Implementation and Analysis of VoIP CPE Management System using TR-069

DARWIS DARWIS

KTH Information and Communication Technology

Master of Science Thesis
Stockholm, Sweden 2008

COS/CCS 2008-26

Implementation and Analysis of VoIP CPE Management System using TR-069

Darwis, Darwis

November 17, 2008

Master of Science thesis project performed at
42networks AB,

Stockholm, Sweden

Supervisor & Examiner:	**Industry Advisor:**
Professor Gerald Q. Maguire Jr.	Olle Cederberg
KTH / Royal Institute of Technology	42networks AB, Vice President R&D

Abstract

Customer Premises Equipment (CPE) management is underestimated by the CPE vendors and services providers while it is in fact one of the most important aspects to ensure the high quality of service. Many people still think CPE management is the same as network management. Thus, they use the Simple Network Management Protocol (SNMP) to manage their CPEs. However, SNMP alone was thought not to scale nor to support the provisioning of the types of services which internet services providers must support today.

This thesis highlights the importance of CPE management, how it is implemented using the TR-069; a CPE management protocol defined by the DSL Forum, and how a management system can be used for VoIP service management, and whether a CPE should implement TR-069 or SNMP as the management system to support. In the addition, the TR-069 will be compared against the SNMP to determine which one is more suitable for CPE management. An interesting conclusion is that while TR-069 does have some advantages over SNMP for managing services rather than simply managing the device, these advantages are not a large as initially believed nor has TR-069 avoided the problem of proprietary management information which SNMP has demonstrated.

Keywords: Customer Premises Equipment, Simple Network Management Protocol, service, Voice Over Internet Protocol, TR-069.

Sammanfattning

Customer Premises Equipment (CPE) skötseln är undervärderad av CPE försäljarna och tjänste leverantörerna meddans det faktiskt är en av de mest viktiga aspekterna för att tillförsäkra hög quality of service. Många personer tror fortfarande att CPE skötseln är det samma som att sköta ett nätverk. Så, de använder Simple Network Management Protocol (SNMP) för att sköta deras CPE:er. Emellertid, SNMP ensamt var inte tänkt att skala eller att ge stöd vid försörjning av typer av tjänster som internet tjänst leverantörer måste stödja idag.

Den här avhandlingen framhäver det väsentliga med CPE skötsel, hur det implementeras vid användande av TR-069; ett CPE skötsel protocol definerat av DSL forum, och hur detta administrations system kan användas för att sköta VoIP tjänster. Tilläggande så kommer avhandligen att jämföra TR-069 och SNMP för att bestämma vilken av dem som är mer lämplig för CPE administration. En intressant sammanfattning är att meddans TR-069 har några fördelar över SNMP för att sköta tjänster hellre än att enkelt sköta enheten, dessa fördelar är inte så stora som man trott från början. Dessutom, TR-069 ser inte ut att kunna övervinna problemet med privatägd (användande av privat MIB) information som SNMP har demonstrerat.

Nickel.ord: Customer Premises Equipment, Simple Network Management Protocol, service, Voice Over Internet Protocol, TR-069.

Acknowledgement

I would like to express my gratitude to my friends and colleagues in 42networks. Their support, advices and encouragement have helped me a lot in doing this master thesis.

I would like to thank Olle Cederberg, the VP R&D in 42networks AB, who gave me a chance to do my thesis at 42networks. Thanks to Peter Larsson also from 42networks, who has shared his ideas about CPE management and DRG parameters. Thanks to Charles Foster who has give me his rough estimation about the memory resource and processing power in the DRG.

My gratitude and respect to Professor Gerald Q. Maguire Jr. from KTH, who was the academic advisor, supervisor, and examiner for this Master Thesis. I would like to thank you for your patience, guidances and valuable advices along the way.

Last but not least, I would like to express my gratitude to my girlfriend Lily Feng who has been always there for me. Thank you for your support from the begining until the end of this thesis.

Table of contents

List of figures

List of tables

List of abbreviations and acronyms

ACS	Access Configuration Server
ADSL	Assymetric Digital Subscriber Line
CPE	Customer Premises Equipment
CWMP	CPE WAN Management Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DRG	Digital Residential Gateway
DRG-EM	Digital Residential Gateway-Element Manager
DSL	Digital Subscriber Line
FQDN	Fully Qualified Domain Name
FTTP	Fiber to the Premises
GSM	Global System for Mobile
GUI	Graphical User Interface
HDSL	High-rate Digital Subscriber Line
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IANA	Internet Assigned Numbers Authority
IDSL	ISDN Digital Subscriber Line
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILBC	Internet Low Bitrate Codec
IP	Internet Protocol
IPTV	Internet Protocol Television
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
JRE	Java Runtime Environment
LAN	Local Area Networks
MAC	Media Access Control
MCC	Multi Cable Converter
MGCP	Media Gateway Control Protocol
MIB	Managed Information Based
NAT	Network Address Translation
NMS	Network Management System
OEM	Original Electronic Manufacturer
OID	Object Identifier
OSI	Open Systems Interconnection
OUI	Origianization Unit Identifier
PBX	Private Branch Exchange
PCM	Pulse Code Modulation
POTS	Plain Old Telephony System
PSTN	Public Switched Telephone Networks
QoS	Quality of Services
RADSL	Rate Adaptive Digital Line Subscriber
REDS	Redirection Server
RPC	Remote Procedure Call
RTCP	Realtime Transport Control Protocol
RTP	Realtime Transport Protocol
SCSI	Small Computer System Interface
SDSL	Symmetric Digital Line Subscriber

SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SOHO	Small Office Home Office
SSL	Secure Socket Layer
STUN	Simple Traversal of UDP through NAT
TCP	Transport Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
UADSL	Universal Assymetric Digital Subscriber Line
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USB	Universal Serial Bus
VAD	Voice Activation Detection
VDSL	Very High bit Digital Subscriber Line
VoIP	Voice over IP
W3C	World Wide Web Consortium
WAN	Wide Area Network
WWW	World Wide Web
XML	Extensible Mark-up Language
XMLNS	Extensible Mark-up Language Name Space

1 Introduction

1.1 Brief introduction to the problem

Managing Customer Premises Equipment (CPE) is an underestimated task. CPE vendors have focused on improving technical functions (i.e. adding more advanced features, providing better receiving quality). While service providers are trying to gain more customers by lowering the price and including more services in the subscription. However, both the CPE vendor and the service provider will gain the maximum benefit from these advanced feature and services ***only*** if there is a good and standardized CPE management system. Standardize in this case means to use CPEs which have common functions and features, common ways of managements and provide same high output quality (with respect to performance as perceived by the users). Here, management of a CPE that the CPE can be suitably controlled, in order to provide the customer (and the service provider) with better quality of service. The tasks of managing a CPE include installation, configuration, error detection and reporting, and troubleshooting.

42networks, a world leading Voice over Internet Protocol (VoIP) CPE vendor, has given serious consideration to CPE management. Their first generation of CPE management system, the Digital Residential Gateway-Element Manager (DRG-EM), was developed about four years ago. However, this is no longer suits the requirements for mass-managing CPEs because managing CPEs today also means ***managing the services offered through the CPE***. DRG-EM was developed based on Simple Network Management Protocol (SNMP) [10] and is only capable for managing 42networks' CPEs. 42networks believes that in order to grow their CPE business, there must be a good CPE management system available to their customers that can manage all of the CPEs in a standardized way.

The Digital Subscriber Line (DSL) Forum is aware of the importance of CPE management. However, the problem of CPE management has been underestimated by the public and many service providers. The DSL Home working group within DSL Forum has defined in a technical report a protocol for managing CPE from the Wide Area Network (WAN) side. This technical report was given the number 69 and finalized on May 2004. The popular name for this report is TR-069 CPE WAN Management Protocol (CWMP). We will refer to it throughout this report simply as TR-069.

1.2 Presentation of 42networks

This thesis had been carried out at 42networks AB, a company in Kista, Stockholm. This company develops and produces products and system solutions for Broadband Telephony, Triple Play for Fiber-to-the-Premises (FTTP), and Embedded VoIP-solutions.

The company was founded in 2003 via a management buy-out from Ericsson Business Innovation AB. Ericsson has a long tradition and deep knowledge of public telephony and had invested many years of research into broadband telephony. This knowledge enabled 42networks to created outstanding quality of service (QoS), to meet strict security requirement, and the robust capacity necessary to support a large installed base of subscribers. Customers of 42networks are telephony service providers and broadband operators. Currently, the company's products are in commercial operation in more than 30 countries.

The competitive advantages of 42networks include:

- VoIP competence,
- World leading partners,
- Early market presence.

The main product of 42networks is a Digital Residential Gateway (DRG). DRG is an analog telephony adapter which enables a user to make telephone call which is transmitted over an Internet Protocol (IP) network. Other product is including DRG management system and embedded VoIP module. Most products are sold through Original Electronic Manufacturer (OEM) partners. Some important OEM partners are Ericsson and Packetfront (for the FTTP), Alvarion, and Dovado.

The head-quarter of 42networks today is located in Stockholm, Sweden. There are also two research and development centers, located in Shanghai, China and London, England. Further information about 42networks can be found at this company website www.42networks.com.

1.3 Goal of this thesis

This thesis studies in depth on how TR-069 is implemented for managing CPE services and the comparison with SNMP. The initial goal was to understand how CPE management based on TR-069 should work, what are the differences between existing management systems and the management systems which TR-069 would or could support. Next, it was important to analyze whether the vendor or service provider could benefit by using TR-069 based CPE Management. Additionally, since this new management system was to replace old one, the new management system must also be able to be used for managing existing CPEs that do not support TR-069 as well as new CPEs that supports the new TR-069.

For the audiences with IP networking knowledge and interested in VoIP and CPE management, especially those who are planning to employ or upgrade an existing CPE management system, they should read start from chapter 2 to the end of this report. But for those who are only intererested in comparison between SNMP and CWMP, they may go directly to chapter 4, section 4.4.

1.4 Thesis Structure

This thesis starts (in Chapter 1) with a brief statement of the problem, a presentation of the company where the research has been carried out, and gives a clear statement of the goal.

Chapter 2 briefly discusses internet access especially via DSL. This chapter provides an introduction to the technology and gives a brief review of the DSL market. This chapter also gives a brief introduction to VoIP and followed by an introduction to just what CPE equipment is and how it can be monitored and managed by a remote management system.

Chapter 3 explains in detail how TR-069 was designed. Relevant services and features are discussed, including the technology to be used in the implementation.

Chapter 4 describes realization of the solution designed in chapter 3. Once the solution was implemented, it was compared with an SNMP management system in a case study. The comparisons consider the difference in resource consumption, cost of implementation, security and feasibility, and scalability relative to the current solution.

Chapter 5 describes the conclusion from this thesis and finally chapter 6 describes the suggestions for future work.

2 Background study

2.1 Internet Access

The internet is a worldwide accessible network consisting of an interconnected computer networks that transmit data packets using the internet protocol. Access to the internet refers to the means by which a user connects to the internet. In the old days, the main purposes for access to internet were to access news website, read email, use a text chat service, remote access to main frame computers, and simple file transfer. At that time, access to internet was mainly done through phone line using an analog dial-up modem with data rates of up to 56kbps. Analog modems satisfied the mass market demand by offering affordable access from home through a telephone line. While the cost was low so were the maximum data rates. Browsing graphical web-pages using an analog modem quickly lead user to frustration. Additionally, accessing the internet through an analog modem was not sufficiently reliable for business use because the connection frequently went down, and then the user had to redial to reconnect. The introduction of a digital subscriber line via ISDN seemed to be quite promising in the beginning. ISDN offered data rates up to 144 kbps. However, this requires the installation of an ISDN line and modem which cost time and money. Additionally, ISDN line installation required telephone technicians who were knowledgeable about data communication - and these technicians were in short supply. In many cases the traditional analogue telephony companies were not prepared for the number of interested subscribers, so they had under dimensioned their ability to accept orders for ISDN and to install it. The result was that most internet services providers were not keen on setting up ISDN connections. Another option for business users was internet access through leased lines. While this seemed to be the best option for high speed internet access, especially for those who communicated on a daily basis, the cost of more than $2000/month was expensive for small-medium size business and for home users.

Unfortunately, the above remained the state of the art for internet access until a major breakthrough in the late 1990s that allowed internet access over a DSL connection. A DSL connection offers higher internet access speeds at an affordable cost, thus satisfying both home and business users. DSL is discussed further in section 2.2 below. The increase in data rates for of access to the internet enabled new services, as no longer was a user limited to just receiving simple text and simple graphic or media; today, a typical broadband internet connection is capable of supporting real-time media such as voice and/or video. This increase in data rates at a decreasing cost and the development of new services attracted very large numbers of end users. Home users today communicate daily via at least voice and in many cases multimedia conferencing. This growth has turned DSL service into a commodity and also means that the average user has no knowledge of how the communication services, networks, or equipment work. At the same time most of the major network operators have decreased their technical knowledge (due to telecommunications deregulation, changes in management, outsourcing, etc.). So neither the end users nor the network operators are prepared to diagnose and fix network problems. It is for this reason that CPE equipment has to be remotely manageable and the CPE management systems have to be more capable.

2.2 DSL Overview

Digital Subscriber Line (DSL) technology offers a solution to the need for high speed, affordable, fixed line internet access for both business and home users. DSL enables a digital signal to be carried over the unused frequency spectrum available on the twisted pair cables between a telephony service provider's central office and the customer's premises. DSL offers much higher data rate for data transmission than analogue modems, ISDN, and leased lines.

2.2.1 DSL Technology

DSL technology originated in the late 1980s, when Joseph Lechleider, an engineer from Bellcore Lab demonstrated the feasibility of sending a broadband signal over a twisted pair through a mathematical analysis. He continued by adapting DSL to carry digital signals over unused frequency spectrum of a twisted pair telephone cable running from the telephone company's central office to the customer's premises. The original plan was to send video signals to implement a Video on Demand service; which needed considerably more downstream than upstream speed. This asymmetry later inspired the idea of asymmetric DSL (ADSL) which allows the simultaneous transmission and reception of data at speeds of up to 10 Mbps downstream and 650 Kbps upstream over 18000 feet (about 5486.4 meters) of twisted pair copper wire without additional signal conditioning.

Initially, telephone cables were designed to limit the transmission to a 300-3400 Hz analog voice channel (with some lower frequencies being used for power and ring signaling). The maximum information rate in a 3400Hz frequency spectrum (as limited by practical power levels) is 56kbps. In order to achieve higher transmission rates, a broader frequency spectrum must be used and therefore, DSL uses a much wider range of frequencies than a voice channel. DSL requires information to be transmitted over a wide range of frequencies from one end of a copper wire pair to the device at the end of the copper loop.

2.2.2 Type of DSL Technology

ADSL (Asymmetric Digital Line Subscriber)
This is the most commonly used form of DSL. Here, asymmetric means that the upstream and downstream data rates are different. The difference can be configured to be up to 6 Mbps. For general internet access, higher downstream than upstream data rates is frequently more appropriate.

HDSL (High Data Rate Digital Line Subscriber)
This type of symmetric DSL enables data transmission from 1.5 Mbps up to 2.3 Mbps. Standard telephony service is not supported over the same line. HDSL considered an economic replacement for E1 or T1 leased lines (of 2Mbps and 1.544Mbps respectively). It uses up to three twisted copper pairs.

SDSL (Symmetric Digital Line Subscriber)
This DSL technology enables data transmission speeds from 128 kbps up to 2.32 Mbps over a single pair of copper wires with a maximum range of 3 km. The upstream and downstream speeds are the same.

RADSL (Rate Adaptive Digital Line Subscriber)
RADSL is a variant of ADSL technology which adapts the upstream speed depending on the distance and line quality of the circuit between the telephone exchange and the modem in an attempt to maintain a specific data rates.

G.Lite (UADSL/DSL-lite)
G.Lite is user-installable and provides speeds of 1.544Mbps downstream and 512Kbps upstream. It is backed by many hardware vendors and the Universal ADSL Working Group.

IDSL (Integrated Service Digital Network Digital Line Subscriber)
IDSL supports symmetric data rates up to 144 kbps using existing phone lines. It differs from ISDN in that it is an always-available service, but is capable of using the same terminal adapter or modem used for ISDN.

VDSL (Very High bit of Digital Line Subscriber)
VDSL is an ADSL technology which theoretically enables up to 52 Mbps downstream and 16 Mbps upstream data transmission over a single twisted pair over a distance of 1,000 feet (~300m). However, these rates are strongly influenced by the properties of the local loop.

2.2.3 DSL Forum

The DSL Forum is a consortium established in 1994 by about 200 companies involved in telecommunication, equipment, computing, networking, and service providers. The Forum continues to drive the development of DSL in order to meet the broadband needs of the mass market. The forum is currently focused on establishing advanced architecture standards, and maximizing the effectiveness in deployment, reach, and application support.

Until 1994, the perception of DSL was negative at best and was declared by many as being dead on arrival. Few industries believed in DSL technology, therefore they would not invest in it. It was due to visionary people from Motorola and Italtel, that an international DSL industry organization was realized. The ADSL Forum was created to define:

- Standards for DSL operations and management
- Standards for DSL network protocols
- Standards for DSL network architecture
- A process for testing equipment interoperability
- A voice for the DSL industry to promote the advance of DSL
- Education about DSL

In 1999, the organization's name changed to DSL Forum to indicate its interest in all variants of DSL technology.

The Forum consists of:

- A board, whose mission is to develop technical specifications and marketing materials that enable the promotion, delivery, and support of profitable broadband products and services.
- A Technical Working Group
  This group is in-charge of developing technical specifications to guide the development of new services. There are currently 3 active technical working groups namely (1).DSLHome-Technical, (2).Architecture and Transport, and (3).Operation & Network Management.
- A Marketing Working Group
  This group is in-charge of marketing related work, public relations, and strategic communications for the DSL industry.

2.3. Voice over Internet Protocol

Voice over Internet Protocol (VoIP) is a technology that allows voice to be transmitted over the internet using the internet protocol. It became popular due to the cost advantages for long distance comparing to the communication using the traditional telephony network. A typical VoIP network is shown in the picture below.

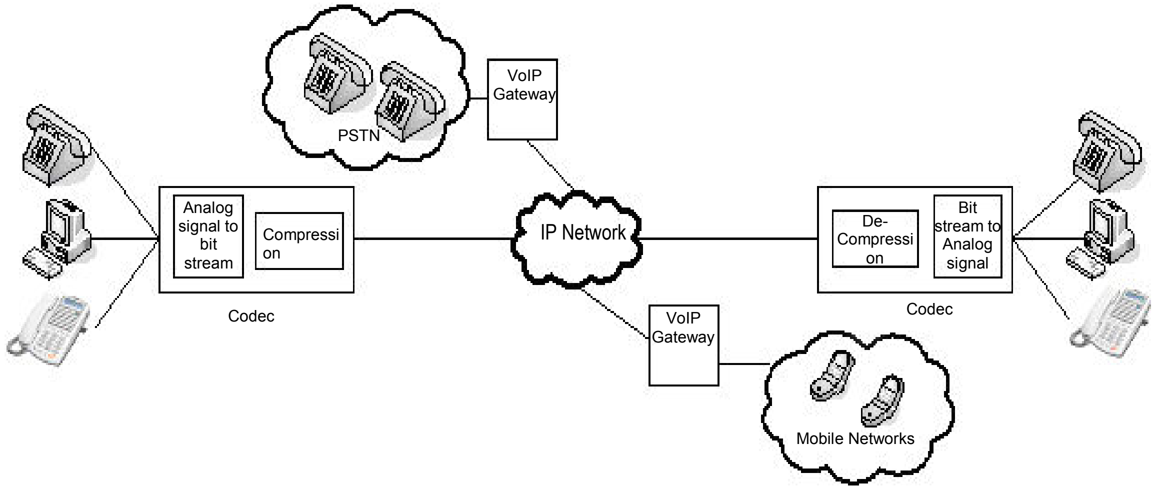


Figure 2-1: Typical VoIP Networks

There are 3 types of end-devices commonly used in VoIP:

- An IP phone
  From the end user's point of view, an IP Phone is a device to provide telephony functionality (such as making and receiving calls), but the device uses IP network connectivity rather than a traditional telephony network. From the network's point of view, an IP phone is simply a network node in an IP network, just like any other host that is connected to this IP network. The IP phone has an IP address and behaves similarly to any other IP host. The advantage of using an IP Phone is that the voice traffic is carried over the IP network, thus the user only needs a single network connection, reducing installation and maintenance expenses. The disadvantage of using an IP phone is that the per-unit price of an IP phone is relatively expensive and sometimes the phone's "advanced-looking" menu that look "too-complicated" to the end user. However, most IP phones today are simple in appearance and nearly indistinguishable from cellular and cordless phones handsets.
- Using a traditional phone connected to a VoIP adapter
  The user can connect their existing analog telephone to a VoIP adapter in order to place and receive calls through the IP network. The adapter converts the analog signal from the phone into a digital signal and transfers it via the IP network. Unlike an IP phone, the phone does not have an IP address but rather the VoIP adapter has an IP address. The VoIP adapter has two connections, one for connection to the analog phone and one for connection to the IP network. Compared to the IP Phone, the use of a VoIP adapter seems less complicated to end users, especially for the elderly. This is simply because they are used to using ordinary analog phone. Another advantage of using such an adapter is that it can be used with the user's existing cordless phone. A VoIP adapter is an example of Customer Premises Equipment (CPE). Section 2.4 describes CPEs, and the specific use we will make of the term in this thesis. Another name for such a VoIP adapter is an "analog telephony adapter".
- Using a computer equipped with a speaker and microphone
  For users whose computers are connected to an IP Network, the addition of speaker and microphone is sufficient hardware for VoIP service. Today a wide variety of handsets and headsets are available which either plug into an audio interface on the user's computer or plug into a USB interface on such a computer. Most of the chat programs currently available offer a "live-chat" service utilizing VoIP.

2.3.1. Signaling Protocol

A signaling protocol is a protocol used for setting up connections between telephones or VoIP terminals. In order to establish, maintain, and terminate these connections there is a variety of signaling which may take place, including: device registration, session initiation, alteration and termination, and capability negotiation. Some of the commonly used signaling protocols for VoIP are:

- Session Initiation Protocol (SIP)
  SIP is a signaling protocol defined by IETF in the mid-1990. It was originally intended to create a mechanism for inviting people to join large-scale multipoint conferences on the Internet Multicast Backbone (Mbone). IP telephony did not really exist at the time. The first draft was known as "draft-ietf-mmusic-sip-00" where the "mmusic" stands for "Multiparty Multimedia Session Control". It included only one request type, which was a call setup request. This draft evolved and in March 1999, the RFC2543 [31] was published establishing first SIP standard. Three years later in 2002, RFC3261 [32] was published to replace RFC2543 [31]. SIP can use a variety of transport protocols, including but not limited to TCP, UDP, TLS, and SCTP. A SIP client typically communicates with SIP servers and other SIP endpoints using port 5060.
- H-323
  H-323 is a recommendation by ITU-T that defines the protocols to provide audio-visual communication sessions on any packet network. The first version of H-323 was published by the ITU in November 1996 with an emphasis of enabling videoconferencing capabilities over a Local Area Network (LAN), but was quickly adopted by the telecommunication industry as a means of transmitting voice communication over IP networks, including WANs and the Internet. One advantage of H-323 was the relatively early availability of a set of standards, not only defining the basic call model, but also the supplementary services needed to address business communication expectations. H-323 standards have been improved over the years with revisions and enhancements necessary to better-enable both voice and video functionality over packet switched networks, with each version being backward-compatible with the previous version. The latest H-323 standard today is H-323v6 that was published in the year 2006.
- Media Gateway Control Protocol (MGCP)
  MGCP is protocol used in VoIP systems. This internal protocol was primarily developed to address the demands of carrier-based IP telephone networks. MGCP is used for controlling telephony gateways by call control elements called media gateway controllers or call agents. A telephony gateway is a network element that provides a conversion between the audio signals carried over telephone circuits and data packets carried over the Internet or other packet networks.

2.3.2. Media stream

When a call session has been established using one of the signaling protocols, the analog voice signal is converted into data packets and transmitted over the internet to the destination as a media stream. This media stream is typically carried in Real-time Transport Protocol (RTP) packets. RTP is a standard packet format for delivering audio, video, and timed text over the Internet. It was developed by the Audio-Video Transport Working Group of the IETF and first published in year 1996 as RFC 1889 [47] and obsoleted by RFC 3550 [48] in year 2003.

One important remark about RTP is although it stands for Real-time Transport Protocol, RTP does *not* ensure real-time delivery. Real-time in this case simply means RTP defines a standard packet format to carry real-time data. In addition to the real-time content, each RTP packet also includes a timestamp and sequence number, in order to enable the receiver to playout the packets in the proper order and with the proper temporal relationship. A companion protocol, the Real-time control transport protocol (RTCP) provide information can be used along with control mechanisms for synchronizing different streams, and understanding whether packets are being delayed or lost.

2.3.3. CODEC

CODEC stands for Encoder-Decoder. A CODEC is used to convert analog signals into a bit stream to be transmitted over some communication channels, and convert the bit stream back to an analog signal on the receiving side. A CODEC must be placed at both ends of the communication path. At the near end, the CODEC transforms the analog signal to a bit stream and at the far end, the CODEC transforms the bit stream back to analog signals. In the world of VoIP, the CODEC is also known as a Vocoder, which stands for "Voice Encoder".

When converting an analog signal to bit stream, the CODEC implements an algorithm which is sometimes includes compression in order to reduce the required communications bandwidth. Some CODECs even support "Voice Activation Detection" (VAD) to avoid coding non-voice audio signals, primarily in conjunction with silence suppression - which avoids sending packets when there is no voice content to be sent. There are many CODECs standards today, because different CODECs were designed to achieve particular purposes, such as to allow very low data rate channels to be able to carry voice, to be robust to particular noise sources, etc.

Table 2-1: List of some well-known CODECs [29]

Codec	Algorithm	Bit Rate (Kbps)	Comments
ITU G.711	PCM (Pulse Code Modulation)	64	G.711 with mu-law used in North America and Japan, while G.711 with A-law used in the rest of the world.
ITU G.722	SBADPCM (Sub-Band Adaptive Differential PCM)	48, 56 and 64	
ITU G.723	Multi-rate Coder	5.3 and 6.4	
ITU G.726	ADPCM (Adaptive Differential PCM)	16, 24, 32, and 40	
ITU G.727	Variable-Rate ADPCM	16-40	
ITU G.728	LD-CELP (Low-Delay Code Excited Linear Prediction)	16	
ITU G.729	CS-ACELP (Conjugate Structure Algebraic-CELP)	8	
ILBC	Internet Low Bitrate	13.33 and 15.20	
Speex	CELP (Code Excited Linear Prediction)	2.15-44.2	Part of the GNU Project and available under the Xiph.org variant of the BSD license
GSM - Full Rate	RPE-LTP (Regular Pulse Excitation Long-Term Prediction)	13	
GSM - Enhanced Full Rate	ACELP (Algebraic CELP)	12.2	
GSM - Half Rate	CELP-VSELP (CELP - Vector Sum Excited Linear Prediction)	11.4	
DoD FS-1016	CELP	4.8	

2.4. Introduction to CPE

Customer Premises Equipment (CPE) is the terminal and associated equipment and inside wiring located at the customer's premises and connected with a common carrier's communication channel at the demarcation point. The word "customer" refers to someone who has subscribed for a service from a telecommunication service provider.

A CPE can be either a gateway or end device. A gateway, such as switch and/or router, connects the end device to the outbound network connection, namely the Wide Area Network (WAN). An end-device is a device providing services to the customer, such as displaying video, capturing video, inputting & outputting, voice and other audio data, etc. A gateway can either be a bridge, a router, or combination of the two.

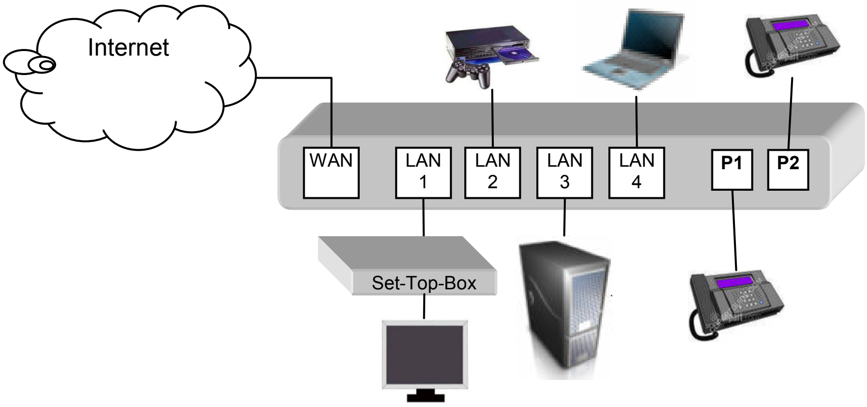


Figure 2-2: A CPE with WAN and LAN connection

When a CPE configured as a bridge, it operates on Layer 2 (of the ISO OSI network layers), and uses Media Access Control (MAC) address learning to determine the devices connected to the WAN and LAN interfaces. All devices on the home network belong to the same subnet as the WAN. LAN side traffic will be sent to the WAN interface when the destination MAC address is not recognized as being local to the LAN. Broadcast traffic on the LAN would also be sent to the WAN.

When a CPE is configured as a router, it separates the home network from the WAN and either the router or a separate Dynamic Host Control Protocol (DHCP) server will typically assign each of local devices a private IP address, while the gateway itself gets a public or private IP depending on the subscriber's subscription and/or network operator. Outbound traffic from the home devices is forwarded by the gateway to the WAN. When private addresses have been used in the local (home) network, the gateway will act as a Network Address Translator (NAT). NAT will be discussed in section 2.5.4.1

When used with a home network broadband internet connection, CPE is often referred to as a "home gateway". As such the device is likely to carry voice, video, and data traffic via a home network broadband internet connection. The CPE is likely to be connected to a VoIP adapter, IP phones, set-top boxes, home computer, etc. This is because common CPE usage in a home environment includes:

- Voice Communication
  The main communication application up to the present is still voice communication. A VoIP adapter or IP phone may be used for those wishing voice service.
- Remote or Home Office
  Utilizing a remote or home office has become increasingly common today. This enables employees to reduce their commuting (with obvious environmental, time, and costs advantages). For users at home or in remote locations this enables them to work, using their computer via a broadband wired or wireless network connection. In addition, many firms are using IP based multimedia conferencing

systems to reduce the amount of travel which their employees do (also for environmental, time, and costs savings reasons).

- Entertainment and information
  Access to entertainment content is one of the largest markets and it is increasingly being offered as part of a broadband package to subscribers (often as part of a so-called "Triple play" package with internet access, IPTV, and IP telephony). This has also enabled increased competition as not only are the traditional telephony companies offering this service via their twisted pair copper wiring plant in competition with cable TV operators, but increasingly in competition with broadband wireless operators.

2.4.1. CPE Requirement

Since the CPE may need to carry voice, video, and data traffic in a high speed broadband network, it must be designed to be able to send and receive the network traffic and able to translate (encode and decode) the data into the required formats and connected via the required interfaces. CPE vendors need to address a variety of requirements during the development phase, such as:

- The device must have standardized interfaces so that other equipment can be easily connected. There are many potential connection interfaces to/from the equipment including coaxial cable, Ethernet, USB, RS-323 serial, IEEE 1394 (also known as Firewire), POTS line, S-Video, SCSI, Bluetooth, etc. For a given product, the vendor must decide which of these interfaces to implement and in many cases how many of these interfaces to implement. For example, a simple configuration of a device designed as a VoIP adapter might utilize one broadband interface and two interfaces for analog telephones.
- Devices must support easy configuration and remote management by the service provider, with sufficient headroom to accommodate new standards and upgrades
- Service-compliant protocols must operate at wire-speed on the attached LAN and the attached WAN, i.e., the CPE should not introduce a performance bottleneck.
- The CPE must interoperate with the access equipment at Layers 1 and 2, and with the edge equipment at Layer 3 and above in the service provider's network. This interoperability is important because the equipment at the customer's premise and the equipment at the operator's side of the copper loop (in the case of DSL) might be from different vendors.
- The CPE must be able to support customer-based and carrier-based virtual private networks (VPN) implementations. For example, the CPE should support IPsec pass-through.
- The equipment must enable low-cost implementation. This is very important as these devices are increasingly being commodity devices. Ideally, the volume of production will be very high, enabling use of highly integrated VLSI components, thus allowing very low parts count in the devices.

A more complete list of CPE requirements can be found in [15].

2.4.2. CPE Business

The worldwide broadband CPE market, which includes DSL and cable modems, (home) gateways, and voice terminal adapters, totaled $4.6 billion in 2007, up 10% from 2006 [28]. Broadband CPE business is expected to continue growing in the years to come and both CPE vendors and service provider's have to do their best to survive in this highly competitive business environment.

The main challenge for the CPE vendor is to develop a CPE to support higher data rates, lower delay, and to provide reliable connectivity. For a service provider, the challenges are to provide more services and stable connectivity to the subscriber (end user) and yet maintain the subscriber's cost at an the affordable level while meeting the profit goals of the service provider. Typically a CPE is provided (and as mentioned earlier often owned) by the service provider. However, increasingly the

CPE is "given” to the customer as part of the subscription. There are several models of subscription options to the customer at the present time. They include:

- The “do-it-yourself" approach where the customer purchases the CPE off-the-shelf. The CPE is deployed, managed, and administered by the customer. In this approach, the service provider is limited to providing only IP connectivity. In this model, the CPE itself is a direct commodity where only the CPE vendor earns a profit. The service provider/operator only earns money based on the end customer’s subscription.
- Alternatively, a service provider might deploy and operate an IP PBX at the customer’s site, with the service provider responsible for ongoing remote management of the platform and service delivery. The service provider provides services such as a premises-based IP Telephony solution including fault monitoring, configuration management, performance management, moved, adds, and changes, etc. Often the customer has the option to buy, lease, or rent the on-site equipment as part of the service agreement.
- Another approach using IP Telephony is a hosted IP PBX Solution. In this case the IP PBX is housed at the service provider’s facility and managed by the service provider. The customer simply rents the service on a per-user basis. The CPE may be bundled as part of the hosted IP PBX service fee, leased, or bought directly by the end customer.
- IP Telephony can also be offered as hosted service from a shared server hosted by the service provider.

When a user gets his or her CPE, he or she needs to subscribe to one or more services provided by a service provider (often thought of as the subscriber's operator). Users of CPE are called “subscribers”. Classically they were categorized based on the how frequently they are access the internet and the services they have subscribed to. However, today a simple way to classify these users is to simply group them as business users (small office home office - SOHO), or residential user [12]. A more detailed classification based on whether a single PC of a group of PCs are connected and the type(s) of applications the end users are using, divides CPE users into the following scenarios:

- Small Office and Home Office (SOHO) user
  This scenario is most likely an office or branch offices with multiple computers organized into one or more workgroups within the same domain. Connections are “always” up towards a main server in a server room located in the main office at the company. Additionally, these sites are “always” connected to the internet.
- Work at Home user
  This is the case when employee, instead of going to the office, works from home and connect their home computer to their office network environment using the some form of internet access. The main concern for these users is to have (if possible) exactly the same office environment when working from home as they would if they were physically connected to the LAN at the company's premises.
- Internet access only user
  Internet access only users are residential users who have a single PC and only accessing the internet reading news, checking email, and sometimes simple online text chat. Such user may have subscriptions to several internet service providers (ISPs), but are very unlikely to connect to different ISPs simultaneously.
- Multi function residential user
  These are advanced residential internet users. They are using high-speed internet access to run sophisticated and high-tech applications such as online games, sharing online content, video and audio streaming, online transactions, etc. These users typically have more than one PC at home and may have access via multiple ISPs and may even be connecting to others via more than one ISP simultaneously.

2.4.3. Digital Residential Gateway

A Digital Residential Gateway (DRG) is a type of CPE developed by 42networks AB. It was designed mainly to provide telephony services and to support low rate data control of home appliances such as alarm systems. The DRG was thought to give the service provider a means of getting into the home networking market. However, it has now evolved into several product lines target towards different customers to support a variety of different purposes. The products range from low cost and low performance up to high end and high performance solutions.

- *DRG11/22*
  DRG11/22 is a cost effective solution to provide IP telephony and broadband services. It support either one POTS port (DRG11) or two POTS ports (DRG22) and one LAN port. DRG11/22 can be operated either as bridge or router (this is configurable via the firmware).
- *DRG3x*
  The DRG3x series offers more connectivity options and features. It has 2 POTS lines and 4 Ethernet LAN ports for the end user. Interfaces to the broadband access network can be to single or multimode fiber or copper. DRG32/34/36/38 allows service providers to offer a range of broadband services, including IP telephony. Existing analog telephones or fax machines can be connected directly to the DRG to provide and IP telephone service. By connecting a set top box to one of the Ethernet ports, Video on Demand (VoD) can be delivered simultaneously with telephony and fast Internet connectivity. The four LAN ports in the DRG provide an end user with the possibility to connect multiple workstations to their home/small office network. Features such as bandwidth shaping, MAC-address restriction, and IGMP snooping can be used to offer a differentiated service suite to a broad range of end users.
- *DRG4x*
  DRG4x has the exactly same technical features and facilities as DRG3x series. The differences are that DRG4x has a different plastic housing (designed to mount on the wall,) and to “clip-on” an IP-TV media converter (for example to convert signals on fiber to RF signals for a coaxial cable or other media) module.
- *DRG340*
  This is a model in the DRG3x series which flash memory has been increased the capacity up to 1.5 Megabytes from the existing 1 Megabyte. From a user interface and features point of view, the DRG340 is similar to DRG3x/4x unit.
- *DRG5x*
  DRG5x series are the current high end products. Each has 2 POTS ports and up to 8 LAN ports. The WAN port supports 100 Mbps or a gigabit connectivity, while each of the LANs only supports up to 100 Mbps. Future product in the 5x series will also provide wireless connectivity (for example Wi-Fi).
  This model is aimed at service providers that provide triple-play solution (high-speed internet, IP Telephone, and IP-TV/Video on Demand). There are two product options: one with telephony support and one without telephony support.
- *DRG Ease*
  DRG Ease is a state-of-the art product, which breaks down a single DRG unit into three detachable modules: a switch module, voice module, and wireless module. The switch module is the main module, while the two other modules are optional for the end-user.

Common features in each of the above CPE platforms are

- Call Signaling Protocol
  All DRG series support H.323, SIP, or MGCP call signaling protocols. The same DRG unit can be used when users decide to change to another call signaling protocol. All the user needs to do is to load the relevant software into the unit.
- Boot time loading of new software or a configuration file
  DRG supports loading software or configuration file (see appendix A.3) through Trivial File Transfer Protocol (TFTP), HTTP, and HTTPS.

- Management and Provisioning system
  All DRG series support the management from the web GUI interface or an SNMP manager. Provisioning (i.e., configuration of services) can be done via SNMP, using a vendor encapsulated option (see appendix A.1) in DHCP or the REDS polling (see appendix A.2).

2.5. CPE Management

The introduction mentioned that most CPE vendors and service providers have underestimated the importance of CPE management. The reasons why managing CPE is often avoided are:

- There is no standardized method of managing CPEs
  Unfortunately, all CPE vendors use proprietary methods for managing their own CPEs and/or other CPEs of their business allies. This makes it difficult for a service provider when expanding their deployment to select a diverse set of products (traditionally this sort of vendor lock-in was done purposely by vendors just so their customers would have to buy all of their equipment from them or face a very large cost of managing another set of incompatible equipment). Although most vendors today use SNMP [10] (Simple Network Management Protocol) to manage CPEs, each does so using their own (often) proprietary vendor options fields and values. There are both advantages and disadvantages in using SNMP to manage CPEs – these will be described in section 2.5.1.
- CPE management adds significant costs and risks
  A high quality - high functionality CPE management system is not "free". Although CPE vendors often include CPE management as part of a packaged business deal involving a large purchase of CPEs, for example with 10000 CPE. Most vendors charge a fixed amount and license their CPE management on a per CPE basis in-additions. There are risks due to incompatibilities between the CPE management system and the CPE. For example specific features from a CPE might not be manageable by the CPE management systems.

However, having and utilizing a CPE management system is no longer avoidable since efficient management of the CPEs plays an important role in ensuring the quality of service which the service provider expects to or is obligated (via a service level agreement) to provide. Some vendors have developed their own CPE management systems based on the available network management standards available today. Before discussing these in more detail in the next section, there will be a brief overview of these existing network management standards.

2.5.1 SNMP based CPE Management

2.5.1.1 SNMP Overview

SNMP [10] [11] has primarily been used to manage network elements (but can also be used to manage individual devices attached as hosts on the network). It utilizes a manager/agent model management; consisting of a manager, an agent, a database of management information, managed objects, and the network protocol. The manager provides the interface between the human network manager and the network management system (NMS). The agent provides the interface between the manager and the physical device(s) being managed.

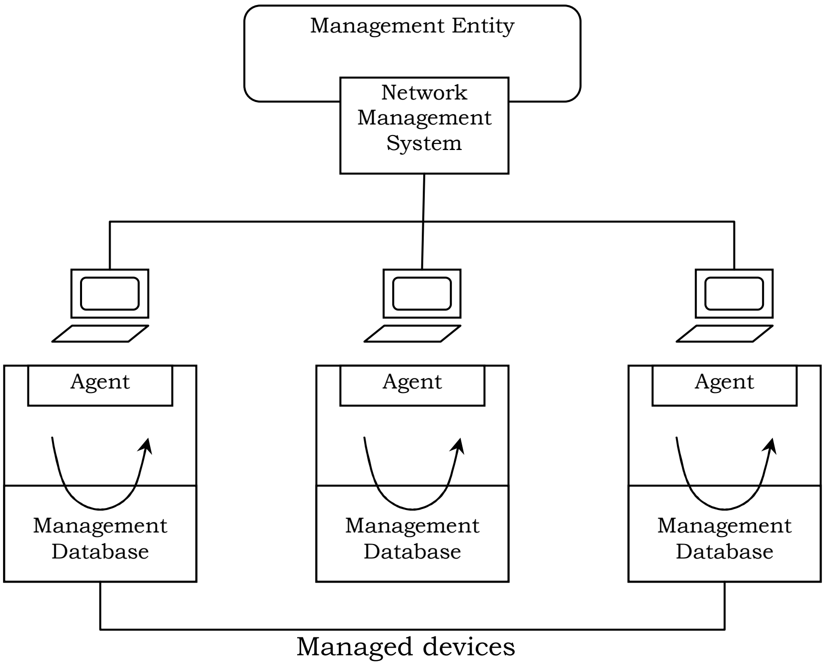


Figure 2-3: An SNMP managed network consists of Managed Device, Agents and Network Management System [38]

SNMP collects and organizes data in the form of Managed Information Bases (MIB). MIBs are a collection of definitions, which define the properties of the managed object within the device to be managed. Every managed device keeps a database of values for each of the definitions written in the MIB.

An MIB is structured as a tree which is globally defined [26]. The top level object in the tree is a root node. A root node has three children. The first one administered by the International Organization for Standardization, with label iso(1); second is administrated by the International Telegraph and Telephone Consultative Committee, with label ccitt(0); and third is jointly administered by the ISO and the CCITT, joint-iso-ccitt(2). Each of these children also has its own children that are managed by other organization. An example of a SNMP MIB tree is shown in Figure 2-4. The numbers in the parenthesis are the numeric value which indicates this branch in the tree.

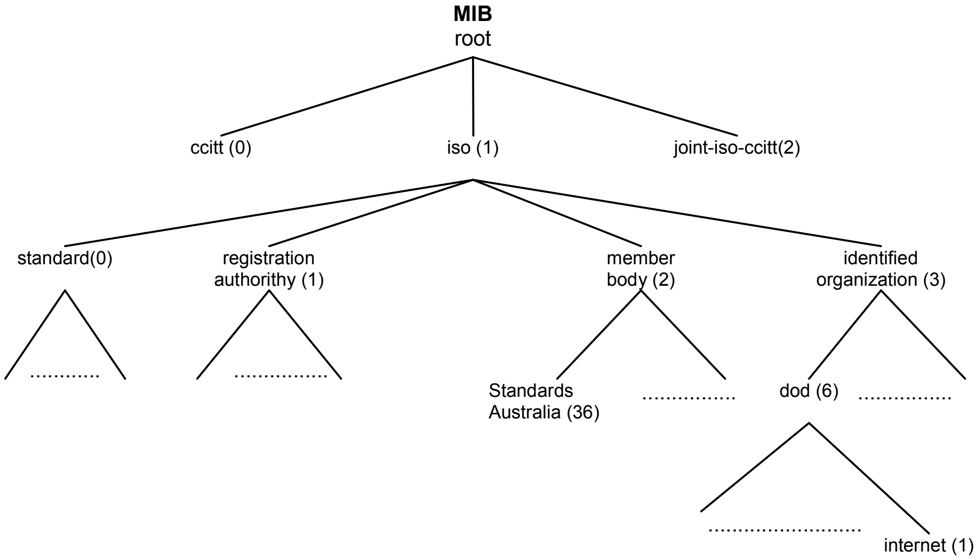


Figure 2-4: SNMP MIB Tree [33]

Every private organization that implements SNMP into its device(s) may get a Private Enterprise Number (PEN) [22] from IANA. Using this Private Enterprise number the organization can create its own MIB, which will be located under this PEN. A hierarchical view of this is shown in Figure 2-5. Note that a PEN is only needed if the organization is going to define its own MIB.

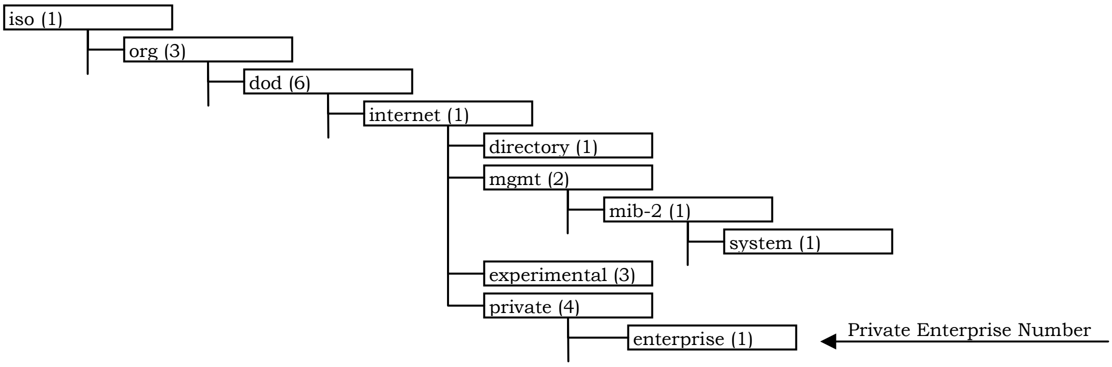


Figure 2-5: Private Enterprise Hierarchy view [10]

SNMP uses five basic messages (GET, GETNEXT, GETRESPONSE, SET, and TRAP) to communicate between the manager and the agent. The GET and GET NEXT message allow the manager to request information for a specific variable of current Object Identifier (OID[1]) or the OID following next to the current OID. The agent, when receiving a GET or GETNEXT message, will respond with a GETRESPONSE message to the manager containing either the information requested or an error indicating why the request cannot be processed. A SET message is used by the manager to modify the value of a specific variable, for example in the case of an alarm setting a variable might operate a relay to power another device on or off. The agent will respond with a GETRESPONSE message indicating the change has been made or an error indicating why the change cannot be made. The TRAP message

[1] An object identifier (OID) is a series of number used to identity each managed object uniquely in the MIB Tree (see Figure 2-4). For example, the managed object of DRGs product name will be assigned .1.3.6.1.4.1.18327.67.2.2.1.0 as its OID.

allows the agent to spontaneously inform the manager upon the occurrence of an important event. Of the five basic SNMP messages, TRAP and GETRESPONSE are the messages that an agent can send. An SNMP agent can initiate the sending of a TRAP while a GETRESPONSE will only be sent when upon receiving GET or GETNEXT request. The other three SNMP messages, GET, GETNEXT and SET are sent only by the manager.

2.5.1.2 Problem of SNMP

Although used widely in network management, there are some known issues with SNMP that make it less than idea for managing CPEs. These issues include:

- SNMP implementation varies across platforms.
  Due to the use of private MIBs by each CPE vendors, in order for a manager to manage a newly introduced CPE, it needs to know about the MIB of that new CPE. This is usually done by loading a MIB file into the manager (assuming that the CPE vendor will provides a MIB file). However, if for some reasons the MIB has been updated (e.g. new services, enhancement of current features, etc.), then the human network manager needs to load an updated MIB file to the NMS. Additionally, some network managed systems only support SNMP version 1 and while new CPE MIBs use SNMP version 2 or even 3. Another problem occurs when a NMS is managing CPEs from 3 different vendors, then it needs to load 3 different sets of MIB files, in which most of the OIDs in the MIB files are for similar services. In addition, often there is no clear means of associating the elements of the MIB with what and how the NMS should utilize this element of the MIB. This is especially hard when the vendor does not provide a proprietary MIB.
- SNMP is mainly thought of as a LAN based management protocol.
  Because SNMP manages network devices within the corporate network, most network administrators block SNMP message from external networks for security reason. This is particularly true for SNMP version 1 as there was very little security (as plaintext passwords were used).
- Problems with Network Address Translation (NAT)
  SNMP MIBs can contain one or more IP addresses. Because of the large variety of messages, formats, and variables possible with SNMP, the NAT cannot easily examine the contents of an SNMP message and translate the IP addresses. Therefore, a NAT can not translate these IP addresses *within* SNMP messages.
- Weak security features in SNMP v1 and v2
  SNMP version 1 and 2 perform authentication using only a plain-text password. This is vulnerable to attacks - given access to the network these passwords can be sniffed. However, this weakness has been improved in SNMP version 3 [11] which provide message integrity, authentication, and encryption. However, SNMPv3 is not very popular hence it has only limited use.
- The two ways of communication from CPE to the SNMP management server are not really two-way communication. Since the CPE is an agent, it is only capable of sending a TRAP as an alarm to the manager. Following this TRAP, the SNMP manager must send a GET or GETNEXT to find out what has actually occurred, and then perform further actions. In any case, the SNMP manager is still performing the action, while the CPE can only send a TRAP. This pattern of communication causes high dependency on the NMS from the CPE and it may consume a lot of resources in the NMS when it receives TRAPs from many CPEs.
- SNMP is a tool for network management, while CPE management not only needs to manage the unit, but also to manage services. The target that SNMP is managing is either a single host or group of hosts in a network (depending on how the network is divided), while service management for CPEs is often for hosts in a separate network that are identified via a subscription. One can still manage services using SNMP, but there will be added complexity due to the need in some case to relay SNMP messages through different networks that are going to use same services (section 4.4.1.1 gives an example of service management using SNMP). It will be even more difficult if one needs to use some other

protocol between a NMS in one network and a CPE in another network - where the network which the CPE is in is not run by the same network operator as the network where the NMS is.

2.5.2 CPE WAN Management Protocol (CWMP)

The DSL Forum through its DSL Home work group formulated the standard for CPE management which is called the CPE WAN Management Protocol (CWMP). This protocol is defined in technical report number 69 (TR-069) [1]. This in the following sections of the report we will refer to this specific version of the CWMP protocol as TR-069.

CPE management in TR-069 is performed through the communication between a CPE and the Auto-Configuration Server (ACS). The ACS is an element in the network whose task is to manage subscribed CPEs in a flexible and systematic way. An ACS in general is a server machine running a manager application. The TR-069 defined the function of ACS:

- *Software and Firmware image management*
  The ACS is in charge in managing software and firmware downloading from a repository to the CPEs. The protocol defines a standard mechanism of identifying the firmware versions, initiating firmware download, and recording the result of a firmware download. It also defines a standard to ensure the integrity of the firmware to be downloaded through the use of a digital signature.
- *Auto Configuration and dynamic service provisioning*
  After the correct firmware is installed in the CPE, the protocol allows the ACS to perform auto-configuration and to provision the CPE with services based on a variety of criteria. The provisioning of a CPE might be done when a CPE initiates a connection to the broadband access network or when the CPE initiating the re-provisioning at any subsequent time. The identification mechanism allows provisioning based on specific requirements of a CPE; group criteria such as vendor, model, software version, or other criteria.
- *Status and performance monitoring*
  Status and performance monitoring is done through a set of defined parameters. These parameters can be standard ones which can be retrieved by any ACS and non-standard parameters which the ACS will require special syntax to retrieve. The standard parameters are defined in TR-098 [17] and the non-standard parameters are defined by each vendor as described in TR-106 [19]. This may sounds similar to the proprietary MIB which is used in SNMP. However, in SNMP, each vendor produces its own MIB. This means, that CPEs from multiple vendors use different OID for the same purpose, for example an SNMP NMS may need to use 1.3.6.1.4.1.3955.2 to identify a CPE from vendor A and use 1.3.6.1.4.1.34523.3.5.2 to identify CPE from vendor B (3955 is the enterprise identification for vendor A and 34523 is the enterprise identification for vendor B). In case of CWMP, the ACS simply observes the “DeviceSummary” to identify the device. There are also parameters that cause the CPE to periodically notify the ACS in order to report its status.
- *Device diagnostics*
  For troubleshooting purposes, the protocol defines a mechanism for the CPE to provide information such as CPE connection status, service issues, etc. This information will be stored as parameters. A CPE can send diagnostic information to an ACS, or the ACS can read this information from a CPE by accessing the relevant parameters. This is different from SNMP in which a CPE only sends a TRAP message to an SNMP manager, and then the manager has to fetch information from CPE.
- *Identity Management for Web Applications*
  There is an optional mechanism for a CPE to allow customization of web-based application content to be accessed via the CPE’s local network.

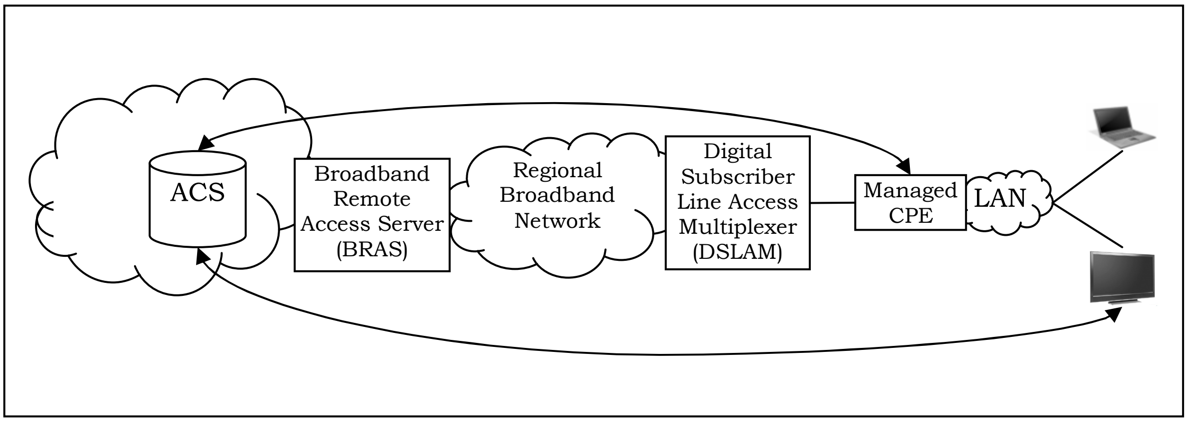

ACS
Broadband Remote Access Server (BRAS)
Regional Broadband Network
Digital Subscriber Line Access Multiplexer (DSLAM)
Managed CPE
LAN

Figure 2-6: CWMP in the network, adapted from Figure-1 in [1]

2.5.2.1 Protocol Components

CWMP is made from several components that are unique to the protocol, while also making use of several standard protocols. The protocol stack is shown in Figure 2-7

CPE / ACS Management Application

RPC

SOAP 1.1

HTTP

SSL/TLS

TCP/IP

Figure 2-7: CWMP Protocol Stack, adapted from Figure-2 in [1]

- *CPE / ACS Management Application*
  This is the DSL Forum defined component for used in a CPE or ACS which runs the CWMP application. This component is an application which invokes each of the CWMP processes. The applications themselves are not considered part of CWMP; hence they are not standardized by CWMP. These applications can be services running on the server or GUI applications.

- *Remote Procedure Call (RPC)*
  CWMP uses RPC [5] for communication between the ACS and the CPE (in fact, RPC can be used in both directions). RPC enables procedures to be called and executed either locally in the same machine or remotely in another machine. With RPC, it is possible to have a procedure realized in only one machine and have the other machines call and execute this procedure remotely with the results sent back to the caller. RPC hides the details of the interface with the network. The transport independence of RPC isolates the application from the physical and logical elements of the data communication mechanisms and allows the application to use a variety of transport protocols. In the case of CWMP, if a CPE initiates a request, it passes both the identification of the procedure to be called and the parameters to this procedure and sends this as a

request to ACS; the ACS unpacks the parameters and passes them as arguments to the identified procedure and packs the results into a response; the response is sent back to the CPE where the return value(s) are unpacked and returned - just as if the procedure had been locally invoked. For more details see [34].

- *Simple Object Access Protocol (SOAP) 1.1*
  In CWMP, every message exchanged between a CPE and ACS (i.e. a request and its response) is encoded in XML. SOAP 1.1 [6] is a protocol for exchanging XML based messages. SOAP is a simple protocol that allows one to access an object across the network. This protocol was first introduced by Microsoft in 1998. Initially, there was no schema language for XML. The early specification of SOAP focused mostly on the defining a type system. The original type system consisted of a small number of primitive types, composites which are accessed by name (structs), and composites accessed by position (array). With these representational types in place, the behavioral types were modeled by defining operations or methods in terms of a pair of struct. In the 4th quarter of 1999, W3C's XML Schema language was released, in it SOAP was integrated into XML. XML was chosen as the standard message format because of its widespread use by major corporations and open source development efforts.

  The reason to use SOAP in CWMP is because SOAP is platform independent and also based on HTTP. The use of HTTP makes CWMP more flexible compared to other distributed object technology such as DCOM, which requires communication through a specific communication port (port 135). A corporate firewall will typically block this port and other ports to prevent malicious access to the web-server in the demilitarized zone (DMZ); however, firewalls frequently opens port 80, the default port for HTTP. Communication in SOAP is done using a Request-Response method. In the Request message, SOAP defines an XML Structure to call a method and pass all the required parameters. In the Response message, SOAP defines an XML structure for the return value. SOAP also defines an XML structure for returning an error value if the request cannot be executed successfully.

  The example below shows a SOAP request which could be use by an ACS to retrieve product information from CPE.

```
<soap:Envelope
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
   <soap:Body>
     <getProductInfo xmlns="http://cds.42networks.com/ws">
       <productID>20630175</productID>
     </getProductInfo>
   </soap:Body>
 </soap:Envelope>
```

  The CPE executes the request received from the ACS and sends responses:

```
<soap:Envelope
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
   <soap:Body>
     <getProductResponse xmlns="http://cds.42networks.com/ws">
      <getProductResult>
       <productName>CPE-53</productName>
        <productDate>2006w30</productDate>
        <productSoftware>DMA0071-R2L100</productSoftware>
        <productDownloader>cxc-132-4888-R2A45</productDownloader>
      <getProductResult>
     </getProductResponse>
   </soap:Body>
 </soap:Envelope>
```

- *HTTP*
  HTTP is a protocol for a distributed, collaborative, hypermedia information system. HTTP has been used since 1990 for transferring text, images and files over the internet. The use of HTTP in CWMP is because HTTP offers a good way to communicate between applications (as it is supported by all Internet browsers and servers) and generally HTTP requests will not be blocked by firewalls. CWMP utilizes HTTP 1.1 as defined in RFC 2616 [7] to carry a SOAP request and its response.

  Below is an example HTTP Response from an ACS containing a SOAP request from a CPE:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: xyz
<soap:Envelope
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
      <soap:Body>
              <Request>Inform</Request>
              <Parameter>x.name=value
                        .........
               </Parameter>
      </soap:Body>
</soap:Envelope>
```

- *SSL/TLS*
  CWMP supports a security scheme using SSL/TLS for communication transport between CPE and ACS. This security scheme provides transaction confidentiality, data integrity, and allows certificate-based authentication between the CPE and ACS. SSL stands for Secure Socket Layer [8]. This protocol was originally deployed by Netscape in 1995. The initial public version was SSLv2, because SSLv1 was never deployed. Microsoft later improved SSLv2 and fixed some security problems, releasing it as a new protocol called Private Communication Technology (PCT). PCT was later improved by Netscape being released as SSLv3. Having three security protocols which were similar to each other, but incompatible was bad for the industry. Due to this the IETF introduced a fourth similar protocol called Transport Layer Security (TLS) [9]. TLS modified the cryptographic algorithm for key expansion and authentication, which made it incompatible with SSLv3. Today nearly all web browsers and web servers support TLS.

  SSL/TLS begins by having both sides exchange their public keys and agrees upon the cipher to be used in the following session. When both sides have agreed on the public keys and the cipher, each can perform a public key encryption using the cipher and used the encryption to secure the entire session.

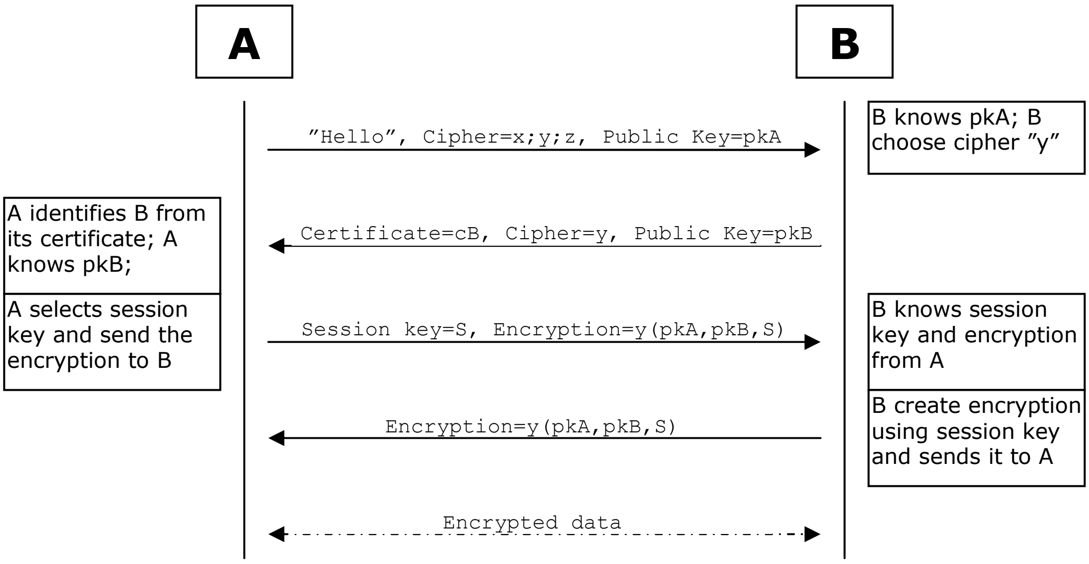

A
B
"Hello", Cipher=x;y;z, Public Key=pkA
B knows pkA; B choose cipher "y"
A identifies B from its certificate; A knows pkB;
Certificate=cB, Cipher=y, Public Key=pkB
A selects session key and send the encryption to B
Session key=S, Encryption=y(pkA,pkB,S)
B knows session key and encryption from A
B create encryption using session key and sends it to A
Encryption=y(pkA,pkB,S)
Encrypted data

Figure 2-8: SSL/TLS Communication

Figure 2-8 illustrates communication when using SSL/TLS. In CWMP, A could be the CPE and B could be the ACS or vice versa. Using SSL/TLS in CWMP, the connection is established between two trusted peers and the subsequent communication is protected by encryption.

However, formally the use of SSL/TLS in CWMP is *recommended*. This is because not all CPEs today support SSL/TLS, and ACS should be able to manage CPS with and without SSL/TLS support. Considering that SSL/TLS provides confidentiality, data integrity, and authentication based on a certificate and shared-secret, not using SSL/TLS means sacrificing some aspects of security.

- *TCP/IP*
  All of the communication described above use TCP in order to have an in order reliable connection that TCP will deliver the bytes in a stream to destination, performing retransmissions if necessary. TCP is also used as the transport protocol, because most firewalls and NAT devices will enable a TCP session initiated from within the local network to receive TCP traffic on the return connection.

2.5.2.2 Security

The security goals of CWMP are to:

- Prevent tampering with the management functions of a CPE or ACS, or the transactions that take place between a CPE and ACS.
- Provide confidentiality for the transactions that take place between a CPE and ACS.
- Allow appropriate authentication for each type of transaction.
- Prevent theft of service.

For better security, it is recommended in CWMP to use SSL 3.0[8]/TLS 1.0[9]. The indication of a request for secure connection establishment occurs when the ACS address is specified as an HTTPS URL. In this case, the CPE must establish the connection using SSL/TLS and ACS must also authenticate the CPE using SSL/TLS. Otherwise the ACS must authenticate CPE using normal HTTP (basic) authentication. The HTTP basic authentication scheme is a method designed to allow a web browser, or other client program, to provide credentials – in the form of a user name and password – when making a request. These credentials will then being transferred in a plaintext, which is easily being intercepted by any third party.

When receiving an incoming connection request from an ACS which was specified with an HTTPS URL, if the CPE supports both SSL and TLS, it must negotiate the capabilities to allow ACS to select one of these protocols. If the CPE only supports one of these two protocols, then it must use the one it supports to open secure connection. When using SSL/TLS, the CPE authenticates the ACS using an ACS - provided certificate against one or more trusted root certificate that are either pre-loaded in the CPE or provided securely to the CPE. CWMP does not specify the mechanism of certificate provision or revocation.

2.5.2.3 CWMP Parameters

CWMP parameters are defined as a data model based on TR-106: Data Model Template for TR-069-Enabled Device [19]. The CWMP parameters have been designed in a hierarchical structure. This data structure in CWMP is presented in the form of an object. Each object contains one parameter or a collection of parameters. Every CWMP compliant device always has a Root Object which is called "Device" for most CPEs or "Internet Gateway Device" for a broadband router. A root-object contains two sub-elements which are the Common-Object and the Services-Object.

Common-Object contains parameters that are common to the CPE; these are used by the ACS for CPE identification. Parameters in the common-object include: DeviceInfo, ManagementServer, GatewayInfo, Time, Config, UserInterface, and LAN. These objects will be described in more detail later in this thesis.

Service-Object contains parameters that show the capability of the CPE to subscribe to the service provided by the ACS. Each CPE can have more than one service objects, if it supports more than one capability. For example, a CPE that is can be use for VoIP and IPTV will have two services objects.

```
Element = Root
        | Root ".DeviceSummary"
        | Root ".Services." ServiceObject "." Instance
        | Root ".Services." ServiceObject "NumberOfEntries"
        | Root ".Services." ServiceObject "." Instance "."
SecondaryCommonObject
        | DeviceRoot "." CommonObject
        | GatewayRoot "." GatewaySpecificObject

Root = DeviceRoot
     | GatewayRoot

DeviceRoot = "Device"

GatewayRoot = "InternetGatewayDevice"

CommonObject = "DeviceInfo"
             | "Config"
             | "UserInterface"
             | "ManagementServer"
             | "GatewayInfo"
             | "Time"
             | "LAN"

SecondaryCommonObject = "DeviceInfo"
        | "Config"
        | "UserInterface"
        | "Time"
        | "LAN"

Instance = NONZERODIGIT [DIGIT]*
```

Figure 2-9: CWMP Data Hierarchy [19]

The ACS defines profile for services management. Each profile represents specific sets of requirements. Parameters for configuring, subscribing, and unsubscribing to a certain service, can be grouped into a profile. A CPE can have more than one profile depending on which services it wants to subscribe. Normally, each CPE has one basic profile which defines its support for basic requirements. One or more complex profiles can be generated based on the basic profile.

The CWMP parameters for are defined in two technical reports. TR-098: Internet Gateway Device Data Model for TR-069 [17] defines the parameters related to IP network configuration, and TR-104: Provisioning Parameters for VoIP CPE [18] specifies telephony configuration related parameters. These two technical reports define a standardized parameter name and object hierarchy that a TR-069 compliant CPE may support.

2.5.3 CWMP Operations

CWMP defines a set of procedures. These are: ACS Discovery, Connection establishment, and File transfer.

2.5.3.1 ACS Discovery

There are several mechanisms a CPE can use to discover its associated ACS:

- Use locally configured URL of the ACS
  The URL of ACS can be configured locally in the CPE. This method can be used if a CPE is set to use a fixed IP address. The CPE can use DNS to resolve the IP address of the ACS based upon the host name component of the URL. This method shall be used only on initial connection to an ACS. After successfully connecting to an ACS, this URL can be modified for subsequent connections.
- Use a DHCP [14] option
  The URL of an ACS can be placed in a DHCP option. When the CPE gets its IP address dynamically (via DHCP), it also receives the ACS URL in a DHCP option. A common option to use is DHCP option 43, Vendor Specific

Information A.1. This option contains free text in which an ACS' URL information can be stored. TR-069 defines two relevant parameters - shown in table below.

Table 2-2: Encapsulated Vendor Specific Options [1]

Encapsulated Option	Encapsulated Vendor Specific Option Number	Parameter
URL of the ACS	1	…ManagementServer.URL
Provisioning code	2	…DeviceInfo.ProvisioningCode

Since DHCP does not provide **any** security mechanism, configuration using a DHCP option should be done via a secure link between the DHCP server and CPE.

- Use stored ACS URL
  Once a CPE has successfully established a connection with an ACS, it retrieves the current ACS's URL and stores it internally in the parameter: ManagementServer.URL. The next connection to an ACS shall use the updated ACSURL.

2.5.3.2 Connection Establishment

Connection establishment can be initiated by either the CPE or ACS. These two alternatives are described below.

- ***Initiated by CPE***

After a CPE discovers the correct address to reach the ACS, then the CPE establishes a connection with this ACS. Connection establishment may occur for any of the following conditions:

- First time boot up during initial installation
  This case occurs when a user has received the CPE and plugs it into the home network for the very first time. Depending on how the home network has been configured, the CPE will use one of the ACS discovery mechanisms (explained above) to learn the address of the ACS.
- On power up
- At periodic time intervals
- When instructed by a human operator (i.e. for device diagnostic purpose) or as a response to certain events.
- Upon receiving a valid connection request from an ACS
- When the address of the ACS changes
  When the ACS changes its address, it may send a request to each of the CPEs that is managing to initiate a new connection. This might occurs if there were a secondary ACS in the network and the primary ACS is down.
- When a parameter has been modified that requires the CPE to inform the ACS of this modification
- Retry upon abnormal session termination
-

If the connection to ACS terminated abnormally, then a CPE must try to reconnect by sending a re-connect request during a time interval as shown in
Table 2-3.

Table 2-3: CPE Reconnect Interval [1]

Post reboot session Retry count	Wait interval range (min-max) seconds
#1	5-10
#2	10-20
#3	20-40
#4	40-80
#5	80-160
#6	160-320
#7	320-640
#8	640-1280
#9	1280-2560
#10 and subsequent	2560-5210

Once a connection has been established, the CPE sends an initial INFORM request to the ACS to initiate a session. If this INFORM request is successful (as indicated in the response from the ACS), then the CPE must consider that a session has been successfully initiated. Now the CPE is ready to accept request from the ACS.

- ***Initiated by ACS***

An ACS can initiate a connection with a CPE using a connection request mechanism. The CPE must support this mechanism in order to accept a connection request from ACS. This connection request can use TCP port 7547 which IANA [20] has assigned for CWMP. A connection request from ACS must use an HTTP 1.1 GET to a specific URL designated by the CPE. This request packet should contain no data in it. Any data in the connection request should be ignored. The CPE must accept any incoming connection request that has a valid authentication parameter. When receiving a connection request, the CPE must use digest authentication to authenticate the ACS before proceeding. The CPE must not initiate a connection if it fails to authenticate the ACS request. If the ACS was successfully authenticated, then the CPE must immediately respond with 200(OK) or 204(No Content) HTTP status code, before continuing to initiate the resulting session.

2.5.3.3 CPE Operation

After successfully initiating a connection with the ACS, a CPE will initiate a transaction session. The success of connection initiation is indicated by a response code sent by the ACS. The CPE initiates a transaction session by sending an initial INFORM request to the ACS. An INFORM request contains the following parameters: DeviceID, Event, MaxEnvelopes, CurrentTime, RetryCount, and ParameterList (a detail explanation of these parameters can be found in appendix A.3.3.1 [16]). By sending an INFORM request, the CPE informs the ACS of its current status and indicates that it is ready to receive further incoming requests from ACS. However, a CPE shall not consider a transaction session to have been successfully initiated **unless** it receives a successful INFORM response from the ACS. Once a transaction session has been established, a CPE can:

- Receive valid incoming requests from an ACS. The CPE must respond to these requests in the next HTTP POST to the ACS.
- Make one or more outgoing requests to the ACS using next HTTP POST message.

When all transactions have been completed, the CPE must terminate the session. A transaction is completed if the CPE has received an empty HTTP response from the ACS or the CPE sends an empty HTTP POST to the ACS after all prior requests from the ACS have been fulfilled. A transaction is successfully terminated if the CPE receives an HTTP response from the ACS within 30 seconds, otherwise the session is considered as unsuccessfully terminated.

2.5.3.4 ACS Operation

The ACS initiates a session after successfully receiving a valid INFORM request from a CPE. The session starts when the ACS responds with an INFORM response. After receiving an incoming SOAP request from a CPE, the ACS must respond to the request in the next HTTP POST to CPE. An ACS can temporarily prevent a CPE from sending requests and later allow it again to send requests by setting HoldRequest in the SOAP envelope to 0 or 1 or by sending an empty HTTP response. If the ACS has one or more requests to send to a CPE, it must sent each request in a new HTTP response.

For the ACS to send an outgoing request, it has to ensure that the most recent HTTP post from a CPE did not contain a SOAP request. When all requests have been fulfilled and there are no pending requests from the ACS to the CPE, and the most recent HTTP POST from the CPE did not contain any SOAP request, then the ACS must send an empty HTTP response to terminate the session.

2.5.3.5 File Transfer

A CPE can be instructed to perform a file transfer to download a file from or upload a file to ACS. The file to be transferred can contain firmware and/or be a vendor specific configuration file. The file transfer is done using HTTP/HTTPS or optionally FTP, SFTP, or TFTP. The transfer should be done when the CPE has successfully established a connection to the ACS. File transfer between a CPE and an ACS can:

- Use the current connection for the file transfer.
  This option is only possible if the file to be transferred is in the same location as the ACS (i.e., the same host component of the HTTP URL). The CPE may send an HTTP GET/PUT over the already established connection. The CPE can send a connection keep-alive message when the file has been successfully transferred in order to maintain the connection.
- Keep the current connection and open a new connection for transferring a file. In this case, both the CPE and ACS may exchange connection keep-alive messages.
- Close the current connection and open a new connection for transferring a file. Once the file transfer has finished, then the CPE can send a new connection request to the ACS to initiate a new connection.

When the file transfer has finished, there should be a status report sent to indicate whether the transfer was finish successfully or failed. If the transfer failed, then some retry policy should be applied, for example to retry the transfer immediately or only after several minutes.

2.5.4 CWMP Operations through a NAT

2.5.4.1 NAT overview

Network Address Translation (NAT) is a method by which IP addresses are mapped from one realm to another, in an attempt to provide transparent routing to hosts - while allowing the use of a private address range inside the realm. Originally, NAT was used to connect and isolate a local network from the public or global network. However, the shortage of public IP addresses has promoted NAT as solution to this perceived IP address shortage.

The IP addressing scheme used up to the present is that of IP version 4 (IPv4). This version of IP uses a 32-bit (4-byte) address, which limits the address space to less than 4 billion unique addresses. Some of these addresses are reserved for special purposes, such as private networks (~18 million addresses), multicast addresses (~1 million addresses), and some reserved for future use. This reduces the number of addresses that can be allocated as public internet addresses to ~3.76 billion. The rapid growth of the internet has required more address than IPv4 is able to provide. IP version 6 (IPv6) increases the address space by increasing the length of IP address to 128 bits, which is able to support roughly ~3.4×10^{38} addresses. However, no one can really say when IPv6 is going to ubiquitous.

Meanwhile, NAT is popular among users because it is simple and seems to solve the IP address shortage problem. A NAT divides the network into public and private networks. The public network is the publicly accessible network of interconnected computer networks that transmit data using IP packets. A private network is a limited access network of interconnected computer networks that transmit data packets using private IP address. These private IP addresses are only usable from within the defined private network. The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets [45]:

```
10.0.0.0        -   10.255.255.255  (10/8 prefix)
172.16.0.0      -   172.31.255.255  (172.16/12 prefix)
192.168.0.0     -   192.168.255.255 (192.168/16 prefix)
```

Public IP addresses are all IP addresses which are not in the range defined above. Each private network utilizes private addresses. Thus, packets which are to be routed to the public network will need to have their private address translated into a public address assigned to the public interface of the NAT device.

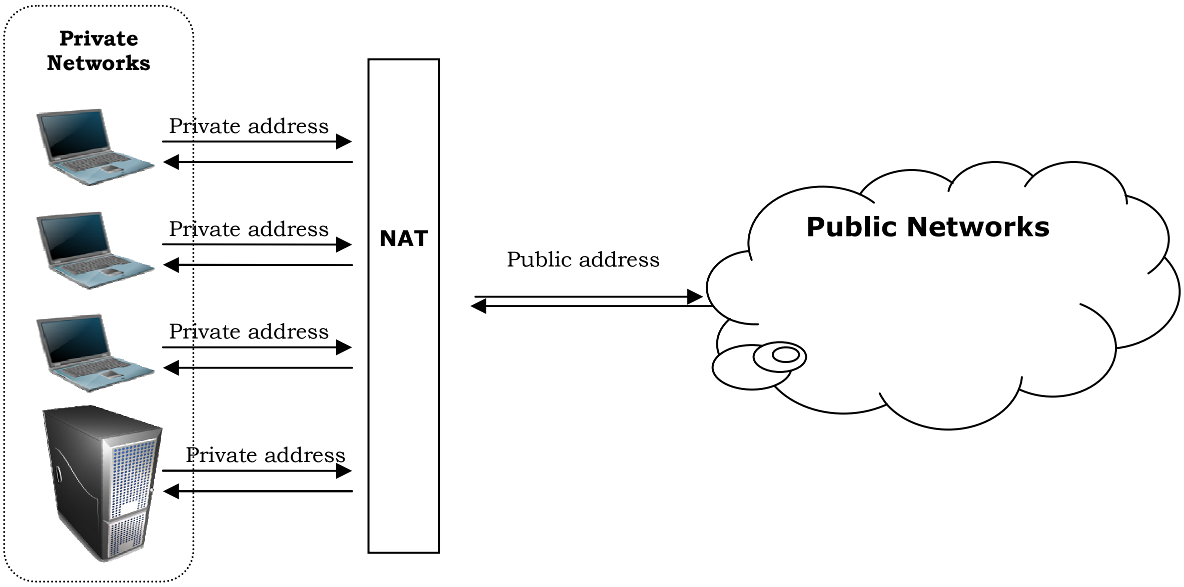


Figure 2-10: NAT

There are different types of NATs:

- Full Cone NAT

A full cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Furthermore, any external host can send a packet to the internal host, by sending a packet to the mapped external address and port.

- Restricted Cone NAT

A restricted cone NAT is one where all requests from the same internal IP address and port are mapped to the same external IP address and port. Unlike a full cone NAT, an external host (with IP address X) can send a packet to the internal host only if the internal host had previously sent a packet to IP address X.

- Port Restricted Cone NAT

A port restricted cone NAT is like a restricted cone NAT, but the restriction includes port numbers. Specifically, an external host can send a packet, with source IP address X and source port P, to the internal host only if the internal host had previously sent a packet to IP address X and port P.

- Symmetric NAT

A symmetric NAT is one where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port. If the same host sends a packet with the same source address and port, but to a different destination, a different mapping is used. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

Although attempting to solve problem of IP shortage, NAT introduces a serious drawback. The hosts behind a NAT do not have true end-to-end connectivity and cannot participate in some Internet protocols. Services that require the initiation of TCP connections from the public network, or stateless protocols such as those using UDP, can be disrupted and may not be able to function. Unless the NAT device makes a specific effort to support such protocols, incoming packets cannot reach their destination. In some protocols, a device behind a NAT requires some means to discover its public address in order to communicate with peers in the public networks. This can be done by using STUN that will be discussed in 0.

2.5.4.2 CWMP and NAT

CWMP can be used to manage a CPE that is connected via a LAN, when the CPE has been configured to use a private IP address (see Figure 2-11).

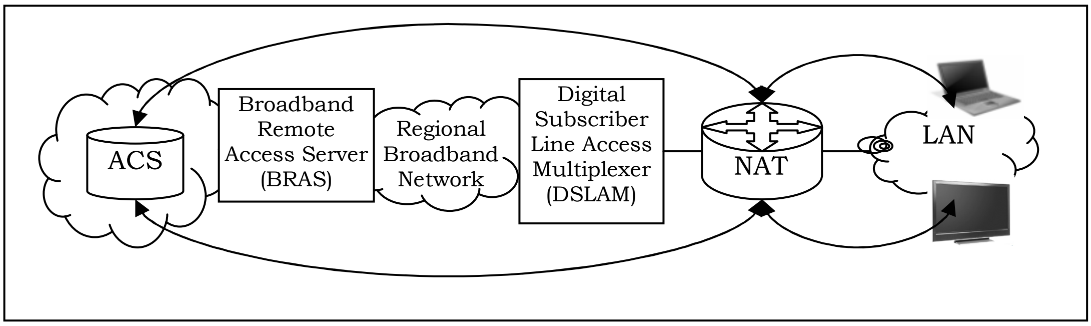

ACS
Broadband Remote Access Server (BRAS)
Regional Broadband Network
Digital Subscriber Line Access Multiplexer (DSLAM)
NAT
LAN

Figure 2-11: CWMP in a NAT environment

There are CWMP capable NAT gateways available today. If the NAT gateway is CWMP capable (i.e., the NAT is aware of and understand the CWMP protocol), then the CPE and the gateway exchange identities through DHCP message exchange. These identities include the manufacturer OUI, serial number, and optional product class. CPE puts its identity in a DHCP option, and sends it to the gateway in a DHCP DISCOVER message when requesting an IP address. When the gateway responds to the CPE, it sends a DHCP RESPONSE that contains an identity (i.e. the identity of the gateway). When both the gateway and CPE know each others identity, gateway can make a connection request to ACS by presenting the identity information from both of them.

If the gateway is not CWMP capable, then a connection request to an ACS can still be done according to the TR-069, but with the limitation that the connection request mechanism in TR-069 for initiating the session cannot be used. In this case, the CPE must first discover to the ACS through the NAT gateway using the correct public IP and selected port. The correct public IP address and port can be discovered via Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) – STUN, which will be discussed in 0. Once the CPE knows a correct public IP address and port, it can establish a connection to the ACS. After the connection has been established, the CPE must keep the NAT binding open to enable the CPE and the ACS sending messages to each other. It can keep this binding open by sending traffic before the binding times out.

2.5.4.3 DHCP Overview

Dynamic Host Configuration Protocol (DHCP) [14] is a protocol used by network nodes to automatically obtain an IP address and other network related parameters (such as subnet mask, DNS server, and gateway address). Using DHCP to assign an IP address ensures every network node has a unique IP address during the specified lease time. The lease time is a time interval during which a network node is assigned a specific IP address.

In order for a network client to get an IP address from the DHCP server, it must first discover the DHCP server. This can be done by broadcasting a DHCP DISCOVER message. The DHCP server responds with a DHCP OFFER message to indicate its existence and to offer (an available) IP address to the client. When the client receives an offer from the DHCP server, it sends a DHCP REQUEST message to the DHCP server indicating the acceptance of the offered IP address. DHCP server shall then send DHCP ACK to acknowledge request from the network client. In some cases, DHCP server may sends a DHCP NAK (not acknowledge) indicating the reason for the client's faulty (i.e. client moved to another subnet) or lease time has expired.

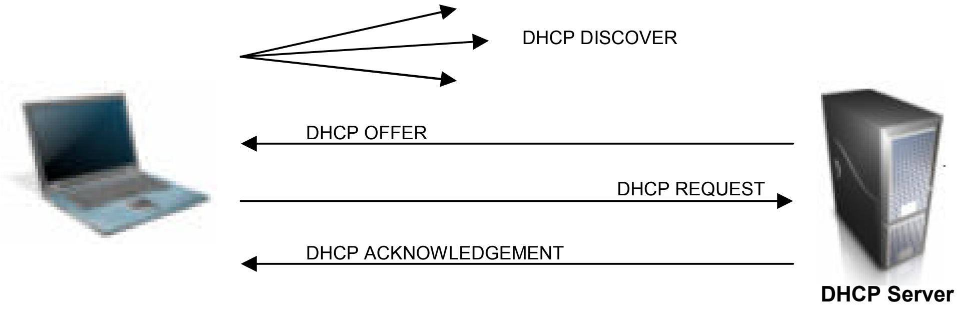


Figure 2-12: DHCP Message Exchange

In addition to assigning an IP address and conveying other network related parameters, a DHCP server can provide optional parameters to the client. These optional parameters are the DHCP OPTIONs which are defined in RFC2132 [14] and RFC-3942[24]. Some of the DHCP options that are used in CWMP are:

Table 2-4: DHCP Options

DHCP Option	Description
Option 43	Vendor Specific Information
Option 60	Vendor Class Identifier
Option 61	Client Identifier
Option 125	Vendor Specific Information
Option 224- 255	site-specific purpose

2.5.4.4 STUN Overview

STUN is a client-server network protocol allowing a client behind a NAT (or multiple NATs) to find out its public address, the type of NAT it is behind, and the (public) internet side port associated by the NAT with a particular local port. A device behind a NAT gateway acts as STUN client to find out its public address. To do this, it sends a binding-request over UDP port 3478 to the designated STUN server. The STUN server is typically located in a public network and use public IP address to communicate to the STUN client. When a Binding Request arrives at the STUN server, it may have passed through one or more NATs between the STUN client and the STUN server. As a result, the source address of the request received by the server is the mapped address created by the NAT closest to the STUN server. The STUN server copies the source IP address and port into a STUN binding-response, and sends these responses back to the source IP address and port of the STUN request. When the STUN client receives the STUN Binding Response, it compares the IP address and port in the packet with the local IP address and port it bound to when the request was sent. If these do not match, then the STUN client is behind one or more NATs. The STUN client will use IP address and port in the STUN response as information on how to reach itself for a peer on the (public) internet. RFC-3489 [25] defines an algorithm for the STUN server to use to discover the presence and type of the NAT (see Figure 2-13).

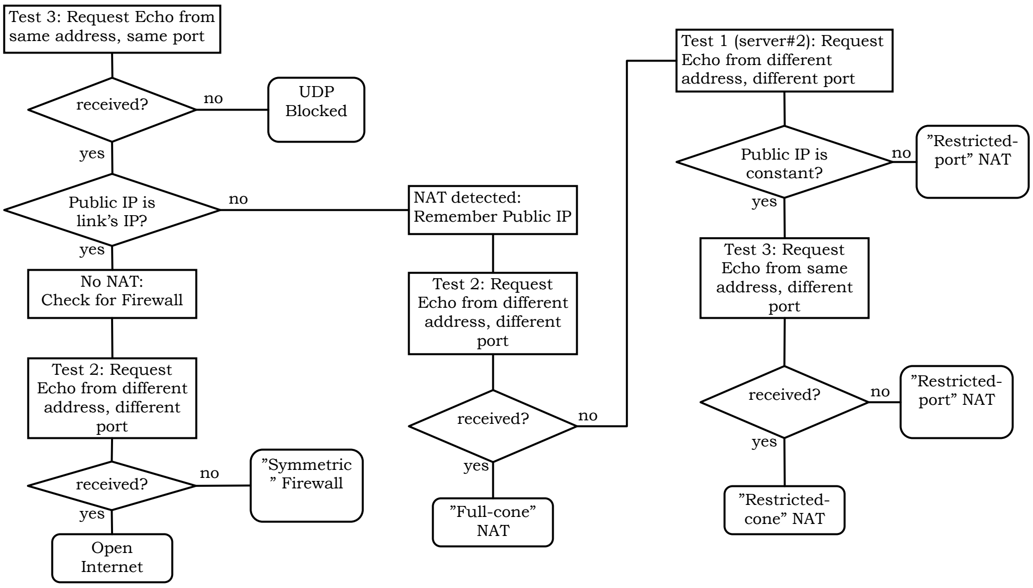

Test 3: Request Echo from same address, same port
received?
no
UDP Blocked
yes
Public IP is link's IP?
no
NAT detected: Remember Public IP
yes
No NAT: Check for Firewall
Test 2: Request Echo from different address, different port
received?
no
"Symmetric" Firewall
yes
Open Internet
Test 2: Request Echo from different address, different port
received?
no
yes
"Full-cone" NAT
Test 1 (server#2): Request Echo from different address, different port
Public IP is constant?
no
"Restricted-port" NAT
yes
Test 3: Request Echo from same address, different port
received?
no
"Restricted-port" NAT
yes
"Restricted-cone" NAT

Figure 2-13: NAT discovery algorithm in STUN server [35]

3 Designing a Solution for CPE Management

We began the process of designing our solution for CPE management by adopting CWMP (specifically TR-069). The primary goal was to build an ACS and to implement TR-069 for the CPE, in order to be able to communicate with an ACS. The secondary goal was to collect all the parameters supported in the CPE which might be relevant for CPE management, and ensure that they can be mapped into parameters as in TR-098 and TR-104 according to the definition in TR-106. Currently, the secondary goal has not been completed and a significant amount of work remains until this can be realized. However, some progress has been made - thus the basic implementation of TR-069 for the CPE can be shown to work (this is evaluated in section 4.4)

Note that we have assumed that we will design, implement, and evaluate both the CPE and ACS portions of CWMP. We have explicitly not included interoperability testing as part of this thesis project, but some results of initial interoperability testing are reported in section 4.4.3.

3.1 Solution Overview

CWMP is naturally divided into a server (ACS) side and a client (CPE) side design and implementation. From the ACS's side, the solution must support responses to an incoming request for registration, firmware, or configuration file download; as well as maintaining the session. From the CPE's side, the protocol must support responses to requests for status reporting and sending informational message to the ACS when a special event occurs or periodically.

The scope of designing CWMP in order to fit the requirement of 42networks' ACS and CPE are included:

- Unit Discovery
- Initiate and Maintaining a Session
- RPC message formatting and exchange
- Parameter Translation

3.2 Unit discovery

The goal of unit discovery is to let the ACS know that there is a new CPE in the network managed by the ACS. The discovery process begins by giving the CPE the information on how to reach the ACS. This information can be the IP address or fully qualified domain name (FQDN) of the ACS. The IP address or FQDN of the ACS can be manually inserted into the CPE by using a parameter in a configuration file. The details of the configuration file are explained in appendix A.3. A basic configuration file would contain a specification of the ACS's URL, such as:

```
[HEADER]
Configuration file sample

[CONFIG]
ACSURL=192.168.18.30
```

Using a configuration file is simple and straight-forward, but this depends on a human administrator to upload (and re-upload in case of modification) the correct configuration file. The additional uploading effort will increases the cost of operation of the service provider.

A better way is to use an automatic mechanism to insert (or modify) the ACS address. One of such approaches is to use DHCP. There are two options to use DHCP:

- Using DHCP Option 43 (Option Vendor Encapsulate)
  DHCP option 43 is a free-text option designed to contain vendor specific information. The ACS address could be placed in this option, for example:

  ```
  Option Vendor-Encapsulated "ACSURL=192.168.18.30"
  ```

- Using a DHCP site specific option from the numeric range 224-254 as defined in RFC-3942 [24]
  The RFC-3942 describes the use of unassigned DHCP option 224-254 for site-specific purposes. One can choose a number from those unassigned option and use it to carry ACS's address. If a vendor has selected an unassigned option and is willing to document that usage, this vendor must inform IANA. In this case the IANA will list these options as "Tentatively Assigned". In this thesis, we will assume that we are using option 230 to carry ACS URL information, for example as:

  ```
  Option ACSURL "192.168.18.30"
  ```

The advantage of using a DHCP option is that a different value can be sent with every DHCP message exchanged when a CPE requests or renew its IP address. The network administrator simply updates the DHCP server's configuration file to modify the option, and cause the DHCP server to read this configuration file - which will now be distributed by the DHCP server. The information contained in the ACSURL is stored in the CPE as an internal parameter and used persistently for communication with ACS.

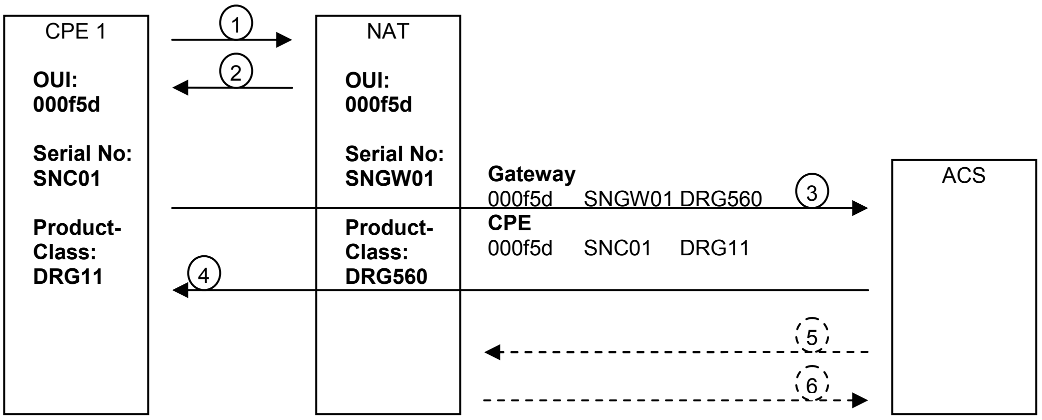
Now that if the CPE knows the address of the ACS, it can send an INFORM message containing its identity and current status. After the ACS receives the INFORM message, it will acknowledge it and add the CPE into the list of devices which it is managing. Thus the ACS has "discovered" the unit to be managed.

3.2.1 Discovery of a CPE behind a NAT

This section describes how an ACS can communicate with a CPE located behind a NAT. The unit discovery for a CPE behind NAT depends upon whether the NAT is CWMP capable or not. A CPE will find out whether its NAT is CWMP capable by checking the NAT's identity which is including, the manufacturer OUI, serial number, and optional product class in the DHCP ACK message. These identities are typically present in DHCP option 43. If these identities are missing, then the NAT is *not* CWMP capable.

3.2.1.1 CWMP Capable NAT

For a NAT which is CWMP capable, a CPE and NAT can establish an association by exchanging identity information with each other using a DHCP message. When a CPE requests its IP address from a DHCP server, it includes its identity information in the DHCP DISCOVER message. This information can be placed in DHCP option 43 or option 125. The CPE identity includes the CPE's manufacturer OUI, CPE's serial number, and an optional product class. This information shall be sent to the NAT which is defined in option 10 in DHCP server's response. The NAT shall store this information from each CPE as rows of data in a ManageableDevice table (see Table 2-2). This process is illustrated in Figure 3-1 .



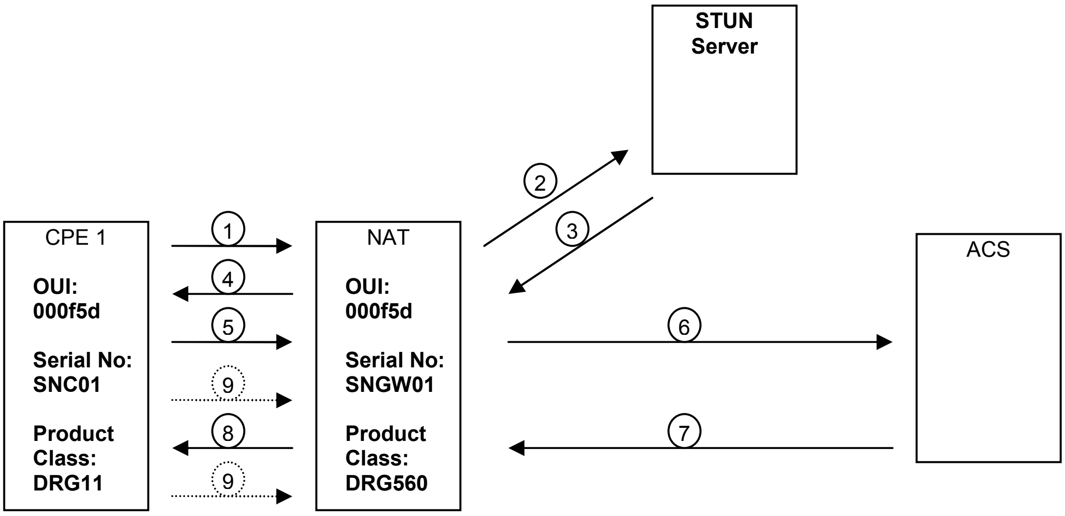
1. The CPE sends a DHCP REQUEST to get an IP address. The DHCP REQUEST from the CPE contains its identities. Assuming the NAT is a DHCP server, it will get identities from the CPE when it receives DHCP REQUEST. Otherwise, the DHCP Server might need to have an extra procedure to send CPEs identities to gateway. The identity information is encoded as shown below:
   ```
   Option Vendor-Encapsulated "DeviceManufacturerOUI=000f5d;
   DeviceSerialNo= SNC01;"
   ```
   The NAT populates a table of ManageableDevice which contains information from all CPEs in the network . This information will later be used to allow ACS to access these CPSs.
2. When a CPE gets DHCP ACK, it also gets gateway's identity along in the packet.
   ```
   Option Vendor-Encapsulated "GatewayManufacturerOUI=00aabb;
   GatewaySerialNo=SNGW01;"
   ```
3. The CPE sends an INFORM message which including its identity and NAT's identity to the ACS.
4. The ACS replies with an INFORM response to CPE through the NAT. At this point the ACS knows the identities of the CPE and its gateway
5. When an ACS is about to access CPE, it start CWMP session to the NAT to learn about the CPE by reading the ManageableDevice table.
6. The NAT responds with a message carrying the content of ManageableDevice Table.

Figure 3-1: CPE behind CWMP capable NAT

3.2.1.2 Using a non-CWMP capable NAT

If the NAT is not CWMP capable, then the CPE uses a different mechanism to establish a connection with the ACS. Since the CPE has been assigned a private IP address, it must first discover how to reach the ACS through the NAT. Following this, the CPE must keep the NAT binding open to ensure the ACS that it can transmit unsolicited packets to the CPE. The CPE must be able to determine the public IP address and port number that it should use to contact the ACS, in order to associate these with the open NAT binding and to communicate this information to the ACS.

This mechanism relies on the use of STUN, which requires the CPE to act as a STUN client. A STUN client can be enabled in CPE by default, either in the source code or by setting the STUNEnable parameter to TRUE. If the CPE is assigned a private IP address, then it will use the designated STUN server search procedure to determine the NAT gateway host address and port and type of the NAT. The designated STUN server is stored in the parameter: STUNServerAddress and STUNServerPort. The CPE will provide the host address and port of the gateway to the ACS. The ACS keeps this host address and port information in a UDPConnectionRequestAddress parameter and uses these values when initiating a connection request to the CPE. The CPE periodically retransmit a binding request to the NAT in order to maintain the availability of the address and port as used by the source address and port on which CPE will listen for UDP connection requests.



1. If STUN is enabled in the CPE, it sends a STUN binding Request to the designated STUN server through the NAT.
2. The NAT forwards the binding request to the designated STUN server
3. The STUN server returns the public IP address and port from gateway as MAPPED-ADDRESS to the CPE
4. The NAT forwards the reply to the CPE. When the CPE receives a reply from ACS through the NAT, it stores the public IP address and port of the NAT.
5. The CPE initiates a connection to the ACS through the NAT. Along with the connection request, CPE also informs the ACS of the MAPPED-ADDRESS of the NAT.
6. The NAT forwards the request to the ACS. When the ACS receives request from the CPE it learns the MAPPED-ADDRESS of the NAT and stores these information in UDPConnectionRequestAddress
7. ACS initiate connection to CPE by sending connection request to address and port stored in UDPConnectionRequestAddress
8. The NAT forwards request to the CPE
9. The CPE periodically transmits a NAT binding to address stored in UDPConnectionRequestAddress to keep the address and port open.

Figure 3-2: CPE behind a non-CWMP capable NAT

3.3 Initiating and maintaining a session

After the CPE and ACS have found each other, they must be able to intiate and maintain a session to communicate with each other. The CPE needs to learn the address of the ACS (as mentioned in the discovery section above) in order to provide its identification and latest status to the ACS. A new CPE sends a BOOTSTRAP INFORM message to ACS to inform the ACS of it's identify and the network address and UDP port number to be used to reach it. The ACS processes this message and adds the CPE into a list of unmanaged CPEs. The CPE stays in this list until the registration process has completed successfully. The registration process includes CPE identification and profile (see section 3.6) assignments by the ACS. Once this is done the CPE will be moved to the list of managed units. This process is shown in Figure 4-7.

The registration session shall be kept-alive as long as the CPE continues sending periodic INFORM message to the ACS. If there is no INFORM message received after the keep-alive time, then the ACS shall note this CPE in the list of lost-registration units, until it receives the next valid INFORM from this CPE.

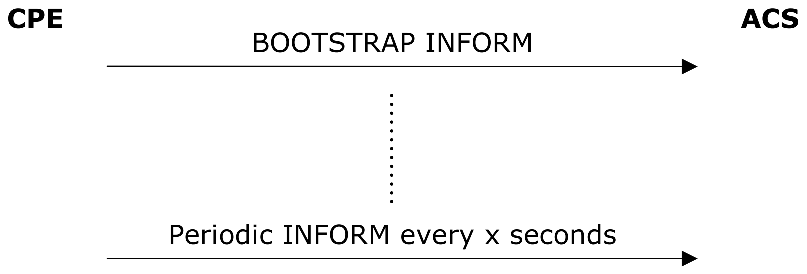


Figure 3-3: BOOTSTRAP and Periodic INFORM

The transactional session starts when either the ACS or CPE sends a request to execute a certain method. Each transactional session allows the occurrence of several transactions. The session maintenance is handled using a cookie across the several transactions. The cookie contains transaction identification, as well as the CPE identification and ACS URL, for example:

```
Set-Cookie: TID=12347339;CPEID=000f5dfefffe;ACSURL=sample.acs.com
```

The session will end when the initiating party receives an empty HTTP response message. When the session ends, the cookie must be discarded. Any expired cookies shall be ignored.

3.4 RPC message format and exchange

CWMP uses an HTTP 1.1 message encoded in XML. In order for both sides to independently interact with each other, this protocol will make use of the Remote Procedure Call (RPC). There are two types of messages in this protocol. The first one is an informational message. Information message is sent from a CPE to an ACS or an ACS to a CPE. Such message does not need a response, because there is no function execution or any return value. An example of an informational message is an INFORM message. INFORM is sent from a CPE to an ACS on first use of device, every boot-up (restart), periodically, or following any triggered event. Information in an INFORM message includes a standard message defined in TR-098. This message contains the following information:

- DeviceSummary
- SpecVersion from the device
- Hardware Version from the device
- Software Version from the device
- Provisioning Code from the device
- Connection Request of URL from the ACS
- Parameter key from previous transaction
- WAN interface IP address

In addition, this message will also contain the following manufacturer defined parameters which are not defined in TR-098:

- Unit identity
  This is the identity of the CPE, which could be its MAC address on the WAN side. A unique serial number or combination of MAC and serial number can also be use as the identity.
- Downloader version
  The downloader in the 42networks' CPE is used to install new software into the CPE. The CPE includes the downloader version when sending INFORM message to the ACS.

- Last Successful INFORM
  This is the time when a CPE had sent a latest INFORM message successfully to the ACS. If the time has exceeded the periodic inform window, then the ACS might suspect an abnormal connection to the CPE and might initiate an immediate monitoring function to check the connection status.
- Event code indicating why the INFORM message was sent.
  This code explains why the INFORM message was sent. The reasons are event occurrences such as:
  - 0 - BOOTSTRAP
    This event occurs when there is update in the DHCP option string. For example, if the unit discovery were using DHCP option 43, then if a CPE notices that the string in option 43 in DHCP was modified, then it will trigger a transmission of INFORM message to the ACS using the latest ACSURL specified in the option 43 string. This most likely happens when the URL of the ACS is modified, thus each CPE will send an INFORM to the newly appointed ACS.
  - 1 - BOOT
    This event occurs when CPE is rebooted for other than one of the reasons below.
  - 2 - PERIODIC
    A periodic INFORM message is sent at a periodic time interval.
  - 4 - VALUE CHANGED
    If the value of a parameter was modified, then the CPE will trigger an INFORM message with 4 - VALUE CHANGED as the event code.
  - 7 - TRANSFER COMPLETED
    This event occurs when download of a firmware or a configuration file is completed.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
    <Request>Inform</Request>
    <Parameter>
    <Device>
        <DeviceInfo>
            <DeviceSummary>DRG-22,Network:DefaultEnable:1.0</DeviceSummary >
            <SpecInfo>1.0</SpecInfo>
            <HardwareVersion>R9A</HardwareVersion>
            <SoftwareVersion>DMA0027-R2K161</SoftwareVersion>
            <ProvisioningCode>XZCO:GRP2</ProvisioningCode>
        </DeviceInfo>
        <ManagementServer>
            <ConnectionRequestURL>http://acs.local</ConnectionRequestURL>
            <ParameterKey>zHbfgjk</ParameterKey>
        </ManagementServer>
        <WANInterface>
            <ExternalIPAddress>DMA0027-R2K161</ExternalIPAddress>
        </WANInterface>
    </Device>
    <X_000f5d>
        <DeviceInfo>
            <MAC>000f5dfe7d77</MAC>
            <DowloaderVersion>cxc_1492_4888_R2A47</DowloaderVersion>
            <LastSuccessInform>17:23 GMT</LastSuccessInform>
            <Event>
                <EventCode>0 BOOTSTRAP</EventCode>
            </Event>
        </DeviceInfo>
    </X_000f5d>
    </Parameter>
</soap:Body>
</soap:Envelope>
```

Figure 3-4: An example of INFORM message

The second message type is a transactional message consisting of a request and response. Each request shall generate one or more response messages. A request is a command to execute a particular method. The response will contain the return value from the execution and/or error codes.

The standard supported RPC methods according to CWMP are:

- GetRPCMethods
- GetParameterNames
- GetParameterValues
- GetParameterAttributes
- SetParameterValues
- SetParameterAttributes
- Download
- Reboot

For those who would like to have more details and examples of these methods, please see the appendix A.4.

In the addition to these standardized methods a number of Vendor-specific methods may be supported. Each method must include a unique identifier in order to distinguish it from the standard methods and other vendor-specific methods. The unique identifier can be either the vendor-name or an officially assigned OUI[2]. The format of a vendor-specific method might looks like the following:

```
X_<VENDOR-IDENTIFIER>_MethodName
```

An example of one of a 42networks' specific ACS methods is:

```
X_000f5d_SetOPURL
```

Each RPC exchanged message is carried as an HTTP message payload as specified in RFC2616 [7]. The session initiator sends messages encapsulated in HTTP Post message and responses in a HTTP response message contain a response code or empty message indicating the end of a session. The syntax of a request message is simply clear text in XML format, followed by the required parameters for the method. Each line is terminated by "CRLF". An example of HTTP POST might looks as the following:

```
POST HTTP/1.1
Host: sample.acs.com
User-Agent: Mozilla/4.0
Content-Length: 27
Content-Type: application/CWMP

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
        <Request>Inform</Request>
        <Parameter>
                section.name=xxx
                . . . .
        </Parameter>
        <soap:/Body>
<soap:/Envelope>
```

3.5 Parameter

One important goal for a fully TR-069 compliant system is to support the parameters as defined in TR-098 [17] and TR-104 [18]. This is still a visionary goal because not all CPEs are able to support all parameters in these two technical reports due to limitations of internal resources and for interoperability reasons. Additionally, not all parameters are applicable in an actual implementation; while

[2] Note that OUIs are assigned by the IEEE.

there are some features that do not have any parameters defined either in TR-098 or TR-104. Thus, the ACS must be designed to be able to manage CPEs that use either CWMP or non-CWMP parameters. A CPE that has any non-CWMP parameters has to translate its non-CWMP parameters into CWMP parameters in order for an ACS to manage it. This task began by classifying the CPE parameters as described in the next section.

3.5.1 Classifying the parameters in a CPE

The first step in the parameter mapping is to separate CPEs parameters into groups of VoIP related parameters and IP related parameters (corresponding to the division in the two technical reports). VoIP related parameters are matched to TR-104 parameters; while IP networking parameters are matched to TR-098 to find the identical functions. The result of this matching is represented in Figure 3-5.

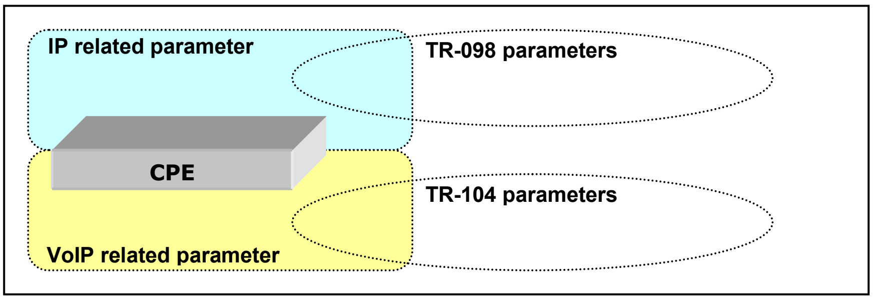


Figure 3-5: Parameters

Figure 3-5 shows a CPE has sets of parameters for configuring services, for example the WAN connection and VoIP. Some of the WAN connection parameters are defined in TR-098 and some of the VoIP parameters in TR-104. However, there are parameters that are not defined in either TR-098 or TR-104. This may due to the CPE support unique capabilities and these capabilities are controlled by vendor specific parameters. If the CPE's parameters were map into TR-098 or TR-104, there may be an intersection of common parameters that provide the same functionality. However, left out of this overlapping set will be the unsupported ones. Unsupported parameters for a CPE are used to provide special or proprietary features. For example, 42networks' DRG supports the Dynamic Resource Allocation Protocol and has the capability to recognize special pulses from the telephony port to indicate the duration of a phone call. These two features are only implemented by 42networks' DRG, making it unique compared to other CPEs. These vendor specific parameters are defined by explicitly attaching the vendor identifier, which can be either the OUI (Organizationally Unique Identifiers) [21] or a domain name (see section 3.4).

3.5.2 Parameters mapping module

The next task is to map the CPE native parameters into CWMP parameters. The first challenge here is that most CPEs today already have their own parameters defined by the vendors. Changing the existing CPE's parameters into CWMP parameters is not feasible. This would require an overhaul of the existing parameters structure; which would lead to increased cost and risk. Even if the parameters were successfully changed, there are the challenges of backward compatibility with older CPEs. In case of DRG for example, if the parameters are changed to CWMP parameters, then all customers must be informed to update their configuration files and a file tool (an application for creating configuration files) must be provided to them. Therefore, mapping parameters is the only feasible way to enable the ACS to manage existing CPEs.

Parameter mapping should be done in a parameter mapping module that maps a CPE's parameter into CWMP compliant parameters within the ACS. The parameter mapping module is located in the ACS and utilizes a table containing the parameter mapping from a CPE to the CWMP parameter used by the ACS and vice versa.

The parameter mapping module works as follows:

- All parameters received from a CPE are non-CWMP parameters; while all parameters received from an ACS are CWMP parameters.
- All parameters coming from CPE shall be mapped into the relevant CWMP parameters and the parameters from the ACS shall be mapped into the CPEs original format (non-CWMP) by the mapping module.
- The ACS should support all capabilities from the registered CPEs. When receiving a request for a certain feature from CPE, the mapping module shall match the parameter to the table of capabilities defined in the ACS. If the capabilities have not been defined in the ACS, then the ACS will return an error code indicating these unsupported capabilities.
- A new mapping table must be created everytime an ACS is configured to support a new type of CPE.
- In order to avoid an incorrect report, the parameter table in the mapping module must be updated every time new CPE parameters are introduced.

The second challenge is caused because each vendor assigned a different name for a parameter although they are basically the same. Below are examples based upon CPEs from some vendors:

- 42networks DRG

```
…
L1ONOFF=ON
L1SIPIP=sip.server.com
L1RTPPORT=8000
L1SIPAUTH=asfd31
L1SIPPASS=****
…
```

- Grandstream

```
…
P31: SIP Registration (0 = do not register, 1 = register)
P34: Authentication Password
P35: SIP User ID
P36: Authenticate ID
P39: Local RTP Port
P40: Local SIP Port
P47: SIP Server (FQDN or IP Address)
…
```

- Linksys Sipura

```
…
Max Redirection
Max Auth
SIP User Agent Name
SIP Server Name
SIP Accept Language
DTMF Relay MIME TYPE
…
```

All of the parameters are assigned flatly. If the value is set, then the features are either enabled or disabled. CWMP uses a different approach when using parameters to configure the CPE. Instead of using parameters to directly configure the CPE, TR-104 defines a parameters group called “VoiceService”. Each VoiceService has a group of parameters describing the CPE’s capability and profiles for service to be assigned to a CPE. The relation between capability and profile is that a profile can be assigned to a CPE if the capability of that profile is set to enable, which means a CPE can sign-up for a SIP profile if the VoiceService supports the SIP capability. As an example of CPE parameters from the different vendors name above, assuming that each CPE has its capability properly defined in VoiceService, the CWMP parameters according to TR-104 would be:

```
…
VoiceService.{i}.VoiceProfile.{i}.SIP.ProxyServer
VoiceService.{i}.VoiceProfile.{i}.SIP.ProxyServerPort
VoiceService.{i}.VoiceProfile.{i}.SIP.InboundAuthUsername
VoiceService.{i}.VoiceProfile.{i}.SIP.InboundAuthPassword
…
```

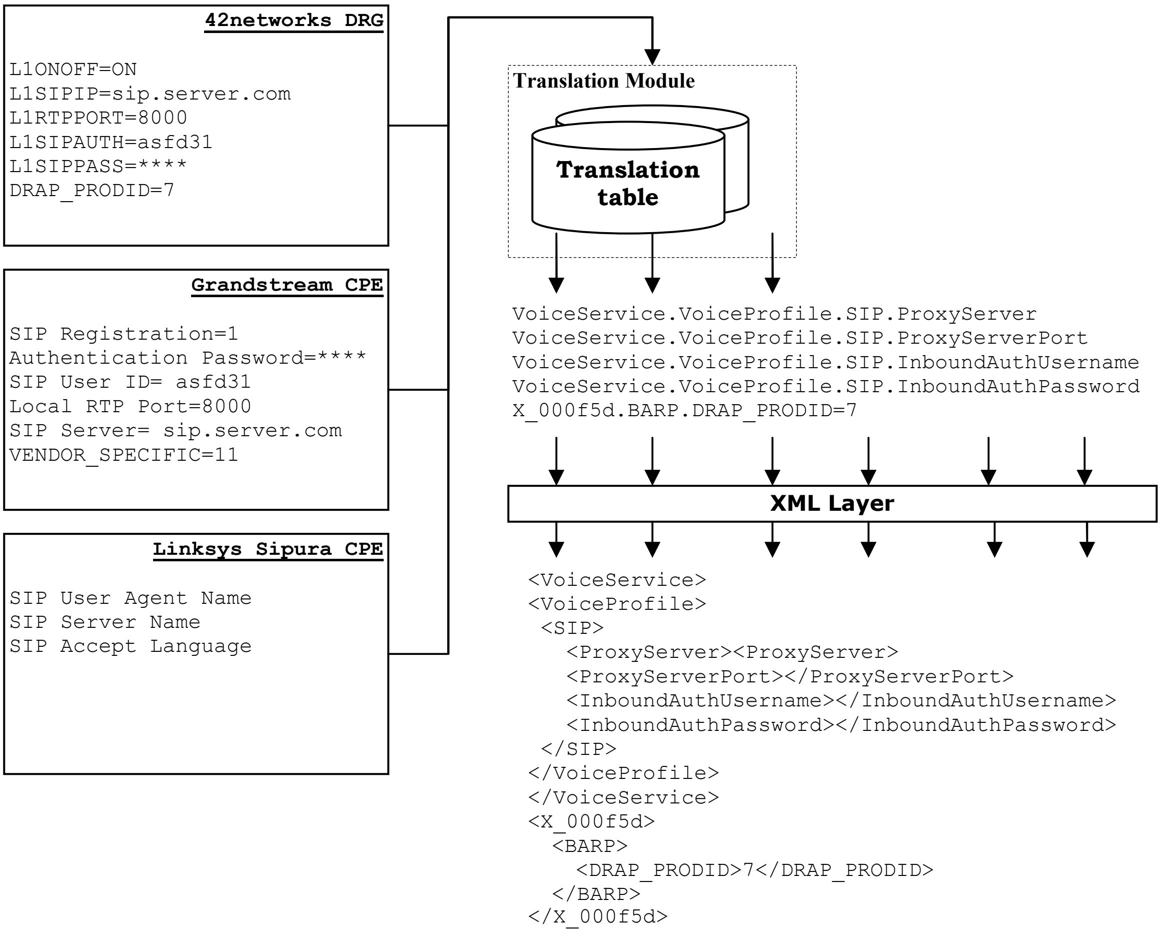
Therefore, the DRGs parameters should be grouped according to its functionality, then match function to either TR-098 or TR-104. When the appropriate group has been matched, we can map the TR-098 or TR-104 into DRG parameters.

There are some parameters that can be mapped directly one to one, for example `VoiceService{i}.VoiceProfiles{i}.Line{i}.Enable` can be directly mapped to `LINE${L}ONOFF` in DRG. Parameters of DRG not in TR-098 or TR-104 will be grouped as vendor specific parameters. A vendor specific parameter is written the same way as writing a vendor specific method as in 3.4, which might looks like the following:

```
X_<VENDOR-IDENTIFIER>_[parameter name]
```

An example of a specific parameter in DRG is the High-Availability mode for each phone line; this would be represented as shown below:

```
X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}HAMODE
```



```
L1ONOFF=ON
L1SIPIP=sip.server.com
L1RTPPORT=8000
L1SIPAUTH=asfd31
L1SIPPASS=****
DRAP_PRODID=7
```

```
SIP Registration=1
Authentication Password=****
SIP User ID= asfd31
Local RTP Port=8000
SIP Server= sip.server.com
VENDOR_SPECIFIC=11
```

```
SIP User Agent Name
SIP Server Name
SIP Accept Language
```

```
VoiceService.VoiceProfile.SIP.ProxyServer
VoiceService.VoiceProfile.SIP.ProxyServerPort
VoiceService.VoiceProfile.SIP.InboundAuthUsername
VoiceService.VoiceProfile.SIP.InboundAuthPassword
X_000f5d.BARP.DRAP_PRODID=7
```

```
<VoiceService>
<VoiceProfile>
 <SIP>
   <ProxyServer><ProxyServer>
   <ProxyServerPort></ProxyServerPort>
   <InboundAuthUsername></InboundAuthUsername>
   <InboundAuthPassword></InboundAuthPassword>
 </SIP>
</VoiceProfile>
</VoiceService>
<X_000f5d>
  <BARP>
    <DRAP_PRODID>7</DRAP_PRODID>
  </BARP>
</X_000f5d>
```

Figure 3-6: Parameter translation process

3.5.3 Data Model

The main goal of data model is able to represent current parameters as classified and able to adapt to future modification for example adding new parameters for new features. The previous chapter has explained and described reasons that current CPE cannot directly supports CWMP parameters. In the other hand, the ACS must support CWMP parameters. Therefore, the data model shall be implemented differently in ACS and CPE.

3.5.3.1 Data Model in ACS

The data-model that is implemented in the ACS uses a hierarchy and structure for the parameters that are defined in TR-106. Thus, each CPE must have one root element that contains information whether the device is a gateway or another device behind the gateway. This is followed by one or more common objects. The device must support at least 1 service which is defined in a service object. An exception to this requirement is for a device which is only serving as an internet gateway, in this case the device will only have one root object. Each service is controlled by one or more parameters.

An Internet Gateway Device that also supports the ABCService and XYZService objects, where XYZService is a VoIP service, can be presented in the data model as the following:

```
InternetGatewayDevice
  DeviceSummary
  DeviceInfo
  ManagementServer
  Time
  UserInterface
  Layer3Forwarding
    LANDeviceNumberOfEntries = 1
    LANDevice.1
    WANDeviceNumberOfEntries = 1
    WANDevice.1
    Services
      ABCServiceNumberOfEntries = 1
      ABCService.1
      ABCServiceSpecificObjects
```

These are gateway connections related parameters which are defined in TR-098. If ABCservice is a connection related service, then its parameters shall also be defined in TR-098.

```
      XYZServiceNumberOfEntries = 1
      XYZService.1
      XYZServiceSpecificObjects
```

The device also supports voice communication through the XYZservice VOIP service. Its parameters shall also be defined in TR-104.

Figure 3-7: CPE Services

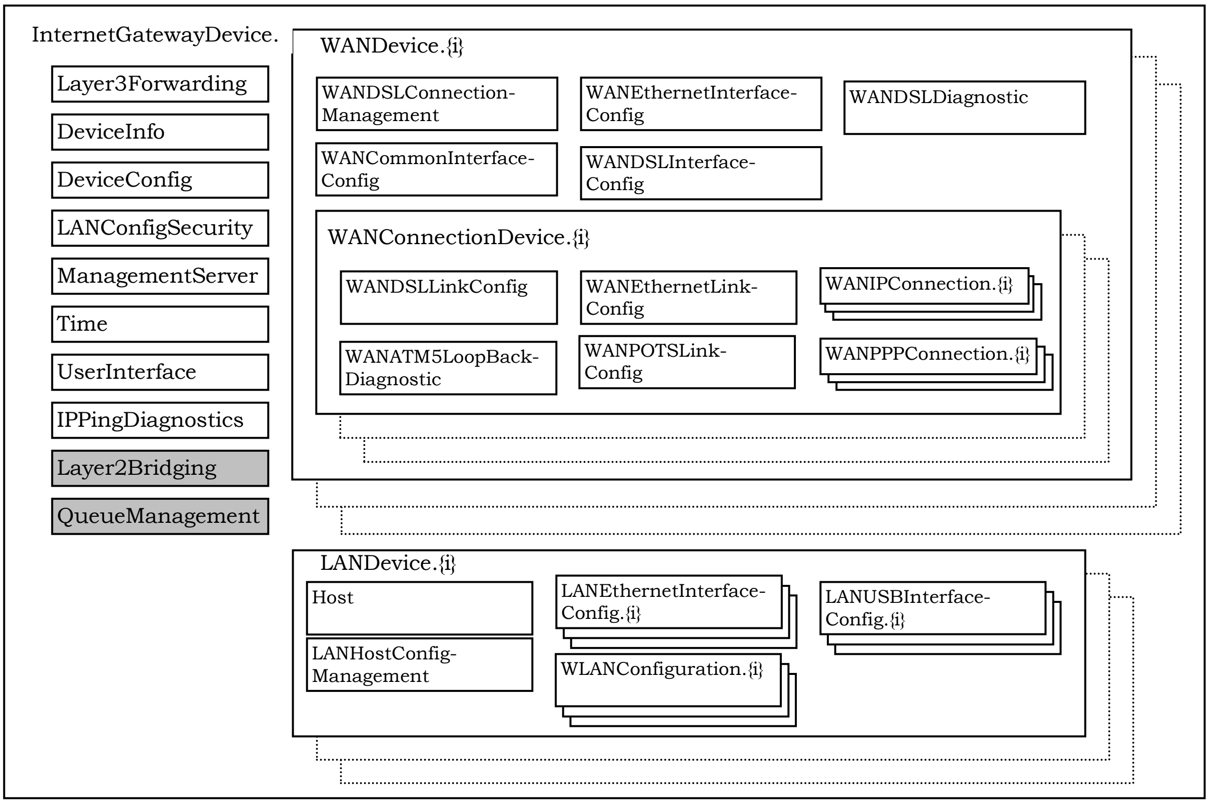
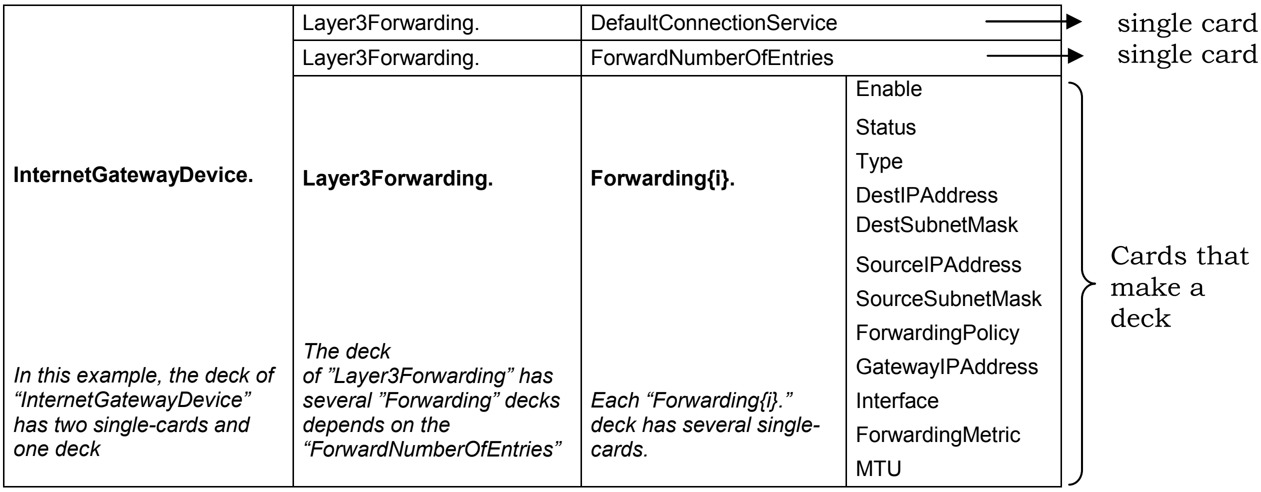


Figure 3-8: TR-098 Data Model [17]

A simple way to understand the structure of the parameter is to look at the parameters organized into a “card-deck”. Every single parameter is a card. If parameters (or cards) that serve similar functions or operate in the same context being put together in a group, then they will make a deck. Decks from a similar context can also be grouped to make a bigger deck. So, the deck consists of cards and/or decks, and every card is a member from a deck. The combination of cards and/or decks that makes a deck with greater scope is the top-level deck “InternetGatewayDevice”.

One service is normally controlled by parameters from one deck. For example, the service of “Layer3Forwarding” is controlled from a deck which contains several cards and decks. As defined in TR-098, the parameters for Layer3Forwarding are shown in Figure 3-9.



	Layer3Forwarding.	DefaultConnectionService	
	Layer3Forwarding.	ForwardNumberOfEntries	
InternetGatewayDevice. *In this example, the deck of "InternetGatewayDevice" has two single-cards and one deck*	**Layer3Forwarding.** *The deck of "Layer3Forwarding" has several "Forwarding" decks depends on the "ForwardNumberOfEntries"*	**Forwarding{i}.** *Each "Forwarding{i}." deck has several single-cards.*	Enable Status Type DestIPAddress DestSubnetMask SourceIPAddress SourceSubnetMask ForwardingPolicy GatewayIPAddress Interface ForwardingMetric MTU

Figure 3-9: A view of Card-Deck

The "card-deck" view also simplifies the profile creation task. Instead of going through the parameters one by one, a profile can be created simply by collecting applicable decks and cards. Of course the deck has to have been previously created. The concept of a profile will be explained further in section 3.6

As in TR-098, the "card-deck" view is applicable as well to the VoiceService parameters in TR-104. Each VoiceService is a deck consisting of decks for capabilities and a profile (see Figure 3-10).

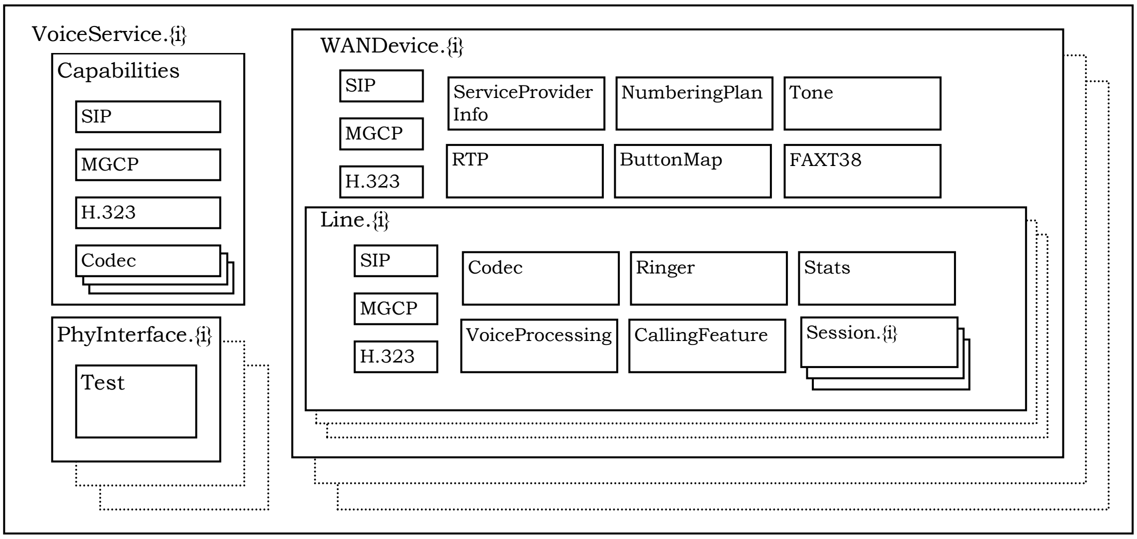


Figure 3-10: TR-104 Data Model [18]

An ACS must support at least the InternetGatewayDevice deck. If an ACS is going to manage a VoIP capable device, it also has to support the VoiceService deck. In addition, the ACS needs to support an additional deck if it is going to manage a device with non-CWMP services and/or a device that does not support CWMP. This deck is a vendor-specific deck and identified by the vendor identifier. A vendor identifier is an officially assigned OUI as explained in 3.4.

3.5.3.2 Data Model in DRG

Since the DRG does not support CWMP parameters, it will not use the "card-deck" for its data model. DRG will use the current data-model. The data model of the current DRG is described in appendix A.3. In order to map the parameters in the DRG to the CWMP parameters, the parameters in the DRG will be divided into a group of “Network” and “VoIP” parameters as in TR-098 and TR-104.

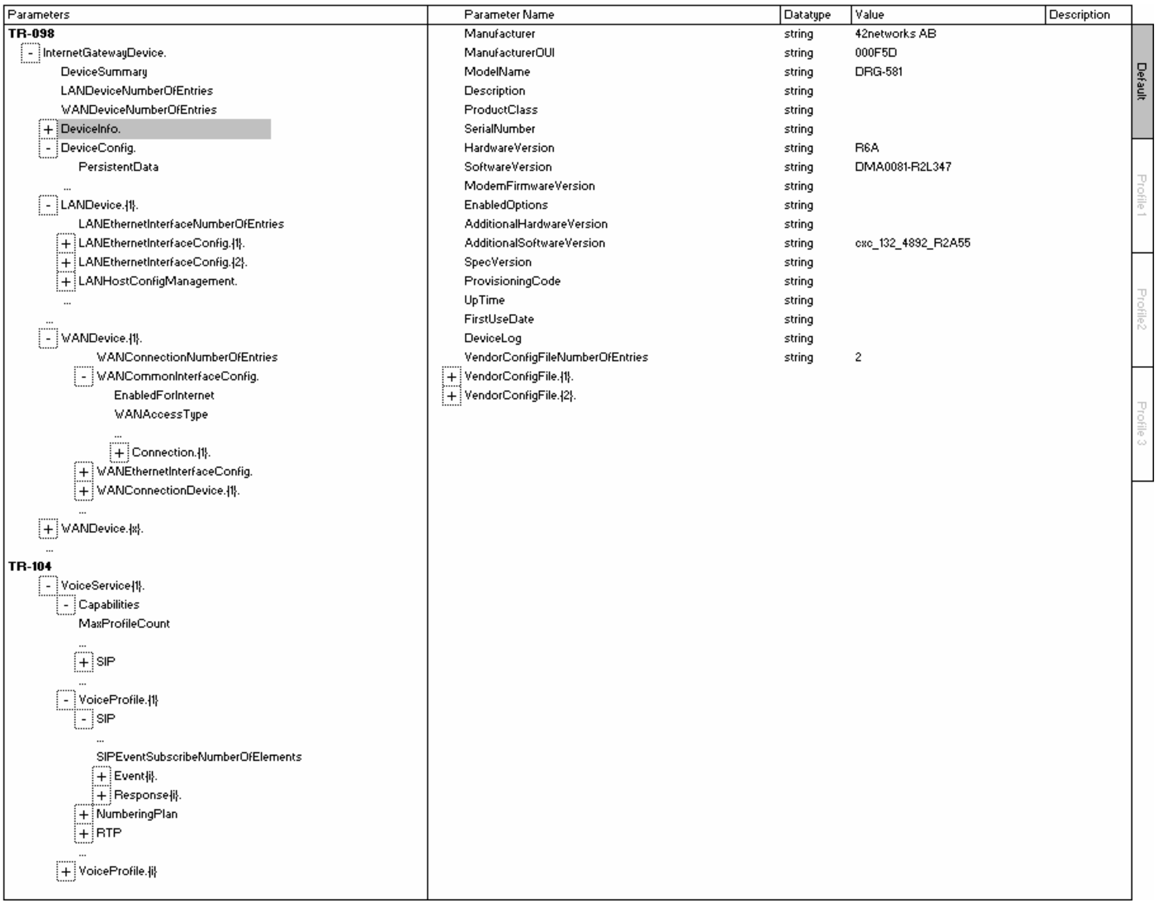
3.6 Service Profile

The service profile is an object consisting parameters that defines one or more services that the ACS is capable of handling and this profile will be assigned to the CPE based upon its subscription. Profiles are created differently depending on the capabilities of the CPE and the complexity of the service. Normally, there is one basic profile and several customized profiles. An ACS can have more than one profile, but only one profile is assigned to a CPE or a group of CPEs, which will be marked as an active profile.

Based on the “card-deck” concept, a profile is basically a deck, which is created by collecting relevant decks and cards. For example, a profile with a VoIP capable CPE with an Ethernet connection for WAN and a LAN port will get the deck from TR-098’s basic profiles: EthernetWAN and EthernetLAN, and a deck from related TR-104’s VoiceService.

A profile can be created in two ways:

- By selecting relevant parameters available in the ACS and grouping them into an object. The ACS supports all TR-098 and TR-104 parameters which are displayed in a tree-view. TR-098 parameters shall be in the network category and TR-104 in the VoIP category. The relevant parameters are selected in a group and saved into a profile. Figure 3-11 shows the default profile is currently selected. A new value can be assigned to each parameter. When the administrator has finished, then all these changes can be saved into the current profile or a new profile.
- Loading a profile file into the ACS.
  A profile can also be created by uploading a profile template file (encoded as an XML file). Every element in the XML file shall be written using exactly the same parameter name as given in TR-098 or TR-104, and following the same structure.



```
Parameters
TR-098
 [-] InternetGatewayDevice.
       DeviceSummary
       LANDeviceNumberOfEntries
       WANDeviceNumberOfEntries
     [+] DeviceInfo.
     [-] DeviceConfig.
           PersistentData
        ...
     [-] LANDevice.{1}.
           LANEthernetInterfaceNumberOfEntries
         [+] LANEthernetInterfaceConfig.{1}.
         [+] LANEthernetInterfaceConfig.{2}.
         [+] LANHostConfigManagement.
        ...
      ...
     [-] WANDevice.{1}.
               WANConnectionNumberOfEntries
             [-] WANCommonInterfaceConfig.
                   EnabledForInternet
                   WANAccessType

                   ...
                 [+] Connection.{1}.
             [+] WANEthernetInterfaceConfig.
             [+] WANConnectionDevice.{1}.
              ...
     [+] WANDevice.{x}.
      ...
TR-104
     [-] VoiceService{1}.
         [-] Capabilities
           MaxProfileCount
              ...
             [+] SIP
              ...
         [-] VoiceProfile.{1}
             [-] SIP
                  ...
               SIPEventSubscribeNumberOfElements
                 [+] Event{i}.
                 [+] Response{i}.
             [+] NumberingPlan
             [+] RTP
              ...
         [+] VoiceProfile.{i}
```

Parameter Name	Datatype	Value	Description
Manufacturer	string	42networks AB	
ManufacturerOUI	string	000F5D	
ModelName	string	DRG-581	
Description	string		
ProductClass	string		
SerialNumber	string		
HardwareVersion	string	R6A	
SoftwareVersion	string	DMA0081-R2L347	
ModemFirmwareVersion	string		
EnabledOptions	string		
AdditionalHardwareVersion	string		
AdditionalSoftwareVersion	string	cxc_132_4892_R2A55	
SpecVersion	string		
ProvisioningCode	string		
UpTime	string		
FirstUseDate	string		
DeviceLog	string		
VendorConfigFileNumberOfEntries	string	2	
[+] VendorConfigFile.{1}.			
[+] VendorConfigFile.{2}.			

Figure 3-11: Creating profile by selecting relevant parameters

Below is the example of how to make a profile from parameter:

```
Parameters:
InternetGatewayDevice.DeviceSummary
InternetGatewayDevice.LANDeviceNumberOfEntries
InternetGatewayDevice.WANDeviceNumberOfEntries
InternetGatewayDevice.DeviceInfo.ModelName
InternetGatewayDevice.DeviceInfo.Description
InternetGatewayDevice.DeviceInfo.ProductClass
InternetGatewayDevice.DeviceInfo.SerialNumber
InternetGatewayDevice.DeviceInfo.HardwareVersion
InternetGatewayDevice.DeviceInfo.SoftwareVersion
InternetGatewayDevice.DeviceInfo.VendorConfigFileNumberOfEntries
InternetGatewayDevice.DeviceInfo.VendorConfigFile.{i}.Name
InternetGatewayDevice.DeviceInfo.VendorConfigFile.{i}.Version
InternetGatewayDevice.DeviceInfo.VendorConfigFile.{i}.Date
InternetGatewayDevice.DeviceInfo.VendorConfigFile.{i}.Description
InternetGatewayDevice.VoiceService{i}.Capabilities.MaxProfileCount
InternetGatewayDevice.VoiceService{i}.Capabilities.MaxLineCount
InternetGatewayDevice.VoiceService{i}.Capabilities.SignalingProtocols
InternetGatewayDevice.VoiceService{i}.Capabilities.SIP.Role
InternetGatewayDevice.VoiceService{i}.Capabilities.SIP.Extensions
InternetGatewayDevice.VoiceService{i}.Capabilities.SIP.Transports
InternetGatewayDevice.VoiceService{i}.Capabilities.SIP.URISchemes
InternetGatewayDevice.VoiceService{i}.VoiceProfiles{i}.Enable
InternetGatewayDevice.VoiceService{i}.VoiceProfiles{i}.Reset
InternetGatewayDevice.VoiceService{i}.VoiceProfiles{i}.NumberOfLines
InternetGatewayDevice.VoiceService{i}.VoiceProfiles{i}.Name
InternetGatewayDevice.VoiceService{i}.VoiceProfiles{i}.SignalingProtocol
```

Card-deck Presentation:

Bold line and font indicate deck. Normal line and font indicate card. Dotted line and font indicate multiple decks.

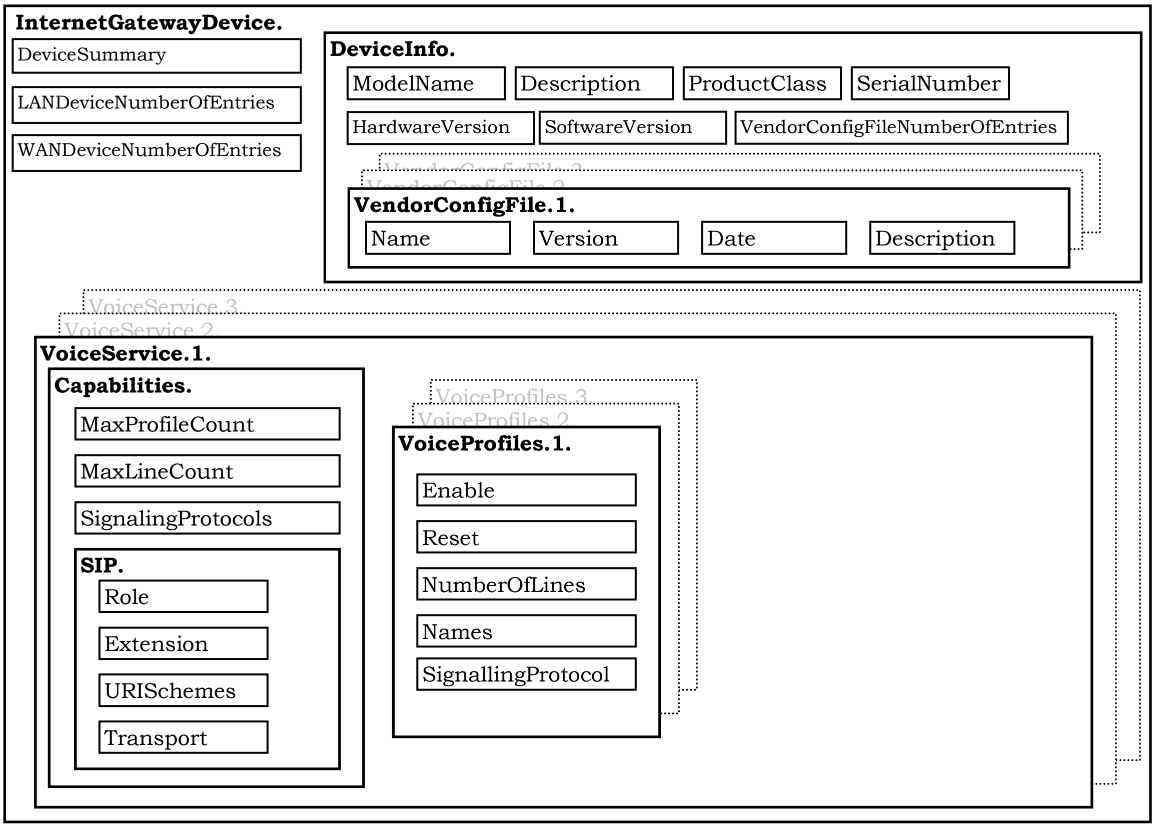


Figure 3-12: Parameters in Deck View

Profile encoded in XML:

```
<InternetGatewayDevice>
    <DeviceSummary></DeviceSummary>
    <LANDeviceNumberOfEntries></LANDeviceNumberOfEntries>
    <WANDeviceNumberOfEntries></WANDeviceNumberOfEntries>
    <DeviceInfo>
        <ModelName></ModelName>
        <Description></Description>
        <SerialNumber></SerialNumber>
        <HardwareVersion></HardwareVersion>
        <SoftwareVersion></SoftwareVersion>
        <VendorConfigFileNumberOfEntries></VendorConfigFileNumberOfEntries>
        <VendorConfigFile index=1>
            <Name></Name>
            <Version></Version>
            <Date></Date>
            <Description></Decsription>
        </VendorConfigFile>
    </DeviceInfo>
    <VoiceService index=1>
        <Capabilities>
            <MaxProfileCount></MaxProfileCount>
            <MaxLineCount></MaxLineCount>
            <SIP>
                <Role></Role>
                <Extensions></Extensions>
                <Transports></Transports>
                <URISchemes></URISchemes>
            </SIP>
        </Capabilities>
        <VoiceProfile index=1>
            <Enable></Enable>
            <Reset></Reset>
            <NumberOfLines></NumberOfLines>
            <Name></Name>
            <SignalingProtocol></SignalingProtocol>
        </VoiceProfile>
    </VoiceService>
</InternetGatewayDevice>
```

4 Implementation, Evaluation, and Analysis

The design as explained in chapter 3 represents the guidelines for implementing CWMP based CPE management. For the CPE, which in this case is the DRG, the implementation has to be able to communicate with the ACS in a CWMP compliant manner. Because the ACS is a new product, we do not have to be concerned about reverse-compatibility and can define the ACS to be CWMP compliant from the start. This chapter will first explain the CWMP implementation in the DRG, then the development of a CWMP compliant ACS.

4.1 DRG side implementation

Due to the limitation of available internal resources, the implementation in the DRG has focused on discovering and communicating with the ACS for service configuration and firmware loading. The service configuration shall be done by setting relevant parameters using "SETPARAMETERSVALUE" method and the firmware loading shall be done using "DOWNLOAD" method. Both the "SETPARAMETERSVALUE" and the "DOWNLOAD" are RPC method which is explained in appendix A.4.DRG uses its own parameters instead of TR-098 and/or TR-104 parameters. When communicating with the ACS, the DRG's parameters are translated into TR-098 and/or TR-104 parameters by the translation module in the ACS.

4.1.1 ACS discovery

The first step in the implementation of the DRG side is informing the DRG of the location of the ACS. The most intuitive way is to put the ACSURL in the DHCP option 43. This option is contained in response to the DHCP REQUEST when a DRG requests an IP address or renews the lease. For example, the ACSURL in DHCP Option 43 might look as follows:

```
Option Vendor-Encapsulated "ACSURL=sample.acs.com"
```

When the DRG receives the ACS address, it shall store it internally and use it as the destination for INFORM messages. These INFORM message shall be sent per-event as explained previously in section 3.4. When the ACS learns of the existence of a DRG, it will initially place the DRG in a list of unmanaged units. The subsequent processing to make this DRG a managed unit will be done later.

4.1.2 Sending a message to the ACS

The communication between the DRG and the ACS shall be based on events. As soon as the DRG learns the location of the ACS, it will send an INFORM message to inform the ACS. An example of such an INFORM message is shown in Figure 4-1.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
    <Request>Inform</Request>
    <Parameter>
    <Device>
        <DeviceInfo>
            <DeviceSummary>DRG-22,Network:DefaultEnable:1.0</DeviceSummary >
            <SpecInfo>1.0</SpecInfo>
            <HardwareVersion>R9A</HardwareVersion>
            <SoftwareVersion>DMA0027-R2K161</SoftwareVersion>
            <ProvisioningCode>XZCO:GRP2</ProvisioningCode>
        </DeviceInfo>
        <ManagementServer>
            <ConnectionRequestURL>http://acs.local</ConnectionRequestURL>
            <ParameterKey>zHbfgjk</ParameterKey>
        </ManagementServer>
        <WANInterface>
            <ExternalIPAddress>80.193.75.3</ExternalIPAddress>
        </WANInterface>
    </Device>
    <X_000f5d>
        <DeviceInfo>
            <MAC>000f5dfe7d77</MAC>
            <DowloaderVersion>cxc-1492-4888-R2A47</DowloaderVersion>
            <LastSuccessInform>2008-11-05 10:20:23</LastSuccessInform>
            <Event>
                <EventCode>0 BOOTSTRAP</EventCode>
            </Event>
        </DeviceInfo>
    </X_000f5d>
    </Parameter>
</soap:Body>
</soap:Envelope>
```

Figure 4-1: An INFORM message sent by the DRG

Up to this point, the DRG and ACS have not yet communicated each other, thus the DRG shall keep sending an INFORM message to the ACS, until the ACS registers the DRG in a list of managed units. If a DRG has been registered in the list of managed units, then it will periodically sends an INFORM message to the ACS to keep the ACS informed that it is still alive. These periodic INFORM messages use event code 2. By default, the periodic INFORM interval is set to 500 seconds.

Besides the INFORM messages, the DRG also communicates with the ACS to execute functions and report its status back to ACS. Example of these messages can be seen in Appendix A.4.

4.1.3 Receiving a Request from the ACS

Upon receiving an incoming request from the ACS, the DRG first identifies the ACS and verifies the session ID by checking the cookie. If the ACS is successfully identified, the DRG must respond with an acknowledgement in the next HTTP POST that it sends to the ACS. If the DRG was unable to correctly respond to the request, then the status of that particular DRG in ACS is marked as “*pending*”.

4.2 ACS side implementation

The 42networks’ ACS has the commercial name Home Device Directory (HDD). However, this report shall continue to refer to it as ACS for consistency. The screen-shots in this chapter are the screen captures from the HDD from different operations.

As described in the TR-069 specification, the functions of the ACS include:

- Auto-configuration and dynamic service provisioning
- Software/firmware image management
- Status and performance monitoring
- Diagnostics

As of today, the current ACS implementation only covers the first three functions above. In terms of corporate priorities, the first two features were the main target for the ACS. As these represent the three most common functions which a service provider requires, it is hoped that this initial implementation will suffice - until the next version which will support diagnostics is commercially released.

4.2.1 ACS Specification

The ACS is implemented as a web application. The implementation consists of a database and an application server, a web-based management Graphical User Interface (GUI), HTTP-based polling and data access interface to the DRG, and some XML based data communication to integrate with an operator specific customer care system. The customer care system keeps track of user subscriptions and provides the data that is used to configure or re-configure a given CPE for the subscribed services. The over-all structure of the ACS system is shown in Figure 4-2.

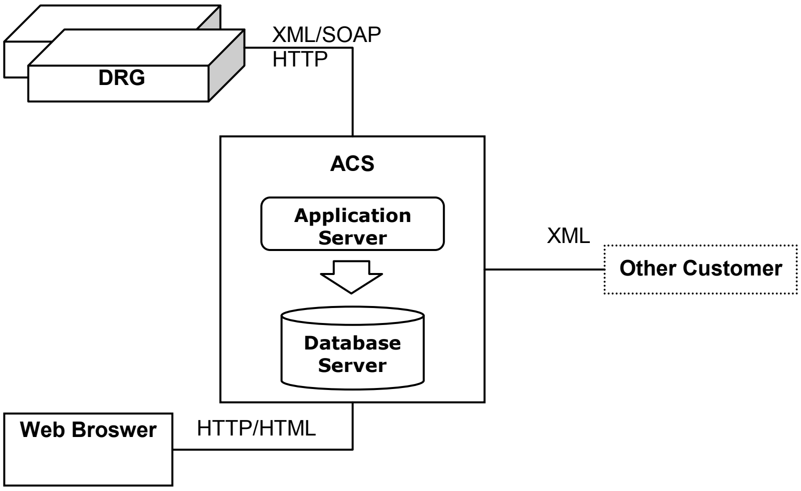
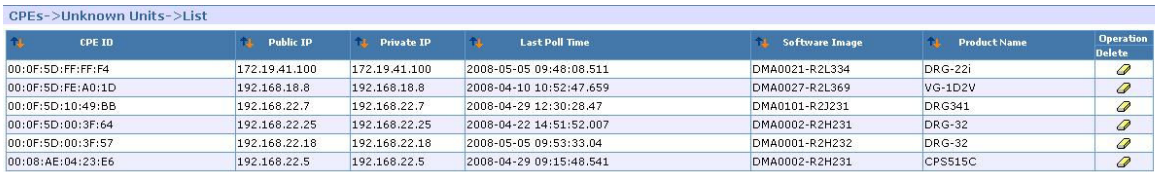


Figure 4-2: ACS Components

The database used is Postgres-8.2.6 and application server is Apache Tomcat 4.1. In order to access the ACS, the (human) CPE management operator's client machine needs a Java Runtime Environment (JRE) 1.4.2_11 (or later) and an internet browser such as Internet Explorer or Firefox.

4.2.2 ACS Operation

ACS operation starts with receiving an initial INFORM message from the DRG that has announced its existence. Initially, the ACS puts the DRG in the "Unknown Units" list when receiving this INFORM message. The "Unknown Units" list in this case is a list of unmanaged units as shown in Figure 4-3.

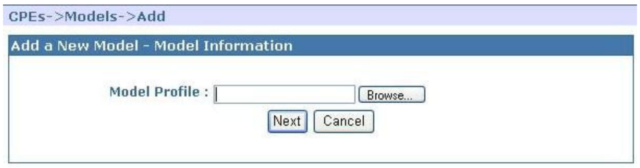


CPEs->Unknown Units->List

CPE ID	Public IP	Private IP	Last Poll Time	Software Image	Product Name	Operation Delete
00:0F:5D:FF:FF:F4	172.19.41.100	172.19.41.100	2008-05-05 09:48:08.511	DMA0021-R2L334	DRG-22i	
00:0F:5D:FE:A0:1D	192.168.18.8	192.168.18.8	2008-04-10 10:52:47.659	DMA0027-R2L369	VG-1D2V	
00:0F:5D:10:49:BB	192.168.22.7	192.168.22.7	2008-04-29 12:30:28.47	DMA0101-R2J231	DRG341	
00:0F:5D:00:3F:64	192.168.22.25	192.168.22.25	2008-04-22 14:51:52.007	DMA0002-R2H231	DRG-32	
00:0F:5D:00:3F:57	192.168.22.18	192.168.22.18	2008-05-05 09:53:33.04	DMA0001-R2H232	DRG-32	
00:08:AE:04:23:E6	192.168.22.5	192.168.22.5	2008-04-29 09:15:48.541	DMA0002-R2H231	CPS515C	

Figure 4-3: The unmanaged CPE List

A DRG that is in the "Unknown-Units" list must be register with the ACS in order to be managed. This registration requires an association of this DRG to an appropriate profile in the ACS which is defined based on the DRG platform (see section 2.4.3. Digital Residential Gateway for a better explanation about the DRG platform). The specific DRG platform is identified by its model. The ACS has to recognize the DRG model in order to manage the DRG (this is because the ACS will have to load the specific translation mapping for this model - as the CPEs do not use CWMP compliant naming). This shall be done by uploading an appropriate profile files (in XML format) into the ACS. The steps required to do this are shown in Figure 4-4, Figure 4-5, and Figure 4-6.

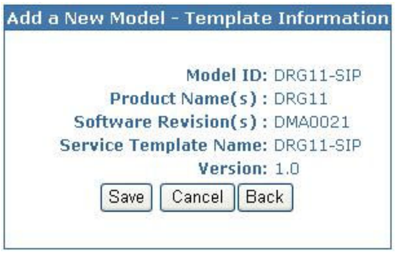


CPEs->Models->Add

Add a New Model - Model Information

Model Profile :

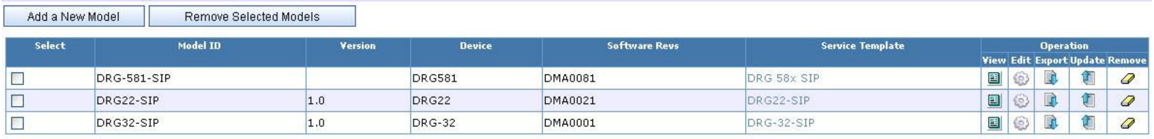
Figure 4-4: Adding a CPE Model



Add a New Model - Template Information

Model ID: DRG11-SIP
Product Name(s) : DRG11
Software Revision(s) : DMA0021
Service Template Name: DRG11-SIP
Version: 1.0

Figure 4-5: Adding a CPE Model

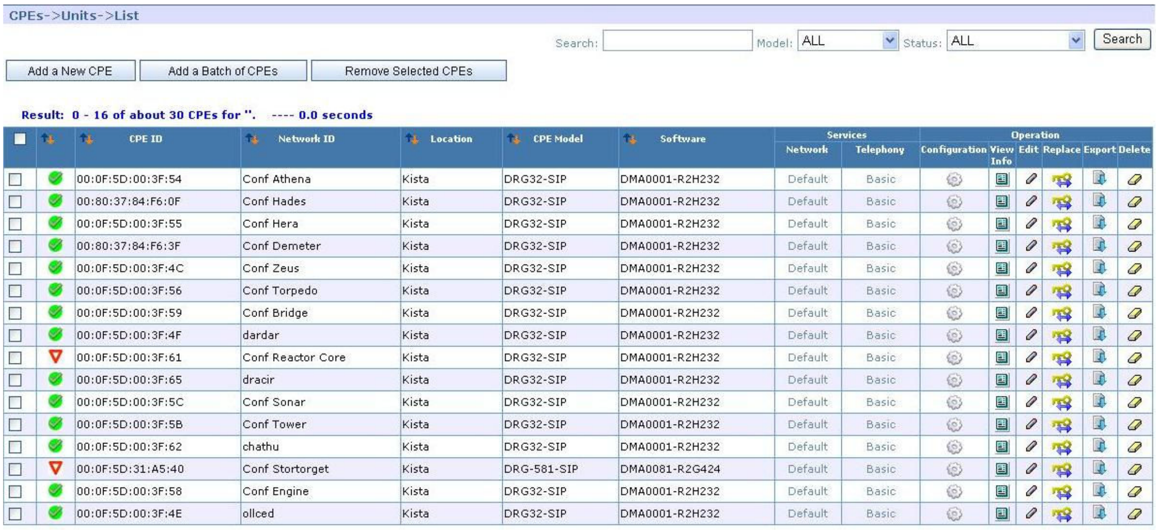


Select	Model ID	Version	Device	Software Revs	Service Template	Operation				
						View	Edit	Export	Update	Remove
□	DRG-581-SIP		DRG581	DMA0081	DRG 58x SIP					
□	DRG22-SIP	1.0	DRG22	DMA0021	DRG22-SIP					
□	DRG32-SIP	1.0	DRG-32	DMA0001	DRG-32-SIP					

Figure 4-6: Adding a CPE Model

4.2.2.1 Assigning Profile and Managing Services

Each platform has a basic profile which is then customized into several service specific profiles. These service specific profiles are sent to the customers who may further customize the profile, for example to indicate what services to configure. During the registration, each DRG is associated with an appropriate profile and is able to use the services listed in the profile (the profile assigned is based upon the subscriber's specific subscription). A list of managed CPEs and their service subscriptions is shown in Figure 4-7.



CPEs->Units->List

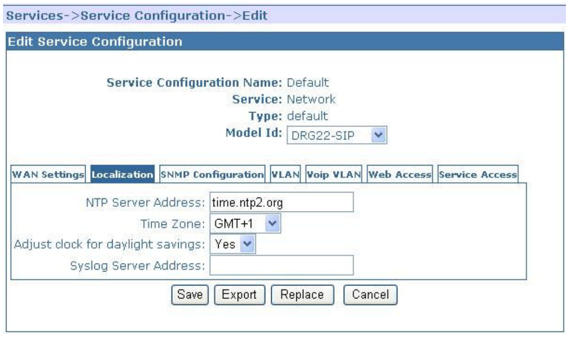
Search: Model: ALL Status: ALL

Result: 0 - 16 of about 30 CPEs for ''. ---- 0.0 seconds

☐		CPE ID	Network ID	Location	CPE Model	Software	Services		Operation					
							Network	Telephony	Configuration	View Info	Edit	Replace	Export	Delete
☐	✓	00:0F:5D:00:3F:54	Conf Athena	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:80:37:84:F6:0F	Conf Hades	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:55	Conf Hera	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:80:37:84:F6:3F	Conf Demeter	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:4C	Conf Zeus	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:56	Conf Torpedo	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:59	Conf Bridge	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:4F	dardar	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	▽	00:0F:5D:00:3F:61	Conf Reactor Core	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:65	dracir	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:5C	Conf Sonar	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:5B	Conf Tower	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:62	chathu	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	▽	00:0F:5D:31:A5:40	Conf Stortorget	Kista	DRG-581-SIP	DMA0081-R2G424	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:58	Conf Engine	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐
☐	✓	00:0F:5D:00:3F:4E	ollced	Kista	DRG32-SIP	DMA0001-R2H232	Default	Basic	⚙	▤	✎	⇄	⇩	✐

Figure 4-7: List of Managed CPE by Service Subscription

Any modifications to the service in the profile will affect all DRGs that have been subscribed to the service with that profile. For example if the time server in "Profile-1" were modified from time.ntp1.org to time.ntp2.org, then this change would propagate to all CPEs that utilized Profile-1. This example of a profile modification is shown in Figure 4-8.



Services->Service Configuration->Edit

Edit Service Configuration

Service Configuration Name: Default
Service: Network
Type: default
Model Id: DRG22-SIP

NTP Server Address: time.ntp2.org
Time Zone: GMT+1
Adjust clock for daylight savings: Yes
Syslog Server Address:

Figure 4-8: Modifying Service

As soon as the modification is confirmed the ACS will set the modified flag to true, and then send the setParameterValue to the relevant DRG(s). Although the parameter of InternetGatewayDevice.Time.NTPServer1 is modified in the ACS, this parameter has to be first sent to the parameter mapping module to translate it into parameters that are known by DRG. Appendix A.6 shows that "NTP1" matches "InternetGatewayDevice.Time.NTPServer1"; therefore, the ACS sends the SetParameterValue for NTP1 to the DRG as shown below.

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<Request>SetParameterValue</Request>
<Parameter>
        NTP1= time.ntp2.org
        …
</Parameter>
```

All DRGs that use the service of a time-server in "Profile-1" will receive this request. When each CPE receives this request, the parameter is changed accordingly and the DRG will respond:

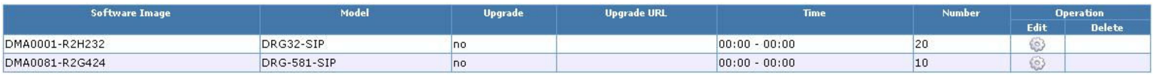
```
POST HTTP/1.1

<Response>SetParameterValue</Response>
<Return>1</Return>
```

The ACS sets the modified flag after receiving all the responses from the relevant DRGs.

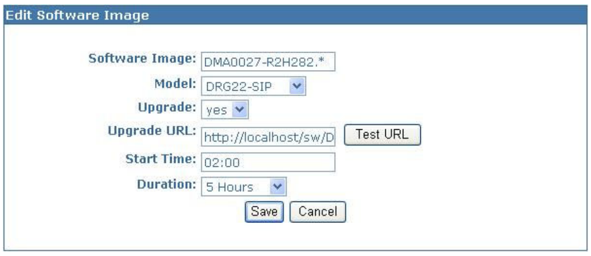
4.2.2.2 Firmware handling

For mass firmware deployment, the ACS determines which units are the deployment targets. These units are normally identified based on the currently installed firmware. The ACS sets the download flag to true for all matching target. Figure 4-9 shows the list of different versions of firmware which are currently deployed. An example of matching the deployed firmware version against a specific version is show in Figure 4-10. One important thing to consider is that loading firmware to DRG will cause the unit to reboot. Therefore, any on going activity will definitely be interrupted. It is highly recommended to upgrade the firmware during the non-busy period, for example at night or at dawn, with prior notification (i.e. phone call, email notification, etc.) to the subscriber. A scheduler function has been created in the ACS to control the actual firmware upgrade action. The task can be defined to take place at a particular day and time when there is less traffic.



Software Image	Model	Upgrade	Upgrade URL	Time	Number	Operation	
						Edit	Delete
DMA0001-R2H232	DRG32-SIP	no		00:00 - 00:00	20	⚙	
DMA0081-R2G424	DRG-581-SIP	no		00:00 - 00:00	10	⚙	

Figure 4-9: List of Firmware



Edit Software Image

Software Image: DMA0027-R2H282.*

Model: DRG22-SIP

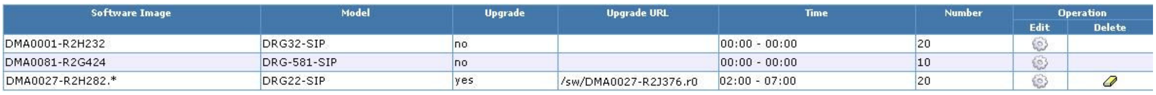
Upgrade: yes

Upgrade URL: http://localhost/sw/D

Start Time: 02:00

Duration: 5 Hours

Figure 4-10: Scheduling Firmware Upgrade



Software Image	Model	Upgrade	Upgrade URL	Time	Number	Operation	
						Edit	Delete
DMA0001-R2H232	DRG32-SIP	no		00:00 - 00:00	20	⚙	
DMA0081-R2G424	DRG-581-SIP	no		00:00 - 00:00	10	⚙	
DMA0027-R2H282.*	DRG22-SIP	yes	/sw/DMA0027-R2J376.r0	02:00 - 07:00	20	⚙	✎

Figure 4-11: List of Firmware with Upgrade Schedule

Figure 4-10 shows the ACS defines a policy that is valid for 5 hours to upgrade DRGs currently installed with DMA0027-R2H282 to DMA0027-R2J376. When a DRG polls during a time that matches the time window (in this case is from 02.00 AM to 5 hours onward) the ACS sends the download command by sending an HTTP POST as the following:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<Request>Download</Request>
<Parameter>
        Filetype=1
        URL=http://192.168.18.10/DMA0028-R2K123.r0
</Parameter>
```

When the download finishes, the DRG will reboot itself and send a response indicating to ACS. The return value will be 1, if download was successful, and other than 1 if the download fails. The example of download response is shown below:

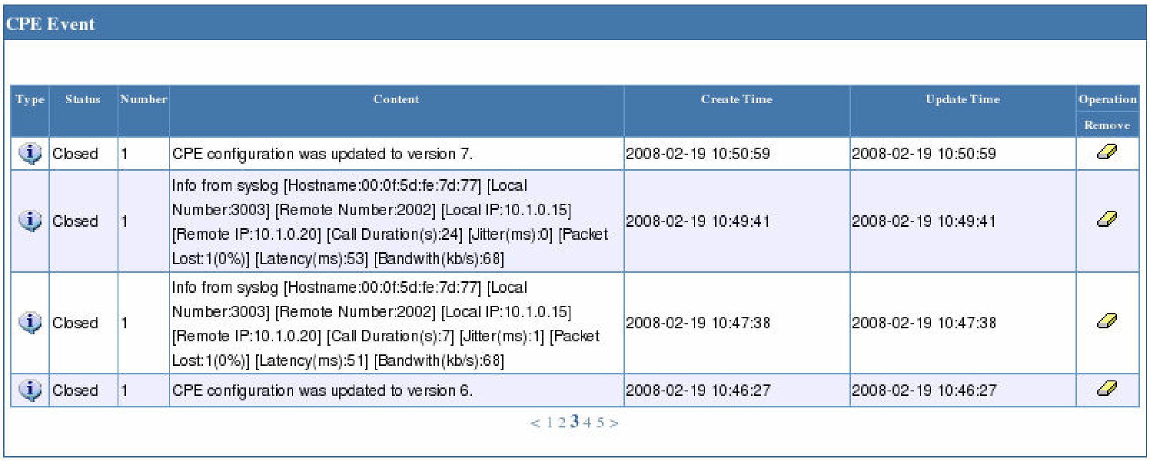
```
POST HTTP/1.1
<Response>Download</Response>
<Return>
        <value>1:Success</value>
</Return>


POST HTTP/1.1
<Response>Download</Response>
<Return>
        <value>9000:File not found</value>
</Return>
```

If, for some reason a DRG does not reboot, then the ACS will not receive any return value. Hence, it does not know the status of the download. If the ACS does not receive a return value from the DRG after firmware download process, then the ACS will immediately resend a REBOOT command to this DRG. After reboot, the DRG sends an INFORM message to ACS with information about the currently loaded firmware. The firmware information can be found in the "SoftwareVersion" under the "DeviceInfo". (see Figure 4-1). If the firmware changes to the newer version, then this means the upgrade has completed successfully. Otherwise, the ACS will resend the DOWNLOAD command to reinitiate the download process if the time-window is still valid.

4.2.2.3 Call Information

The ACS also include feature to present the statistic of every phone call from each DRG. This statistic includes information of the MAC address and the IP address of the DRG, the origin and the destination of the call, the duration of the call (in the unit of second), and call quality related information such jitter, packet-loss, latency and bandwidth.



CPE Event

Type	Status	Number	Content	Create Time	Update Time	Operation
						Remove
ⓘ	Closed	1	CPE configuration was updated to version 7.	2008-02-19 10:50:59	2008-02-19 10:50:59	✎
ⓘ	Closed	1	Info from syslog [Hostname:00:0f:5d:fe:7d:77] [Local Number:3003] [Remote Number:2002] [Local IP:10.1.0.15] [Remote IP:10.1.0.20] [Call Duration(s):24] [Jitter(ms):0] [Packet Lost:1(0%)] [Latency(ms):53] [Bandwith(kb/s):68]	2008-02-19 10:49:41	2008-02-19 10:49:41	✎
ⓘ	Closed	1	Info from syslog [Hostname:00:0f:5d:fe:7d:77] [Local Number:3003] [Remote Number:2002] [Local IP:10.1.0.15] [Remote IP:10.1.0.20] [Call Duration(s):7] [Jitter(ms):1] [Packet Lost:1(0%)] [Latency(ms):51] [Bandwith(kb/s):68]	2008-02-19 10:47:38	2008-02-19 10:47:38	✎
ⓘ	Closed	1	CPE configuration was updated to version 6.	2008-02-19 10:46:27	2008-02-19 10:46:27	✎

Figure 4-12: Call information

4.3 Mapping DRG Parameter to ACS

There is no clear categorization of DRG parameters. However, based on their technical functionalities, parameters of DRG are classified into group: "Network" and "VoIP".

4.3.1 Mapping Networking Parameters

Parameters in "Network" group are those related with network traffic connections including DEVICE-INFO, LAN, ROUTE, WAN, PPPOE, DIFFSERV, PORT, SNMP, ACCESS, TIME, VLAN, PRIORITY QUEUE, and BARP. After the grouping, parameters from both sides were manually matched and linked if they perform the same function. Below are some examples of parameter mappings:

```
DRG                    TR-098
DEVICE-INFO.           InternetGatewayDevice.DeviceInfo.
       PRODSN                 SerialNumber
       PRODNO                 ModelName
       PRODNAME               Description
       PRODREV                HardwareVersion
       CFREF                  ProductClass
       SW                     SoftwareVersion
       PRODWEEK        X_000f5d.InternetGatewayDevice.DeviceInfo.PRODWEEK
… (see Appendix A.6 for a complete mapping)
```

The DRG and TR-098 parameters are used in completely different ways. In the DRG, parameters are loosely related to each other. The only rules were the sections (see appendix A.3) to place the parameter and the datatype into correct section. However, this rule is no longer followed today. On the other hand, TR-098 parameters are defined in a clear structure and tightly related to each other. For example, the number of LANDevice {i} and WANDevice {i} depend upon LANDeviceNumberOfEntries and WANDeviceNumberOfEntries respectively. Fortunately, TR-098 defines a baseline profile for an InternetGatewayDevice. Unfortunately, DRG does not currently support the baseline profile. Thus, it will be necessary to add new parameters according to InternetGatewayDevice profile 1.1. In addition to this, the DRG must also add support for the EthernetWAN and EthernetLAN profiles for both the WAN and LAN ports and a POTS profile for supporting VoIP.

4.3.2 Mapping VoIP Parameters

The groups of VoIP parameters from a DRG are: GENERAL-VoIP, CALL-FEATURE, LINE-CONFIGURATION, ACCOUNT-SUBSCRIPTION, FAX, RINGTONES, and RINGSIGNAL. As in section 4.3.1, the DRG's VoIP parameters also take a different approach than the TR-104 parameters. TR-104 defines the capabilities as based on the profile's creation. However, this is not applicable in the DRG as the parameters are defined flatly, without any relation to the capability. If the parameter and value are defined correctly, then the feature shall work accordingly. Therefore, parameters from DRG shall be first mapped directly to those in VoiceProfile, and consider as a single VoiceProfile. The capabilities shall be defined after the first mapping as the reference for VoiceProfile creation in the future. Below is example of the parameters mappings:

```
DRG                    TR-104
                       VoiceService{i}.VoiceProfiles{i}.SIP.
L${L}SIPPIP                   ProxyServer
L${L}DOMAINNAME               UserAgentDomain
LINE${L}AUTHUSER              AuthUserName
                       X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.
L${L}SIPSIP                   L${L}SIPSIP
L${L}SIPSPORT                 L${L}SIPSPORT
… (see Appendix A.6 for complete mapping)
```

TR-104 allows a CPE to have one or more instances of the VoiceService object. Therefore, it is possible to have several VoiceServices in order to support different VoIP protocols. For example, a DRG58x has two phone lines. It can support SIP, H323, or MGCP. Thus, there should be three sets of VoiceServices – one for each different protocol in the ACS. However, there can be *only* one active VoiceService depending on the protocol being used. There can also be several different active VoiceProfiles for each service provider that offers phone service. As shown in Figure 4-13, each phone line in the DRG58x can be subscribed to service from different service providers.

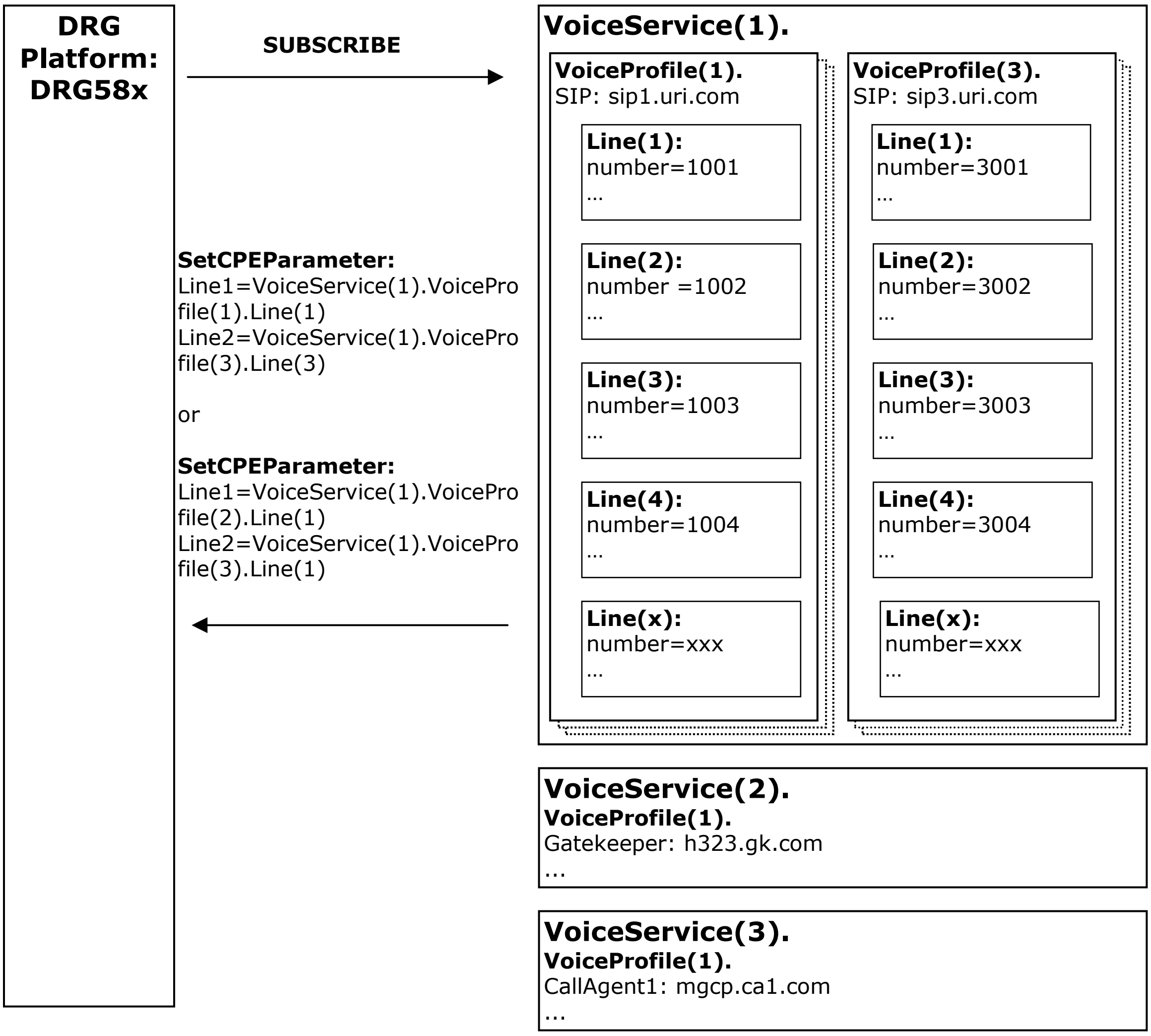


Figure 4-13: Managing VoIP Subscription from ACS

4.4 Analysis of CWMP in the comparison to the SNMP

SNMP is the most commonly used network management system today. Section 2.5.1.2 described its limitations and the reasons that SNMP is not suitable for CPE management. However, SNMP has been and is still being used it to manage CPEs. This section describes a simple scenario where an operator is going to deploy a network that offers services through a CPE. This scenario focuses *only* on service delivery and management. It assumes the network has already been properly setup and that the CPE has been connected properly at the customer's premises. Each customer is offered triple play services – specifically with an internet connection with different maximum data rates, VoIP with different types of features, and IPTV with different sets of channels. The CPE is a typical 42networks' DRG that is use in many triple-play environments today, i.e. the DRG 54x.

The DRG54x has a WAN port, 4 LAN, ports, and 2 telephony ports. The typical connections in a home network are displayed in Figure 4-14.

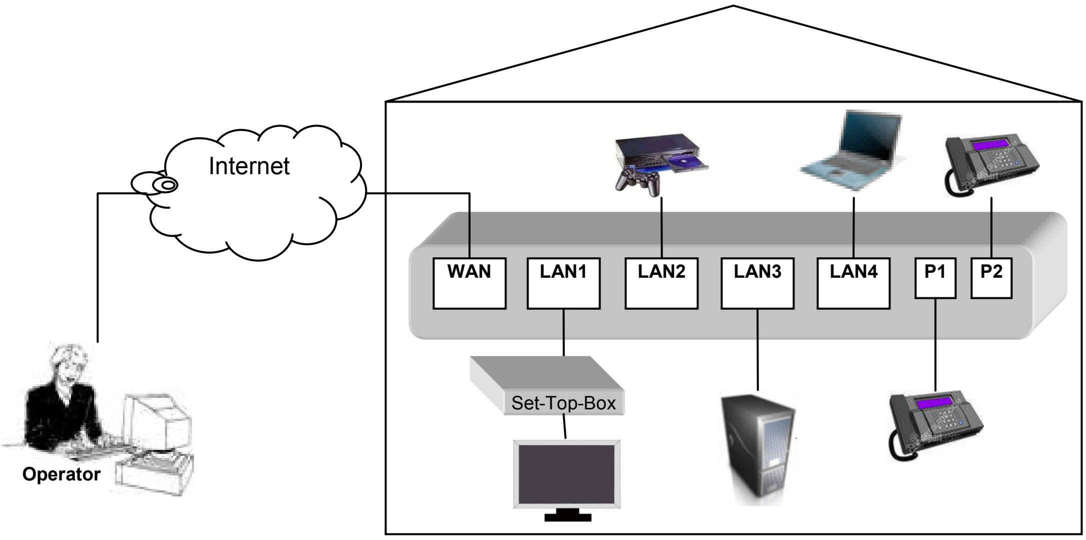


Figure 4-14: Home network with each device connected to a CPE

In this configuration, the service provider's operator has defined 3 types of services for the customer to subscribe to. These three alternatives are Basic, Medium and Advanced (see Table 4-1):

Table 4-1: Example of typical triple-play services

Type of subscription	Internet	IP Telephony	IPTV
Basic	2 Mbps	Basic telephone	Channel 1-8
Medium	8 Mbps	With incoming caller ID	Channel 1-16
Advanced	24 Mbps	With incoming caller ID and Voice mail for 50 messages	Channel 1-50

4.4.1 Solution with SNMP

In this solution, all services that are offered by the operator must be configurable in the DRG through SNMP OIDs. Any modification to the services can causes modification to the OID. There must be 3 groups of OIDs for the 3 types of services above. An SNMP manager uses different groups of OIDs to assign different services to the DRG. For example:

- Basic

```
2 Mbps internet: 1.3.6.1.4.1.18327.67.2.1 integer 2000
enable telephone: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3 integer 1
TV channels:
•  Enable channel 1: 1.3.6.1.4.1.18327.67.2.2.1 integer 1
•  Enable channel 2: 1.3.6.1.4.1.18327.67.2.2.2 integer 1
•  ...
•  Enable channel 8:1.3.6.1.4.1.18327.67.2.2.8 integer 1
```

- Medium

```
8 Mbps internet: 1.3.6.1.4.1.18327.67.2.1 integer 8000
enable telephone: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3 integer 1
incoming caller ID: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3.5 integer 1
TV channels:
•  Enable channel 1: 1.3.6.1.4.1.18327.67.2.2.1 integer 1
•  Enable channel 2: 1.3.6.1.4.1.18327.67.2.2.2 integer 1
•  ...
•  Enable channel 16:1.3.6.1.4.1.18327.67.2.2.16 integer 1
```

- Advance

```
24 Mbps internet: 1.3.6.1.4.1.18327.67.2.1 integer 24000
enable telephone: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3 integer 1
incoming caller ID: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3.5 integer 1
voice mailbox: 1.3.6.1.4.1.18327.67.2.8.5.1.2.3.4 integer 1
TV channels:
  • Enable channel 1: 1.3.6.1.4.1.18327.67.2.2.1 integer 1
  • Enable channel 2: 1.3.6.1.4.1.18327.67.2.2.2 integer 1
  • ...
  • Enable channel 50:1.3.6.1.4.1.18327.67.2.2.50 integer 1
```

4.4.1.1 Assigning a Service using SNMP

Once each service has been defined, it is now time to assign each of them to the DRG. In a general network configuration, any DRG in the network can be discovered and configured by sending SNMP commands to the DRG. In order to automate the service assignment, there will be a set of SNMP scripts for each service assignment. This script is sent to each DRG from an SNMP manager upon receiving an SNMP restart trap from DRG.

One way to simplify the service assignment to each DRG is to locate groups of DRG into different sub-networks based on the service. For example, an operator can divide the network 192.168.0.0/16 into 192.168.1.0/24 for basic service, 192.168.2.0/24 for medium service, and 192.168.3.0/24 for advanced service. So, each subnet shall have set of SNMP scripts related to the service to be assigned. Using this method a DRG only needs to be assigned to the right network address to get the right service. Another advantage of this is that when the service provider needs to modify the service, the service provider can simply ask the network administrator to set the appropriate OID and broadcast it to the relevant subnet. For example, if basic service gets a service upgrade to maximum of 3 Mbps, then the SNMP script is simply updated to “1.3.6.1.4.1.18327.67.2.1 integer 3000” and broadcasted to 192.168.1.255.

The communication between the manager and the CPE is done using a TRAP message based on some event. The SNMP TRAP events are:

- coldStart(0),
- warmStart(1),
- linkDown(2),
- linkUp(3),
- authenticationFailure(4),
- egpNeighborLoss(5),
- enterpriseSpecific(6)

4.4.1.2 Advantages and Disadvantages of using SNMP

This section describes the advantages and disadvantages of using SNMP to manage services. The advantages are:

- SNMP is a well-known method for managing networks today. Hence, it should not be difficult to find an SNMP expert to employ as a manager.
- An SNMP script is straight forward, as it is written in clear text.
- A small scale SNMP manager tool can be found easily and downloaded freely from the internet. The license for a commercial SNMP manager costs around US$400 plus US$50 annual support fee - which is considered to be low cost.

The disadvantages are:

- Although written in clear text and quite understandable, writing an SNMP script can be a troublesome and frustrating job, especially when the script grows larger to cover more services and details.
- The SNMP manager depends on access to the appropriate MIB file. Without an appropriate MIB file, the OID is only presented as dotted digits which are

not easy to understand. If services are modified, then the MIB file must be adapted based upon the modification.

- The SNMP commands are sent in clear text, thus they are vulnerable to interception by any third party. Anyone who manages to receive an SNMP message, for example using a network traffic capture tool, can possibly see information inside the network packet and modify it to enable new services for which the subscriber has not subscribed. The example below is a capture of an SNMP-SET command from a network capture tool.

```
Simple Network Management Protocol
    Version: 1 (0)
    Community: private
    PDU type: SET (3)
    Request Id: 0x00005dac
    Error Status: NO ERROR (0)
    Error Index: 0
    Object      identifier      1:      1.3.6.1.4.1.18327.67.2.8.5.1.10.3
(SNMPv2-SMI::enterprises.18327.67.2.8.5.1.10.3)
    Value: STRING: "192.168.18.15"
```

One way to prevent this is to use a dynamic SNMP community (especially for the write community). However, using a dynamic SNMP community name only minimizes the risk. If someone manages to get the community name, he/she can still try to forge an SNMP packet before the community name changes.

- For the setup, as the example in section 4.4.1.1 where a DRG is assigned to the network based on its service subscription, if a user decides to change his or her subscription, then the DRG needs to be moved to another network location. In a more complex situation where a user would like to have faster internet data rates, but less telephony features, then the service provider has to create a new service assignment script and possibly create a new network segment for this group of users. The conclusion from this case analysis is that SNMP is capable, but not ideal for service management.

4.4.2 Solution with CWMP

A CWMP management system defines a service in terms of a subscription. For example, the services can simply be called basic, medium, and advance service. Each service consists of parameters controlling the offered features:

- Basic

```
Internet        : THRUPUTBITMAX=2000
Telephony       : Line1_status=enable
TV              : ch1_status=enable
                  ch2_status=enable
                  …
                  ch8_status=enable
```

- Medium

```
Internet        : THRUPUTBITMAX=8000
Telephony       : Line1_status=enable
                  Line1_CID=enable
TV              : ch1_status=enable
                  ch2_status=enable
                  …
                  ch16_status=enable
```

- Advanced

```
Internet        : THRUPUTBITMAX=24000
Telephony       : Line1_status=enable
                  Line1_CID=enable
                  Line1_VoiceMail=enable
TV              : ch1_status=enable
                  ch2_status=enable
                  …
                  ch50_status=enable
```

4.4.2.1 Assigning a Service using CWMP

In CWMP, a service is defined as a subscription. Each service available for subscription shall be defined in the ACS as profile. When a DRG boots up, it sends a BOOTSTRAP message to the ACS. The DRG uses the mechanism described in section 4.1.1 to reach the ACS. The DRG is placed in the unmanaged list because there is no service assigned. This is usually the case for a DRG installed in a new or empty apartment. If someone moves into the apartment, then the DRG will be subscribed to a service selected by the tenant. Following this, the ACS sends the related parameters to the DRG based on its subscription. Those parameters will be sent to each DRG in a HTTP packet as described in section 4.2.2.1.

For example, a DRG that is subscribed to the basic service will get a basic profile consisting of the basic service parameters as shown below:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<Request>SetParameterValue</Request>
<Parameter>
        THRUPUTBITMAX=2000
        Line1_status=enable
        ch1_status=enable
        ch2_status=enable
        …
        ch8_status=enable
</Parameter>
```

If one of the features in the service is modified, for example the internet basic service is upgraded to 3Mbps, then the basic service profile will be modified in the ACS and sent to each relevant DRG.

4.4.2.2 Advantages and Disadvantages of using CWMP

This section describes the advantages and disadvantages of using CWMP to manage services. The advantages are:

- CWMP is designed for service management. There is no IP address or other network address restrictions because the management is purely management of the service. A CPE can be placed anywhere in the network. As long as the CPE is able to communicate using HTTP, it should be able to subscribe to a service managed by an ACS.
- CWMP is less affected by a NAT-ed environment.
- When SSL/TLS is implemented, CWMP has better security that offered by SNMP (v1 and v2).

The disadvantages are:

- CWMP is still a fairly new protocol. An ACS is not currently available for free download, even for managing a simple service. In most cases, the vendors build their own ACS to manage their own CPEs or their business allies' CPEs. There was an open source project developing a light-weight ACS [36]. But this project was stop halfway and has been idled for almost a year since November 2007 (according to the development blog).
- Not all standard services as defined in TR-098 or TR-104 are compatible with the native format in existing CPEs. For the existing CPEs, support for CWMP parameters needs to be implemented. There is a parameter translation module, but if every vendor relies on the translation module, there will be a lot of proprietary sets of parameters that at the end of the day, ACS might not be able to handle all of them.

4.4.3 Analysis and Comparison

This section presents the analysis and comparisons of SNMP and CWMP based on the research and test using the DRG.

4.4.3.1 Protocol flexibility

A potential problem in CWMP is that the standard (TR-069) is “too-strict”. Unlike other standards that are mainly concern with communication mechanisms and “how-to-do things”, TR-069 also defines “what-to-use” in the protocol stack due to the inclusion of SOAP1.1 and SSL. Those who want to be fully compatible with CWMP standard will have to use SOAP1.1 for data exchange and SSL for security. This can be a problem because any improvement to CWMP will be also restricted.

On the other hand, the SNMP standard does not define what tools to use for the implementation. Vendors are free to decide which tool to use or to develop their own SNMP application. This gives the protocol greater flexibility because a vendor is not tied to a tool.

4.4.3.2 Operating through NAT

SNMP can only manage their CPEs when they are in the closed networks (when no NAT is present). Using SNMP in a NAT-ed environment is not possible because SNMP does not have any NAT traversal capability. One can argue that it would be possible to use STUN [25] to traverse the NAT. In reality, STUN is only used for discovering the external interface for sending traffic to the public network. Unfortunately, it does not keep the NAT binding open. In the addition, STUN cannot be used in several conditions, such as in symmetrical NAT. There are more limitations which can be found in section 14 of RFC3489 [25]. On the other hand, CWMP has NAT traversal capability because the CPE sends an INFORM message and keeps the NAT binding open. Therefore, the ACS can perform provisioning and management while the NAT binding is open. This is perhaps the only function in CWMP that SNMP can not perform.

4.4.3.3 Cost of implementation

In a CPE which has limited amount of resources and low processing power, the decision of whether to support SNMP or CWMP depends significantly upon the cost of implementation. Based on the studies done by 42networks, the CWMP implementation in a CPE seems to cost more than the SNMP implementation.

An SNMP implementation in the DRG consists of an SNMP agent, MIB database, and MIB handler which requires approximately 80 kilobytes. This 80 kilobytes can be broken down into 20 kilobytes for the SNMP agent, 50 kilobytes for the MIB database, and 10 kilobytes for the MIB handler. No special parser is required because the SNMP message is in a plain-text format.

A complete implementation of CWMP in the other hand consists of an HTTP-client, SOAP parser, and SSL stack which together require approximately 1 megabyte. The SSL stack takes about 100 kilobytes, the SOAP parser is about 200-600 kilobytes, and the rest is for the HTTP-client. The requirement for the SOAP parser may not be a problem for some CPEs which already support an XML parser in their web interface application. However, for those CPEs that do not support an XML parser today, CWMP will have a high cost of implementation, in term of memory resource.

The amount of the required resource will increase when the CPE is required to support CWMP parameters. A rough estimate for supporting basic profile in TR-098 and TR-104 is about 300 kilobytes.

4.4.3.4 Processing Power

The SNMP requires less processing power than CWMP for two reasons. The first reason is because SNMP message uses plain-text. Thus it does not need any parser because all applications recognize and processed the plain-text with a very small processing power. The CWMP message on the other hand is encoded in SOAP. A CPE will require more processing power to first parse the SOAP message, before executing the actual action.

The second reason is because SNMP message exchange between the agent and manager is a one-to-one communication. The SNMP agent sends one message at a time and the manager replies with one message. There is neither session control nor a requirement to keep a connection open during the message exchange. Unlike the SNMP, the CWMP message exchange needs session control and keeps the TCP connection open during the entire transaction. This obviously requires the CPE to expend more processing power for the CWMP

4.4.3.5 Traffic in the network

Referring to the example of the DRG service in section 4.4, there are approximately 120 parameters that need to be configured to subscribe a single DRG to the basic service. This means 120 OIDs in case of SNMP. In order to set 120 OIDs, the SNMP manager has to send 120 SNMP-SET commands. If the average size of one SNMP packet is 110 bytes (42 bytes header and the rest is payload), the amount of network traffic to assign basic service is two times (including the response from the DRG) 110 bytes multiply by 120 OID which is equal to 26,400 bytes (26.4 kbps).

In comparison with the CWMP, one parameter and the value is roughly 30 bytes, each packet including HTTP and SOAP header is about 600 bytes. Unlike SNMP, CWMP configures all the parameters of the DRG at once using a SetParameterValue command. The amount of data sent over the network from the ACS to the DRG is 30 bytes times 120, plus 800 bytes which equals to 4,400 bytes. Upon receiving the parameters, the DRG replies with a SetParameterValueResponse with a packet size of about 480 bytes. Total traffic for both directions is 4,400 plus 480, equals to 4,880 bytes (4.88 kbps).

Table 4-2: Network traffic when subscribing to Basic Service

Basic Service			
Number of CPE	Number of Parameters	Total bytes (SNMP)	Total bytes (CWMP)
1	120	26400	4880
2	240	52800	8480
3	360	79200	12080
4	480	105600	15680
5	600	132000	19280
6	720	158400	22880
7	840	184800	26480
8	960	211200	30080
9	1080	237600	33680
10	1200	264000	37280

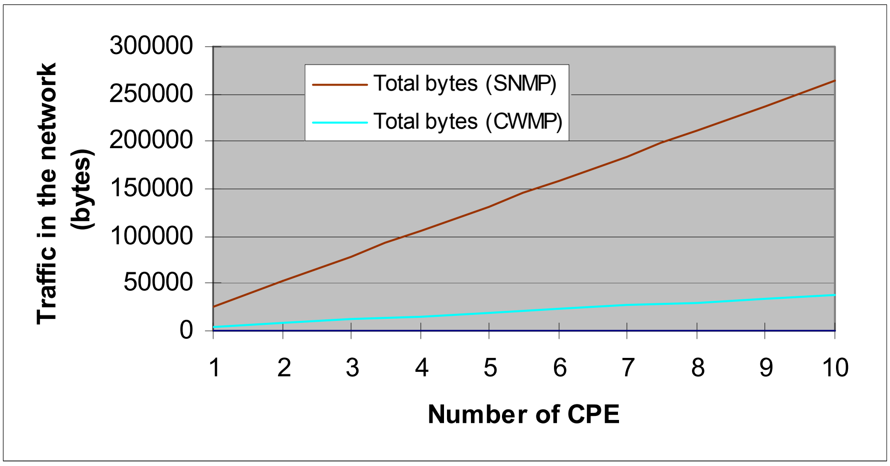


Figure 4-15: Chart of network traffic when subscribing to Basic Service

The simple comparison between SNMP and CWMP above shows that CWMP generates less network traffic when only considered service assignment. While 26.4 kbps and 4.88 kbps may not be a burden to the subscribers -- especially those who have several megabit per-second connections. However, in a large network that consists of thousands of subscribers, the amount of traffic will multiply and will lead to a significant burden on the server (either the ACS or the SNMP manager).

A DRG that use CWMP periodically sends INFORM packet which is about 400 bytes per-packet and ACS sends HTTP 200 OK replies which are roughly 200 bytes per-packet. If the periodic interval is set to be too small, the DRG could generate a lot of traffic. Thus, the DRG sets the interval for sending a periodic INFORM to be 900 seconds by default. This means a single DRG will be responsible for sending or receiving 600 bytes every 900 seconds i.e., 2/3 bytes per second - even if it is not doing anything. The interval value of 900 seconds is configurable.
Table 4-3 and Figure 4-16 shows amount of traffic per-DRG per-second when the interval is set to different value.

Table 4-3: The amount of traffic per-DRG, per-second when periodic INFORM message is configured from 100 to 1000 seconds

Interval (seconds)	Traffic per-second (bytes)
100	6.00
200	3.00
300	2.00
400	1.50
500	1.20
600	1.00
700	0.86
800	0.75
900	0.67
1000	0.60

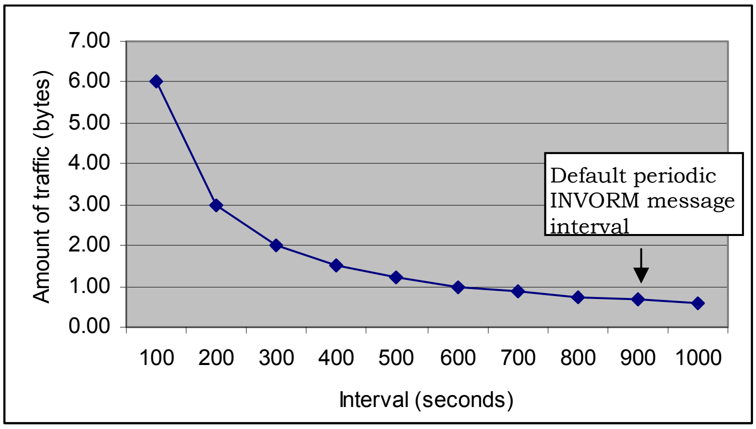


Figure 4-16: The amount of traffic decreases when the interval of periodic message is set to larger value

This type of traffic may not be a problem when the CWMP's ACS is used for managing a network with a small number of DRGs. For example, if there is 100 CPEs in the network then the average data traffic from the periodic INFORM per-second is 100 x (2/3) bytes, which is equal to 66.67 bytes. The traffic will be increased linearly as the number of managed CPEs is increasing. The Table 4-4 and Figure 4-17 shows amount of traffic per-second when number of DRG in the network is increasing.

Table 4-4: The amount of traffic per-second from different number of DRG, assuming the periodic INFORM interval is 900 seconds

Amount of DRGs	Traffic per-second (bytes)	Traffic per-second (**kbytes**)
100	66.67	0.07
200	133.33	0.13
300	200.00	0.20
400	266.67	0.27
500	333.33	0.33
600	400.00	0.40
700	466.67	0.47
800	533.33	0.53
900	600.00	0.60
1000	666.67	0.67
10000	6666.67	6.67

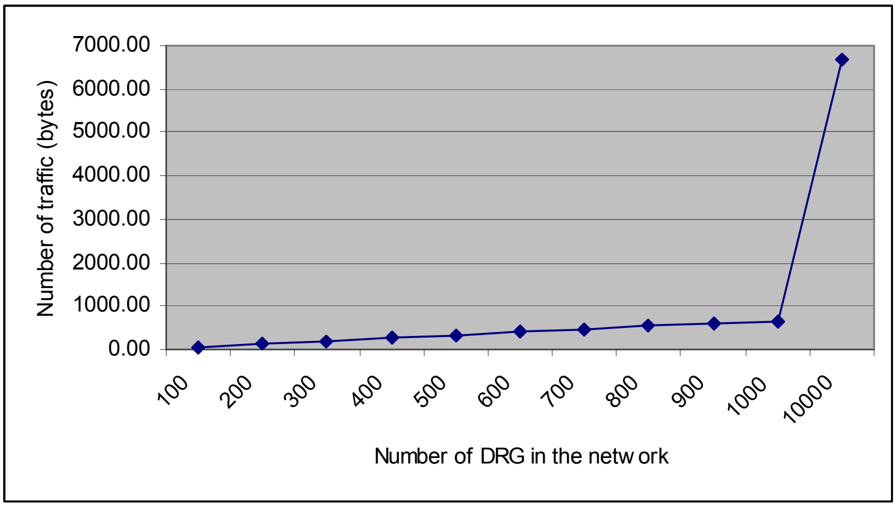


Figure 4-17: The amount of the traffic increases when number of DRGs is increasing

The conclusion from these charts is when the ACS manages more DRGs, then larger interval value should be used to send periodic INFORM messages. This will cause less traffic in the network. However, if the interval is set to a very high number, for example 3,600 seconds (1 hour), then the ACS may have less updated information from the DRG. In this case, if a DRG loses connection after 200 seconds from the last INFORM message, then the ACS will find out this problem after 3400 seconds after the DRG loses connection. Therefore, the interval of the periodic INFORM may need to be set to an appropriate value depending on the network conditions. For example, after a major DRG upgrade in the network, the periodic INFORM interval may need to be set to a lower value (i.e. 300 seconds) to monitor the stability after the upgrade. If the upgrade results in the networking being is stable, then the interval can be set back to the normal value (i.e.900 seconds).

4.4.3.6 Scalability

Scalability is an important consideration when deciding upon a CPE management system. Scalability considerations include when the number of CPEs increases, when the number of services increases, and when new features are added into the CPE. When the number of CPEs increases, the traffic in the network will also increase during the service subscription phase (see Table 4-2 and Figure 4-15).

The number of the service increases when a service provider offers more services for subscription. This can be due to new features being implemented in the CPE or simply repackaging of existing services. For example in Table 4-1, the service provider offers are three types of subscriptions, which means there are 3 profiles available for to be assigned to the CPE. If the service provider decides to break down the original subscription into sub-services and combines these sub-services into new types of subscription, then the service provider will offer more services to the customer. One possible new service is for example a “Basic-plus” package which offers 2 Mbps Internet, IP Telephony with Voice Mailbox, and 16 Channels IPTV. Other new services can be created similarly by combining sub-services from Internet, IP telephony, and IPTV respectively. In this case, the number of new services can be calculated to be 3^3, which is equal to 27 new services. Table 4-5 and Figure 4-18 below shows the increase in number of services due to repackaging and the introduction of new features.

Table 4-5: Services increasing due to repackaging and/or new features

	Possible Services
Original Service	3
Service Repack	27
New Feature 1	64
New Feature 2	125
New Feature 3	216
New Feature 4	343
New Feature 5	512
New Feature 6	729
New Feature 7	1000
New Feature 8	1331
New Feature 9	1728
New Feature 10	2197

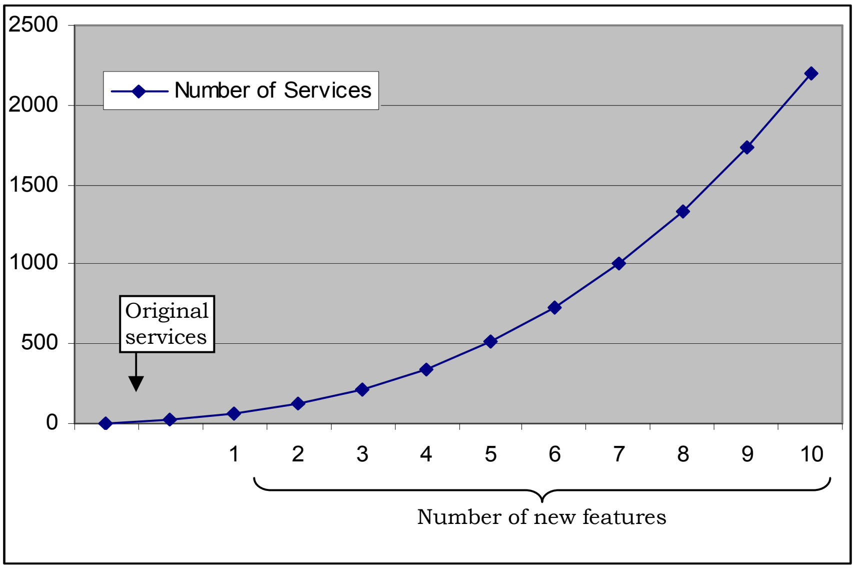


Figure 4-18: Services increasing due to repackaging and/or new features

The impact on the CPE management system will be that it has to be able to handle a large number of different services and to maintain these service subscriptions, while making modifications as simple as possible.

In an SNMP management system, every new feature implemented in CPE will be assigned one or more new OIDs for its configuration. The management system will need to set certain values for these OIDs everytime there is a service subscription or modification. In this case, the subscription script has to be modified. In a CWMP management system, assuming that the new features have been well implemented in the CPE, only the profile in the management system need to be modified in order to support the configuration of the new services. Once the profile has been modified, all subscribed CPEs that are using this profile shall be modified accordingly. In terms of scalability, CWMP appears to be better than SNMP for CPE a management system.

4.4.3.7 Managing CPEs from different vendors

Managing CPEs from different vendors is one of the strengths of the ACS. The CPEs from different vendors have their own vendor specific features and capabilities. In an SNMP management system, these specific features and capabilities are identified and managed using a set of private MIBs. Each private MIB is a collection of OIDs that are placed under the vendor's private enterprise number (see Figure 2-5). If an SNMP manager is to manage a CPE from multiple vendors it requires the different sets of private MIB from each vendor. In a CWMP ACS, each CPE from different vendors will have its own base profile (see section 3.6). Every profile will have parameters to configure both the standard and the specific features in the CPE. The CPE has to be subscribed to this profile in order to be managed.

Managing CPEs from different vendors has not yet been tested in the 42networks' ACS. The main reason is because the current target of the 42networks' ACS is to manage DRGs and its own services. There are also some APIs that are available today in order to improve the interoperability of the ACS with other CWMP CPE management system. Another reason is because managing CPEs from other vendors requires information (i.e. design guidelines, or even source-code) from the vendors in order to make the ACS fully compatible. Such information is normally given by a vendor to its partner and/or other vendors under a special agreement.

Hypothetically, it should not be a problem for the CPE to discover the ACS, as long as the ACSURL can be configured in the DHCP option (see section 4.1.1). There will be potential problem when managing a vendor specific profile that has vendor specific features because the ACS may need the relevant parameters in its database and will also need the vendor specific parameter in the translation module. Moreover, the different ways of writing the SOAP message can cause problems when parsing the SOAP message. However, if these vendor specific parameters and methods are written correctly following the rule in TR-106 [19], then the ACS should be able to manage them correctly.

5 Conclusions

A CPE management system is often overlooked and its importance is underestimated by service providers and network operators. It is in fact one of the most important keys to ensuring high quality service delivery. CPE vendors have been trying to develop management systems, but there were no real standards for CPE management systems. They each ended up developing their own management system which later turned out to be a proprietary product that could only be used by their own CPE. Most of these CPE management systems are based on SNMP, which is the most popular standard for network management today. However, SNMP is more suitable for network management where each network element is being managed individually, than for CPE management where lots of CPEs are being managed in the same way. Managing each element individually means that as the number of units increase, the management task also increases.

The DSL Forum defined a standard for CPE management which is formulated in their Technical Report number 69 (TR-069) called CPE WAN Management Protocol (CWMP). Instead of managing each CPE individually, CWMP manages a CPE based on its service subscription. Managing CPEs based on its service subscription simplifies the management process and makes CPE management easier for service providers. When the number of CPE increases, the service provider only need to make sure that each CPE is correctly subscribed to one of the available services.

Based the research and implementation done in 42networks concerning the ACS and CPE shows that both SNMP and CWMP can be used for VoIP CPE management. Each method has been studied and compared, and the results show that SNMP is capable of performing most of CWMP's functions, although CWMP seems to be better when managing services. The only task that SNMP cannot do when compared to CWMP is to manage CPEs which are located behind the NAT. This occurs because SNMP cannot traverse the NAT. Other than that, SNMP seems be more flexible, requires a smaller memory footprint and has a lower cost of implementation – the later feature is important for CPEs with low processing power.

CWMP defines standard parameters in TR-098 for Internet Gateway Devices, and TR-104 for the VoIP CPE. However, parameters from these two TRs are not applicable to most CPEs today. In addition, most CPEs have one or more vendor specific (non-standard) features which are neither covered in TR-098 nor TR-104. The non-standard features have to be written according to the definition in TR-106 by prepending "X_" and the vendor identity to the parameter's name.

Moreover, the possibility of defining non-standard parameters has exposed CWMP to the risk of reproducing the private MIB problems of SNMP. A CPE vendor can simply rewrite all parameters in its CPE according to TR-106 instead of adopting TR-098 and/or TR-104 and proclaim that their CPE is TR-069 compliant. When more than one CPE vendor does this, then the CPE management system will become a non-standard system. This will reproduce exactly the same problem as has occurred in SNMP where each vendor has defined its own private MIBs.

6 Suggested Topics for Future Work

The CWMP study in this report is based on CPE management system developed by 42networks. The system today only supports Auto-configuration, dynamic service provisioning, and software/firmware image management. Status and performance monitoring has only been implemented for VoIP calls in order to get the call result and voice call quality. Diagnostics have not yet been implemented. Another round of comparison with SNMP should be made when the diagnostic part is implemented. It would be interesting to see which protocol is more effective, i.e. whether using an SNMP TRAP or CWMP INFORM message is more suitable for diagnostics. A better comparison between the use of SNMP and CWMP should be made by measuring the server load when using an SNMP manager versus the CWMP ACS.

The communication over a secured channel (using SSL/TLS) is one of the CWMP features. This feature is not yet being implemented in 42networks' ACS because it was in lower priority in the product roadmap. However, supporting SSL/TLS is important in terms of CWMP standards compliance. Therefore, there should be another study of CWMP when both ACS and CPE support secure communication as in SSL/TLS. One interesting question is whether it is possible to have different levels of security as services that could be offered to customers.

Section 4.4.3, describes the traffic caused by CWMP message exchange. The conclusion shown in Table 4-4 and Figure 4-17 is that the traffic increases when an ACS manages more CPEs, even when the CPEs are doing nothing. There has been a study conducted by A.E. Nikolaidis, G.A. Doumenis, G.I. Stassinopoulos, et al. [39] that describes the problem of increments in packet size for remote configuration and management causing problems in a point-to-multipoint arrangement comprising of an automatic configuration server and thousands of home gateways and multimedia devices. In their report, they proposed to use XML compression to minimize the size of the CWMP packets transmitted between ACS and CPE, and vice versa. An interesting study in the future is to investigate whether it is possible to use other compression mechanism such as HTTP compression [42] or some other mechanism to reduce the size of the packets being sent while using CWMP.

The first SNMP problem as described in section 2.5.1.2 is SNMP implementation which is varies across platforms due to the use of private MIB. In contrast, the CWMP has two additional standards (TR-098 and TR-104) which define the standard parameters for gateway device and VoIP CPE provisioning. It should be possible for an SNMP implementation to adopt parameters from those two TRs and to redefine them as public MIB entries which would make SNMP an attractive option for standard CPE management. This may not prevent those who insist upon using private MIB entries, but the existence of public MIB entries should at least encourage some CPE vendors to start using standardized MIB entries for VoIP CPE management. This will be an interesting research topic because one has to consider whether the data model used in TR-098 and TR-104 can be directly translated to an SNMP MIB. Additionally, a comparison of the cost of implementation and how it will affect the current public MIB will need to be made.

References

[1] DSL Home Technical Working Group, "CPE WAN Management Protocol", TR-069, May 2004, <http://www.dslforum.org/techwork/tr/TR-069.pdf> (August 2006).

[2] "The DSL Source Book", Paradyne Corporation, 2000.

[3] The International Engineering Consortium, *Web ProForum Tutorials*, 2005, "The Evolution of Broadband", <http://www.iec.org/online/tutorials/evolution/> (12 August 2006).

[4] Curt Franklin, "How DSL Works", <http://www.howstuffworks.com/dsl.htm> (20 August 2006).

[5] Suns Microsystems, "RPC: Remote Procedure Call Protocol specification", RFC 1050, IETF, April 1988, <http://www.faqs.org/rfcs/rfc1050.html> (10 September 2006).

[6] "Simple Object Access Protocol (SOAP) 1.1", W3C Note, 08 May 2000 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508> (12 September 2006).

[7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616,IETF, June 1999,<http://www.ietf.org/rfc/rfc2616.txt> (18 September 2006).

[8] Netscape Communications, *"The SSL Protocol, Version 3.0"*, <http://www.netscape.com/eng/ssl3/draft302.txt> (18 September 2006).

[9] T. Dierks and C. Allen, "The TLS Protocol, Version 1.0", RFC 2246, IETF, January 1999, <http://www.ietf.org/rfc/rfc2246.txt > (18 September 2006).

[10] William Stallings, "SNMP, SNMP v2, SNMP v3, and RMON 1 and 2 (Third Edition)", Addison-Wesley, Reading, Mass., 1999.

[11] D. Levi, P. Meyer, and B. Stewart, "SNMPv3 Application", RFC 2273, IETF, January 1998, <http://www.ietf.org/rfc/rfc2273.txt > (19 September 2006).

[12] DSL Home Technical Working Group, "References and Requirements for DRG Architectures for Data Access", Technical Report, TR-018, DSL Forum, March 1999, <http://www.dslforum.org/techwork/tr/TR-018.pdf> (20 September 2006).

[13] Charlie Kauffman, Radia Perlman, and Mike Speciner "Networks Security Private Communication in a Public World", Prentice Hall PTR, 2002.

[14] S. Alexander and R. Droms, "DHCP Options and BOOTP Vendor Extensions*"*, RFC 2132, IETF, March 1997, <http://www.ietf.org/rfc/rfc2132.txt > (8 November 2006).

[15] FS-VDSL Working Group: DRG, "Customer Premises Equipment Specification", FS-VDSL Specification Part 3, ITU-T, February 2003 < http://www.fs-vdsl.net/Specifications/FSVDSL-FGTS-Part01-OperReqs_v03.pdf> (9 November 2006).

[16] DSL Home Technical Working Group, "DRG WAN Management Protocol", Technical Report, TR-069 Amendment 1, DSL Forum, November 2006, <http://www.dslforum.org/techwork/tr/TR-069.pdf> (April 2007).

[17] DSL Home Technical Working Group, "Internet Gateway Device Version 1.1 Data Model for TR-069", Technical Report, TR-098, DSL Forum, September 2005, <http://www.dslforum.org/techwork/tr/TR-098.pdf> (January 2007).

[18] DSL Home Technical Working Group, "Provisioning Parameters for VoIP CPE", Technical Report, TR-104, DSL Forum, September 2005, <http://www.dslforum.org/techwork/tr/TR-104.pdf> (January 2007).

[19] DSL Home Technical Working Group, "Data Model Template for TR-069-Enabled Devices", Technical Report, TR-106, DSL Forum, September 2005, <http://www.dslforum.org/techwork/tr/TR-104.pdf> (April 2007).

[20] Internet Assigned Numbers Authority (IANA), TCP and UDP Port Numbers, Internet Assigned Numbers Authority (IANA), <http://www.iana.org/assignments/port-numbers> (last updated 2008-10-10).

[21] IEEE, Organizationally Unique Identifiers (OUIs), IEEE, March 7, 2005 at 12:08:59, <http://standards.ieee.org/faqs/OUI.html> (May 2007).

[22] Private Enterprise Number (PEN) Request Template <http://pen.iana.org/pen/PenApplication.page> (May 2007).

[23] Peter Larsson, "CDSP2 Design Protocol Revision PA4", Packetfront AB, June 2007.

[24] B.Volz, "Reclassifying Dynamic Host Configuration Protocol version 4 (DHCPv4) Options", RFC 3942, IETF, November 2004, <http://www.ietf.org/rfc/rfc3942.txt > (19 July 2007).

[25] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, IETF, November 2003, <http://www.ietf.org/rfc/rfc3489.txt > (20 July 2007).

[26] M. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, IETF, May 1990, <http://www.ietf.org/rfc/rfc1155.txt > (19 November 2007).

[27] Alistair Cockburn "Writing Effective Use Cases", Addison-Wesley, 2001.

[28] Infonetics Research: Broadband CPE market <http://www.reuters.com/article/pressRelease/idUS247646+12-Mar-2008+MW20080312> (May 2008)

[29] Henning Schulzrinnem, "Audio codecs", Department of Computer Science, Columbia University. Last updated Wed, 09 Jan 2008 <http://www.cs.columbia.edu/~hgs/audio/codecs.html >(May 2008)

[30] Free images from http://www.clipartguide.com/ (October 2008)

[31] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, IETF, March 1999, <http://www.ietf.org/rfc/rfc2543.txt > (20 September 2008).

[32] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler , "SIP: Session Initiation Protocol", RFC 3261, IETF, June 2002, <http://www.ietf.org/rfc/rfc3261.txt > (20 September 2008).

[33] MIB tree, <http://www.smartbridges.com/new_images/36_MIB.jpg> (5 October 2008).

[34] Dave Marshall, "Remote Procedure Calls (RPC)", May 1999, <http://www.cs.cf.ac.uk/Dave/C/node33.html> (7 October 2008.)

[35] Wikipedia, "Simple traversal of UDP over NATs", Wikipedia, <http://en.wikipedia.org/wiki/STUN#Algorithm > (6 October 2008)

[36] Dobrica Pavlinušić, "Lightweight CWMP server written in perl", Google code, <http://code.google.com/p/perl-cwmp/> (10 October 2008)

[37] "BOOTP / DHCP options", <http://www.networksorcery.com/enp/protocol/bootp/options.htm> (10 October 2008)

[38] Cisco Systems, Inc, "Simple Network Management Protocol", Cisco Systems, Inc, Internetworking Technology Handbook <http://www.cisco.com/en/US/docs/internetworking/technology/handbook/SNMP.html > (11 October 2008)

[39] A.E. Nikolaidis, G.A. Doumenis, G.I. Stassinopoulos, et al., "Management traffic in emerging remote configuration mechanisms for residential gateways and home devices", Communications Magazine, IEEE, Volume 43, Issue 5, May 2005, pages 154 – 162

[40] Axiros, TR-069 interoperable vendors, <http://www.axiros.com/products/cpe-dsl-management/tr-069-interoperability.html > (20 October 2008)

[41] Dimark, Dimark Management System , <http://www.dimark.com/tr-069_client.html > (20 October 2008)

[42] Constantin Rack , HTTP Compression, <http://www.http-compression.com/> (20 October 2008)

[43] ITU-T Recommendation T.140, "Protocol for Multimedia Application Text Conversation" (February 1998) and Addendum 1

[44] A. van Wijk, G. Gybels, "Framework for Real-Time Text over IP Using the Session Initiation Protocol (SIP) ", RFC 5194, IETF, June 2008, <http://www.ietf.org/rfc/rfc5194.txt > (20 September 2008).

[45] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot, "Address Allocation for Private Internets" RFC 1597, IETF, March 1994, <http://tools.ietf.org/html/rfc1597 > (4 November 2008).

[46] BOOTP / DHCP options, <http://www.networksorcery.com/enp/protocol/bootp/options.htm> (8 November 2008).

[47] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, IETF, January 1996, <http://www.ietf.org/rfc/rfc1889.txt > (4 November 2008).

[48] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, RFC 3550, IETF, July 2003, <http://www.ietf.org/rfc/rfc3550.txt> (4 November 2008).

Appendix

A.1Vendor Encapsulated option

A vendor encapsulation option is option number 43 out of over 200 options in DHCP [14]. This implementation is often called “option43”. This option is used for exchanging vendor specific information in DHCP. It can contain any free text not longer than 255 bytes. Some operators use this option to load default DRG configuration when the DRG gets its IP address from the DHCP server.

A.2REDs Polling

The REDs [3]polling is 42networks proprietary mechanism to automatically provision default configuration to a DRG. It uses HTTP to request a configuration file to be loaded into a DRG. The basic idea is when a user gets his DRG from an operator, he should only need to do is to connect his DRG to his home network. After some time (time is configurable, by default 15 minutes), if DRG has not been managed by any other means (using SNMP or any configuration file request using TFTP or HTTP) then it will send a request to the default configuration server to get the configuration file which is identified by the MAC address of DRG. After DRG being configured, it will poll periodically (time is configurable, by default 15 minutes) to check whether there is a newer configuration file to be loaded.

The default configuration server set in DRG was http://reds.42networks.com; however operators have to have their own default configuration server in the actual implementation. Before deploying the solution, operator must provide configuration files in the server for all DRGs out in the field.

This feature is no longer used today and the REDs server had been deactivated.

A.3Configuration file

The most common method currently used to widely deploy DRG uses a configuration file. The provisioning of configuration file can be either from SNMP, web GUI, option43, or REDs polling. The configuration file is an encoded text-file, normally not more than 2 kilobytes. It consists of sections depends on the information category to be loaded to DRG. In each section, there are parameters name and values. The sections are:

- Header
- Application
- Config
- Ringtones
- Ringsignals
- End

Sample of a configuration file:

```
[HEADER]
Configuration file sample
[APPLICATION]
OPDEF=sample.ini
FLASHHOOKMINTIMER=30
FLASHHOOKTIMER=140

[CONFIG]
DIALPLAN=(xx.#|xx.T)
L1ONOFF=ON
L1SIPPIP=10.0.0.10

[RINGTONES]
DIALTONE=425@-5#ON(1000),R
```

[3] REDs is the abbreviation from ”redirects”.

```
RINGTONE=425@-5#ON(1000),OFF(500),R

[RINGSIGNALS]
RING_CADENCE_1=ON(1000),OFF(500),R

[END]
```

A.4RPC

RPCs are used in bidirectional communication between DRG to ACS. This section describes some currently supported RPC methods both in DRG and ACS.

- GETRPCMETHODS

  This function must be supported by both ACS and DRG. It may be used for both ACS and DRG to list all the supported method (standard and proprietary methods) from each other. This method has no argument. Here is the example:

  Request:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<Request>GetRPCMethods</Request>
```

  Response:

```
POST HTTP/1.1
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Response>GetRPCMethodsResponse</Response>
<Return>
        <MethodList>
                GetRPCMethods[CR]
                GetParameterValues[CR]
                GetParameterNames[CR]
                GetParameterAttributes[CR]
                SetParameterValues[CR]
                SetParameterAttributes[CR]
                Download[CR]
                Reboot
        </MethodList>
</Return>
<soap:/Body>
<soap:/Envelope>
```

- GETPARAMETERVALUES

  This function allows the ACS to query parameters and values, a list of requested parameters are passed as argument. For example:

  Request:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Request>GetParameterValue</Request>
<Parameter>Config.L1SIPIP[CR]
            Config.L2SIPIP
</Parameter>
<soap:/Body>
<soap:/Envelope>
```

  Response:

```
POST HTTP/1.1

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Response>GetParameterValue</Response>
<Return>Config.L1SIPIP=192.168.18.10[CR]
        Config.L2SIPIP=192.168.18.10
</Return>
<soap:/Body>
<soap:/Envelope>
```

- SETPARAMETERSVALUE
  This function allows ACS to set values of the parameters in DRG.
  Request:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Request>SetParameterValue</Request>
<Parameter>
        RINGTONES.CALLWAITING=425@-10#ON(500),OFF(500),ON(500)[CR]
        RINGTONES.NETWORK_BUSY=425@-5#ON(250),OFF(250),R[CR]
        CONFIG.DIALPLAN=(0[1-7]xxxxxxxx|08[1-9]xxxxxxx|x.T|x.#) [CR]
        CONFIG.CLIR=ON[CR]
        CONFIG.CLIR_PREFIX=*31*[CR]
        CONFIG.CALLERID1ONOFF=ON[CR]
        CONFIG.CALLERID2ONOFF=ON[CR]
</Parameter>
<soap:/Body>
<soap:/Envelope>
```

  Response:

```
POST HTTP/1.1

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
        <Response>SetParameterValue</Response>
        <Return>1</Return>
<soap:/Body>
<soap:/Envelope>
```

- DOWNLOAD
  Download command is placed in a Request message and result in the response message. If the result of the execution is success, then it will return value 1 as status code. Otherwise it will directly reply the status code which represents the cause of the failure. Here is the example:
  Request:

```
HTTP/1.1 200OK
Date: Sat, 10 Aug 2007 23:43:47 GMT

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Request>Download</Request>
<Parameter>
        Filetype=1
        URL=http://192.168.18.10/DMA0028-R2K123.r0
</Parameter>
```

  Response on success:

```
POST HTTP/1.1

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Response>Download</Response>
<Return>
        <value>1:Success</value>
</Return>
<soap:/Body>
<soap:/Envelope>
```

  Response on failure:

```
POST HTTP/1.1

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Response>Download</Response>
<Return>
        <value>9000:File not found</value>
</Return>
<soap:/Body>
<soap:/Envelope>
```

- REBOOT
  This function is for ACS to instruct a DRG to reboot. To protect DRG from unauthorized or fake reboot instruction, REBOOT command must come with a valid reboot-string which indicates that the reboot comes from an authorized ACS. A simple reboot-string can be an encrypted ACS identity. Here is the example:
  Request:
  ```
  HTTP/1.1 200OK
  Date: Sat, 10 Aug 2007 23:43:47 GMT

  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
  <Request>Reboot</Request>
  <Parameter>
  	Reboot-String=jxze345324eklrnf
  </Parameter>
  ```

  Response on success:
  ```
  POST HTTP/1.1

  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
  <Request>Reboot</Request>
  <Return>1</Return>
  <soap:/Body>
  <soap:/Envelope>
  ```

  Response on failure:
  ```
  POST HTTP/1.1

  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
  <Request>Reboot</Request>
  <Return>-1</Return>
  <soap:/Body>
  <soap:/Envelope>
  ```

A.5List of DHCP Option

This list below contains complete DHCP options. Some modification made to the list to add relevant and remove the irrelevant information.

Code	Description	References
0	Pad.	RFC 2132
1	Subnet Mask.	RFC 2132
2	Time Offset (deprecated).	RFC 2132
3	Router.	RFC 2132
4	Time Server.	RFC 2132
5	Name Server.	RFC 2132
6	Domain Name Server.	RFC 2132
7	Log Server.	RFC 2132
8	Quote Server.	RFC 2132
9	LPR Server.	RFC 2132
10	Impress Server.	RFC 2132
11	Resource Location Server.	RFC 2132
12	Host Name.	RFC 2132
13	Boot File Size.	RFC 2132
14	Merit Dump File.	RFC 2132
15	Domain Name.	RFC 2132
16	Swap Server.	RFC 2132
17	Root Path.	
18	Extensions Path.	
19	IP Forwarding enable/disable.	
20	Non-local Source Routing enable/disable.	
21	Policy Filter.	
22	Maximum Datagram Reassembly Size.	
23	Default IP Time-to-live.	
24	Path MTU Aging Timeout.	
25	Path MTU Plateau Table.	
26	Interface MTU.	
27	All Subnets are Local.	
28	Broadcast Address.	
29	Perform Mask Discovery.	
30	Mask supplier.	
31	Perform router discovery.	
32	Router solicitation address.	
33	Static routing table.	RFC 2132
34	Trailer encapsulation.	RFC 2132
35	ARP cache timeout.	RFC 1533, RFC 2132
36	Ethernet encapsulation.	RFC 1533, RFC 2132
37	Default TCP TTL	RFC 1533, RFC 2132
38	TCP keepalive interval.	RFC 1533, RFC 2132
39	TCP keepalive garbage.	RFC 1533, RFC 2132
40	Network Information Service domain.	RFC 1533, RFC 2132
41	Network Information Servers.	RFC 1533, RFC 2132
42	NTP servers.	RFC 1533, RFC 2132
43	Vendor specific information.	RFC 1533, RFC 2132
44	NetBIOS over TCP/IP name server.	RFC 1533, RFC 2132
45	NetBIOS over TCP/IP Datagram Distribution	RFC 1533, RFC 2132

	Server.	
46	NetBIOS over TCP/IP Node Type.	RFC 1533, RFC 2132
47	NetBIOS over TCP/IP Scope.	RFC 1533, RFC 2132
48	X Window System Font Server.	RFC 1533, RFC 2132
49	X Window System Display Manager.	RFC 1533, RFC 2132
50	Requested IP Address.	RFC 1533, RFC 2132
51	IP address lease time.	RFC 1533, RFC 2132
52	Option overload.	RFC 1533, RFC 2132
53	DHCP message type.	RFC 1533, RFC 2132, RFC 3203, RFC 4388
54	Server identifier.	RFC 1533, RFC 2132
55	Parameter request list.	RFC 1533, RFC 2132
56	Message.	RFC 1533, RFC 2132
57	Maximum DHCP message size.	RFC 1533, RFC 2132
58	Renew time value.	RFC 1533, RFC 2132
59	Rebinding time value.	RFC 1533, RFC 2132
60	Class-identifier.	RFC 1533, RFC 2132
61	Client-identifier.	RFC 1533, RFC 2132, RFC 4361
62	NetWare/IP Domain Name.	RFC 2242
63	NetWare/IP information.	RFC 2242
64	Network Information Service+ Domain.	RFC 2132
65	Network Information Service+ Servers.	RFC 2132
66	TFTP server name.	RFC 2132
67	Bootfile name.	RFC 2132
68	Mobile IP Home Agent.	RFC 2132
69	Simple Mail Transport Protocol Server.	RFC 2132
70	Post Office Protocol Server.	RFC 2132
71	Network News Transport Protocol Server.	RFC 2132
72	Default World Wide Web Server.	RFC 2132
73	Default Finger Server.	RFC 2132
74	Default Internet Relay Chat Server.	RFC 2132
75	StreetTalk Server.	RFC 2132
76	StreetTalk Directory Assistance Server.	RFC 2132
77	User Class Information.	RFC 3004
78	SLP Directory Agent.	RFC 2610
79	SLP Service Scope.	RFC 2610
80	Rapid Commit.	RFC 4039
81	FQDN, Fully Qualified Domain Name.	RFC 4702
82	Relay Agent Information.	RFC 3046, RFC 5010
83	Internet Storage Name Service.	RFC 4174
84	Unused option. *	RFC 3679
85	NDS servers.	RFC 2241
86	NDS tree name.	RFC 2241
87	NDS context.	RFC 2241
88	BCMCS Controller Domain Name list.	RFC 4280
89	BCMCS Controller IPv4 address list.	RFC 4280
90	Authentication.	RFC 3118
91	client-last-transaction-time.	RFC 4388
92	associated-ip.	RFC 4388
93	Client System Architecture Type.	RFC 4578
94	Client Network Interface Identifier.	RFC 4578

95	LDAP, Lightweight Directory Access Protocol.	RFC 3679
96	Unused option. *	RFC 3679
97	Client Machine Identifier.	RFC 4578
98	Open Group's User Authentication.	RFC 2485
99	GEOCONF_CIVIC.	RFC 4776
100	IEEE 1003.1 TZ String.	RFC 4833
101	Reference to the TZ Database.	RFC 4833
102 - 111	Unused options. *	RFC 3679
112	NetInfo Parent Server Address.	RFC 3679
113	NetInfo Parent Server Tag.	RFC 3679
114	URL.	RFC 3679
115	Unused option.	RFC 3679
116	Auto-Configure	RFC 2563
117	Name Service Search.	RFC 2937
118	Subnet Selection.	RFC 3011
119	DNS domain search list.	RFC 3397
120	SIP Servers DHCP Option.	RFC 3361
121	Classless Static Route Option.	RFC 3442
122	CCC, CableLabs Client Configuration.	RFC 3495, RFC 3594, RFC 3634
123	GeoConf.	RFC 3825
124	Vendor-Identifying Vendor Class.	RFC 3925
125	Vendor-Identifying Vendor-Specific.	RFC 3925
126	Unused options. *	RFC 3679
127	Unused options. *	RFC 3679
128	TFPT Server IP address.	RFC 4578
129	Call Server IP address.	RFC 4578
130	Discrimination string.	RFC 4578
131	Remote statistics server IP address.	RFC 4578
132	802.1P VLAN ID.	RFC 4578
133	802.1Q L2 Priority.	RFC 4578
134	Diffserv Code Point.	RFC 4578
135	HTTP Proxy for phone-specific applications.	RFC 4578
136	OPTION_PANA_AGENT.	
137 - 149	Unused options. *	RFC 3942
150	TFTP server address. Etherboot. GRUB configuration path name.	
151 - 174	Unused options. *	RFC 3942
175	Etherboot.	
176	IP Telephone.	
177	Etherboot. PacketCable and CableHome.	
178 - 207	Unused options. *	RFC 3942

208	pxelinux.magic (string) = F1:00:74:7E (241.0.116.126).	RFC 5071
209	pxelinux.configfile (text).	RFC 5071
210	pxelinux.pathprefix (text).	RFC 5071
211	pxelinux.reboottime (unsigned integer 32 bits).	RFC 5071
212 - 219	Unused options. *	RFC 3942
220	Subnet Allocation.	
221	Virtual Subnet Selection.	
222	Unused options. *	RFC 3942
223	Unused options. *	RFC 3942
224 - 254	Private use.	RFC3942
255	End.	RFC 2132

* Unused DHCP options are now owned by IANA for further assignment in the future. They are either assigned before but later revoked by IANA or never been assigned at all.

A.6Parameter Mapping List

1. Network parameter mapped to TR-098 parameter.

DEVICE-INFO	
PRODSN	InternetGatewayDevice.DeviceInfo.SerialNumber
PRODNO	InternetGatewayDevice.DeviceInfo.ModelName
PRODNAME	InternetGatewayDevice.DeviceInfo.Description
PRODWEEK	X_000f5d.InternetGatewayDevice.DeviceInfo.PRODWEEK
PRODREV	InternetGatewayDevice.DeviceInfo.HardwareVersion
CFREF	InternetGatewayDevice.DeviceInfo.ProductClass
SW	InternetGatewayDevice.DeviceInfo.SoftwareVersion
INI	If DRG has been configured with any INI file, then the "InternetGatewayDevice.DeviceInfo.VendorConfigFileNumberOfEntries" will contain non zero (1 or larger) value, which means there will be entries for "InternetGatewayDevice.DeviceInfo.Vendor-ConfigFile.{i}.Name, InternetGatewayDevice.DeviceInfo.Vendor-ConfigFile.{i}.Version, InternetGatewayDevice.DeviceInfo.Vendor-ConfigFile.{i}.Date, InternetGatewayDevice.DeviceInfo.Vendor-ConfigFile.{i}.Description"
WAN	
CICUSTOM	X_000f5d.InternetGatewayDevice.Layer3Forwarding.WANIPConnection.{i}CICUSTOM
IF${I}CICUSTOM	X_000f5d.InternetGatewayDevice.Layer3Forwarding.WANIPConnection.{i}IF${I}CICUSTOM
IF${I}DHCP	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.AddressingType
IF${I}DNSDOMAINNAME	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.DNS Servers
IF${I}DNSHOSTNAME	X_000f5d.InternetGatewayDevice.Layer3Forwarding.WANIPConnection.{i}IF${I}DNSHOSTNAME
IF${I}ENABLED	InternetGatewayDevice.WANDevice.{i}.WANEthernetInterfaceConfig.Enable
IF${I}IPADDRESS	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.ExternalIPAddress
IF${I}IPGATEWAY	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.DefaultGateway
IF${I}IPNETMASK	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.SubnetMask
IF${I}L3PROT	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnectionNumberOfEntries **or** InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnectionNumberOfEntries
IF${I}NETCONF	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANIPConnection.{i}.ConnectionType
PPPOE	
IF${I}ENABLED	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.Enable
AUTH	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}..WANPPPConnection.{i}PPPAuthenticationProtocol
IF${I}PPP_ECHOCOUNT	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.PPPLCPEcho
IF${I}PPP_ECHOTO	
IF${I}PPP_IDLETO	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.IdleDisconnectTime
IF${I}PPP_PASSWORD	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.Password
IF${I}PPP_USERNAME	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.Username
IF${I}PPPOE_AC	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.PPPoEACName
IF${I}PPPOE_SRV	InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}.PPPoEServiceName
CHAP	CHAP or PAP authentication is depending on "InternetGatewayDevice.WANDevice.{i}.WANConnectionDevice.{i}.WANPPPConnection.{i}. PPPAuthenticationProtocol" If the selection is CHAP then the user name and password following are CHAP, and if it is PAP then the username and password follows accordingly.
CHAPNAME	
CHAPSECRET	
PAP	
PAPPASSWD	
PAPUSERID	

LAN	
IF${I}ENABLED	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.Enable
IF${I}DHCP	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.IPInterface AddressingType
IF${I}IPADDRESS	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.IPInterfaceI PAddress
IF${I}IPNETMASK	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.IPInterface.{i}.IPInterface SubnetMask
DHCPDOMAIN	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DomainName
DHCPPOOLMAX	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.MaxAddress
DHCPPOOLMIN	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.MinAddress
DHCPSERV	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DHCPServerEnable
DHCPSTATICID	X_000f5d.InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagementDHCPSTATICID
DHCPSTATICIDTYPE${S}	X_000f5d.InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagementDHCPSTATICID TYPE${S}
DHCPSTATICIP${S}	X_000f5d.InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagementDHCPSTATICIP ${S}
IPDNS[1-2]	InternetGatewayDevice.LANDevice.{i}.LANHostConfigManagement.DNSServers
ROUTE	
ROUTEDESTIP${R}	InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.DestIPAddress
ROUTEGATEIPIP${R}	InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.GatewayIPAddress
ROUTEINT$${R}	InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.Interface
ROUTEMETRIC[1-8]	InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.ForwardingMetric
ROUTESUBNETMASK${R}	InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}.DestSubnetMask
ROUTERX	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}ROUTERX
ROUTETX	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}ROUTETX
PORTFWDIP${PORTFWD}	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}PORTFWDIP${PORTFWD }
PORTFWDMAX${PORTFWD}	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}PORTFWDMAX${PORTF WD}
PORTFWDMIN${PORTFWD}	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}PORTFWDMIN${PORTFW D}
PORTFWDPROT${PORTFWD}	X_000f5d.InternetGatewayDevice.Layer3Forwarding.Forwarding.{i}PORTFWDPROT${PORTF WD}
DIFFSERV	
DS_DEFAULT	X_000f5d.DIFFSERV.DS_DEFAULT
DS_RTP	X_000f5d.DIFFSERV.DS_RTP
DS_SIP	X_000f5d.DIFFSERV.DS_SIP
DS_SNMP	X_000f5d.DIFFSERV.DS_SNMP
CALLSIG_TOS	X_000f5d.DIFFSERV.CALLSIG_TOS
RTP_TOS	X_000f5d.DIFFSERV.RTP_TOS
PORT	
MAU${M}	InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.Enable InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.MaxBitRate InternetGatewayDevice.LANDevice.{i}.LANEthernetInterfaceConfig.{i}.DuplexMode
MAU${M}AUTONEG	
MAU${M}AUTONEG100FD	
MAU${M}AUTONEG100HD	
MAU${M}AUTONEG10FD	
MAU${M}AUTONEG10HD	
MAU${M}DEFTYPE	
MAU${M}FC	
THRUPUTBITMAX	X_000f5d.THRUPUT.THRUPUTBITMAX
THRUPUTPKTMAX	X_000f5d.THRUPUT.THRUPUTPKTMAX
THRUPUTWINSIZE	X_000f5d.THRUPUT.THRUPUTWINSIZE
UPLIMIT	X_000f5d.THRUPUT.UPLIMIT
UPTHRHOLD	X_000f5d.THRUPUT.UPTHRHOLD
SNMP	
AUTHKEY	X_000f5d.SNMP.AUTHKEY
ENAUTHFAILTRP	X_000f5d.SNMP.ENAUTHFAILTRP

PRIVKEY	X_000f5d.SNMP.PRIVKEY
RESTARTTRAP	X_000f5d.SNMP.RESTARTTRAP
SNMPENABLE	X_000f5d.SNMP.SNMPENABLE
SNMPENGBOOTS	X_000f5d.SNMP.SNMPENGBOOTS
SNMPLAN	X_000f5d.SNMP.SNMPLAN
SNMPWAN	X_000f5d.SNMP.SNMPWAN
SNMPREADCOMMUNITY	X_000f5d.SNMP.SNMPREADCOMMUNITY
SNMPWRITECOMMUNITY	X_000f5d.SNMP.SNMPWRITECOMMUNITY
T${T}	X_000f5d.SNMP.T${T}
TRAPHOSTCOMMUNITY	X_000f5d.SNMP.TRAPHOSTCOMMUNITY
TRAPHOSTIPADDRESS	X_000f5d.SNMP.TRAPHOSTIPADDRESS
FILE CONTROL	
HTTPFILE	Firmware and configuration file loading are not handled in service related parameter. Both of them now are triggered directly from the RPC method.
SWUGTYPE	
TFTPFILE	
TFTPRETRIES	
TFTPSERVER	
ACCESS	
OPMGMT	InternetGatewayDevice.ManagementServer.URL
POLLING	InternetGatewayDevice.ManagementServer.PeriodicInformEnable
POLLINTERVAL	InternetGatewayDevice.ManagementServer.PeriodicInformInterval
$PWVAL	X_000f5d.ACCESS.$PWVAL
$PWVAL-AD	X_000f5d.ACCESS.$PWVAL-AD
.CONFIGCLEAR.	
.PASSWORD	
WEB_ROOT	X_000f5d.ACCESS.WEB_ROOT
WEB_USER	X_000f5d.ACCESS.WEB_USER
HTTPSERVERPORT	X_000f5d.ACCESS.HTTPSERVERPORT
WWWONOFF	X_000f5d.ACCESS.WWWONOFF
TIME	
NTPSERVERIP	InternetGatewayDevice.Time.NTPServer1
TIMEZONE	InternetGatewayDevice.Time.LocalTimeZoneName
DST	InternetGatewayDevice.Time.DayLightSavingUsed
VLAN	
V${V}	X_000f5d.VLAN.V${V}
IF${I}PRIORITYTAG	X_000f5d.VLAN.IF${I}PRIORITYTAG
IF${I}VLANTAG	X_000f5d.VLAN.IF${I}VLANTAG
MAU${M}VID	X_000f5d.VLAN.MAU${M}VID
PRIORITYTAG_CALL	VoiceService{i}.VoiceProfiles{i}.SIP.EthernetPriorityMark
PRIORITYTAG_RTP	VoiceService{i}.VoiceProfiles{i}.RTP.EthernetPriorityMark
VLANTAG_CALL	VoiceService{i}.VoiceProfiles{i}.SIP.VLANIDMark
VLANTAG_RTP	VoiceService{i}.VoiceProfiles{i}.RTP.VLANIDMark
PRIORITY QUEUE	
PQ_DSCP ${PDRG3X}	X_000f5d.PQ.PQ_DSCP ${PDRG3X}
PQ_DSCP_MAP	X_000f5d.PQ.PQ_DSCP_MAP
PQ_PORT${PDRG3X}	X_000f5d.PQ.PQ_PORT${PDRG3X}
PQ_TYPE	X_000f5d.PQ.PQ_TYPE
PQ_VLAN${PDRG3X}	X_000f5d.PQ.PQ_VLAN${PDRG3X}
PQ_VLAN_THRESHOLD	X_000f5d.PQ.PQ_VLAN_THRESHOLD
BARP	
BARP_ENABLE	X_000f5d.BARP.BARP_ENABLE
BARP_BARS_IPADDRESS	X_000f5d.BARP.BARP_BARS_IPADDRESS
BARP_CCLEAR	X_000f5d.BARP.BARP_CCLEAR

BARP_CRETRY	X_000f5d.BARP.BARP_CRETRY
BARP_PREALLOCATE	X_000f5d.BARP.BARP_PREALLOCATE
BARP_RTP_PACKING	X_000f5d.BARP.BARP_RTP_PACKING
BARP_TACK	X_000f5d.BARP.BARP_TACK
BARP_TDSC	X_000f5d.BARP.BARP_TDSC
BARP_UDPPORT	X_000f5d.BARP.BARP_UDPPORT
BARP_VOCODER	X_000f5d.BARP.BARP_VOCODER
DRAP_PRODID	X_000f5d.BARP.DRAP_PRODID

2. Telephony parameter mapped to TR-104 parameter.

GENERAL VoIP	
DIALPLAN	VoiceService{i}.VoiceProfiles{i}.NumberingPlan
DIALPULSE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.DIALPULSE
DIALTIMEOUT	VoiceService{i}.VoiceProfiles{i}.NumberingPlan.InterDigitTimerStd
POUNDSPEEDDIAL	VoiceService{i}.VoiceProfiles{i}.NumberingPlan
G723ON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.G723ON
G729ON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.G729ON
RTPPORTSTART	VoiceService{i}.VoiceProfiles{i}.RTP.LocalPortMin
RTPPORTEND	VoiceService{i}.VoiceProfiles{i}.RTP.LocalPortMax
SI3210ENHENABLE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SI3210ENHENABLE
SI3210IMPEDSYNTH	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SI3210IMPEDSYNTH
RING_FREQ	X_000f5d.VoiceService{i}.VoiceProfiles{i}.RING_FREQ
RING_AMPLITUDE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.RING_AMPLITUDE
USERAGENT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.USERAGENT
TELEVENTPAYLOAD	VoiceService{i}.VoiceProfiles{i}.RTP.TelephoneEventPayloadType
PARKTONUMBER${L}	X_000f5d.VoiceService{i}.VoiceProfiles{i}.PARKTONUMBER${L}
ALLOWMWITONE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.ALLOWMWITONE
CALLSIGPORT${P}	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CALLSIGPORT${P}
JB_TYPE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.JB_TYPE
AJB_MAXDELAY	X_000f5d.VoiceService{i}.VoiceProfiles{i}.AJB_MAXDELAY
FJB_DELAY	X_000f5d.VoiceService{i}.VoiceProfiles{i}.FJB_DELAY
AUTO_JB_SWITCH	X_000f5d.VoiceService{i}.VoiceProfiles{i}.AUTO_JB_SWITCH
ALLOWMWIBLINK	X_000f5d.VoiceService{i}.VoiceProfiles{i}.ALLOWMWIBLINK
COUNTRY	X_000f5d.VoiceService{i}.VoiceProfiles{i}.COUNTRY
INBANDDTMF	VoiceService{i}.VoiceProfiles{i}.DTMFMethodG711.
KEYPADTYPE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.KEYPADTYPE
OUTOFBANDDTMF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.OUTOFBANDDTMF
PULSE_METER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.PULSE_METER
SIP	
L${L}SIPPIP	VoiceService{i}.VoiceProfiles{i}.SIP.ProxyServer
L${L}SIPPPORT	VoiceService{i}.VoiceProfiles{i}.SIP.ProxyServerPort
L${L}SIPSIP	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.L${L}SIPSIP
L${L}SIPSPORT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.L${L}SIPSPORT
L${L}DOMAINNAME	VoiceService.{i}.VoiceProfile.{i}.SIP.UserAgentDomain
LINE${L}AUTHUSER	VoiceService{i}.VoiceProfiles{i}.Line{i}.SIP.AuthUserName
LINE${L}AUTHPSWD	VoiceService{i}.VoiceProfiles{i}.Line{i}.SIP.AuthPassword
LINE${L}NUMBER	VoiceService{i}.VoiceProfiles{i}.Line{i}.DirectoryNumber
LINE${L}PORT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.LINE${L}PORT
LINE${L}MSGACCOUNT	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.MessageWaiting

INCSTANDPORT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.INCSTANDPORT
SIP_INVITE_NO_SDP	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.SIP_INVITE_NO_SDP
SIP_INVITE_TIMER	VoiceService{i}.VoiceProfiles{i}.SIP.InviteExpires
SIP_TEL_URI	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.SIP_TEL_URI
SIP_URI_USER_PARAM	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.SIP_URI_USER_PARAM
SIP_SESSION_TIMER	VoiceService{i}.VoiceProfiles{i}.SIP.ReInviteExpires
SIP_NOTIFY_KEEPALIVE	VoiceService{i}.VoiceProfiles{i}.SIPEventSubscribeNumberOfElements.
SIP_NOTIFY_NAT_MAPPING_TIMEOUT	VoiceService{i}.VoiceProfiles{i}.SIPEventSubscribeNumberOfElements.
SIP_SEND_PRACK	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.SIP_SEND_PRACK
STUNCLIENTMODE	VoiceService{i}.VoiceProfiles{i}.STUNEnable.
STUNSERVERADDR	VoiceService{i}.VoiceProfiles{i}.STUNServer.
STUNSERVERPORT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.STUNSERVERPORT
STUNDEFSERVERI	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.STUNDEFSERVERI
STUNDEFSERVERII	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.STUNDEFSERVERII
STUNDEFSERVERIII	X_000f5d.VoiceService{i}.VoiceProfiles{i}.SIP.STUNDEFSERVERIII
CALL FEATURE SETTING	
FLASHHOOKMAXTIMER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.FLASHHOOKMAXTIMER
FLASHHOOKMINTIMER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.FLASHHOOKMINTIMER
CONF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CONF
DROP	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.DROP
FLASH	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.FLASH
HOLD	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.HOLD
L${L}_3PC	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.MaxSessions
CCBSOFF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CCBSOFF
CCBSON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CCBSON
L${L}C5S	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}C5S
L${L}CCBSINTERVAL	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}CCBSINTERVAL
L${L}CCBSDURATION	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}CCBSDURATION
CFNOANSWEROFF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CFNOANSWEROFF
CFNOANSWERON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CFNOANSWERON
CFUNCONDITIONALOFF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CFUNCONDITIONALOFF
CFUNCONDITIONALON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CFUNCONDITIONALON
L${L}CF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}CF
L${L}CFUNCOND	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.CallForwardUnconditionalEnable
L${L}CFUNCONDNUM	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.CallForwardUnconditionalNumber
L${L}CWTONE	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.CallWaitingEnable
CWOFF	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CWOFF
CWON	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CWON
CWSTAT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CWSTAT
CLIR	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.AnonymousCalEnable
CLIR_PREFIX	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.CLIR_PREFIX
L${L}ANONYMOUS_FROM_HEADER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}ANONYMOUS_FROM_HEADER
L${L}ANONYMOUS_TO_HEADER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}ANONYMOUS_TO_HEADER
L${L}ANONYMOUS_DISPLAY_NAME	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}ANONYMOUS_DISPLAY_NAME
L${L}PROXY_REQUIRE_PRIVACY	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}PROXY_REQUIRE_PRIVACY
L${L}SUSPENDTIMER	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}SUSPENDTIMER
L${L}_LOCAL_RINGING	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}_LOCAL_RINGING
L${L}MEDIADIRECTION	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}MEDIADIRECTION
L${L}REVPOLFORPAY	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}REVPOLFORPAY
LINE SETTING	
LINE${L}ONOFF	VoiceService{i}.VoiceProfiles{i}.Line{i}.Enable
L${L}HAMODE	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}HAMODE

L${L}TRANSPORT_TYPE	VoiceService{i}.VoiceProfiles{i}.SIP.Transport
L${L}FAXT38	VoiceService{i}.VoiceProfiles{i}.FAXT38.Enable
RINGSIGNAL${L}	VoiceService{i}.VoiceProfiles{i}.Line{i}.Ringer.Pattern{i}
KEEPALIVETIME${L}	VoiceService{i}.VoiceProfiles{i}.SIP.RegistrationPeriod
CALLERID${L}ONOFF	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.CallerIDEnable
CALLERIDNAME${L}	VoiceService{i}.VoiceProfiles{i}.Line{i}.CallingFeature.CallerIDName
ALERTINGPARK${L}	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.ALERTINGPARK${L}
L${L}CODEC${F3CALS}	VoiceService{i}.VoiceProfiles{i}.Line{i}.Codec.List{i}.Enable
	VoiceService{i}.VoiceProfiles{i}.Line{i}.Codec.List{i}.SilenceSuppression
	VoiceService{i}.VoiceProfiles{i}.Line{i}.VoiceProcessing.EchoCancellationEnable
	VoiceService{i}.VoiceProfiles{i}.Line{i}.Codec.List{i}.PacketizationPeriod
	VoiceService{i}.VoiceProfiles{i}.Line{i}.Codec.List{i}.DTMFMethod
	VoiceService{i}.VoiceProfiles{i}.Line{i}.Codec.List{i}.Priority
L${L}FLASH_OOB	X_000f5d.VoiceService{i}.VoiceProfiles{i}.CallingFeature.L${L}FLASH_OOB
T38 FAX	
T38RMAN	X_000f5d.VoiceService{i}.VoiceProfiles{i}.FAXT38T38RMAN
T38_ECC_COUNT	VoiceService{i}.VoiceProfiles{i}.FAXT38.LowSpeedRedundancy
T38PROT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.FAXT38T38PROT
T38ECT	X_000f5d.VoiceService{i}.VoiceProfiles{i}.FAXT38T38ECT
RINGSIGNAL	
RING_CADENCE_${RING_C}	VoiceService.{i}.VoiceProfile.{i}.Line.{i}.Ringer
RINGTONES	
CALLFORWARD	Tone configuration and assignment in TR-104 is completely different from DRG. In TR-104, tones are taken from the defined patterns and description instead of direct assignment as in DRG. Example below is assigning DIALTONE and BUSYTONE for DRG using TR-104 parameter. In DRG, the assignment will be: DIALTONE=425@-5#ON(1000),R BUSY=425@-5#ON(250) OFF(250),R
DTMF_D	
DTMF_C	
DTMF_A	
BUSY	
DTMF_B	
RINGBACK	
DTMF_POUND	
CALLWAITING	
OFF_HOOK_WARN	
NETWORK_BUSY	
REORDER	
RINGTONE_${RING_T}	
DTMF_STAR	
CUSTOM_${C}	
STUTTER_DIAL	
MSG_WAIT_INDICATOR	
CONFIRM	
DTMF_CONT_STAR	
DIALTONE	
DTMF_${DTMF_T}	
DTMF_CONT_${DTMF_T}	
DTMF_CONT_POUND	

Sample of RINGTONE creation:

VoiceService.{i}.VoiceProfile.{i}.
VoiceService.{i}.VoiceProfile.{i}.Tone.EventNumberOfEntries=21
VoiceService.{i}.VoiceProfile.{i}.Tone.Event.{1}.Function=Dialing
VoiceService.{i}.VoiceProfile.{i}.Tone.Event.{1}.ToneID=**101**
VoiceService.{i}.VoiceProfile.{i}.Tone.Event.{2}.Function=UserBusy

VoiceService.{i}.VoiceProfile.{i}.Tone.Event.{2}.ToneID=**104**
Description.
VoiceService.{i}.VoiceProfile.{i}.Tone.DescriptionNumberofEntries=6
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{1}.EntryID=**101** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{1}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{1}.TonePattern=**1011** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneName=NormalDialtone VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{1}.ToneFile= VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{1}.ToneRepetition=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.EntryID=**102** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.TonePattern=**1012** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneName=USDialtone VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneFile= VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneRepetition=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{3}.EntryID=**103** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{3}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{3}.TonePattern=**1013** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{2}.ToneName=Howlertone VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{3}.ToneFile= VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{3}.ToneRepetition=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{4}.EntryID=**104** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{4}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{4}.TonePattern=**1014** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{4}.ToneFile= VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{4}.ToneRepetition=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.EntryID=**105** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.TonePattern=**1012** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.ToneFile= VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.ToneRepetition=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{6}.EntryID=**106** VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{6}.ToneEnable=true VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{6}.TonePattern=0 VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{5}.ToneFile=dialtone_default.wav VoiceService.{i}.VoiceProfile.{i}.Tone.Description.{6}.ToneRepetition=0
Pattern.
VoiceService.{i}.VoiceProfile.{i}.Tone.PatternNumberOfEntries=5
VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{1}.EntryID=**1011** VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{1}.ToneOn=true VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{1}.Frequency1=425 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{1}.Power1=-15 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{1}.Duration=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.EntryID=**1012** VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.ToneOn=true VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.Frequency1=625 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.Power1=-15 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.Frequency2=425 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.Power2=-15 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{2}.Duration=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{3}.EntryID=**1013** VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{3}.ToneOn=true VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{3}.Frequency1=625

VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{3}.Power1=-3 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{3}.Duration=0
VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.EntryID=**1014** VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.ToneOn=true VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.Frequency1=425 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.Power1=-15 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.Duration=500 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{4}.NextEntryID=**1015**
VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.EntryID=**1015** VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.ToneOn=true VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.Frequency1=0 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.Power1=0 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.Duration=500 VoiceService.{i}.VoiceProfile.{i}.Tone.Pattern.{5}.NextEntryID=**1014**

COS/CCS 2008-26

www.kth.se