E-Line Package Support in E-MTA

by

Ada Sin Mei Pang
B.A.Sc. (Electrical Engineering), Simon Fraser University, 2001

PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING

In the
School
of
Engineering Science

SIMON FRASER UNIVERSITY

Summer 2005

Approval

Name: Ada Sin Mei Pang

Degree: Master of Engineering

Title of report: E-Line Package Support in E-MTA

Examining Committee:

Chair: Dr. P. Ho
Professor
School of Engineering Science, SFU

____

Dr. R. Vaughan
Professor
School of Engineering Science, SFU

____

Dr. C. Scratchley
Lecturer
School of Engineering Science, SFU

____

Mr. C. Griffiths
Engineering Manager
Broadcom Corp.

____

Mr. T. Pang
Senior Staff Software Engineer
Broadcom Corp.

Date Approved: July 28th, 2005

Abstract

The exponential growth of packet-based data networks has led to an increasing demand for VoIP. Although VoIP was initially intended to bypass costly international tolls, this technology has since matured to the point where its voice services are comparable to those offered by the PSTN.

To compete with satellite and telephone companies, cable operators are working towards incorporating VoIP into their infrastructure. However, in order to support VoIP, new protocols must be introduced. The protocol responsible for relaying information regarding the provision of voice services is the call signaling protocol. The information that is relayed via this protocol is categorized into packages, with each package being designed to support a specific service. The E-Line package is an analog access line package designed for supporting international telephony services. This thesis project focuses on the successful design and implementation of a subset of the E-Line package on Broadcom's VoIP residential broadband gateway.

Acknowledgments

I have been fortunate to receive invaluable guidance and suggestions from numerous people in preparation for this thesis project. I would like to express my gratitude to Dr. Rodney Vaughan, Dr. Craig Scratchley, and Mr. Chad Griffiths. In addition, I would like to thank Dr. Paul Ho for being the chair of my thesis project and Mr. Thomas Pang for being my external examiner.

Table of Contents

List of Figures

List of Tables

Glossary

ANS	Announcement Server
CableLabs	Cable Television Laboratories, Inc
CAS	Channel Associated Signaling
CLASS	Custom Local Area Signaling Services
CM	DOCSIS Cable Modem
CMS	Call Management Server
CMTS	Cable Modem Termination System
Codec	Coder-Decoder
CPE	Customer Premise Equipment
DOCSIS	Data Over Cable System Interface Specification
DSP	Digital Signal Processing
E-MTA	Embedded MTA
ETSI	European Telecommunications Standards Institute
FQDN	Fully Qualified Domain Name
HFC	Hybrid Fiber/Coaxial Cable
IETF	Internet Engineering Task Force
IC	Integrated Circuits
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-T	Telecommunications Standardization Sector of the ITU
LCS	Line Control Signaling
LE	Local Exchange
MG	Media Gateway
MGC	Media Gateway Controller
MGCP	Media Gateway Control Protocol
MTA	Multimedia Terminal Adapter
NCS	Network-Based Call Signaling
NTFY	Notification
OSI	Open Systems Interconnection
OSS	Operational Support Systems
PCO	Public Call Office
QoS	Quality of Service
PSTN	Public Switched Telephone Network
RQNT	Notification Request
RTP	Real-time Transport Protocol
SG	Signaling Gateway
SG9	Study Group 9
SLIC	Subscriber Line Interface Circuit
SNMP	Simple Network Management Protocol
USB	Universal Serial Bus
VoD	Video on Demand
VoIP	Voice over IP

1 Introduction

While traditional cable networks were designed for one-way broadcasting services, many cable operators have upgraded their infrastructures to provide bi-directional high-speed packetized services. This broadband capacity enables operators to offer enhanced services such as Voice over IP (VoIP) and real-time streaming. Furthermore, with the advancement of multimedia technology, the next generation cable networks will offer high-grade packetized video services.

To standardize packetized voice traffic, organizations such as the International Telecommunication Union (ITU) and the European Telecommunications Standards Institute (ETSI) have introduced specifications that outline the end-to-end architecture as well as protocol interfaces. Designed to allow interoperability in a highly-integrated environment, these specifications define several management services that are required to support voice traffic. One of the management services is call signaling, which is responsible for call management. For instance, the dial tone that is heard when a user picks up the telephone is triggered via call signaling protocols.

Call signaling protocols are responsible for relaying events and signals between the user and the service provider. These relayed events and signals are grouped into packages. Typically, different packages are used to support different types of endpoints[1]. For instance, telephony events such as off-hook and on-hook are collectively grouped into a package that is associated with telephony lines. Similarly, a different package is being developed to support video endpoints. New packages can be easily added to the call signaling protocol as new media formats and services are introduced.

This thesis focuses on a new telephony package, the E-Line package. Intended as an extension to the default telephony package, the E-Line package addresses international support for telephony devices. This new package is designed to be used on the

---

[1] An endpoint identifies the terminating interface. For instance, an endpoint can be an analog access line connected to a telephone or a facsimile machine.

Embedded Multimedia Terminal Adapter (E-MTA). The E-MTA is a device residing in the customer premises, which provides VoIP support for telephony equipment over cable networks.

Although this thesis only focuses on a subset of the E-Line package, the design and the implementation of this thesis must be scalable to support the remaining package and to account for new extensions. In addition, a thorough verification process must be carried out to ensure compliance with the package definition as well as interoperability with other network elements within the cable infrastructure.

The next section provides an overview of the cable infrastructure and the interface specifications. Section 3 outlines the hardware platform of the E-MTA developed by Broadcom. Section 4 offers an overview of package definitions, while Section 5 focuses on the E-Line package. In Section 6, the design and the implementation of the E-Line package are presented. Finally, results and analysis of the E-Line package are presented in Section 7.

2 Cable Architectural Background

To compete with new services offered by satellite and telephone companies, cable operators are introducing new features to enhance the cable infrastructure. Among the list of new features such as VoIP, Video on Demand (VoD), and wireless connectivity, the addition of VoIP to the cable infrastructure is the primary focus for many cable operators.

Cable modems have come a long way since 1996 when they were mostly proprietary solutions that could only operate if the cable headend[2] was supplied by the same vendor. To eliminate proprietary systems and to allow interoperability between different vendors, extensive standardization efforts have been made to regulate all entities within the cable infrastructure. This standardization effort was led by Cable Television Laboratories, Inc. (CableLabs), a non-profit research and development consortium focused on cable telecommunications development [9]. The following sections provide an overview of the intricate cable infrastructure and the upgrades that are required to support VoIP services.

2.1 DOCSIS

One of the first projects that CableLabs focused on was the introduction of the Data Over Cable Service Interface Specification (DOCSIS). Designed to control high-speed data access within cable networks, this specification manages IP traffic between the subscriber and the cable headend. In particular, DOCSIS focuses on the Physical and the Data-link layer of the Open Systems Interconnection (OSI) Reference model [3].

The major components of the DOCSIS network consist of the Cable Modem (CM) resided in the customer premises and the Cable Modem Terminating System (CMTS) located at the cable headend. Figure 1 provides the reference architecture of the DOCSIS network.

---

[2] Cable headend is the control site where central cable equipment resides.

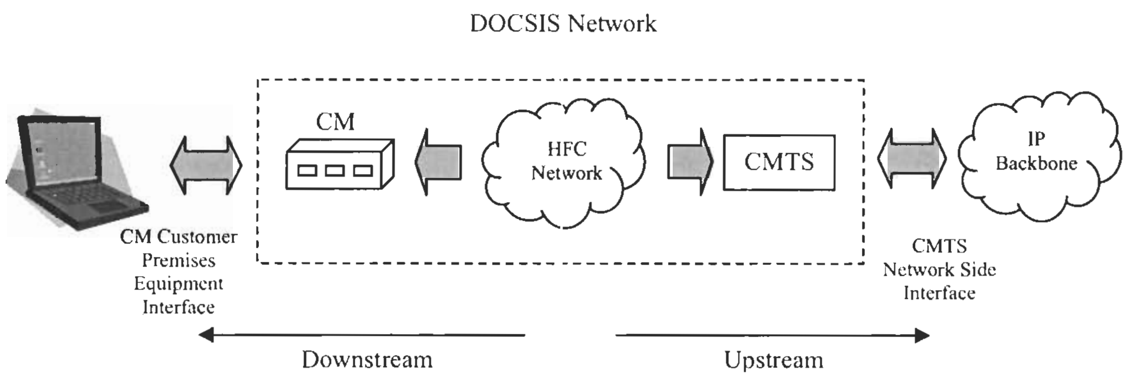


Figure 1. DOCSIS Reference Architecture [3]

The CM is a modulator/demodulator that encapsulates IP packets for transmission over the DOCSIS network. Typically terminated on an Ethernet or a Universal Serial Bus (USB) interface, the CM provides networking capability for IP-enabled devices in the customer premises [3].

The CMTS is the hub for all traffic within the DOCSIS network. Responsible for bandwidth management, the CMTS provides connectivity for upstream and downstream traffic between the IP Backbone and the Hybrid Fiber/Coaxial (HFC) network. In addition, the CMTS maintains Quality of Service (QoS) within the DOCSIS network [3].

Connecting the CMTS and the CM is the HFC network. The HFC network provides high-speed reliable transport between the cable headend and the customer premises.

2.2 PacketCable

The successful standardization of DOCSIS provided foundation for new packetized services within cable networks. To help identify and qualify these new IP-based services, CableLabs created the PacketCable project. Aimed at supporting real-time multimedia services such as IP telephony, interactive gaming, and multimedia conferencing, the PacketCable project defines interface specifications for packetized voice, video, and other high-speed services [4].

The PacketCable reference architecture, as illustrated in Figure 2, is essentially divided into three networks: the DOCSIS HFC Access Network, the Managed IP Backbone, and the Public Switched Telephone Network (PSTN). The DOCSIS HFC Access Network is similar to the DOCSIS network described in Section 2.1 except that it now includes the Multimedia Terminal Adapter (MTA). The Managed IP Backbone provides interconnections between functional components of the PacketCable architecture. Finally, the PSTN is included in the PacketCable architecture to provide communications between packet-switched and circuit-switched clients.

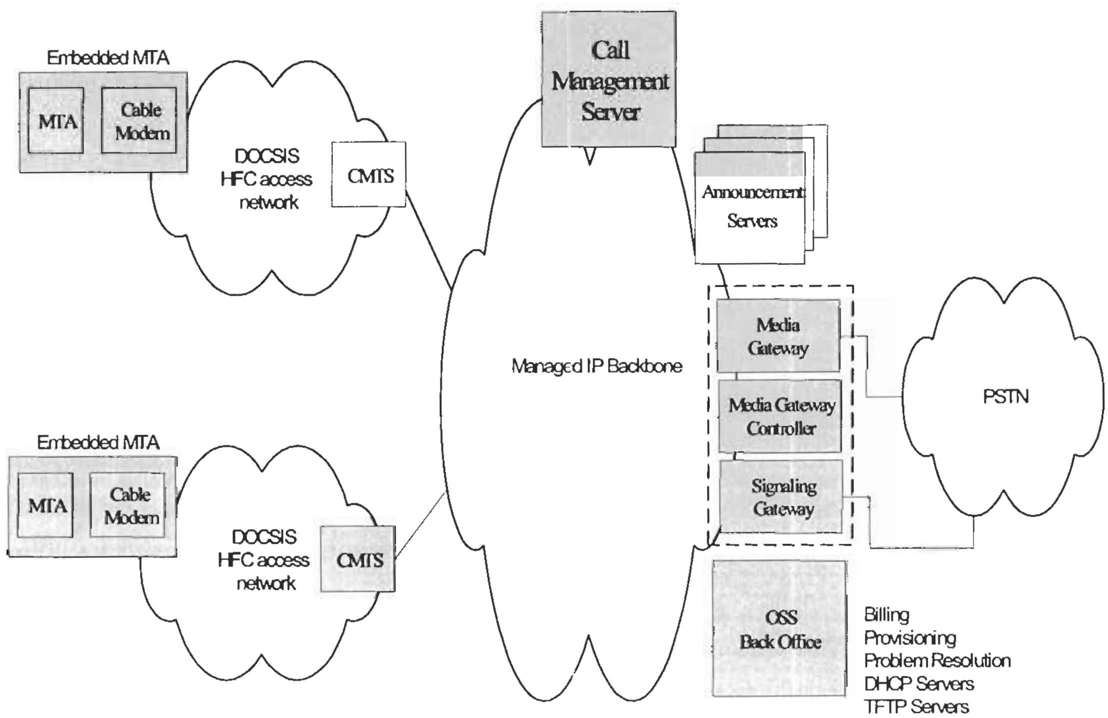


Figure 2. PacketCable Reference Architecture [4]

The MTA in the DOCSIS HFC Access Network is a PacketCable client device that enables customer premise equipment (CPE) such as a telephone to utilize VoIP. Main responsibilities of the MTA include signal generation, voice codec selection, and call signaling. Working in conjunction with the CM, the MTA transmits data to other

PacketCable devices via the DOCSIS network. When a PacketCable MTA is incorporated with a DOCSIS CM in a single hardware device, the integrated component is an Embedded MTA (E-MTA) [4].

Traffic management within the PacketCable architecture is achieved by servers residing in the Managed IP Backbone. The Call Management Server (CMS) is responsible for call control. Being a trusted entity within the Managed IP Backbone, the CMS provides signaling services for the MTA, the CMTS, and PSTN devices. Essentially, it has control over every connection within the PacketCable network. Another management server within the PacketCable architecture is the Announcement Server (ANS). It is responsible for supplying and managing all announcements within the PacketCable network [4].

The Operational Support Systems (OSS) back office contains the core business processes such as accounting, device configuration, and security management. Working together with other management servers within the PacketCable architecture, the OSS back office provides a scalable, robust, and secured PacketCable network [4].

Interoperability between the Managed IP Backbone and the PSTN is achieved via the PSTN gateway. The PSTN gateway consists of three functional components: the Signaling Gateway (SG), the Media Gateway (MG), and the Media Gateway Controller (MGC). The SG and the MG are respectively responsible for signaling and media conversions between PSTN and IP traffic. The MGC maintains the overall behavior of the PSTN gateway [4].

2.3 IPCablecom

The focus of both the DOCSIS and the PacketCable projects are on high-speed data access primarily within North America. However, due to deviations in traditional telecommunication implementations, these projects are not completely applicable in other places such as Europe. For instance, the Custom Local Area Signaling Services (CLASS) used for generating Caller ID messages deviates between different countries. The Bellcore standard in North America regulates that the Caller ID message must be generated using 1200 baud Bell 202 tone modulation, while British Telecom developed their own standard encoding the Caller ID message in CCITT V23 modem tones [1]. To address these service discrepancies, the IPCablecom project was created.

The IPCablecom project was created by Study Group 9 (SG9, Integrated Broadband Cable Networks and Television and Sound Transmission) of ITU-T, the Telecommunications Standardization Sector of the ITU. Based heavily on the DOCSIS and the PacketCable specifications, the IPCablecom project provides annexes to address regional differences in the cable infrastructure. In particular, the IPCablecom project has currently been extended to include both European and Asian specifications [7].

The PacketCable reference architecture, as illustrated in Figure 2, is also employed in the IPCablecom project. The work for this thesis focuses on a new annex that addresses the call signaling discrepancies between different countries.

3 Hardware Platform

The hardware platform used for this thesis is the BCM93349VBCM VoIP residential broadband gateway developed by Broadcom. Since this thesis is primarily software-based, hardware constraints such as the required DSP power and the number of supported lines on the gateway are separated from the design and the development of this thesis. Hence, this section only provides a brief overview of the hardware context for this thesis. The reference design of the BCM93349VBCM is provided in Figure 3.

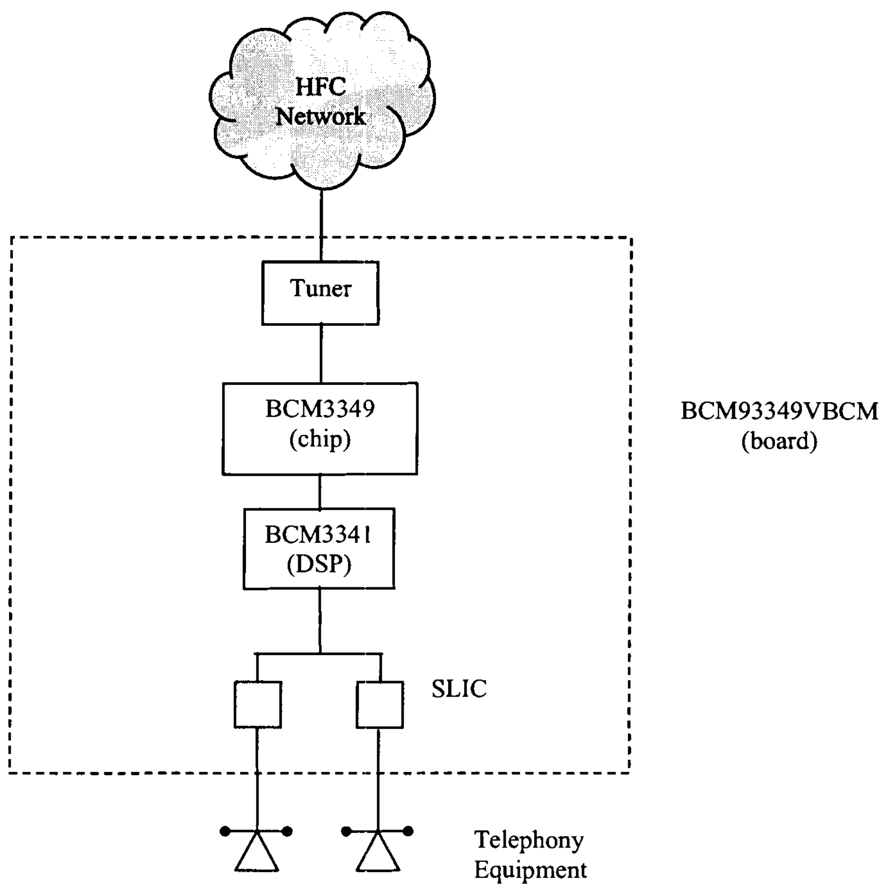


Figure 3. Hardware Platform

The fundamental components that make up the BCM93349VBCM are the BCM3349 and the BCM3341. The BCM3349 and the BCM3341 are single-chip Integrated Circuits (IC) offering cable modem and VoIP functionalities respectively. Essentially, this combined solution forms a PacketCable E-MTA.

Interface to the HFC network is achieved via the tuner. The tuner is responsible for locking onto the selected carrier frequency and applying the proper filtering for channel decoding. Interface to telephone equipment is achieved via the Subscriber Line Interface Circuit (SLIC). Working in conjunction with the BCM3341, the SLIC is responsible for ring generation and hook state detection. As illustrated in Figure 3, the BCM93349VBCM is capable of supporting 2 CPEs per gateway.

4 Technical Overview

4.1 Call Signaling Overview

Call signaling is the process responsible for call management. When a user picks up the telephone and dials the telephone number of a remote party, call signaling protocols are employed to establish the connection with the remote end. Similarly, when a user completes a call and hangs up the telephone, call signaling protocols are utilized to delete the connection and to release committed resources. Thus, call signaling protocols must coordinate with many network entities to handle call processing. This section provides an overview of the intricate signaling interfaces within the IPCablecom infrastructure.

To allow for interoperability between call management services, signaling interfaces have been defined for network entities within the IPCablecom architecture. As illustrated in Figure 4, the call signaling interface is basically divided into two frameworks: the Network-Based Call Signaling (NCS) Framework and the PSTN Signaling Framework [4].

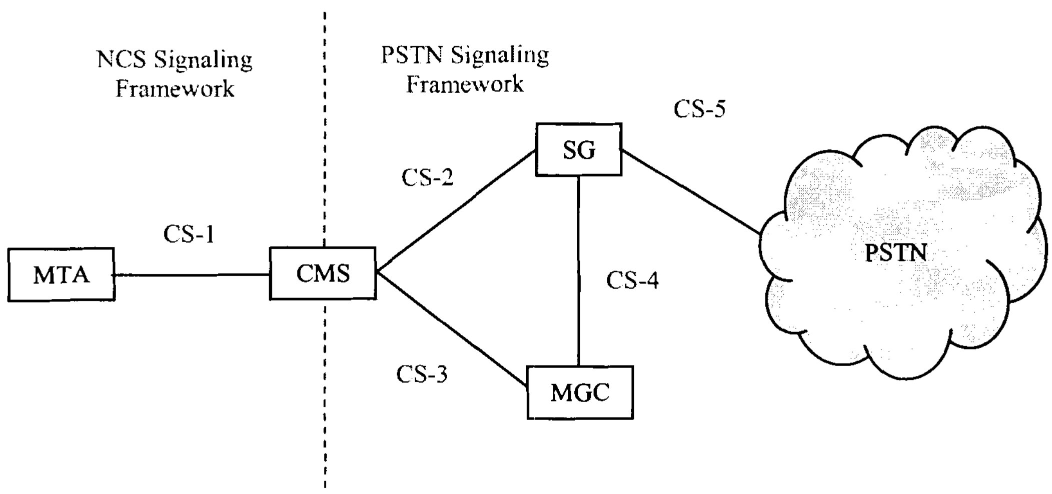


Figure 4. Call Signaling Interface [4]

The Network-Based Call Signaling (NCS) Framework contains only the CS-1 interface. Communications via this interface are achieved using the Network-Based Call Signaling (NCS) protocol. The NCS protocol employs a centralized architecture, where call state intelligence and feature implementations reside in the CMS while the MTA is only responsible for local device handling. In essence, the MTA is analogous to a "dummy device", informing the CMS of any device events and responding to CMS commands. Information relayed via the CS-1 interface reflects the current state of the MTA (telephone is ringing, telephone is off-hook, etc.). More details on the NCS protocol is presented in Section 4.2.

The second framework in Figure 4 is the PSTN Signaling Framework. To access both PSTN-based services and to bridge PSTN subscribers with the IPCablecom network, this second framework requires more signaling interfaces than the NCS Framework. Call signaling to the PSTN is accomplished by the SG via CS-5. Since the SG does not contain call control intelligence, PSTN signaling must be initiated by either the CMS or the MGC via the CS-2 or the CS-4 interface. Finally, the last interface, CS-3, between the MGC and the CMS is currently undefined within the IPCablecom architecture. The MGC and the CMS are often combined within a single entity, making this interface internal to the component.

A particular contribution of this thesis is the changes required in the NCS framework to account for signaling discrepancies between different countries. Thus, the next section describes the NCS protocol in details.

4.2 Network-Based Call Signaling Protocol

The Network-Based Call Signaling (NCS) protocol is an extended variant of the Media Gateway Control Protocol (MGCP) created by Internet Engineering Task Force (IETF). Optimized for cable access environment, the NCS protocol provides integrated QoS and security support [5]. The protocol employs a comprehensive list of commands to manage voice calls within cable networks. To examine the complete list of NCS commands is beyond the scope of this thesis. Rather, the commands that are of interest in this thesis are presented below:

- Notification (NTFY)
    - Notifies the CMS upon detection of an observed event
- Notification Request (RQNT)
    - Requests the MTA to generate a signal and/or a notification upon events detection

The NTFY and the RQNT commands are used in conjunction with each other to handle MTA operations. Figure 5 provides a simple call flow that illustrates how these two commands are used. The call flow starts off with the MTA originating a call, triggering an off-hook event and dial tone generation.

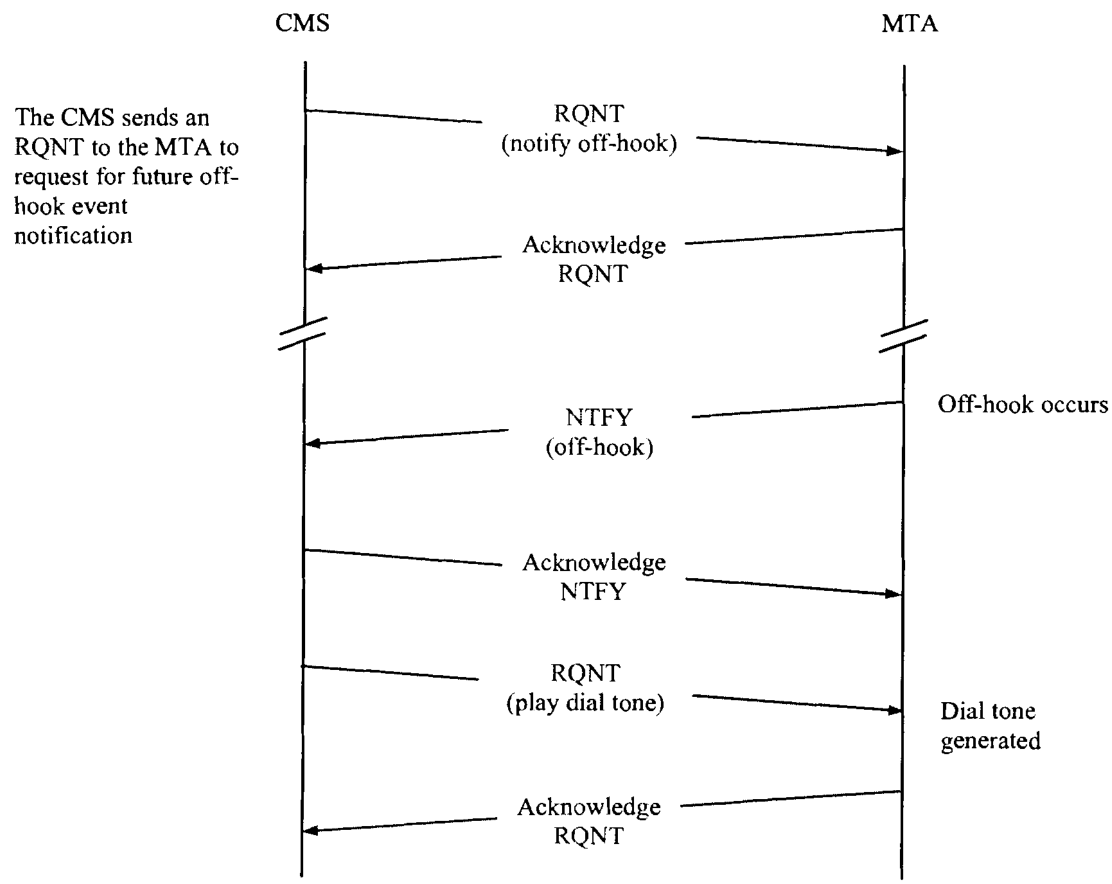


Figure 5. NCS Simplified Call Flow

The type of information requested by the CMS via the RQNT command depends highly on the underlying signaling package. A package defines the events and the signals that are relayed via call signaling. Section 4.3 presents a few packages that have been introduced in the NCS protocol to meet different endpoint requirements.

4.3 NCS Basic Packages

The previous section describes the signaling process used within the NCS framework to notify events and to generate signals. These relayed events and signals are collectively grouped into packages. Instead of changing the basic protocol when new functionality is required to support new services, new packages are introduced. The next sections present the developed packages for the NCS protocol.

4.3.1 NCS Supported Packages

PacketCable currently defines three packages for the NCS protocol: L-Line, Line Control Signaling (LCS), and Fax Relay (FXR). All three packages have been incorporated into the IPCablecom architecture to support telephony services [6].

The L-Line package was designed specifically for analog endpoints. Since this package provides basic signaling functionalities to interact with telephony equipment such as telephone, facsimile machine, and analog modem, the L-Line package is mandatory for both PacketCable and IPCablecom networks.

The LCS package was introduced to relay line supervision signals such as ring, on-hook, and off-hook. This package allows events to be relayed via the media stream instead of via the signaling protocol to allow interoperability with legacy equipment.

The FXR package provides signaling functionalities for fax relay services. Based on the ITU-T T.38 specification, the fax relay service provides the transfer of facsimile documents over IP. Instead of using the pass-through fax service which encodes fax signals with voice codecs, the fax relay service transfers fax data using a protocol that is specifically designed for fax[3].

Since this thesis focuses on the signaling interface for analog access lines, details of the L-Line package are presented in the next section.

4.3.2 L-Line Package

The L-Line package contains all the basic events/signals that can be detected/generated on analog access lines. Table 1 provides an example of the supported events and signals in the L-Line package. For a complete list of all the supported events and signals, please refer to Appendix A.

---

[3] Fax protocols are very sensitive to jitter and packet loss.

Table 1. Subset of the L-Line Package Definition [5]

Code	Description	Event	Signal
dl	Dial tone		√
rg	Ring		√
hd	Off-hook transition	√	
hu	On-hook transition	√	

Application of the L-Line package is best illustrated by an example. Recall from Section 4.2 that the RQNT command is used to request for events notification and/or signals generation. Thus, if the CMS requests the MTA to ring a telephone, the following NCS message[4] is sent:

RQNT 1000 aaln/1@mta1.pclabs.com
X: 1500
S: rg
R: hd

The above message requests the MTA with a Fully Qualified Domain Name (FQDN) of mta1.pclabs.com to generate a ring signal on its first endpoint. Typically, any signals or events that are associated with a package are preceded by a package name. Thus, in the above NCS command, the signal request should have been denoted by "S: L/rg" instead of "S: rg". However, because the L-Line package is the default NCS package, the default package is always assumed in the absence of a package name. In addition, the above NCS message requests the MTA to notify when the telephone on its first endpoint transitions off-hook. Subsequently, when the MTA detects the off-hook event, the

[4] NCS command explanation [5]:
RQNT <Transaction ID> <endpoint type>/<endpoint number>@<mta name>
X: <request ID>
S: <requested signal>
R: <requested event>
aaln = analog access line
rg = ring
hd = off-hook

following NTFY message[5] is triggered:

NTFY 2000 aaln/1@mta.pclabs.com
X: 1500
O: hd

The preceding two examples describe only a small subset of the NCS signaling call flow. Detailed analysis of the complete call flow is beyond the scope of this thesis. However, the two examples are sufficient to illustrate how a package is integrated with the signaling protocol. Having presented the basic telephony package in this section, the next section describes the E-Line package, which is an extension to the L-Line package.

---

[5] NCS command explanation [5]:
NTFY <Transaction ID><endpoint type>/<endpoint number>@<mta name>
X: <request ID>
O: <observed event>

5 E-Line Package

Due to deviations in telecommunication implementations between different countries, the E-Line package was added to the IPCablecom project. Based on the default L-Line package described in Section 4.3.2, the E-Line package is an extension that specifically addresses regional differences. Using the same signaling mechanism for the L-Line package, the E-Line package introduces two new groups of signals, as illustrated in Table 2:

Table 2. E-Line Package Definition [7]

Code	Description	Event	Signal
cr(x)	Cadence-ringing		√
ps	Pulsed Signal		√

The first group is the Cadence-ringing signal. The focus of this group is to provide 128 new ring cadences to comply with different country specifications. The second group of new signals is the Pulsed Signal. The Pulsed Signal group is designed to provide configurable interfaces for pulse signal generation [7]. Since both the Cadence-ringing signal and the Pulsed Signal are the focus of this thesis, the next two sections describe these signals in detail.

5.1 Cadence-ringing

Although the basic L-Line package provides several interfaces to generate different kinds of ring signals on an endpoint, these interfaces are not sufficient to meet the diverse ring requirements from different countries. For instance, the Germany specification mandates that the ring duration[6] must be 13.2 seconds long. Thus, the default L-Line package cannot be used in Germany since the default package only supports the maximum ring duration of 6 seconds.

---

[6] Ring duration = the time between the initial application of the ring signal (including the ring and idle periods) and when the ring pattern repeats.

To provide for a flexible interface in supporting ring cadences per international specifications, the E-Line package added the Cadence-ringing signal group. Denoted by the signaling parameter *cr(x)* where *x* is any number between 0 and 127, the Cadence-ringing signal group provides 128 new ring signals [7]. The following NCS message provides an example to request for the first Cadence-ringing signal using the RQNT command:

```
RQNT 1000 aaln/1@mta1.pclabs.com
X: 1500
S: E/cr(0)
```

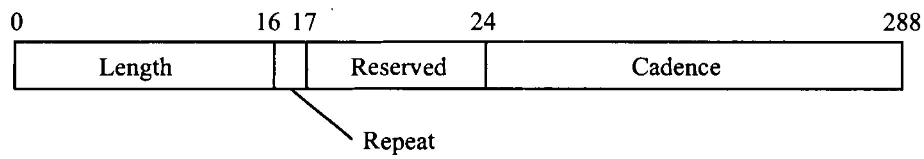
In the above NCS message, the E-Line package name "E" must precede the Cadence-ringing signal since the E-Line package is not the default signaling package.

When the E-Line package was added to the IPCablecom project, default ring patterns for the first five Cadence-ringing signals were provided. However, since these default values are intended only for initial product configurations, they are not presented in this report. Instead, this project examines the provisioning process, which is a management process designed to configure device parameters. As the next section describes, the provisioning process provides a very simple mean to configure the Cadence-ringing signal.

5.1.1 Cadence-ringing Provisioning

To allow flexibility in device configurations, provisioning interfaces have been incorporated into the IPCablecom architecture. Each provisioning interface employs the Simple Network Management Protocol (SNMP) to configure device parameters. For details concerning SNMP, please refer to Appendix B. As an example of configuring a device via the provisioning interface, MTA settings such as ring cadence values can be controlled via this interface.

When IPCablecom created the Cadence-ringing signal group, a new object was also added to the provisioning database for signal configuration. Identified by the object identifier *pktcSigDevRingCadenceTable*, this new object allows cable operators to configure each of the 128 Cadence-ringing signals. A single entry within the *pktcSigDevRingCadenceTable* object specifies the ring cadence for one Cadence-ringing signal. Each ring cadence is defined using a 288-bit pattern, as illustrated in Figure 6 [2].



Bits	Field Name	Description
0 – 15	Length	Identifies the length in bits of the ring cadence duration
16	Repeat	Represents the repeatable characteristics. A value of 0 indicates a repeatable ring pattern and 1 denotes a non-repeatable one
17 – 23	Reserved	Reserved for future expansion
24 – 287	Cadence	Defines the ring cadence. Each bit denotes 50 millisecond (msec) duration, where a '1' represents ring and a '0' represents silence

Figure 6. Ring Cadence Bit Definition [2]

As an example, to generate a repeatable Cadence-ringing signal of 2-seconds ring and 4-seconds idle, the following ring pattern, specified using hexadecimal notation, should be provisioned:

00 78 00 FF FF FF FF FF 00 00 00 00 00 00 00 00 00 00

The *pktcSigDevRingCadenceTable* object provides a ring cadence resolution of 50 msec. This resolution is much finer than the 100-msec resolution provided by the default L-Line package. Furthermore, since the actual ring cadence pattern is 264 bits long, ring duration of 13.2 seconds can be supported. Thus, countries such as Germany which mandates long ring duration can be generated by the Cadence-ringing signal.

5.1.2 Cadence-ringing Call Flow

The application of the Cadence-ringing signal is best illustrated by a call flow. Figure 7 presents the call flow for generating the first Cadence-ringing signal.

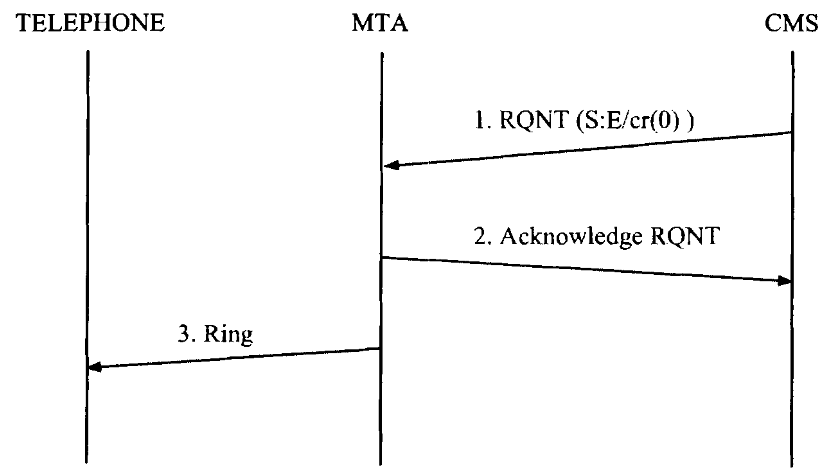


Figure 7. Cadence-ringing Call Flow

1. The CMS sends the following RQNT message to the MTA to trigger the first Cadence-ringing signal:

   RQNT 1000 aaln/1@mta1.pclabs.com MGCP 1.0 NCS 1.0
   S: E/cr(0)
   X: 1500

2. The MTA acknowledges the RQNT message.
3. The MTA retrieves from its provisioning database the definition of the first Cadence-ringing signal. It then applies the ring cadence to its first endpoint.

5.2 Pulsed Signal

The second component of the E-Line package is the Pulsed Signal group. Although the Pulsed Signal group is comprised of nine different signals, only the Metering Pulse Burst (mpb) signal is examined. As the next section describes, the Metering Pulse Burst signal is used to account for metering pulse changes.

5.2.1 Metering Pulses

Metering is the real-time process used by the customer to track the cost of an outgoing telephone call. Most commonly used in the Public Call Office (PCO) such as public payphones, the metering process employs the concept of a pulse rate to determine call charges[7]. For instance, using a pulse rate of 180 seconds, a 3-minute call will result in a call charge of one unit. The PCO will then map this one unit to the proper charging rate, and notify the user of the estimated call charges or the approximated remaining call duration. Similarly, when the call duration is between 3 to 6 minutes, a call charge of 2 units is applied. Non-PCO users, such as residential telephony subscribers, can also estimate the call charges by installing metering equipment in their homes. Although metering equipment in customer premises can be used to estimate the charge of each call, reports generated by this equipment are not used for billing. Meters in the telephone exchange[8] are used to calculate the actual call charges [8].

The pulse rate is determined based on a number of factors:

- Distance between the calling and the called party
- Time of the day
- Day of the year

For long-distance calls, the pulse rate is typically much smaller than local calls. Hence, the number of chargeable units applied for a long-distance call is much higher than a local call. In addition, depending on whether the call is made during peak hours or during holidays, the pulse rate for the call varies. Table 3 provides an example of the pulse rates between Meerut and Jaipur city in India [8].

---

[7] Call charges defined here and onwards are the cost of outgoing telephone calls to the customer.
[8] Also known as switchboard, a telephone exchange is a central office containing switches and other equipment to establish connections between telephones.

Table 3. Pulse Rate Example [8]

Description	Time	Pulse Rate (seconds)
Full Pulse Rate	9 am – 10 pm	15
Half Pulse Rate	10 pm – 9 am	30
Sundays and National Holidays	24 hours	30

To relay pulse rate information between the telephone exchange and the end user[9], a 16kHz pulse (metering pulse) is generated every 'pulse rate' seconds by the telephone exchange. Upon detection of this metering pulse, metering equipment at the end user can calculate the call charges accordingly. Metering pulses are initiated immediately after the remote end answers the call, and are terminated when either party disconnects. Not only is the centralized architecture of 16kHz Metering easy for maintenance, it provides accurate metering results since the metering pulses are aligned with the actual pulse rate for billing.

5.2.1.1 Metering Pulses at the MTA

Although metering pulses were designed to work for the PSTN, they can also be applied in the IPCablecom network. However, instead of relaying metering pulses through the intricate networks of the PSTN and the IP Backbone, metering pulses are generated locally by the MTA. Triggered via special signaling parameters from the CMS, the MTA is capable of generating the exact metering pulses that are experienced by PSTN clients.

A metering pulse feature that is examined in this thesis is the ability to handle occasional metering pulse changes at the MTA. For instance, if the subscriber is re-routed to an operator during a call, special metering pulses must be generated to reflect the additional operators' charge. The metering pulse change feature can be simulated at the MTA via the Metering Pulse Burst (mpb) signal. Upon receiving the *mpb* signal request, a burst of metering pulses can be generated so that the metering equipment attached to the endpoint can account for the supplementary charge [7].

---

[9] End users defined here and onwards are PCO users as well as residential users.

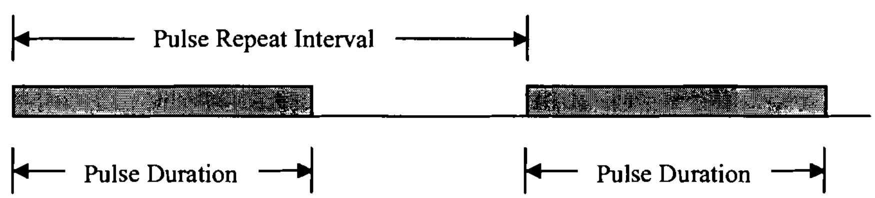
The following RQNT command provides an example to generate the default *mpb* signal, which is a single 150-msec 16kHz pulse:

RQNT 1000 aaln/1@mta1.pclabs.com
X: 1500
S: E/ps(lt=mpb)

Unlike Cadence-ringing signals which can only be defined via the provisioning interface, signals from the Pulsed Signal group can be configured via both the call signaling and the provisioning interface.

5.2.2 Pulsed Signal Parameters via the Signaling Interface

To assist in the application of the *mpb* signal, new signaling parameters have been introduced. The three new parameters are illustrated in Figure 8 [7]:



Pulsed Signal Parameter	Description
Pulse Duration (pd)	Defines in msec the length of a single pulse
Pulse Repeat Interval (pr)	Defines in msec the repeat interval of a single pulse
Repetitions (rep)	Identifies the total number of pulses to apply (including the initial and the repeated pulses)

Figure 8. Pulsed Signal Parameters

For example, to request for a Metering Pulse Burst signal composed of 2 pulses with a pulse length of 100 msec and a pulse repeat interval of 1000 msec, the following NCS message can be triggered:

```
RQNT 1000 aaln/1@mta1.pclabs.com
X: 1500
S: E/ps(lt=mpb, pd=100, pr=1000, rep=2)
```

In the absence of Pulsed Signal parameters in the NCS message, default configurations for the Pulsed Signal apply.

Apart from configuring the signals from the Pulsed Signal group via the call signaling interface, signal parameters can also be modified via the provisioning interface. As the next section illustrates, the provisioning interface provides additional configurable parameters over the call signaling interface.

5.2.3 Pulsed Signal Provisioning

To provision the signals from the Pulsed Signal group, a new object called *pktcSigPulseSignalTable* was added to the provisioning database. The parameters that can be provisioned are [2]:

- Pulse Duration – defines in msec the length of a single pulse
- Pulse Repeat Interval – defines in msec the repeat interval of a single pulse
- Repetitions – identifies the total number of pulses to apply (including the initial and the repeated pulses)
- Pulse Frequency – identifies the frequency of the generated pulse
- Pulse DB Level - specifies the decibel level of the generated pulse

The first three provisional parameters are analogous to the 'pd', 'pr', and 'rep' parameters from the call signaling interface. The Pulse Frequency and the Pulse DB Level parameters allow cable operators to modify the Metering Pulse Burst signal from the default 16kHz pulse with an amplitude of -13.5 dBm [2].

When Pulsed Signal parameters are changed simultaneously via both the signaling and the provisioning interface, the values specified via the signaling interface take precedence.

5.2.4 Pulsed Signal Events

The CMS can monitor the MTA's behavior by observing events reported by the MTA. A total of three different events can be reported in association with the Pulsed Signal: Operation Complete (oc), Operation Failure (of), and Pulse Complete (pc) [7].

The Operation Complete (oc) event is triggered when a signal has been completed on an endpoint. For instance, once the entire Metering Pulse Burst signal (including the number of repetitions) has been applied, the *oc* event is reported to indicate that Pulsed Signal operation has been completed.

In contrast to the *oc* event, an Operation Failure (of) event can also be generated by the MTA. Typically, the *of* event is triggered when the signal generation was interrupted either by receiving a new signal generation requests or by local events such as on-hook.

The Pulse Complete (pc) event is triggered after the generation of individual pulses. For instance, if the Metering Pulse Burst signal is requested with the Pulsed Signal parameters *rep=2*, then two *pc* events will be reported.

To request for the notification of any Pulsed Signal events, the CMS can include them in the Requested Event parameter (R:) of the NCS message. As an example, to request for the notification of an Operation Complete event upon the completion of the *mpb* signal, the following RQNT message can be triggered:

```
RQNT 1000 aaln/1@mta1.pclabs.com
X: 1500
S: E/ps
R: oc
```

After the MTA completes applying the Metering Pulse Burst signal, the following notification command is sent to the CMS:

NTFY 2000 aaln/1@mta.pclabs.com
X: 1500
O: E/pc(mpb)

Having discussed the various configurable properties and the reported events associated with the Pulsed Signal, the call flow for this signal is presented in the next section.

5.2.5 Pulsed Signal Call Flow

Figure 9 illustrates the call flow for generating the Metering Pulse Burst signal at the MTA.

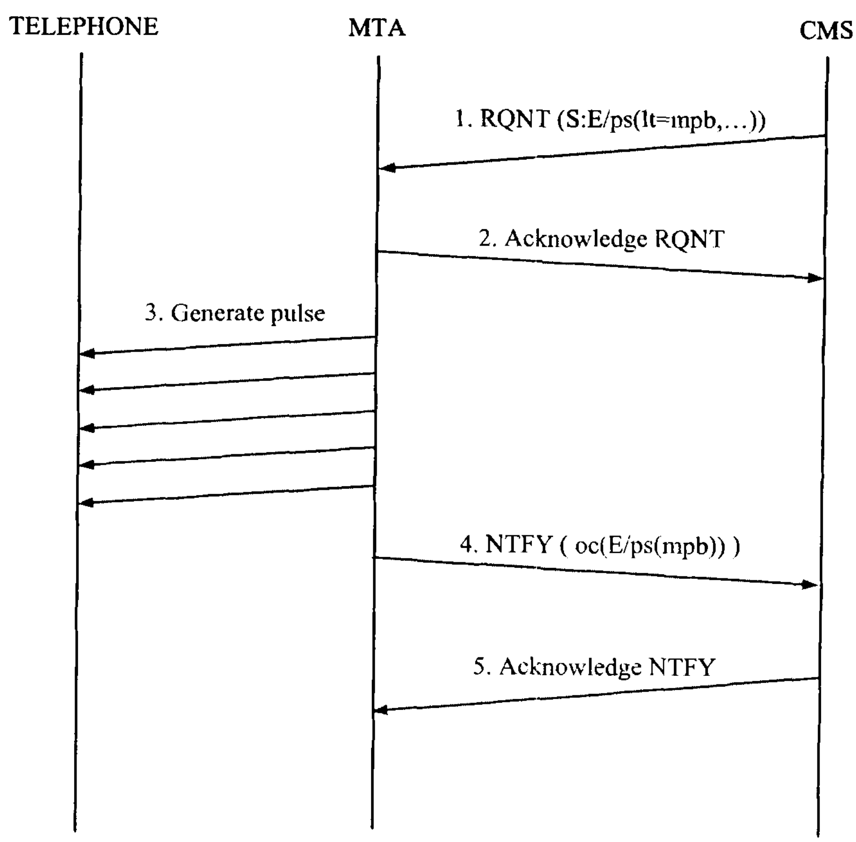


Figure 9. Metering Pulse Burst Call Flow

1. The CMS sends the following RQNT message to the MTA to trigger the Metering Pulse Burst signal:

   RQNT 1000 aaln/1@mta1.pclabs.com MGCP 1.0 NCS 1.0
   S: E/ps(lt=mpb, pd=100, pr=1000, rep=5)
   X: 1500
   R: oc

2. The MTA acknowledges the RQNT.
3. The MTA begins generation of the metering pulses on its first endpoint.
4. Once the MTA completes the application of the five metering pulses, the MTA notifies the CMS of the Operation Completion event. This notification is triggered because the CMS had requested for the *oc* event in the previous RQNT message:

   NTFY 5600 aaln/1@mta1.pclabs.com MGCP 1.0 NCS 1.0
   X: 1500
   O: oc(E/ps(mpb))

5. The CMS acknowledges the Operation Complete event.

6 Design and Implementation

The preceding sections illustrate the process involved in utilizing the E-Line package. This section provides an overview of the modifications required on the MTA to support this new signaling package.

6.1 E-Line Package Interface Specification

Since the E-Line package employs the same signaling interface as the L-Line package, changes for the E-Line package must be properly integrated into the MTA's existing signaling architecture. Furthermore, any modifications made for the E-Line package must not affect the basic operation of the L-Line package. Therefore, when designing the signaling component of the E-Line package, backwards compatibility is an important consideration.

Another important consideration in the design of the E-Line package is code sharing. Since some of the core processing code on the MTA is used by various embedded telephony solutions at Broadcom, the design of the E-Line package must promote code reusability and facilitate simple upgrades.

With the design requirements defined, an understanding of the MTA processing is required before proceeding with the implementation. Since the project involves the interactions between many different MTA modules such as the signaling stack, endpoint processing, and provisioning, the interfaces between each of these modules must be well understood.

6.2 E-Line Package Signaling Flow

To illustrate the procedure involved in processing a signal request for the E-Line package, Figure 10 provides an overview of the signaling flow.

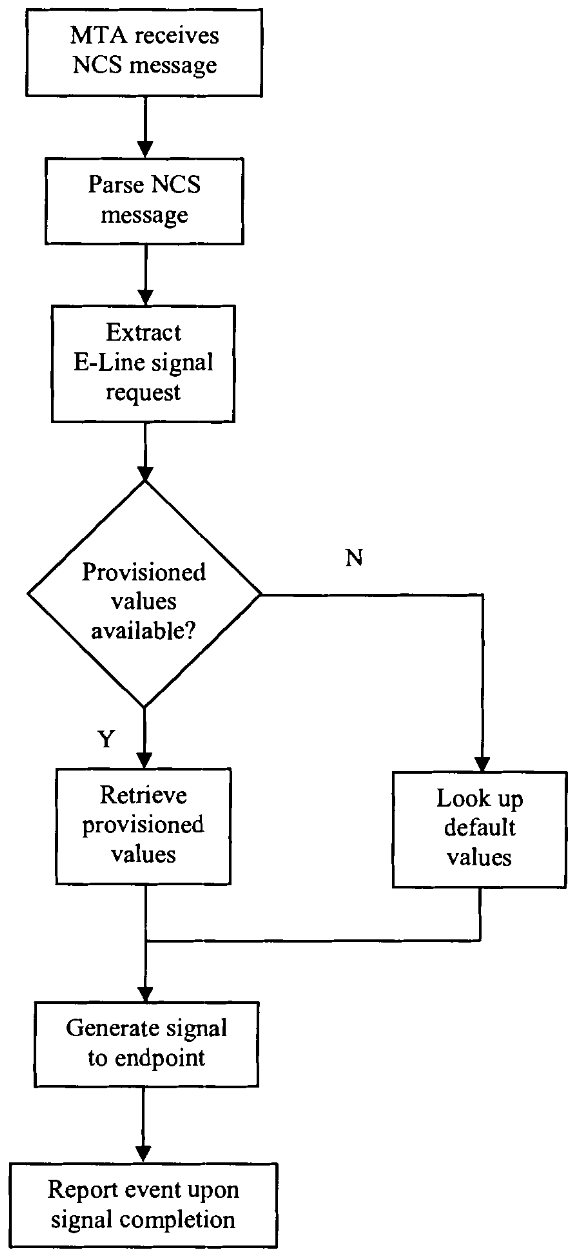


Figure 10. E-Line Package Signaling Flow

To ensure proper functioning of the E-Line package, error checks have been incorporated within each stage of Figure 10. Any errors that are encountered during each stage must be gracefully handled. For instance, if an NCS message includes the Cadence-ringing signal request of *cr(130)*, this message must be rejected since the Cadence-ringing index is beyond the supported range. Any resources that are allocated to process this erroneous message must be released and the system reverts to the original state.

6.3 Signal Generation

Signals are generated at the MTA via two separate paths. As illustrated in Figure 11, the two distinct paths serve very different purposes and require the interactions of different MTA components.

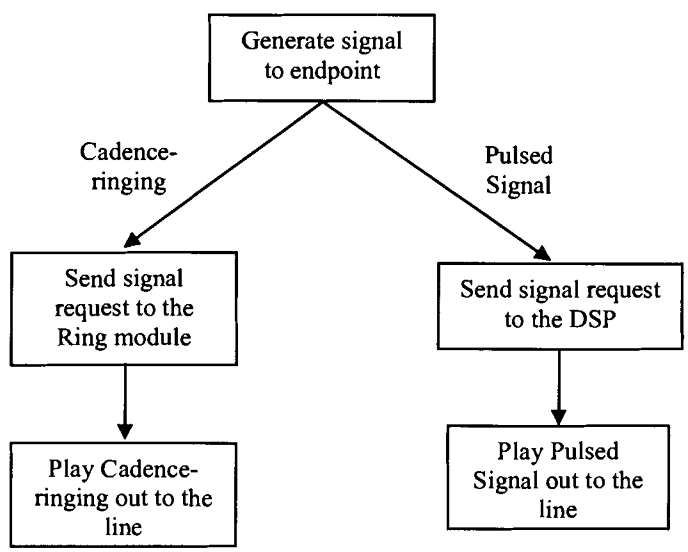


Figure 11. Signal Generation

The left path of Figure 11 is used to generate ring signals via the Ring module. The Ring module, better known as the Channel Associated Signaling (CAS) control module, is responsible for low level messaging and event handling with analog or digital devices. In particular, the CAS module communicates with the SLIC to trigger ring generation and hook state detection. The Cadence-ringing signal of the E-Line package is generated via the Ring module. Since the CAS module was originally designed to support only the short-duration ring signals from the L-Line package, this module has been expanded in this project to support both short- and long-duration ring signals.

The Pulsed Signal, on the other hand, does not require interaction with the CAS module. As illustrated in the right path of Figure 11, the generation of the Pulsed Signal requires interface to the DSP. Because of the definition of the Pulsed Signal, a state machine has been incorporated in the DSP to handle the various signal states. The possible states of the state machine are:

- On – Pulsed Signal is being generated: tone on
- Off – Pulsed Signal is being generated: tone off
- Idle – no Pulsed Signal generation

Prior to examining the Pulsed Signal State Machine in details, an understanding of the state machine terminologies is essential. Table 4 provides an overview of the state machine parameters relating to the Pulsed Signal.

Table 4. Pulsed Signal State Machine Parameters

State Machine Parameters	Definitions
duration	Specifies the duration of the "on" tone
period	Defines the repeat interval – includes both "on" and "off" tone
sampleCount	Denotes the running count on the number of Pulsed Signal (in 1ms resolution) that have been generated. sampleCount is re-initialized once a single Pulsed Signal pulse has been generated
pulseCount	Represents the number of remaining Pulsed Signal pulses to generate

The following figure depicts the state transition diagram for the Pulsed Signal State Machine.

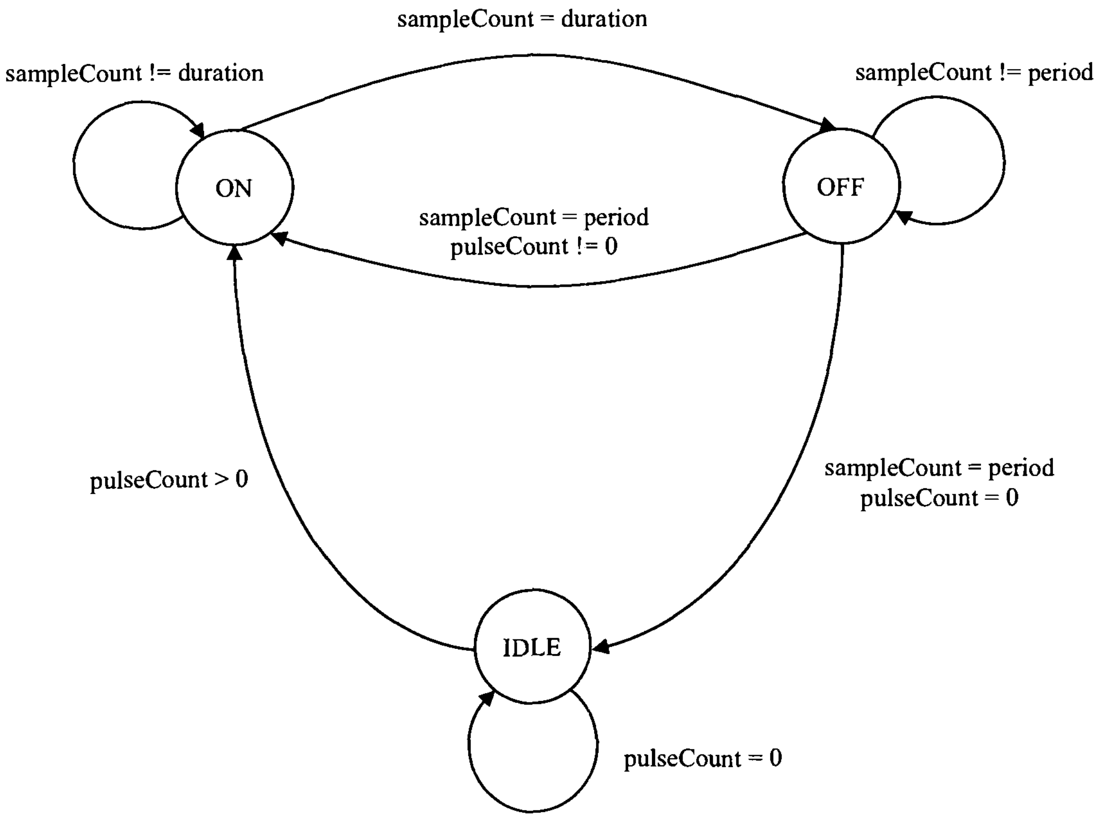


Figure 12. Pulsed Signal State Transition Diagram

The Pulsed Signal State Machine is executed every 1 msec[10], with the sampleCount parameter being incremented every tick. The parameters in Table 4 are then examined after each tick to determine the next state and the appropriate actions.

Even though the Pulsed Signal State Machine is currently only used to generate the Metering Pulse Burst signal, it was designed and implemented with considerations that other signals from the Pulsed Signal group can be applied using a similar approach.

---

[10] The DSP processing rate was determined to be at 1msec to satisfy the conflicting requirements of memory consumption and instruction cycle utilization. If the processing rate was set too small, large buffers would have been required to store the processed data. If, however, the processing rate was set too large, too much time would have been spent in context switching instead of actual data processing.

7 Results and Analysis

To validate the E-Line package, a thorough testing process has been implemented. A summary of the various testing procedures and the collected results are presented in this section.

Testing was accomplished by compiling the E-Line package in debug mode. Inserting appropriate breakpoints throughout the implementation, the output of each functional block was examined. In doing so, individual features such as the handling of signaling messages and the control from the provisioning module were verified. Furthermore, signals generated at the endpoint were examined via tools such as an oscilloscope and Cool Edit Pro[11]. The gathered results indicated that the signals were generated with good accuracy levels. The measured signal amplitude was within 0.96% of the expected signal amplitude. In addition, the timing and the frequency of the recorded signal were respectively within 0.05% and 0.28% of the signal requirements[12]. The discrepancies in the gathered results can be attributed by the precision of the measuring tools. Better accuracy levels can be obtained by using measuring equipment with finer resolution.

The second phase of testing was Integration Testing. Since the software involves the interaction between many different components, Integration Testing is essential to validate the compatibility between different modules. For instance, to trigger the Cadence-ringing signal *cr(0)*, the call signaling module must interact with the endpoint module to ensure that the proper signal is triggered. Furthermore, the endpoint module must coordinate with the provisioning module such that the correct ring cadence is retrieved for signal generation. Through a series of in-depth testing similar to the preceding example, all the components created for the E-Line package were confirmed to be properly integrated.

---

[11] Cool Edit Pro is professional software used for audio recording and creation.
[12] Even though the IPCablecom specification did not impose a tolerance range for E-Line package signals, the measured error range was confirmed to be acceptable after interoperability tests with other equipment.

Finally, the last phase of testing involved System Testing. The purpose of this testing phase is to validate that the complete system can operate as a whole. To initiate this process, the system as illustrated in Figure 13 was set up.

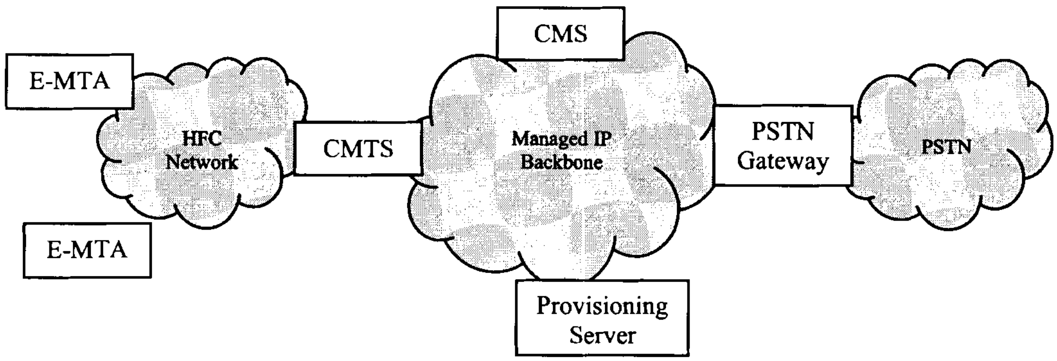


Figure 13. System Testing Environment

The system illustrated in Figure 13 is a scaled-down version of the IPCablecom reference architecture from Figure 2. The complete IPCablecom reference architecture is not required for System Testing because entities such as the Announcement Server are not required to validate the proper functioning of the E-Line package. With the exception of the PSTN Gateway, all of the necessary equipment from Figure 13 is readily available in the laboratory at Broadcom.

The System Testing phase was divided into two parts: On-Net Testing and Off-Net Testing. On-Net Testing involves the verifications of components that reside only in the cable infrastructure. For instance, a telephone call made between two E-MTAs is termed on-net call. Off-Net Testing requires the interaction with the PSTN interface. An off-net call denotes a telephone call between an E-MTA and a PSTN device.

On-Net Testing was done within the laboratory at Broadcom. Triggering the E-Line package at the CMS, the expected signals were confirmed at the endpoint. Additional testing was performed to ensure that error scenarios were properly handled. For instance, when the CMS requests for signal generation on an endpoint that was not available[13], an appropriate error response was generated.

Since a PSTN gateway was not readily available in Broadcom laboratories, off-net testing was accomplished in off-site facilities. Call flows that are similar to that from Figure 9 were used to validate the E-Line package.

[13] An endpoint can be configured to be unavailable via the provisioning interface.

8 Conclusion

With the exponential growth of the packet-based data network, VoIP is gaining mass popularity with the community. As new signaling packages are introduced to align and to exceed the PSTN experience, the general transition from PSTN to VoIP is imminent. This thesis contributes to that transition. The focus of this thesis has been the addition of a subset of the E-Line package to the E-MTA. To provide the context for this, the cable infrastructure and the package definitions are introduced, and a thorough examination of the E-Line package is provided. The design, implementation, and verification of the E-Line package are also presented. Through a series of extensive testing, the implementation of the E-Line package was verified to be compliant with the specifications created by IPCablecom. With the implementation successfully achieved, the E-Line package was integrated into Broadcom's VoIP residential broadband gateway. The stage has also been set for exploring new packages for enhancing the telephony experiences in the cable infrastructure. Future development includes adding the complete E-Line package to the E-MTA.

9 References

[1] M. Ainslie, "Caller ID FAQ". Online. Internet. Available http://artofhacking.com/files/callerid/CLI_FAQ.HTM#Q_6

[2] G. Beacham, S. Kumar, et. all. "Network-Based Call Signaling (NCS) Signaling MIB for PacketCable and IPCablecom Multimedia Terminal Adatpers (MTAs)", October 2004

[3] Cable Television Laboratories, "Data-Over-Cable Service Interface Specifications", July 2000

[4] Cable Television Laboratories, "PacketCable 1.0 Architecture Framework Technical Report", December 1999

[5] Cable Television Laboratories, "PacketCable Network-Based Call Signaling Protocol Specification", April 2004.

[6] Cable Television Laboratories, "PacketCable NCS Basic Packages Technical Report", March 2002.

[7] European Telecommunications Standards Institute, "Digital Broadband Cable Access to the Public Telecommunications Network; IP Multimedia Time Critical Services; Part 4: Network-Based Call Signaling Protocol", Dec 2002.

[8] Pep Infotech Limited, "Public Call Office General Information". Online. Internet. Available http://www.pepteller.com/pep04.htm

[9] Trans-Video, "Cable Modems". Online. Internet. Available http://www.trans-video.net/cablemodems.html

Appendices

Appendix A L-Line Package Definition

The L-Line package provides a comprehensive list of events and signals that are applicable to analog access lines. Since this package is the default telephony package, the application of this package is sufficient to support basic telephony operations such as fax handling, caller ID, and call waiting. The following table provides the complete package definition for the L-Line package.

Table 5. L-Line Package Definition [5]

Code	**Description**	**Event**	**Signal**	**Additional Information**
0-9,*,#,A,B,C,D	MFPB (DMTF) tones	√		
bz	Busy tone			
cf	Confirmation tone			
ci(ti, nu, na)	Caller Id			'ti' denotes time, 'nu' denotes number, and 'na' denotes name
dl	Dial tone		√	
ft	Fax tone	√		
hd	Off-hook transition	√		
hf	Flask hook	√		
hu	On-hook transition	√		
L	MFPB (DMTF) long duration	√		
ld	Long duration connection	√		
ma	Media start	√		
mt	Modem tones	√		
mwi	Message waiting indicator		√	
oc	Operation complete	√		
of	Operation failure	√		
osi	Open interval		√	
ot	Off-hook warning tone		√	
r0, r1, r2, r3, r4, r5, r6 or r7	Distinctive ring (0..7)		√	
rg	Ring		√	
ro	Reorder tone			
rs	Ringsplash		√	

Code	**Description**	**Event**	**Signal**	**Additional Information**
rt	Ring back tone		√	
sl	Stutter dial tone		√	
t	Timer		√	
TDD	Telecomm Devices for the Deaf (TDD) tones	√		
vmwi	Visual message waiting indictor		√	
wt1, wt2, wt3, t4	Call waiting tones		√	
X	MFPB (DTMF) tones wildcard	√		

Details concerning the individual events and signals of Table 5 are available in [5].

Appendix B SNMP

SNMP is a widely used protocol for configuration management. This application-layer protocol enables network administrators to manage the information exchange between network devices. The SNMP protocol employs a request-reply model to allow for the interaction between a management station and multiple agents.

A management station contains intelligence that enables network administrators to perform network management operations. Typically, a management station resides in a Provisioning Server inside the network core. Agents are devices containing configurable parameters that can be changed by a management station via the SNMP interface. Examples of agents are routers and hubs where parameters such as link speed and performance statistics can be changed.

Information managed via the SNMP interface is arranged in a virtual information database called the Management Information Base (MIB). Typically organized in a hierarchically fashion, the MIB categorizes each configurable parameter into an object for easier access. Each managed object is identified by an object identifier.