - to - end digital service such as ISDN with G.711, an R - factor of 93.2 is possible as the highest value. In VoIP service, several impediments contribute to the degradation of the R - factor. In the first version of G.107 recommendation, the R - factor under ideal network conditions was considered to be 94.2. This value was revised to 93.2 in the later versions of G.107. From the R - factor, additional parameters can be derived such as MOS, minimum percentage of people able to say Good or Better (GoB), maximum percentage of people that report as Poor or Worse (PoW) quality, and so on. The R - factor is mapped to a subjective voice quality measure MOS using the following equations:

$$\mathbf{MOS = 1; R \leq 0}$$
$$\mathbf{MOS = 1+0.035R+R(R\text{-}60)(100\text{-}R)(7x10^{-6}) ;R= [1,100]}$$

R - value ranging from 50 to 100 in selected steps. The corresponding MOS, GoB, PoW, and qualitative user satisfaction limits are given. It can be observed that

R from 90 to 100 corresponds to best quality, 80 to 89 is high quality, 70 to 79 is medium, 60 to 69 is low, and 50 to 59 is poor. A rating below 50 indicates unacceptable quality.

5.7 R-factor calculations

The R - factor is a transmission - rating factor for the quality of the voice in VoIP[11]. The R - factor is a scalar prediction that ranges from 0 to 100 for narrowband voice communication. The end equipment used, room noise, losses in the network, delay, packet loss, and compression algorithms used affect the R - value. The value of the R - factor can be computed by the following equation:

$$\mathbf{R} = R_0 \text{ - } I_s \text{ - } I_d \text{ - } I_e \text{ - } \mathbf{eff} + \mathbf{A.}$$

R_0 is the highest value of R that takes into account mainly the signal - to noise ratio (SNR) value, including noise sources such as circuit noise and room noise and subscriber line noise.

R_0 is a function of

$$(N_c, \mathbf{SLR}, P_s, D_s, \mathbf{RLR}, P_r, \mathbf{LSTR}).$$

I_s comprises impairments that occur simultaneously with the voice signal. The major factors that contribute to this impairment are loudness ratings of the telephone set, number of quantization distortion units, and side - tone loudness rating.

I_s is a function of

$$(R_0, \mathbf{SLR, RLR, STMR, TELR, qdu})$$

I_d comprises impairments caused by delay. The factors that contribute to these impairments are the amount of delay present in the network and the values of talker and listener echo loudness ratings.

I_d is a function of

$$(\mathbf{T},\ T_r,\ T_a,\ \mathbf{RLR},\ \mathbf{STMR},\ \mathbf{TELR},\ \mathbf{WEPL})$$

I_{e-eff} is the equipment impairment factor, which mainly comprises impairments caused by distortion. The main parameters that contribute to the I_{e-eff} are the voice compression codec and end - to - end the packet impediments.

I_{e-eff} is a function of

$$(I_e,\ B_{pl},\ P_{pl}\ )$$

A is the advantage factor, which represents the user tolerance to the degradation of the voice quality. The value of A is governed by the end - user communication interface. For wire - bound communication, the value of A is zero.

5.8 Measuring MOS values practically

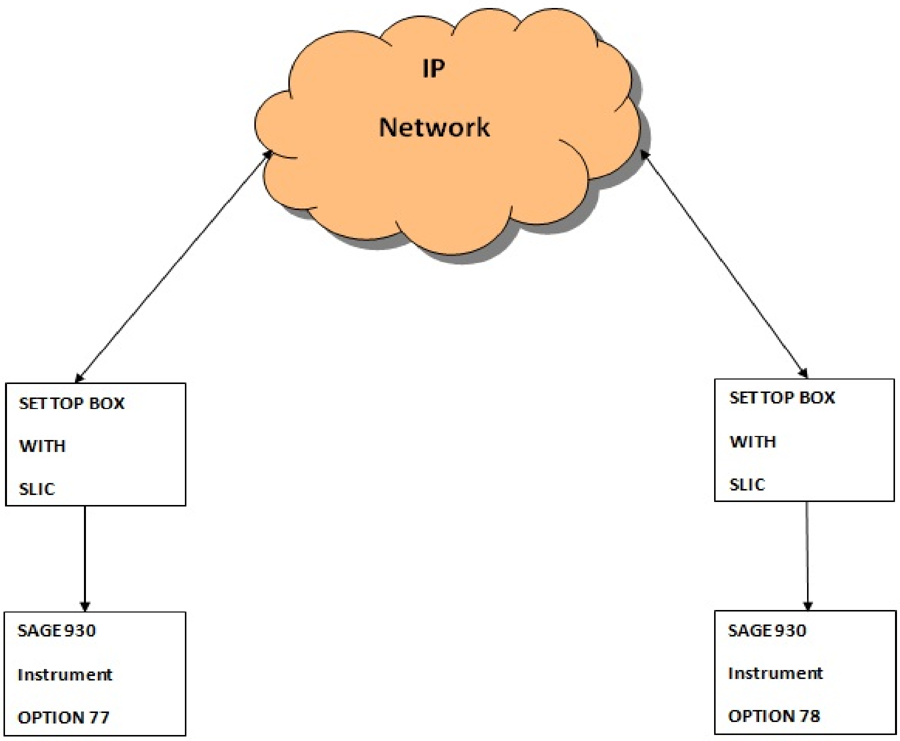

Figure 5.4: MOS value calculation

MOS value can be measured with the help of the SAGE instrument by setting the setup as shown in the figure. In this test we will make one side as the director and other side as the responder. Measuring the MOS value for the different case has been shown in the Table 5.2.

Table 5.2: MOS values for the different experiment

		Sample testapp.	**Sample testapp.**
NF _CODEC	**FN _CODEC**	**NF _MOS**	**FN _MOS**
PCM	PCM	4.15	4.01
PCM	PCM	3.96	4.14
PCM	PCM	4.36	3.99
PCM	PCM	4.15	4.14
PCM	PCM	3.96	4.19

Chapter 6

Jitter Buffer

6.1 Basic Jitter Buffer Operation

In VoIP, voice packets from the sender will travel through the IP network to reach to the desired destination. These IP packets will be carrying voice frames in form of RTP packets as their payload. While travelling in the network voice packets goes through the different impairments. Packet drops, packet fixed delay, and delay in arrival of the packet known as jitter and packet fragmentation is some of the example of these packet impairments[11].

Ideally delay in arrival between the two consecutive received packets should be constant. But this is not actually the case. This variation in arrival of the packets is known as the jitter. To reduce this variation in the arrival of the packets jitter buffer is used. Jitter buffer will always be used at the receiver side before giving a packet to the decoder for further processing. Packets from the network will arrive to the receiver with variable amount of delay. So the input to the jitter buffer will be at irregular intervals. Jitter buffer tries to remove the jitter by holding them in the buffer for the several milliseconds or for certain voice frame intervals. Thus output of the jitter buffer will be at regular interval of codec frame interval. Size of jitter buffer will be dependent on the overall one way delay of the system. For a comfortable voice conversation in VoIP one way (or mouth to ear) delay should be less than 150 ms[3]. Delay more than this can lead to degradation in voice quality. The size of the jitter buffer should be derived by considering this condition. In order to achieve desired amount of delay jitter buffer may drop some of the packets. Larger value of jitter buffer can reduce the amount of packet loss but at the same time it will increase the delay. So there is always a trade-off between the amount of jitter buffer delay and packet loss.

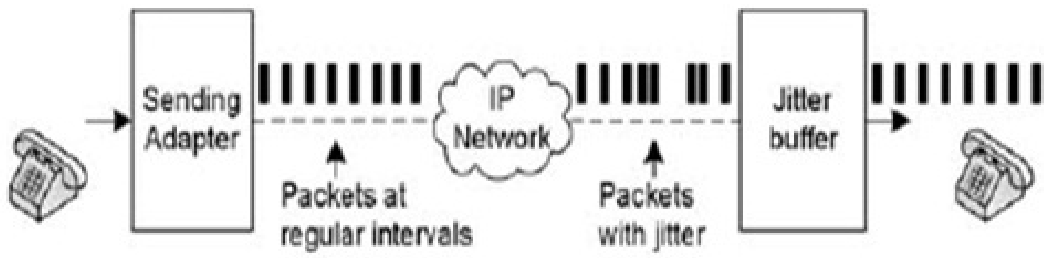

Figure 6.1: Basic Jitter Buffer Operation[11]

Jitter buffers are broadly classified as Fixed Jitter Buffer (FJB) and Adaptive Jitter Buffer (AJB). Fixed Jitter buffer is generally used for the purpose of the fax and modem application. As name suggests Fixed Jitter Buffer will hold the voice packets for the fixed amount of time regardless of the network condition. If the network condition is good than Fixed Jitter Buffer will be a good option to chose. Under impaired network condition Fixed Jitter Buffer will not be able to adapt the network condition and the packet loss would increase from the normal range. This will degrade the overall voice quality in ongoing conversation.

Jitter buffer which adapts its size according to the changing network condition is referred to as the Adaptive Jitter Buffer (AJB). Adaptive Jitter Buffer will use the input of the RTP, RTCP and RTCP-XR parameters to adjust the size of the jitter buffer. Fixed Jitter Buffer does not use the RTCP and RTCP-XR parameters [2] [3].

6.2 Delay Components in packet flow

Consider there are two parties A and B communicating with each other. Both of the parties will go through the different delays. Consider that packets are flowing from A to B for understanding this scenario [6]. Now starting from the party A, voice from the A will get converted from analog form to the digitized form in the form of the sampling. These samples then get quantized with the help of the encoder which uses compression codec algorithm like G711 and G729. These compressed voice payload as frames are packetized in the RTP packets. These RTP packets are then gets converted into the IP packets in order to flow into the IP network. Delay up to this point of time is of fixed nature and is of order of some milliseconds. Small delay variations are possible but that can be neglected as compared to overall delay. After IP packet creation these IP packets will travel through IP network to reach to the desired destination and this IP network creates the major impediments. At the destination depacketization will reproduce RTP packets from the IP packets. This delay will be of order of few milliseconds. RTP packets will be given to the jitter buffer in order to buffer the packets. Jitter buffer in this manner converts the variable delay caused by the IP network to the fixed delay and try to compensate for the IP network. Output of the jitter buffer will be at the regular decoder interval.

6.3 Adaptive Jitter Buffer

To understand the mechanism of the Adaptive Jitter Buffer assumes that the sender and the receiver both are maintaining the same timing reference. For the figure shown below say that if a packet is transmitted at the time t_i. Same packet arrives at the receiver at time a_i and will be buffered in the jitter buffer. Buffered packet will be played out at time p_i. From the figure we can say that each of the packets arriving at the jitter buffer will experience the minimum delay of the D_{prop}. The variable limit of the arrival time of a packet is in the range of a_{imin} to a_{imax}. So we can say that the span of the jitter buffer is a_{imax} - a_{imin}. From figure we can say that the play out time p_i and a_{imax} are same. So packets coming after a_{imax} will be discarded by the jitter buffer and will not be played. When the group of the packets arrives after the significant amount of delay followed by the burst of the packets, this condition is known as the spike. Under spiky condition adaptive jitter buffer will try to adjust with the spike and minimize the amount of the packet drops during the spike.

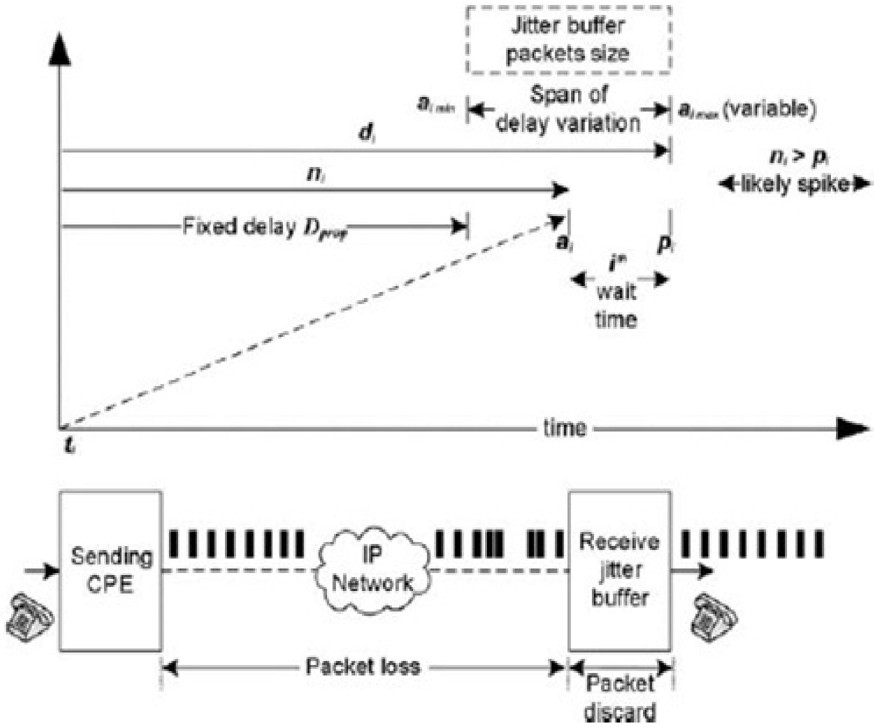

Figure 6.2: Adaptive Jitter Buffer[11]

Playout adjustment algorithms are basically of two types:

- *Talk spurt based algorithm* which tries to adjust silence periods. Algorithm detects the silence period in the speech and adjusts this silence period.

- *Non talk spurt based algorithm* which tries to adjust the play out time is adjusted for each packet with the help of some available parameters.

Adaptive Jitter Buffer will have the values of how much it can adapt in different conditions. When the variation in delay of the arrival of the packet is very small, Adaptive Jitter Buffer will buffer the packet for the minimum time. When the jitter present is more than that of minimum value of jitter, Jitter Buffer starts increasing its size so as to accommodate the late arrived packet. If the jitter associated with the packet is more than maximum the jitter buffer can tolerate, at that time the packet will be considered as the late packet and will be discarded. Lost packet in the network will not be taken care by the jitter buffer.

6.4 Adaptive Jitter Buffer in PacketCable

Adaptive Jitter Buffer in PacketCable uses the RTP parameters of received packets RTCP -XR parameters to adjust the size of the jitter buffer. Jitter buffer for the PacketCable adjust the size of jitter buffer on arrival of each packet. So we can say that jitter buffer in PacketCable is of Non-Talk spurt based algorithm. For the given case assume that the used codec in the PacketCable is G711.

Initially at the start of the conversation jitter buffer will be initialized with jitter control information parameters. These parameters include type of codec use, Packetization period, minimum and maximum value of jitter, jitter buffer can tolerate in late arrival of the packet, size of the minimum and maximum jitter buffer size, number of packets delay in play out of the packet. Minimum size of the jitter buffer is 20msec and maximum size jitter buffer can adapt is 420msec. Jitter Buffer in the case of the PacketCable is a circular buffer. It can accommodate maximum of 512msec of G711 frames.

Packets containing voice will be received from the network. These packets will be IP packets. From these IP packets RTP packets will be extracted. RTP packets always contain timestamp and sequence number in its header. Timestamp is a 32 bits unsigned number which gives the information about a packet when it was sampled with respect to the first sample. Value of the first sample in the timestamp will be a randomly selected 32 bits unsigned number. Timestamp increases by the amount of size of voice payload in packet. Sequence number is of 16 bit selected randomly for the first packet and increased by one for the next packet in the sequence. These two parameters are very useful for adjusting jitter buffer. So input to the jitter buffer will be RTP packets received from the network. These packets not necessarily are received in the same sequence as they were transmitted from the sender. Jitter buffer just accepts the packet it has received and make a decision whether to discard it or store it in the jitter buffer.

Received RTP packets with the timestamp and sequence number will be given to the jitter buffer. First of all the received frame will be stored in the jitter buffer and then after data will be read from the jitter buffer at regular interval of the time. To store the received packet for a received packet with a timestamp and sequence number, calculate present size of the jitter buffer when the packet has been received. If the size of the jitter buffer is more than 420msec discard extra packet. For size less than 420msec and if the packet has the timestamp which is larger than packets already read by the jitter and cleared, copy the frames in the packet to the jitter buffer and update the flags related to the jitter buffer. From the received packet sequence number and expected packet sequence number verify whether any packet is lost or not. If lost, increment the number of packets lost. Now check for the packet being a late packet or early packet. Accordingly adjust the counter for late packet. Margin for a packet if is in the acceptable range of 0 to 30msec and if counter exceeds the threshold value treat the packet as late packet and accept it. But if the margin is greater than 30msec and counter exceeds the threshold value than treat that packet as very late packet and discard that packet. For the packet that has arrived 90msec before its expected time of arrival than the packet will be considered as the early packet. Such 10 packets if continuously arrive before their expected time start discarding packet one by one for next coming early packet. If the current packet has the highest timestamp received replace it with the older highest timestamp. If current frame is the first frame received and store in the jitter buffer than send the control signal to indicate that the jitter buffer has the data store in it and is ready for read.

Once the packet has been stored in the jitter buffer and is ready to read from the jitter buffer. From the value of the timestamp calculate the starting address of the packet from where the stored data in the jitter buffer has to be read. Also calculate the end of the packet which is generally starting address plus the size of the payload in the RTP packet. For the stored frames in the jitter buffer check for the jitter buffer associated flag. This flag checks for whether the received frame are in sequence or not. Flag also check for the received frames within the packet that whether received frame is a silence frame or data frame. Calculate the packet loss parameters for the received packets. Copy the frames which are stored in the jitter buffer of packet size to the desired destination buffer. Increase the value of timestamp expected to receive next in the sequence for reading.

Adaptive Jitter Buffer uses RTCP-XR parameter for the packet loss calculation. PacketCable uses RF3611 A.2 algorithm for the packet loss calculation. Algorithm uses two phenomenon gap and burst. Gaps can be interpreted as the period of time during which isolated packet loss occurs. Burst corresponds to the period of time during which packet loss is significant such that it can degrade the voice quality. With the help of this algorithm calculate the burst density, gap density, gap duration, burst duration, loss rate and discard rate.

Now to understand the algorithm [7] let us take four states:

- state 1 = received a packet during a gap
- state 2 = received a packet during a burst
- state 3 = lost a packet during a burst
- state 4 = lost an isolated packet during a gap

"C" corresponds to the state transition e.g. C_{14} = transition from state 1 to state 4.

Nbreceivedpackets = number of packets received.

Lostcounttemp = number of packets lost in the current burst.

Packetlosacc = total number of packets lost

Packetdiscardcount = total number of packets discarded

```
if (Nbreceivedpackets ≥ 16)
{
        If(Lostcounttemp == 1)
        {
                C14++;
        }
        else
        {
                C13++;
        }
        Lostcounttemp++;
        C11 += Nbreceivedpackets;
}
else
{
        Lostcounttemp++;
        if(Nbreceivedpackets == 0)

                C33++;
```

else

{

C_{23} ++;

C_{22} += Nbreceivedpackets;

}

}

Nbreceivedpackets == 0;

$C_{31} = C_{13}$;

$C_{32} = C_{23}$;

Totalpacket = $C_{11}+C_{13}+C_{14}+C_{22}+C_{23}+C_{32}+C_{33}$

Totalpacket += Nbreceivedpackets;

/ $*$ Calculate burst density and gap density $*$ /

$P_{32} = C_{32}$ / (C_{31} + C_{32} + C_{33});

if ((C_{22} + C_{23}) < 1)

{

$P_{23} = 1$;

}

else

{

P_{23} = 1 - C_{22}/(C_{22}+C_{23});

}

burst_density = 256 $*$ P_{23} / (P_{23} + P_{32});

gap_density = 256 $*$ C_{14}/(C_{11}+C_{14}+Nbreceivedpackets);

/$*$calculate burst and gap durations in ms$*$/

m = frameDuration_in_ms $*$ framesPerRTPPkt;

gap_length = (C_{11} +C_{14} + C_{13}+Nbreceivedpackets) $*$ m / C_{13};

burst_length = Totalpacket $*$ m / C_{13} - gap_length;

/$*$ calculate loss and discard rates $*$/

loss_rate = 256 $*$ Packetlossacc / Totalpacket;

discard_rate = 256 $*$ Packetdiscardcount / Totalpacket;

With the help of the above algorithm we calculate the possible parameters which can gives the idea of packet loss in the network due to congestion and the packets discarded in the jitter buffer. We also can calculate the densities at which these packet loss/packet discard occurs. The value of this parameters actually shows that how the jitter buffer can actually accommodate the packets and how it adapts the network conditions.

Chapter 7

Conclusion and Future scope

7.1 Conclusion

The thesis has shown the latest technology which provides the different interactive services over the existing cable network. At present it has been limited to the telephony services. Architecture of the PacketCable for different version has been available with PacketCable 1.0 the first version available for the simple telephony to the latest version PacketCable 2.0 which can be useful for the residential telephony, video call, and other interactive services. PacketCable architecture has been explored in detail.

To provide the PacketCable services over the existing cable network SLIC-SLAC device has been attached with the Set Top Box to perform the BORSCHT functionality. Configuration of this SLIC SLAC device has been performed with the proper parameters setting.

Provisioning of the MTA and Cable modem has been performed with the help of the provisioning server. Proper configuration and parameter setting for the particular MTA and Cable Modem has been done in the provisioning server.

Validation of the PacketCable for different scenario has been shown with the setup required. Call establishment using SIP signaling has been shown for a given scenario.

Voice digitization and packetization in the form of the IP packets had been studied for the PacketCable. Delay measurement has been measured for different cases. Results shows that the overall delay is in the range of 110-130msec. at the same time delay for the PCM driver path is 70-80msec. So from this we can say that the PCM driver takes the major amount of time in the overall delay.

Measurement of voice quality in the form of the MOS value has been performed for the PacketCable. Results for the MOS values are within the acceptable range.

PacketCable uses the adaptive jitter buffer to take care of the jitter. Adaptive jitter buffer with its mechanism of handling the packets has been explained. Packet discard and packet loss are also calculated using the algorithm.

7.2 Future Scope

Present work has been done for the purpose of the analog telephony to provide the telephony services. Future work includes replacing analog telephony with the cordless telephony. Providing the same functionalities which are provided by the PSTN analog telephony. Interfacing the SLIC-SLAC with the set top box. Validate the functionality with the same.

References

[1] E.Miller, F. Andreasen, G.Russell, "The Emergence of Integrated Broadband Cable Networks ,The Packetcable Architecture", *IEEE Communication Magazine*, June 2001.

[2] L.Atzori, M.Lobina and Marco Corona "Playout buffering of speech packets based on Quality maximization approach", *IEEE transactions on multimedia*, April 2006

[3] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks" *Wide Area Networks Proceedings of IEEE INFOCOM*, 1994

[4] A.Shallwani, "An Adaptive Playout Algorithm with Delay Spike Detection for Real-Time VoIP", a report, October 2003

[5] J.Pinto and K. Christensen "An Algorithm for Playout of Packet Voice based on Adaptive Adjustment of Talk spurt Silence Periods", *Proceedings of 24th Annual IEEE International Conference on Local Computer Networks*, 1999

[6] K.McNeill, M. Liu and J.Rodriguez, "An adaptive jitter buffer play-out scheme to improve voip quality in Wireless networks", *Military communication Conference (MILCOM)*, 2006

[7] T.Friedman, R.Caceres and A.Clark, "RTP Control Protocol Extended reports(RTCP XR)", *RFC 3611*, Novenber 2003

[8] D.Hardman, "Noise and Voice quality in VoIP environment", *white paper*, April 2003

[9] J.Anderson, "Addressing VoIP speech quality with Non-intrusive measurements", *white paper*, September 2003.

[10] J.Riddel, "PacketCable Implementation", *Cisco press*, February 2007

[11] S.Nagireddii, "VoIP Voice and Fax Signal Processing ", *John Wiley and Sons, Inc., publication, New jersey*, 2008

[12] PacketCable 2.0 architecture: http://www.cablelabs.com/specifications/PKT-TR-ARCH-FRM-V06-090528.pdf