國立臺灣大學電機資訊學院電機電信電子產業研發碩士專班

碩士論文

Industrial Technology R&D Master Program in Electrical, Communication and Electronics Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於用戶端設備廣域網路管理協定的設備管理架構

Device Management Framework Based on CPE WAN Management Protocol

林信丞

Hsing-Cheng Lin

指導教授：郭斯彥 博士

Advisor: Sy-Yen Kuo, Ph.D.

中華民國 九十七 年 六 月

June, 2008

誌謝

本論文能順利圓滿完成，首要感謝指導教授郭斯彥教授兩年來的指導與教誨，提供了完善的研究環境、資源讓學生能完成兩年碩士班的學業和論文。也要感謝口試委員雷欽隆教授、顏嗣鈞教授、陳英一教授和呂毓榮博士對於學生論文上的指正與建議。

在研究所期間參與寬廣計畫收穫匪淺，要感謝呂毓榮學長的帶領、提供研究方向，並且時時提供問題的解決方針、指點迷津，並且提供寶貴的意見給我參考。感謝炯霖學長和彥呈學長，提供我遇到問題的解答或是解決方向，並且予以協助。參與寬廣計畫，對於論文的完成貢獻良多，謝謝你們的指導，讓我得以完成此論文。感謝昀諭、彥能同學、承宇學弟，和我一起在計畫及課業上研究、互相學習及幫助。

最要感謝的是我的父母，感謝你們不辭辛勞的養育，讓我能在良好、衣食無虞的環境下學習、成長，感謝父親的諄諄教誨，讓我依照我自己的興趣去學習、發展，充分支持我的決定。感謝母親無微不至的呵護，讓我能有更多的時間在課業上學習。感謝哥哥，你是我從小學習的標竿。在此，我把這本論文以及這份榮耀獻給你們，謝謝你們。

中文摘要

一般來說，家用網路設備實際安裝於用戶家中，市面上的設備大都有區域網路下的管理介面，傳統做法為服務供應商的技術人員到府服務，透過區域網路的管理介面，進行家用網路設備安裝配置或故障檢測排除。

TR－069 是由 DSL Forum 所開發的技術規範之一，其全稱為「CPE 廣域網管理協議」。它提供了對下一代網絡中家庭網絡設備進行管理配置的通用框架和協議，用於對家庭網絡中的網管、路由器、機頂盒等設備進行遠程集中管理。

這些設備不論是在最開始安裝的時候還是在後期運行中的業務配置變更或是出現故障需要維護的時候，都需要通過管理接口對設備進行配置或是診斷。現階段的設備大都提供在 LAN 側的管理配置接口和界面，因此傳統的做法是運營商的維護人員上門進行安裝或調整設備，通過 LAN 側管理接口做一些設備配置或故障診斷的工作。但是，這種一對一的人工服務方式顯然運行效率不高而且需要花費大量的人力。隨著運營商家庭網絡業務的開展，將會有大量的設備需要安置在用戶家中，採用過去人工方式對這些設備進行維護和管理將會成為一個巨大的負擔。

本篇論文的目的在於瞭解用戶端設備廣域網管理協議，詳細討論自動配置伺服器和用戶端設備廣域網管理協議之間的連接架構，傳輸方式，並以 TR－069 為基礎，設計一個架構可將自動配置伺服器及用戶端設備分別置入所需設備中。

關鍵字：

閘道器、簡單網絡管理協議、簡單物件接取協定、可擴展標示語言、動態網域名稱服務、網路語音通訊協定、用戶端設備廣域網路管理協定、自動配置伺服器

Abstract

Generally speaking, the home network equipment is installed in user's family actually, the equipment on the market mostly have a management interface under regional networks, the traditional method is gone to the office to serve for the service supplier's technical staff, pass the management interface of the regional network, install and configure the home network equipment or the trouble is measured and got rid of.

TR- 069 is one of the technical specifications developed by DSL Forum, its full name' CPE WAN Management Protocol'. It has offered and carried on the frame and protocol; in common use disposed in management to the family network equipment in the network of future generation, used for carrying on long-range centralized management to such equipment as the network management, router, Set Top box in family's network, etc.

No matter the equipment is in the beginning to install or change the configuration on business while operating on later stage. When the equipment breakdown needs to repair, all required to configuration or diagnose the equipment through the managing port. The equipment of the present stage mostly offered the managing port and interface on the

LAN-side. Therefore the traditional method is that the distributor’s repairer went to your place and installed or debugged the equipment through the managing port on the LAN-side to configure the equipment or diagnose the breakdown. However, it is obvious that the one by one service method is low efficiency and need a lot of manpower. With the development of operator's family's network business, have a large amount of equipment need to dispose at the user's place. Adopting artificial way to maintain and manage the equipment becomes an enormous burden.

The purpose of this thesis lies in understanding users end equipment WAN manage the agreement, ,discuss in detail that disposes the server and users end equipment WAN and manages the connection structure between the agreements automatically, transmission means, based on TR- 069, design one structure can dispose server and user end equipment put into the necessary equipment respectively automatically.

Keywords: Gateway, SNMP, SOAP, XML, Dynamic DNS, VoIP, CPE WAN Management Protocol, ACS.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

The recent trend to integrate networking technologies into home devices is paving the way for significant enhancements to the user's entertainment experience. In parallel, it creates new business opportunities for Service Providers who can exploit the rich capabilities of modern home devices to offer various services through the internet. Consumers will be able to gain access to a wide range of multimedia services provided not only from within the home network (local services) but increasingly also from the public network (remote services).

A key factor to achieve the massive deployment of these network originated services is the level of convenience regarding the initial installation, configuration and life-cycle management of the hosting home devices. Minimal or even zero user intervention is an essential requirement. Universal Plug and Play (UPnP) [1] is the most widely adopted interoperability technology that addresses automatic installation and configuration of devices and services. It uses existing Internet standards (i.e. IP, TCP, UDP, HTTP, SOAP and XML) and defines mechanisms for discovery, control, event and presentation of devices and services. However, the UPnP protocol was initially designed having in mind the proximity networks. The broadcast messages for discovery (SSDP) and notification (GENA) purposes, the use of unreliable UDP protocol and the lack of security mechanisms are some of the main reasons that make UPnP unsuitable to support the management of devices and services through the public network.

In contrast to UPnP's local scope, the Open Services Gateway initiative (OSGi) seems to be the most widely adopted architecture for enabling the remote delivery of services. It defines a standard platform, where various software components (bundles) can be added to, updated or removed "on the fly" from a device without ever disrupting its operation. OSGi is a Java-based framework and focuses rather on services, than on the actual devices through which the services are consumed. Therefore, it does not address issues like device discovery, event, model and control for remote management purposes.

The Simple Network Management Protocol (SNMP) has been for long the standard solution for the remote management of network entities. However, a set of disadvantages like the use of unreliable UDP protocol [6] and the inability to describe complex structures [7], make it unsuitable to support the next generation devices on a massive scale and within multi-service environments.

Addressing primarily the remote delivery of services, the DSL Forum attempted to fill all the above gaps by specifying the CPE WAN Management Protocol (or simpler TR-069) [9]. This protocol seems to be a promising solution for the configuration and management of CPEs not only in DSL, but also in every environment that can offer multiple services to home networks. The Customer Premises Equipment (CPE) term typically involves residential gateways, PCs, set-top boxes, Video/VoIP phones, cameras, stereos and many other conceivable networked consumer electronics devices. Moreover, a family of interrelated standards for LAN-side (i.e. UPnP-based) [10], [11] and WAN-side (i.e. TR-069 based) [13]-[16] configuration of CPE's expands the scope and importance of the DSL Forum's normative work.

1.1 Background and Motivation

Generally speaking, the home network equipment is installed in user's family actually, the equipment on the market mostly have a management interface under regional networks, the traditional method is gone to the office to serve for the service supplier's technical staff, pass the management interface of the regional network, install and configure the home network equipment or the trouble is measured and got rid of.

The traditional view thinks, home Gateway is the last network node that the telecommunication service network entered the family, since these home network equipment lie in user's family, regard home Gateway as the general peripheral device of computer, if the display, outer burning and recording machine, etc., take charge of management by user, service supplier need not take charge of managing to the peripheral device lying telecommunications in the family, in order to save expensive cost of labor and make business simple to take .

In recent years, with flourishing development, telecommunication of service item, home network heighten to telecommunication service demand that quality have concurrently, home Gateway is no longer the final network node that telecommunication service network entered the family, but become the assembling and clicking of service item of innovative telecommunications, and derive more relevant business from this, create more additional value. Because users are the general people, if family network equipment is managed under the care of user's proper motion, the relatively unable professional and technical personnel like the project department of enterprises, can manage the home networking equipment according to the standard operation procedure, and users generally lack the concept of the information safety, therefore may cause a lot of online security questions.

Thus, the service supplier of telecommunications will be unable to guarantee telecommunication service quality, estimate and safeguard the cost unable to be accurate; even will limit the development and innovation of this field. Nowadays, the service supplier of telecommunications carries to the service quality of the end in order to guarantee the telecommunication service, not only should include home Gateway in managing but also act as agent through home Gateway conduct, further management is in the home equipment of following numerous connections network, such as several plane box, VoIP, etc., his unified centralized management.

Therefore **TR-069** (short for Technical Report 069) is the way to reach this purpose. It defines an application layer protocol for remote management of end-user devices.

1.2 Research Contribution

The main goal of this thesis is to provide a structure for communication between a CPE (Customer Premises Equipment) and Auto-Configuration Server (ACS). Thus, the thesis introduces the three features: (1) CPE WAN Management Protocol, (2) Auto-Configuration Server (ACS), and (3) CPE (Customer Premises Equipment). Also, UPnP, OSGi and SNMP will be described.

1.3 Thesis Outline

This thesis is constructed as follows.

Chapter 1 introduces the situation of remote configuration today. It describes the background, motivation and goal of this thesis. Then the research contribution of this thesis is stated. Finally it introduces the outline of the thesis.

Chapter 2 introduces the main technologies and protocol used in the thesis. Features of SOAP and SNMP are introduced. Readers who are familiar with these technologies could skip this chapter.

Chapter 3 introduces the CPE WAN Management Protocol which is intended to support a variety of functionalities to manage a collection of CPE, security goal and full protocol support.

Chapter 4 we realize the Protocol Component, Security Mechanisms and Architectural Components.

Chapter 5 Integration the TR-069 Protocol and understand the all procedures on CPE WAN Management Protocol. Then design the gateway architecture and system architecture.

Chapter 6 summarizes the results of the thesis and indicates the future works and suggests some further research topics.

Bibliography lists all standards, RFCs, documents, even web sites, which have been referenced.

Appendix A provides a glossary of the terms used in this thesis.

Chapter 2
Background and Related Works

In this chapter, we will introduce the main technologies and protocol used in the thesis. Features of UPnP, OSGi, SOAP, NAS and SNMP are introduced. Readers who are familiar with these technologies could skip this chapter.

2.1 SOAP

SOAP is a protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP forms the foundation layer of the web services protocol stack providing a basic messaging framework upon which abstract layers can be built.

There are several different types of messaging patterns in SOAP, but by far the most common is the Remote Procedure Call (RPC) pattern, in which one network node (the *client*) sends a request message to another node (the *server*) and the server immediately sends a response message to the client. SOAP is the successor of XML-RPC, though it borrows its transport and interaction neutrality and the envelope/header/body from elsewhere, probably from WDDX.

2.1.1 Transport methods

SOAP makes use of an Internet application layer protocol as a transport protocol. Critics have argued that this is an abuse of such protocols, as it is not their intended purpose and therefore not a role they fulfill well. Backers of SOAP have drawn analogies to successful uses of protocols at various levels for tunneling other protocols.

Both SMTP and HTTP are valid application layer protocols used as transport for SOAP, but HTTP has gained wider acceptance as it works well with today's internet infrastructure; specifically, HTTP works well with network firewalls. SOAP may also be used over HTTPS (which is the same protocol as HTTP at the application level, but uses an encrypted transport protocol underneath) in either simple or mutual authentication; this is the advocated WS-I method to provide web service security as stated in the WS-I Basic Profile 1.1. This is a major advantage over other distributed protocols like GIOP/IIOP or DCOM which are normally filtered by firewalls. XML was chosen as the standard message format because of its widespread use by major corporations and open source development efforts. Additionally, a wide variety of freely available tools significantly eases the transition to a SOAP-based implementation.

The somewhat lengthy syntax of XML can be both a benefit and a drawback. While it promotes readability for humans, it can retard processing speed and be cumbersome.

For example, CORBA, GIOP, ICE, and DCOM use much shorter, binary message formats. On the other hand, hardware appliances are available to accelerate processing of XML messages. Binary XML is also being explored as a means for streamlining the throughput requirements of XML.

2.2 Network-attached storage

Network-attached storage (NAS) is a file-level computer data storage connected to a computer network providing data access to heterogeneous network clients.

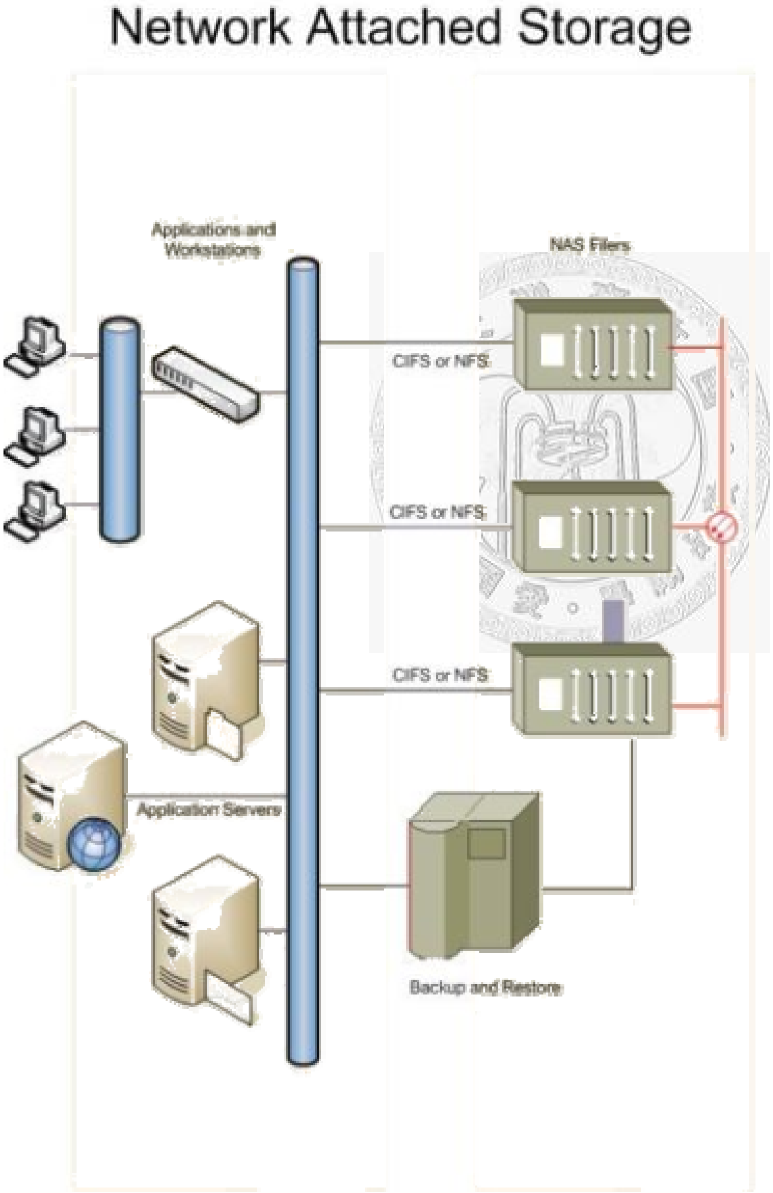
NAS hardware is similar to the traditional file server equipped with direct attached storage. However it differs considerably on the software side. The operating system and other software on the NAS unit provide only the functionality of data storage, data access and the management of these functionalities. Use of NAS devices for other purposes (like scientific computations or running database engine) is strongly discouraged. Many vendors also purposely make it hard to develop or install any third-party software on their NAS device by using closed source operating systems and protocol implementations .In other words, NAS devices are server appliances. NAS units also usually have a web interface as opposed to keyboard/video/mouse. Often minimal-functionality or stripped-down operating systems are used on NAS devices.

For example FreerNAS, which is open source NAS software meant to be deployed on standard computer hardware, is in fact a "leaned-out" version of FreeBSD. Likewise, NexentaStor is based upon the core of the NexentaOS, an open source hybrid operating system with an OpenSolaris core and a Linux user environment. NAS systems usually contain one or more hard disks, often arranged into logical, redundant storage containers or RAIDs (redundant arrays of independent disks), as do traditional file servers. NAS removes the responsibility of file serving from other servers on the network. NAS uses file-based protocols such as NFS (popular on UNIX systems) or SMB (Server Message Block) (used with MS Windows systems). NAS units rarely limit clients to only one protocol.

NAS provides both storage *and* file system. This is often contrasted with SAN (Storage Area Network), which provides only block-based storage and leaves file system concerns on the "client" side. SAN protocols are SCSI, Fibre Channel, iSCSI, ArTA over Ethernet, or HyperSCSI.

The boundaries between NAS and SAN systems are also starting to overlap, with some products making the obvious next evolution and offering both file level protocols (NAS) and block level protocols (SAN) from the same system. However a SAN device is usually served through NAS as one large flat file, not as a file system. An example of this is Open filer, a free product running on Linux.

Fig. 1 Network Attached Storage



2.3 SNMP

SNMP [9] is a protocol that can manage and monitor network devices and get information and conditions about devices. It can manage single device or multiple devices. It consists of a set of standards for network management, including an Application Layer protocol of the OSI model, a database schema, and a set of data objects. Management data are in the form of variables on the managed systems, which can describe the system configuration. These variables can then be queried and set by our managing applications.

SNMP has five core protocol data units shown as follows.

1. GET REQUEST - used to retrieve a piece of management information.

2. GETNEXT REQUEST - used iteratively to retrieve sequences of management information.

3. GET RESPONSE - used by the agent to respond with data to get and set requests from the manager.

4. SET REQUEST - used to initialize and make a change to a value of the network element.

5. TRAP - used to report an alert or other asynchronous event about a managed system.

A SNMP-managed network is made up with three key components. They are managed devices, agents and network-management systems (NMSs).

A managed device contains a SNMP agent. A SNMP agent is a software module that can store and collect management information and make this information available to NMSs. Agents can translates that information into a form compatible with SNMP. Managed devices can be routers and gateways, switches and bridges, hubs, or computer hosts. A NMS contains a software module that monitor and control managed devices. The management network is shown in Figure 2.

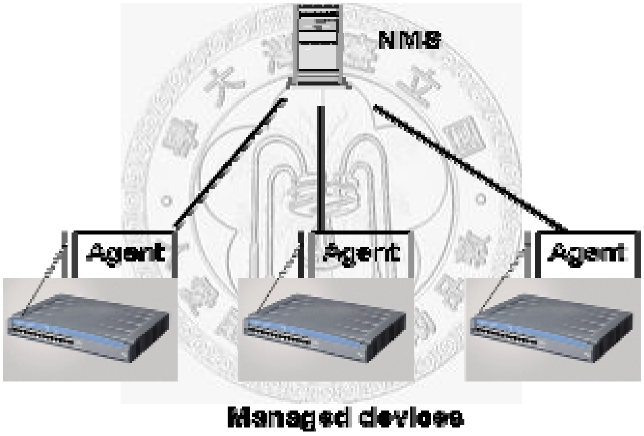


Fig. 2 SNMP management network

Agent will manage and store information in the MIBs (Management Information Bases [10]) in volatile memory. When getting SET REQUEST or other request from NMSs, agents will access the MIBs in volatile memory to reply the request.

2.4 Summary

In this chapter, we introduce the main technologies and protocol used in the thesis. Features of SOAP and SNMP are introduced. Readers who are familiar with these technologies could skip this chapter. This chapter introduces the basic knowledge and next two chapters we will introduce TR-069, realize the CWMP's architecture and requirement based on this knowledge.

Chapter 3　TR-069 Introduction

TR- 069 is one of the technical specifications developed by DSL Forum, its full name' CPE WAN Management Protocol'. It has offered and carried on the frame and protocol; in common use disposed in management to the family network equipment in the network of future generation, used for carrying on long-range centralized management to such equipment as the network management, router, Set Top box in family's network, etc.

No matter the equipment is in the beginning to install or change the configuration on business while operating on later stage. When the equipment breakdown needs to repair. All required to configuration or diagnose the equipment through the managing port. The equipment of the present stage mostly offers the managing port and interface on the LAN-side. Therefore the traditional method is that the distributor's repairer went to your place and installed or debugged the equipment through the managing port on the LAN-side to configure the equipment or diagnose the breakdown. However, it is obvious that the one by one service method is low efficiency and need a lot of manpower. With the development of operator's family's network business, have a large amount of equipment need to dispose at the user's place. Adopting artificial way to maintain and manage the equipment becomes an enormous burden.

The appearance of TR- 069, just in order to solve the difficult problem of such a service, in TR- 069 frames defined, include two kinds of logic equipment mainly: Receive the equipment of user of management and management server (ACS). Under the environment of family's network, need to carry on the equipment of disposing and management from the network side, equipment usually directly correlated with operator's business such as turning off, the roof box, IP telephone terminal station, etc. Such work as and all disposition, diagnosis correlated with user's equipment, upgrading, etc. are finished by unified management server ACS.

3.1 Functional Components

The CPE WAN Management Protocol is intended to support a variety of functionalities to manage a collection of CPE, including the following primary capabilities:

3.1.1 Auto-Configuration and Dynamic Service Provisioning

The CPE WAN Management Protocol allows an ACS to provision a CPE or collection of CPE based on a variety of criteria. The provisioning mechanism includes specific provisioning parameters and a general mechanism for adding vendor-specific provisioning capabilities as needed.

The provisioning mechanism allows CPE provisioning at the time of initial connection to the broadband access network, and the ability to re-provision at any subsequent time. This includes support for asynchronous ACS-initiated re-provisioning of a CPE.

The identification mechanisms included in the protocol allow CPE provisioning based either on the requirements of each specific CPE, or on collective criteria such as the CPE vendor, model, software version, or other criteria.

The protocol also provides optional tools to manage the CPE-specific components of optional applications or services for which an additional level of security is required to control, such as those involving payments.

The provisioning mechanism allows straightforward future extension to allow provisioning of services and capabilities not yet included in this version of the specifications.

3.1.2 Software/Firmware image Management

The CPE WAN Management Protocol provides tools to manage downloading of CPE software/firmware image files. The protocol provides mechanisms for version identification, file download initiation (ACS initiated downloads and optional CPE initiated downloads), and notification of the ACS of the success or failure of a file download.

The CPE WAN Management Protocol also defines a digitally signed file format that may optionally be used to download either individual files or a package of files along with explicit installation instructions for the CPE to perform. This signed package format ensures the integrity of downloaded files and the associated installation instructions, allowing authentication of a file source that may be a party other than the ACS operator.

3.1.3 Status and Performance Monitoring

The CPE WAN Management Protocol provides support for a CPE to make available information that the ACS may use to monitor the CPE’s status and performance statistics. The protocol defines a common set of such parameters, and provides a standard syntax for vendors to define additional non-standard parameters that an ACS can monitor. It also defines a set of conditions under which a CPE should actively notify the ACS of changes.

3.1.4 Diagnostics

The CPE WAN Management Protocol provides support for a CPE to make available information that the ACS may use to diagnose connectivity or service issues. The protocol defines a common set of such parameters and a general mechanism for adding vendor-specific diagnostic capabilities.

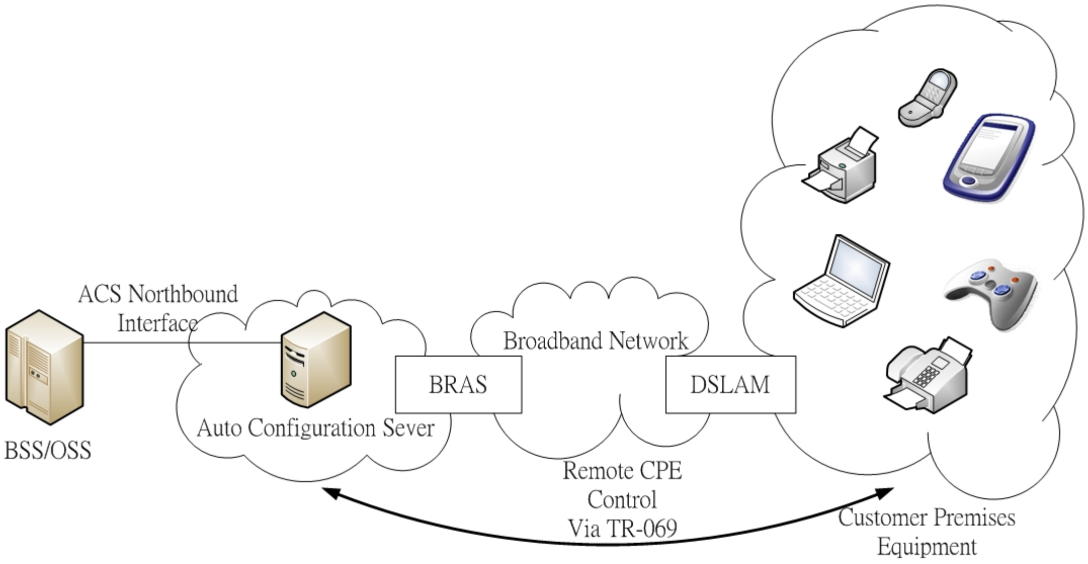
3.2 Positioning in the Auto-Configuration Architecture

TR-046[2] describes the overall framework for B-NT auto-configuration. This process consists of three sequential stages, each of which is focused on a specific aspect of the overall B-NT auto-configuration process.

The procedures for the first two stages of B-NT auto-configuration are specified in TR-062[3] and TR-044[4]. These define the ATM layer and IP layer auto-configuration procedures, respectively, used to initiate basic broadband connectivity.

The third stage of auto-configuration defined in TR-046 covers “auto-configured complex services.” In the case of a B-NT, the CPE WAN Management Protocol relates primarily to this third stage. Specifically, the CPE WAN Management Protocol is proposed as the protocol to be used on the ACS-Southbound Interface between an Auto-Configuration Server (ACS), and a B-NT as shown in Figure 3.

Fig. 3 Positioning in the Auto-Configuration Architecture



While the CPE WAN Management Protocol is targeted at management of B-NTs, this protocol may be used to manage other types of CPE as well, including stand-alone routers and LAN-side client devices, as also shown in Figure 1. Unless otherwise indicated, the CPE WAN Management Protocol as defined in this specification applies to any such managed device. Portions of this specification that apply only to a B-NT are explicitly indicated in the text. This specification includes a complete definition of the CPE parameter model for a B-NT. The corresponding parameter model for other specific device types is beyond the scope of this specification.

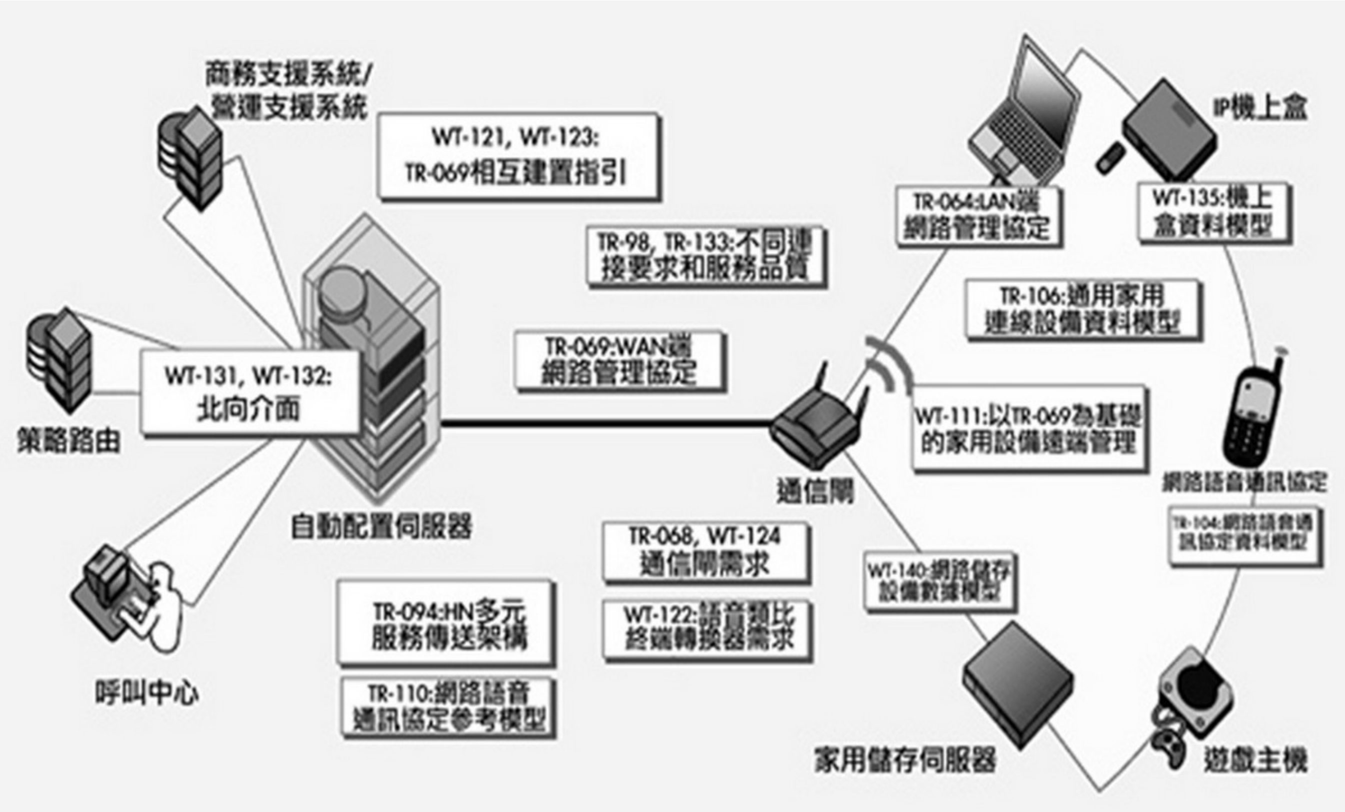
3.2.1 Full Protocol Support

The key functions offered by these protocols are as follows:

- TR-069 provides the extensible, secure, communications layer, while also providing basic gateway router and Wi-Fi configuration and management functionality.
- TR-098 provides QoS functionality as well configuration profiles to ease management and deployment.
- TR-104 and TR-110 combine to provide remote VoIP device configuration and management.
- TR-106 and TR-111 combine to allow the remote management of devices on a LAN, even those using the private IP space behind a NAT gateway.
- WT-107 provides a data model for MoCA devices, as well as other functionality
- TR-135 provides for the configuration and management of Set Top Boxes (STB). Note, unless the STB is an edge device, TR-106 and TR-111 support will also be required.

- TR-140 provides for the configuration and management of Network Attached Storage (NAS). Note, unless the NAS is an edge device, TR-106 and TR-111 support will also be required.

Fig. 4 Full Protocol Support

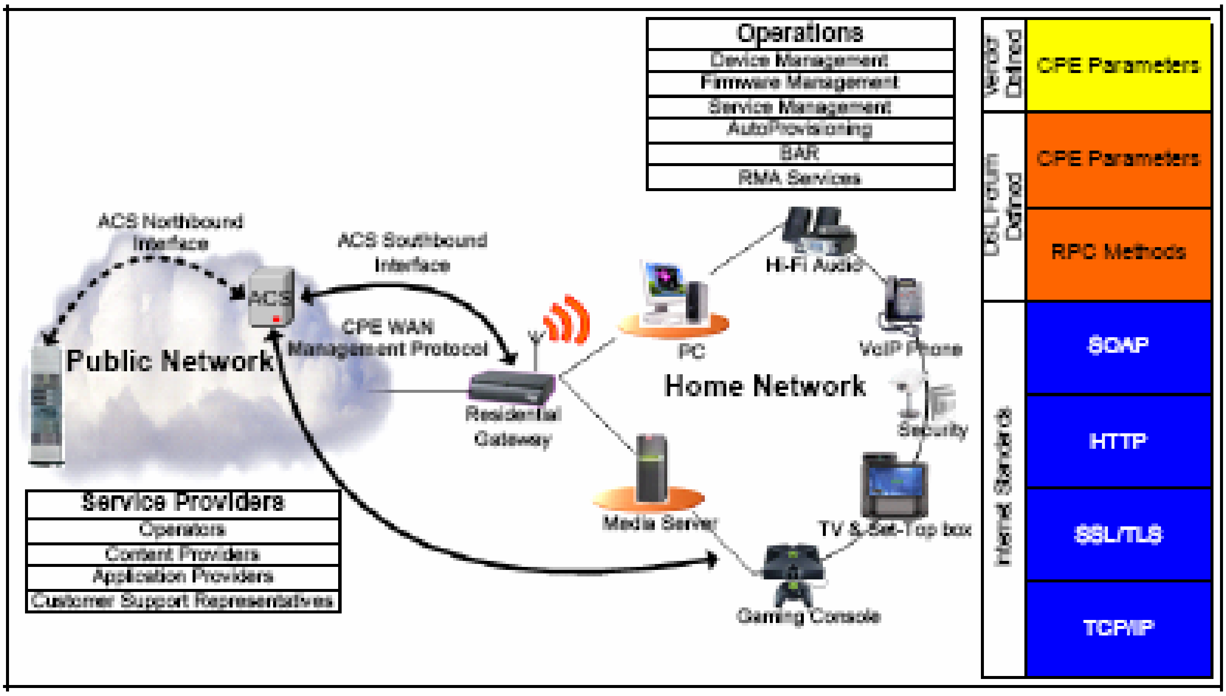


3.3Summary

In this chapter, we introduce the CPE WAN Management Protocol which is intended to support a variety of functionalities to manage a collection of CPE , security goal and full protocol support.

Chapter 4
TR-069 Architecture

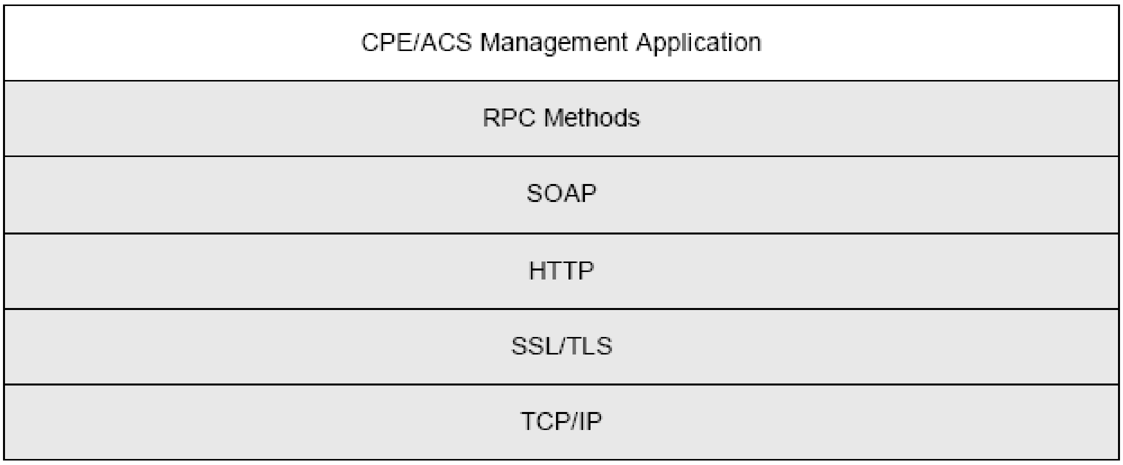
Fig. 5 Target Environment and Protocol Stack of the CPE WAN Management Protocol



4.1 Protocol Component

The CPE WAN Management Protocol comprises several components that are unique to this protocol, and makes use of several standard protocols. The protocol stack defined by the CPE WAN Management Protocol is shown in Figure 6. A brief description of each layer is provided in Table 1.

Fig. 6 Protocol stack



CPE/ACS Management Application

RPC Methods

SOAP

HTTP

SSL/TLS

TCP/IP

Table. 1

Layer	Description
CPE/ACS Application	The application uses the CPE WAN Management Protocol on the CPE and ACS, respectively. The application is locally defined and not specified as part of the CPE WAN Management Protocol.
RPC Methods	The specific RPC methods that are defined by the CPE WAN Management Protocol. This includes the definition of the CPE Parameters accessible by an ACS via the Parameter-related RPC Methods. The specific Parameters defined for an Internet Gateway Device.
SOAP	A standard XML-based syntax used here to encode remote procedure calls.
HTTP	HTTP 1.1.
SSL/TLS	The standard Internet transport layer security protocols. Specifically, either SSL 3.0 (Secure Socket Layer), or TLS 1.0 (Transport Layer Security) .Use of SSL/TLS is RECOMMENDED but is not required.
TCP/IP	Standard TCP/IP.

4.2 Security Mechanisms

The CPE WAN Management Protocol is designed to allow a high degree of security in the interactions that use it. The CPE WAN Management Protocol is designed to prevent tampering with the transactions that take place between a CPE and ACS, provide confidentiality for these transactions, and allow various levels of authentication.

The following security mechanisms are incorporated in this protocol:

- The protocol supports the use of SSL/TLS for communications transport between CPE and ACS. This provides transaction confidentiality, data integrity, and allows certificate-based authentication between the CPE and ACS.
- The HTTP layer provides an alternative means of CPE authentication based on shared secrets.

The protocol includes additional security mechanisms associated with the optional signed Voucher mechanism and the Signed Package Format.

4.2.1 Security Initialization Models

Initialization of the security mechanisms is described in the context of various business models for CPE distribution. Three models are considered:

- Distribution of CPE by the service provider associated with the ACS.
- Retail distribution of the CPE, where association of the CPE with the service provider

and customer is done at the time of purchase.

- Retail distribution where no pre-association with the CPE is done.

In the first two cases, the specific identity of the CPE can be known to the ACS before the CPE is first used. In these cases, the following mechanisms may be used in Table.2:

Table. 2

Authentication of	**Type used**	**Description**
ACS	Shared secret	Shared secret must be pre-loaded into CPE before the first use of the CPE.
	Certificate	Discovery of the ACS URL uniquely identifies the identity of the ACS for the purpose of certificate validation.
CPE	Shared secret	Shared secret must be provided to the ACS before the first use of the CPE.
	Certificate	The CPE may use online certificate enrollment with the CA associated with the ACS. The CPE must be provided with the information needed to contact this CA.

In the latter case of retail distribution of the CPE, there is no possibility of pre-association of the CPE with a particular ACS. The following table presents possible approaches to accommodating this case, but does not attempt to mandate a specific approach in Table.3:

Table. 3

Authentication of	**Type used**	**Description**
ACS	Shared secret	Not appropriate for this case.
	Certificate	Discovery of the ACS URL uniquely identifies the identity of the ACS for the purpose of certificate validation.

CPE	Shared secret	Possible alternatives, outside the scope of this specification: • Establish a common server for secure distribution of CPE shared secrets among multiple service providers. • Initial CPE to ACS connection of an unrecognized CPE could be allowed without authentication. ACS would then set the shared secret Parameter for subsequent access. Care in ACS implementation would be required to prevent denial of service attacks.
	Certificate	The CPE may use online certificate enrollment with the CA associated with the ACS. The CPE must be provided with the information needed to contact this CA, which could be incorporated into the discovery process.

4.3 Architectural Component

4.3.1 Parameters

The RPC Method Specification defines a generic mechanism by which an ACS can read or write Parameters to configure a CPE and monitor CPE status and statistics. The particular list of defined Parameters for an Internet Gateway Device.

Each Parameter consists of a name-value pair. The name identifies the particular Parameter, and has a hierarchical structure similar to files in a directory, with each level separated by a "." (dot). The value of a Parameter may be one of several defined data types.

Parameters may be defined as read-only or read-write. Read-only Parameters may be used to allow an ACS to determine specific CPE characteristics, observe the current state of the CPE, or collect statistics. Writeable Parameters allow an ACS to customize various aspects of the CPE's operation. All writeable Parameters must also be readable. The value of some writeable Parameters may be independently modifiable through means other than the interface defined in this specification (e.g., some Parameters may also be modified via a LAN side auto-configuration protocol).

The protocol supports a discovery mechanism that allows an ACS to determine what Parameters a particular CPE supports, allowing the definition of optional parameters as well as supporting straightforward addition of future standard Parameters.

The protocol also includes an extensibility mechanism that allows use of vendor-specific Parameters in addition to those defined in this specification.

4.3.2 File Transfer

The RPC Method Specification defines a mechanism to facilitate file downloads or (optionally) uploads for a variety of purposes, such as firmware upgrades or vendor-specific configuration files.

When initiated by the ACS, the CPE is provided with the location of the file to be transferred, using HTTP or, optionally, HTTPS, FTP, or TFTP as the transport protocol. The CPE then performs the transfer, and notifies the ACS of the success or failure.

Downloads may be optionally initiated by a CPE. In this case, the CPE first requests a download of a particular file type from the ACS. The ACS may then respond by initiating the download following the same steps as an ACS-initiated download.

The CPE WAN Management Protocol also defines a digitally signed file format that may optionally be used for downloads.

4.3.3 CPE Initiated Notification

The RPC Method Specification defines a mechanism that allows a CPE to notify a corresponding ACS of various conditions, and to ensure that CPE-to-ACS communication will occur with some minimum frequency.

This includes mechanisms to establish communication upon initial CPE installation, to ‘bootstrap’ initial customized Parameters into the CPE. It also includes a mechanism to establish periodic communication with the ACS on an ongoing basis, or when events occur that must be reported to the ACS (such as when the broadband IP address of the CPE changes). The ACS must be aware of this event in order to establish incoming connections to the CPE.

In each case, when communication is established the CPE identifies itself uniquely via manufacturer and serial number information so that the ACS knows which CPE it is communicating with and can respond in an appropriate way.

4.3.4 Asynchronous ACS initiated Notifications

An important aspect of service auto-configuration is the ability for the ACS to notify the CPE of a configuration change asynchronously. This allows the auto-configuration mechanism to be used for services that require near-real-time reconfiguration of the CPE. For example, this may be used to provide an end-user with immediate access to a service or feature they have subscribed to, without waiting for the next periodic Inform interval.

The CPE WAN Management Protocol incorporates a mechanism for the ACS to issue a Connection Request to the CPE at any time, instructing it to establish a communication session with the ACS.

While the CPE WAN Management Protocol also allows polling by the CPE in lieu of ACS-initiated connections, the CPE WAN Management Protocol does not rely on polling or establishment of persistent connections from the CPE to provide asynchronous notification.

4.4 Summary

In this chapter, we realize the Protocol Component, Security Mechanisms and Architectural Components.

Chapter 5
Integration of the structure between CPE and ACS

5.1 ACS Discovery

The CPE WAN Management Protocol defines the following mechanisms that may be used by a CPE to discover the address of its associated ACS:

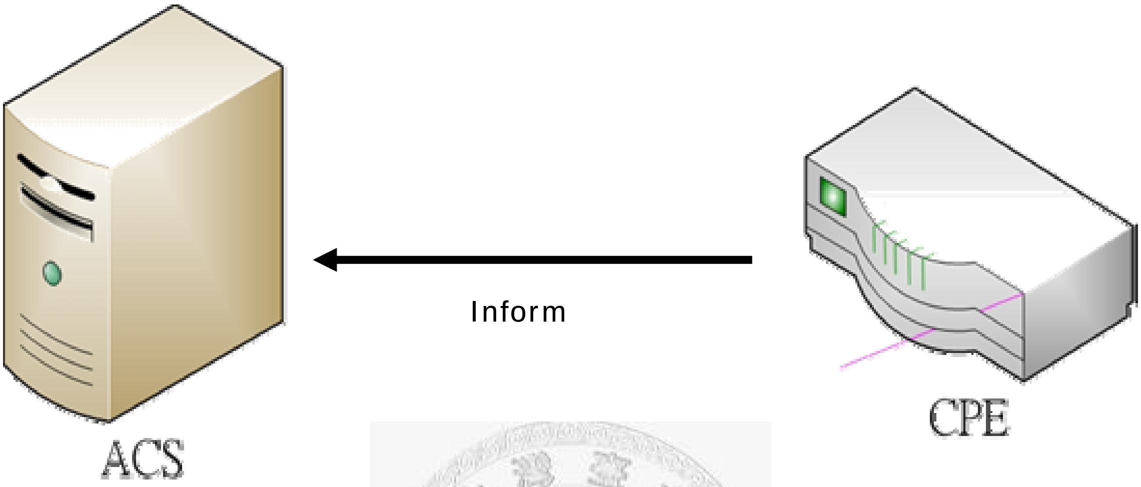
1. The CPE may be configured locally with the URL of the ACS. For example, this may be done via a LAN-side CPE auto-configuration protocol. The CPE would use DNS to resolve the IP address of the ACS from the host name component of the URL.

2. As part of the IP layer auto-configuration, a DHCP server on the access network may be configured to include the ACS URL as a DHCP option [12]. The CPE would use DNS to resolve the IP address of the ACS from the host name component of the URL. In this case a second DHCP option MAY be used to set the ProvisioningCode, which may be used to indicate the primary service provider and other provisioning information to the ACS.

A CPE identifies itself to the DHCP server as supporting this method by including the string “dslforum.org” (all lower case) anywhere in the Vendor Class Identifier (DHCP option 60).

3. The CPE may have a default ACS URL that it may use if no other URL is provided to it.(shown in Fig.7)

Fig. 7 The CPE may have a default ACS URL



The ACS URL MUST be in the form of a valid HTTP or HTTPS URL [5]. Use of an HTTPS URL indicates that the ACS supports SSL. If an HTTPS URL is given, and the CPE that does not support SSL, it MAY attempt to use HTTP assuming the remainder of the URL is unchanged. Once the CPE has established a connection to the ACS, the ACS may at any time modify the ACS address Parameter stored within the CPE (Internet Gateway Device. Management Server .URL). Once modified, the CPE MUST use the modified address for all subsequent connections to the ACS.

The “host” portion of the ACS URL is used by the CPE for validating the certificate from the ACS when using certificate-based authentication. Because this relies on the accuracy of the ACS URL, the overall security of this protocol is dependent on the security of the ACS URL.

The CPE SHOULD restrict the ability to locally configure the ACS URL to mechanisms that require strict security. The CPE MAY further restrict the ability to locally set the ACS URL to initial setup only, preventing further local configuration once the initial connection to an ACS has successfully been established such that only its existing ACS may subsequently change this URL.

The use of DHCP for configuration of the ACS URL SHOULD be limited to situations in which the security of the link between the DHCP server and the CPE can be assured by the service provider. Since DHCP does not itself incorporate a security mechanism, other means of ensuring this security should be provided

5.2 Connection Establishment

5.2.1 CPE Connection Initiation

The CPE may at any time initiate a connection to the ACS using the pre-determined ACS address. A CPE MUST establish a connection to the ACS and issue the Inform RPC method under the following conditions:

- The first time the CPE establishes a connection to the access network on initial installation

• On power-up or reset

• Once every PeriodicInformInterval.

• When so instructed by the optional ScheduleInform method

• Whenever the CPE receives a valid Connection Request from an ACS

• Whenever the URL of the ACS changes

• Whenever a parameter is modified that is required to initiate an Inform on change. In the case of an Internet Gateway Device, this includes changes to the following

- IP address of the default broadband connection
- Management IP address (associated with the Connection Request URL)
- Provisioning code
- Software version

• Whenever the value of a parameter that the ACS has marked for “active notification” via the SetParameterAttributes method is modified by an external cause (a cause other than the ACS itself). Parameter changes made by the ACS itself via SetParameterValues MUST NOT cause a new session to be initiated. If a parameter is modified more than once before the CPE is able to initiate a session to perform the notification, only one notification is performed. If a parameter is modified by an external cause while a session is in progress, the change causes a new session to be established after the current session is terminated (it MUST not effect the current session).

In order to avoid excessive traffic to the ACS, a CPE MAY place a locally specified limit on the frequency of parameter change notifications. This limit SHOULD be defined so that it is exceeded only in unusual circumstances. If this limit is exceeded, the CPE MAY delay by a locally specified amount initiation of a session to notify the ACS. After this delay, the CPE MUST initiate a session to the ACS and indicate all relevant parameter changes (those parameters that have been marked for notification) that have occurred since the last such notification.

The CPE SHOULD NOT maintain an open connection to the ACS when no more outstanding messages exist on the CPE or ACS.

5.2.2 ACS Connection Initiation

The ACS at any time request that the CPE initiate a connection to the ACS using the Connection Request notification mechanism.(shown in Fig.8) Support for this mechanism is REQUIRED in a CPE, and is RECOMMENDED in an ACS.

This mechanism relies on the CPE having an IP address that is routable from the ACS.

If the CPE is behind a firewall or NAT device lying between the ACS and CPE, the ACS may not be able to access the CPE at all. In this case, only CPE connection initiation is possible.(shown in Fig.9)

Fig. 8

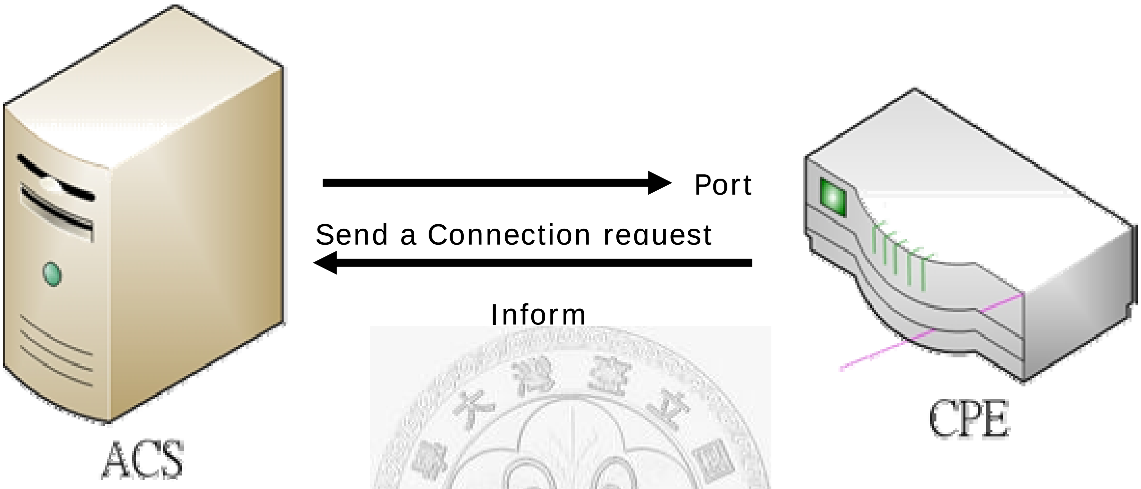
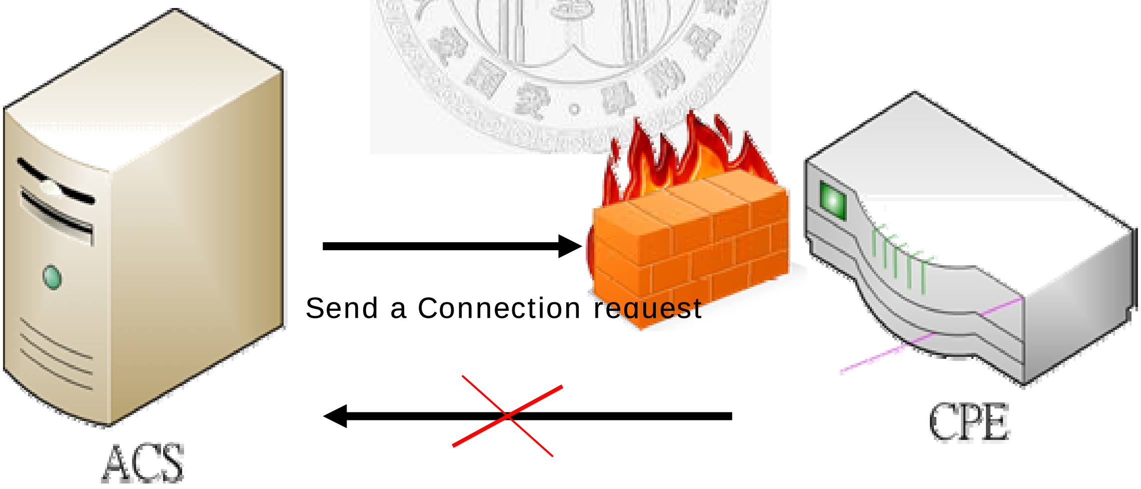


Fig. 9



The Connection Request notification mechanism is defined as follows:

- The Connection Request notification is an HTTP Get to a specific URL designated by the CPE. The URL value is available as read-only Parameter on the CPE. The path of this URL value SHOULD be randomly generated by the CPE so that it is unique per CPE.

- The Connection Request notification MUST make use of HTTP, not HTTPS. The associated URL MUST be an “http” URL.
- No data is conveyed in the Connection Request HTTP Get notification. Any data that might be contained SHOULD be ignored by the CPE.
- The CPE SHOULD use digest-authentication to authenticate the ACS before proceeding—the CPE SHOULD NOT initiate a connection to the ACS due to an unsuccessfully authenticated request. The shared-secret used to authenticate the ACS is available as a modifiable Parameter on the CPE.
- The CPE SHOULD restrict the number of Connection Request notifications it accepts during a given period of time in order to further reduce the possibility of a denial of service attack.
- The successful authentication of an HTTP Get to the designated port and URL causes the CPE to perform a fixed action: it establishes a session with the pre-determined ACS address, and once connected, it sends an Inform message.
- If the CPE is already in a session with the ACS when it receives a Connection Request notification, it MUST NOT terminate that session prematurely as a result.

This mechanism relies on the ACS having had at least one prior communication with the CPE via a CPE-initiated interaction. During this interaction, if the ACS wishes to allow future ACS-initiated transactions, it would read the value of the Internet GatewayDevice. Management Server. Connection Request URL Parameter. If the URL used for management access changes, the CPE must notify the ACS by issuing an Inform message indicating the new management IP address, thus keeping the ACS up-to-date.

5.2.3 File Transfers

If the CPE is instructed to perform a file transfer via the Download or Upload request from the ACS, and if the file location is specified as an HTTP URL with the same host name as the ACS, then the CPE MAY choose any of the following approaches in performing the transfer:

- The CPE MAY send the HTTP get/post over the already established connection. Once the file has been transferred, the CPE MAY then proceed in sending additional messages to the ACS while continuing to maintain the connection.
- The CPE MAY open a second connection over which to transfer the file, while maintaining the session to the ACS over which it may continue to send messages.

- The CPE MAY terminate the session to the ACS and then perform the transfer.

If the file location is not an HTTP URL or is not in the same domain as the ACS, then only the latter two options are available to it.

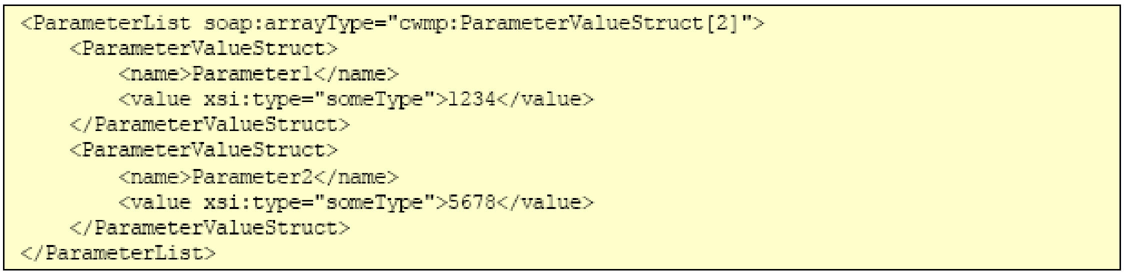
5.3 Use of SOAP

The CPE WAN Management Protocol defines SOAP 1.1 [8] as the encoding syntax to transport the RPC method calls and responses defined in Appendix A.

The following describes the mapping of RPC methods to SOAP encoding:

- The encoding must use the standard SOAP 1.1 envelope and serialization namespaces:
  - Envelope namespace identifier "http://schemas.xmlsoap.org/soap/envelope/"
  - Serialization namespace identifier "http://schemas.xmlsoap.org/soap/encoding/"
  - All elements and attributes defined as part of this version of the CPE WAN Management Protocol are associated with the following namespace identifier:
    - “urn:dslforum-org:cwmp-1-0”
- The data types used in Appendix A correspond directly to the data types defined in the SOAP 1.1 serialization namespace.
- For an array argument, the given argument name corresponds to the name of the

overall array element. No names are given for the individual member elements, so these should be named by their type. For example, an argument named ParameterList, which is an array of ParameterValueStruct structures, would be encoded as:



```
<ParameterList soap:arrayType="cwmp:ParameterValueStruct[2]">
    <ParameterValueStruct>
        <name>Parameter1</name>
        <value xsi:type="someType">1234</value>
    </ParameterValueStruct>
    <ParameterValueStruct>
        <name>Parameter2</name>
        <value xsi:type="someType">5678</value>
    </ParameterValueStruct>
</ParameterList>
```

- Regarding the SOAP specification for encoding RPC methods, each argument listed in the method call represents an [in] parameter, while each argument listed in the method response represents an [out] parameter. There are no [in/out] parameters used.

- The RPC methods defined use the standard SOAP naming convention whereby the response message corresponding to a given method is named by adding the “Response” prefix to the name of the method.

- A fault response MUST make use of the SOAP Fault element using the following conventions:

  - The SOAP faultcode element MUST indicate the source of the fault, either Client or Server, as appropriate for the particular fault. In this usage, Client represents the originator of the SOAP request, and Server represents the SOAP responder.

• The SOAP faultstring sub-element MUST contain the string “CWMP fault”.

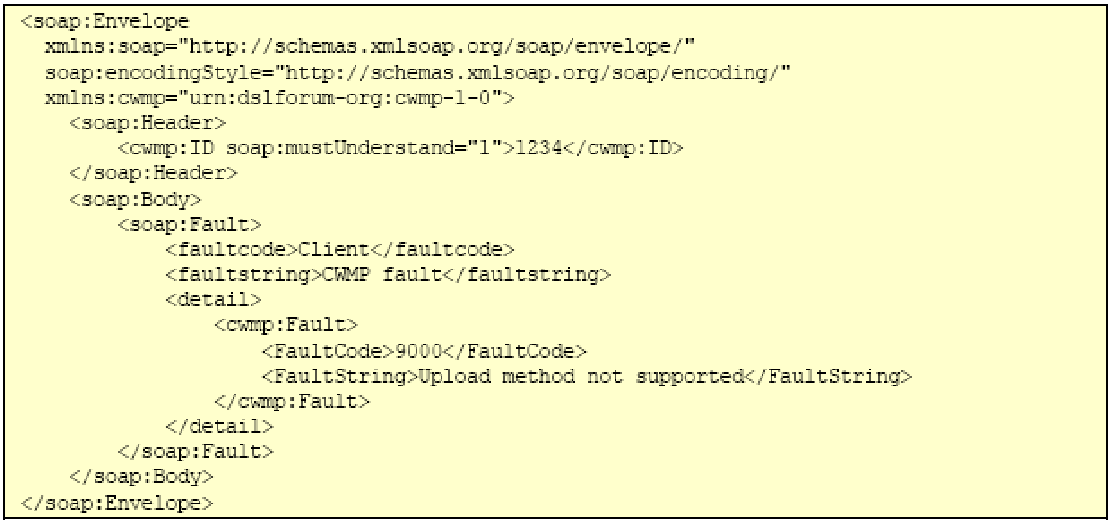
• The SOAP detail element MUST contain a Fault structure defined in the “urn:dslforum-org:cwmp-1-0” namespace. This structure contains the following elements:

- A FaultCode element that contains a single numeric fault code.
- A FaultString element that contains a human readable description of the fault.
- A SetParameterValuesFault element, to be used only in an error response to the SetParameterValues method, that contains a list of one or more structures indicating the specific fault associated with each parameter in error. This structure contains the following elements:
- A ParameterName element that contains the full path name of the parameter in error.
- A FaultCode element that contains a single numeric fault code that indicates the fault associated with the particular parameter in error.
- A FaultString element that contains a human readable description of the fault for the particular parameter in error.

The following is an XML-schema segment that defines the Fault structure:

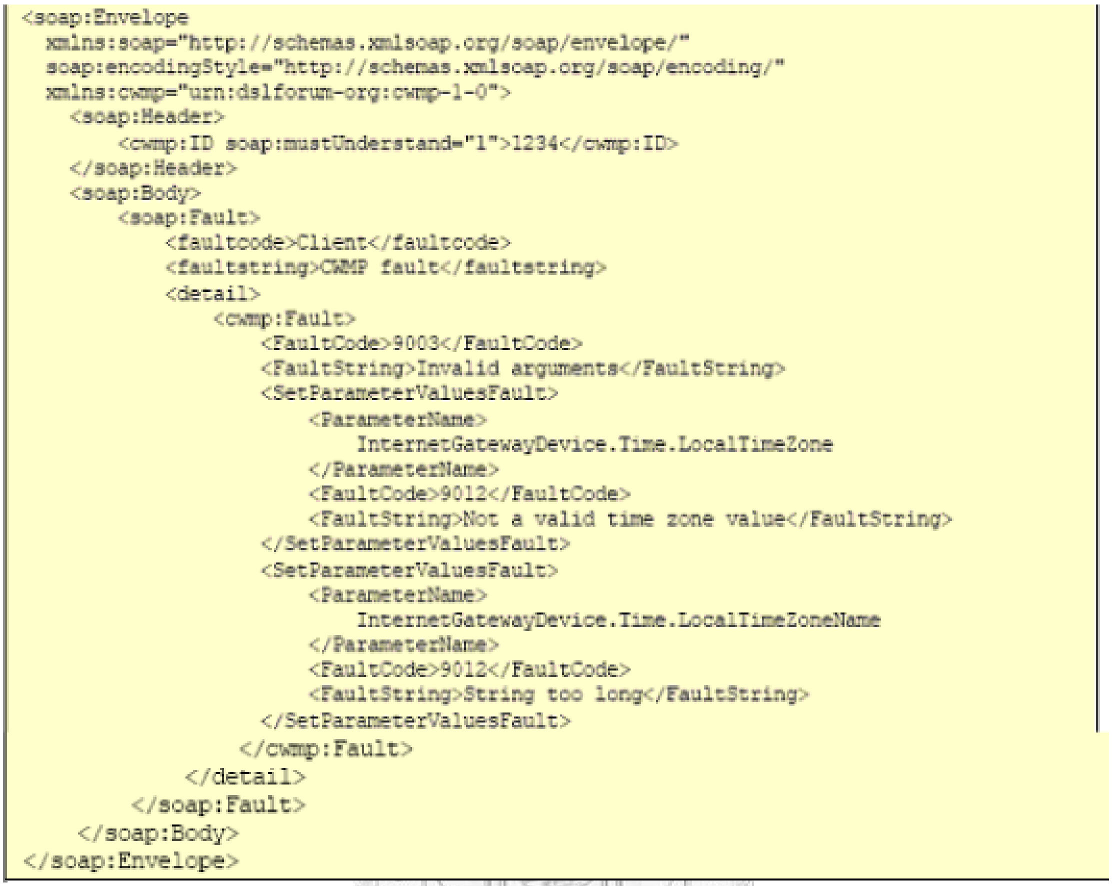
```
<xs:element Name="Fault">
     <xs:complexType>
     <xs:sequence>
          <xs:element Name="FaultCode" Type="unsignedInt"/>
          <xs:element Name="FaultString" Type="string" minOccurs="0"/>
          <xs:element Name="SetParameterValuesFault" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
                    <xs:sequence>
                         <xs:element Name="ParameterName" Type="string"/>
                         <xs:element Name="FaultCode" Type="unsignedInt"/>
                         <xs:element Name="FaultString" Type="string" minOccurs="0"/>
                    </xs:sequence>
               </xs:complexType>
          </xs:element>
     </xs:sequence>
     </xs:complexType>
</xs:element>
```

Below is an example envelope containing a fault response:



```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
    <soap:Header>
        <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
    </soap:Header>
    <soap:Body>
        <soap:Fault>
            <faultcode>Client</faultcode>
            <faultstring>CWMP fault</faultstring>
            <detail>
                <cwmp:Fault>
                    <FaultCode>9000</FaultCode>
                    <FaultString>Upload method not supported</FaultString>
                </cwmp:Fault>
            </detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

Below is an example envelope containing a fault response for a SetParameterValues method call:



```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
    <soap:Header>
        <cwmp:ID soap:mustUnderstand="1">1234</cwmp:ID>
    </soap:Header>
    <soap:Body>
        <soap:Fault>
            <faultcode>Client</faultcode>
            <faultstring>CWMP fault</faultstring>
            <detail>
                <cwmp:Fault>
                    <FaultCode>9003</FaultCode>
                    <FaultString>Invalid arguments</FaultString>
                    <SetParameterValuesFault>
                        <ParameterName>
                            InternetGatewayDevice.Time.LocalTimeZone
                        </ParameterName>
                        <FaultCode>9012</FaultCode>
                        <FaultString>Not a valid time zone value</FaultString>
                    </SetParameterValuesFault>
                    <SetParameterValuesFault>
                        <ParameterName>
                            InternetGatewayDevice.Time.LocalTimeZoneName
                        </ParameterName>
                        <FaultCode>9012</FaultCode>
                        <FaultString>String too long</FaultString>
                    </SetParameterValuesFault>
                </cwmp:Fault>
            </detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

The CPE WAN Management Protocol defines a series of SOAP Header elements as specified in Table.4:

Table. 4 SOAP Header Element

Tag Name	Description
ID	This header element MAY be used to associate SOAP requests and responses using a unique identifier for each request, for which the corresponding response contains the matching identifier. The value of the identifier is an arbitrary string and is set at the discretion of the requester. If used in a SOAP request, the ID header MUST appear in the matching response (whether the response is a success or failure). Because support for this header is required, the mustUnderstand attribute MUST be set to “1” (true) for this header.

HoldRequests	This header MAY be included in envelopes sent from an ACS to a CPE to regulate transmission of requests from the CPE. This header MUST NOT appear in envelopes sent from a CPE to an ACS. This tag has Boolean values of “0” (false) or “1” (true). If the tag is not present, this is interpreted as equivalent to a “0” (false). The behavior of the CPE on reception of this header . Support in the CPE for this header is REQUIRED. If an ACS must update the flow-control state but has no other message to send, it may send an envelope containing only this header and an empty body. Because support for this header is required, the mustUnderstand attribute MUST be set to “1” (true) for this header.
NoMoreRequests	This header MAY be included in envelopes sent by an ACS or a CPE to explicitly indicate to the recipient whether or not it will not be sending any more requests during the remainder of the session. This tag has Boolean values of “0” (false) or “1” (true). If the tag is not present, this is interpreted as equivalent to a “0” (false). This may be set to true in an envelope that contains the final request or in any subsequent envelope. Once set to true during a session, it SHOULD be set to true in the remaining envelopes sent, and the sender MUST NOT send additional request messages during that session. The behavior of the CPE on reception of this header . Support in the CPE for transmission or reception of this header is OPTIONAL. The behavior of the ACS on reception of this header. Support in the ACS for transmission or reception of this header is OPTIONAL. Because support for this header is optional, the mustUnderstand attribute MUST be either absent or set to “0” (false) for this header.

5.4 Transaction Session Procedure

All transaction sessions MUST begin with an Inform message from the CPE contained in the initial HTTP post. This serves to initiate the set of transactions and communicate the limitations of the CPE with regard to message encoding.

The session ceases when both the ACS and CPE have no more requests to send, no responses remain due from either the ACS or the CPE. At such time, the CPE may close the connection.

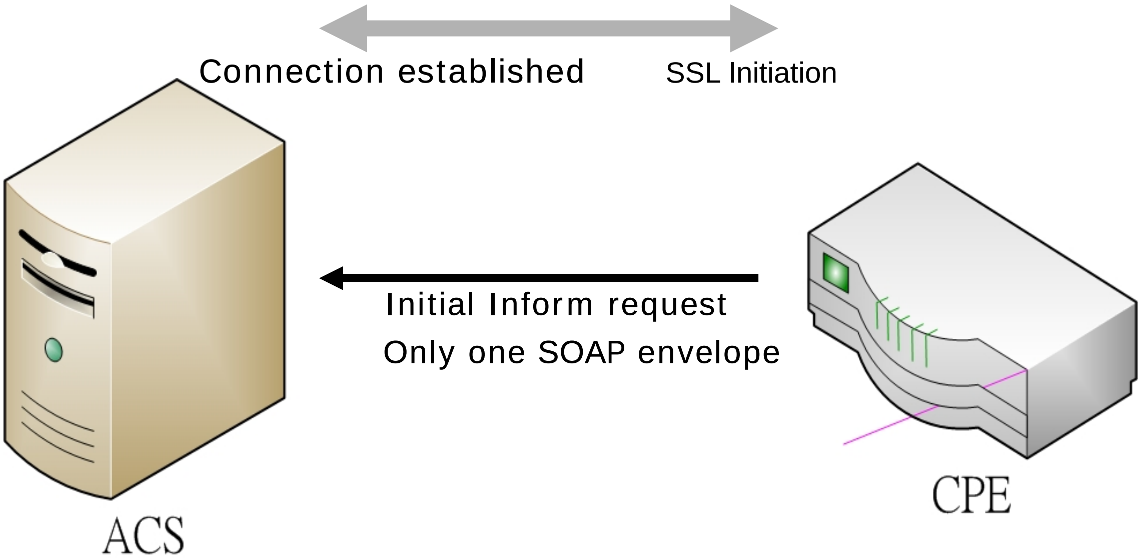
No more than one transaction session between a CPE and its associated ACS may exist at a time.

5.4.1 CPE Operation

5.4.1.1 Session Initiation

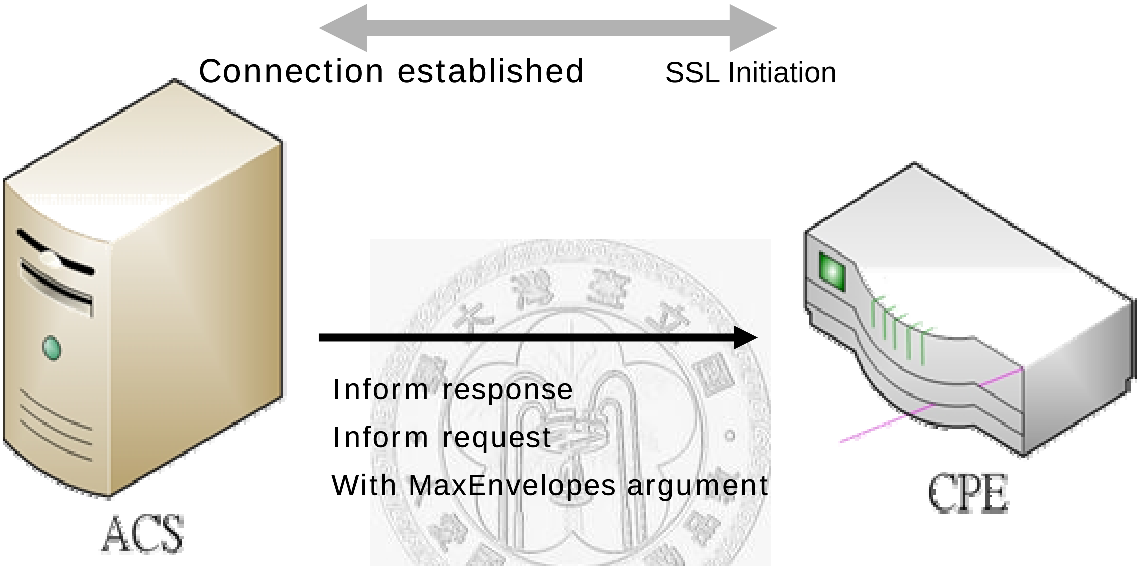
The CPE will initiate a transaction session to the ACS. Once the connection to the ACS is successfully established, the CPE initiates a session by sending an initial Inform request to the ACS. This indicates to the ACS the current status of the CPE and that the CPE is ready to accept requests from the ACS.(shown in Fig.10)

Fig. 10



In this initial HTTP post carrying the Inform request, only one SOAP envelope is allowed. The MaxEnvelopes argument in the Inform response indicates the maximum number of envelopes that may be carried by each subsequent HTTP post.(shown in Fig.11)

Fig. 11



The CPE SHOULD initiate a session only when it has locked the Parameters accessible through this interface to ensure they cannot be changed via any other mechanism. The CPE SHOULD maintain this lock until the session is terminated.

5.4.1.2 Incoming Request

On reception of SOAP requests from the ACS, the CPE MUST respond to each request in the order they were received. This definition of order places no constraint on whether multiple responses are sent in a single HTTP post (if the ACS can accept more than one envelope), or distributed over multiple HTTP posts.

To prevent deadlocks, the CPE MUST NOT hold off responding to an ACS request to wait for a response from the ACS to an earlier CPE request.

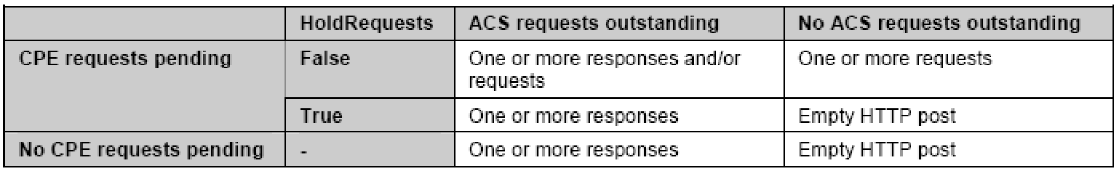
5.4.1.3 Outgoing Request

When the CPE has request messages to send (after the initial Inform request), it may send these in any order with respect to responses being sent by the CPE to the ACS. That is, the CPE may insert one or more requests at any point in the sequence of envelopes it transmits to the ACS. There is no specified limit to the number of requests a CPE may send prior to receiving responses (the number of outstanding requests). A CPE MAY incorporate a locally specified limit if desired.

If the CPE receives an envelope from the ACS (either request or response) with the HoldRequests header equal to true, the CPE MUST NOT send any requests in subsequent HTTP posts.(shown in Fig.12) The CPE may restart sending envelopes only when it subsequently receives an envelope with the HoldRequests header equal to false (or equivalently, no HoldRequests header). In determining whether it may send a request, the CPE MUST examine all envelopes received through the end of the most recent HTTP response. Only the last envelope in an HTTP response determines whether requests are allowed on the next HTTP post. If the CPE receives an empty HTTP response from the ACS, this may be interpreted as HoldRequests equal false.

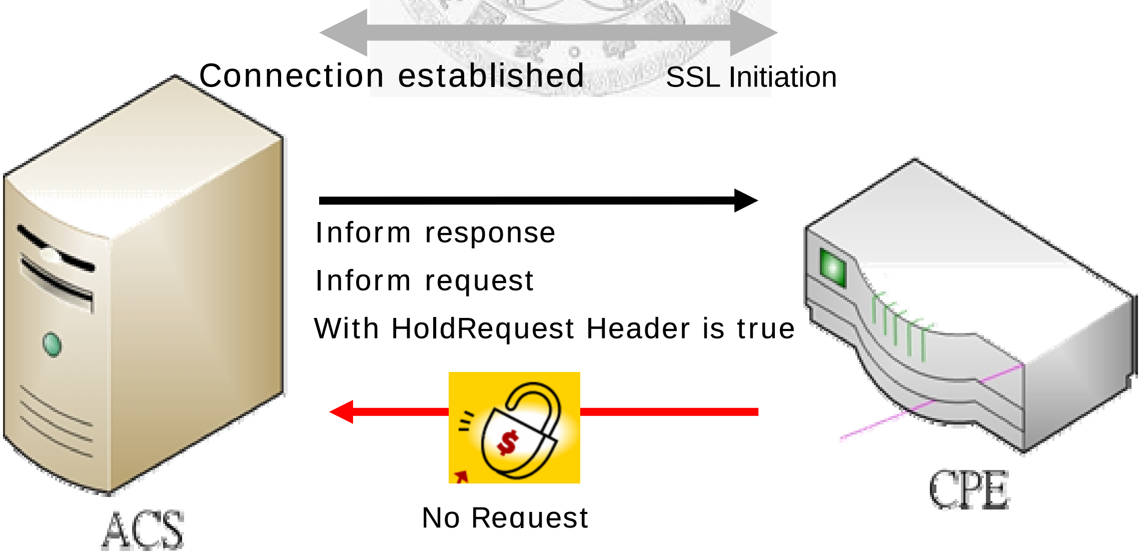
If there are one or more outstanding requests from the ACS, or if the CPE has one or more outstanding requests and HoldRequests is false, then the CPE MUST send at least one request or response in any HTTP post sent to the ACS. An empty HTTP post MUST be sent if the ACS has no requests or responses outstanding. Table 5 lists the complete set of constraints on what a CPE MUST send while a session is in progress.

Table. 5 CPE Message Transmission Constraints



	HoldRequests	ACS requests outstanding	No ACS requests outstanding
CPE requests pending	False	One or more responses and/or requests	One or more requests
	True	One or more responses	Empty HTTP post
No CPE requests pending	-	One or more responses	Empty HTTP post

Fig. 12



5.4.1.4 Session Termination

The CPE MUST terminate the transaction session when all of the following conditions are met:

1) The ACS has no further requests to send the CPE. The CPE concludes this if either one of the following is true:

a) The most recent HTTP response from the ACS contains no envelopes.

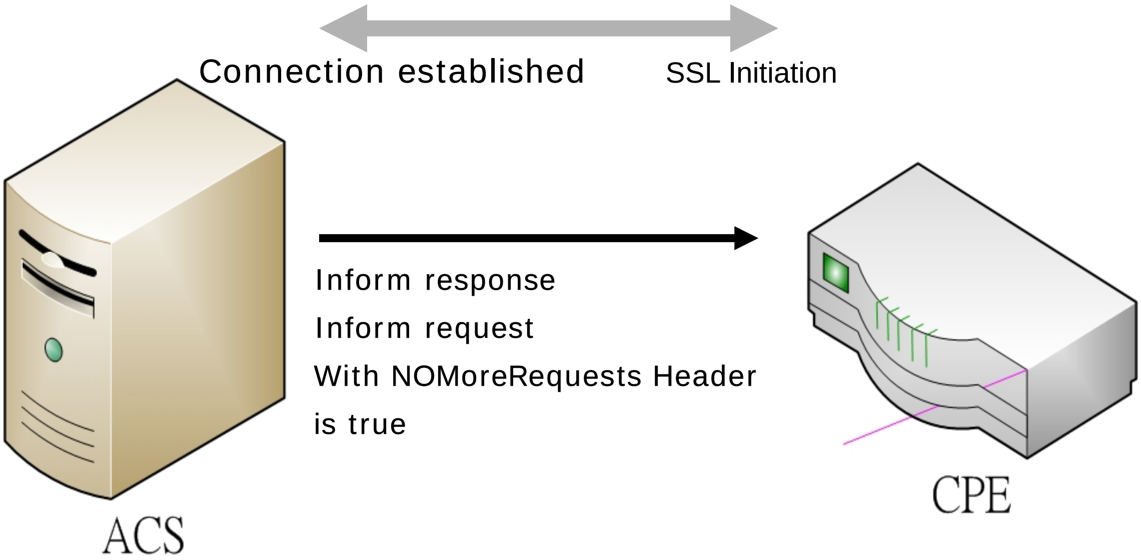
b) The most recent envelope received from the ACS includes a NoMoreRequests header equal true. Use of this header by a CPE is OPTIONAL.(shown in Fig.13)

2) The CPE has no further requests to send to the ACS.

3) The CPE has received all outstanding response messages from the ACS.

4) The CPE has sent all outstanding response messages to the ACS resulting from prior requests.

Fig. 13



The CPE MUST also terminate a session if it has received no HTTP response from an ACS for a locally determined time period of not less than 30 seconds.

If the above conditions are not met, the CPE MUST continue the session.

If one or more messages exchanged during a session results in the CPE needing to reboot to complete the requested operation, the CPE MUST wait until after the session has cleanly terminated based on the above criteria before performing the reboot.

If the session terminates unexpectedly, the CPE SHOULD attempt to establish a new session, starting the session establishment procedure from the beginning. The CPE MAY place locally specified limits on the number of times it attempts to reestablish a session in this case.

5.4.2 ACS Operation

5.4.2.1 Session Initiation

Upon receiving the initial Inform request from the CPE, the ACS MUST respond with an Inform response. The ACS may follow this with series of requests sent to the CPE.

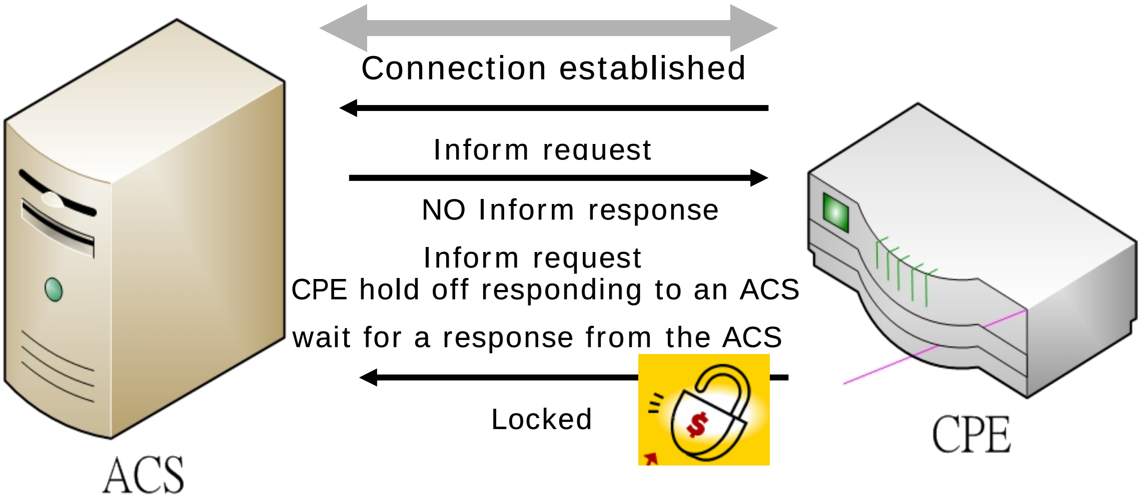
The MaxEnvelopes argument in the Inform request indicates the maximum number of envelopes that may be carried by each HTTP response sent by the ACS to the CPE. If the CPE can accept more than one envelope, the initial HTTP response carrying the Inform response may also carry additional requests up to the total limit imposed by MaxEnvelopes

5.4.2.2 Incoming Requests

On reception of SOAP requests from the CPE, the ACS MUST respond to each request in the order they were received. This definition of order places no constraint on whether multiple responses are sent in a single HTTP response (if the CPE can accept more than one envelope), or distributed over multiple HTTP responses.

To prevent deadlocks, the ACS MUST NOT hold off responding to a CPE request to wait for a response from the CPE to an earlier ACS request.(shown in Fig.14)

Fig. 14



If the ACS wishes to prevent the CPE sending requests during some portion of the session, it may do so by setting the HoldRequests header to true in each envelope transmitted to the CPE until the ACS again wishes to allow requests from the CPE. The ACS MUST allow CPE requests before completion of a session (this may be done either explicitly via the HoldRequests header or implicitly by sending an empty HTTP response).

5.4.2.3 Outgoing Requests

When the ACS has request messages to send, it may send these in any order with respect to responses being sent by the ACS to the CPE. That is, the ACS may insert one or more requests at any point in the sequence of envelopes it transmits to the ACS (after the Inform response). There is no specified limit to the number of requests an ACS may send prior to receiving responses (the number of outstanding requests). An ACS MAY incorporate a locally specified limit if desired.

If the ACS has one or more requests remaining to be sent and/or one or more responses outstanding from earlier requests from the CPE, the ACS MUST send at least one request or response in any HTTP response sent back to the CPE. An empty HTTP response is only allowed if the ACS has no more requests or responses outstanding.

5.4.2.4 Session Termination

Since the CPE is driving the HTTP connection to the ACS, only the CPE is responsible for connection initiation and teardown.

The ACS may consider the session terminated when all of the following conditions are met:

1) The CPE has no further requests to send the ACS. The ACS concludes this if either one of the following is true:

   a) The most recent HTTP post from the CPE contains no envelopes.

   b) The most recent envelope received from the CPE (in the order defined in section 3.4.1) includes a NoMoreRequests header equal true. Use of this header by an ACS is OPTIONAL.

2) The ACS has no further requests to send the CPE.

3) The CPE has sent all outstanding response messages to the ACS resulting from prior requests.

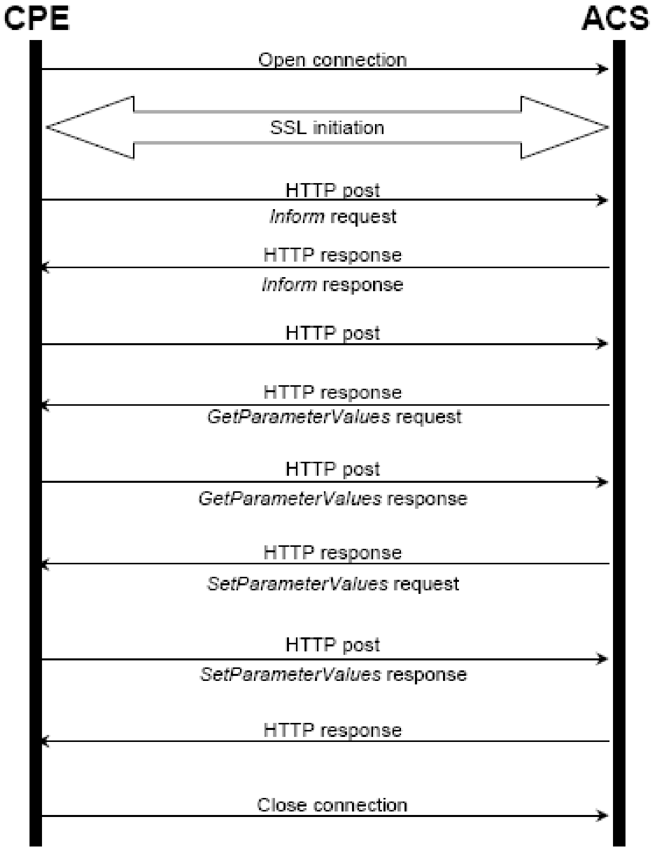
4) The ACS has received all outstanding response messages from the CPE.

If the above criteria have not all been met, but the ACS has not received an HTTP post from a given CPE within a locally defined timeout of not less than 30 seconds, it may consider the session terminated. In this case, the ACS MAY attempt to reestablish a session by performing a Connection Request.

5.4.3 Transaction Examples

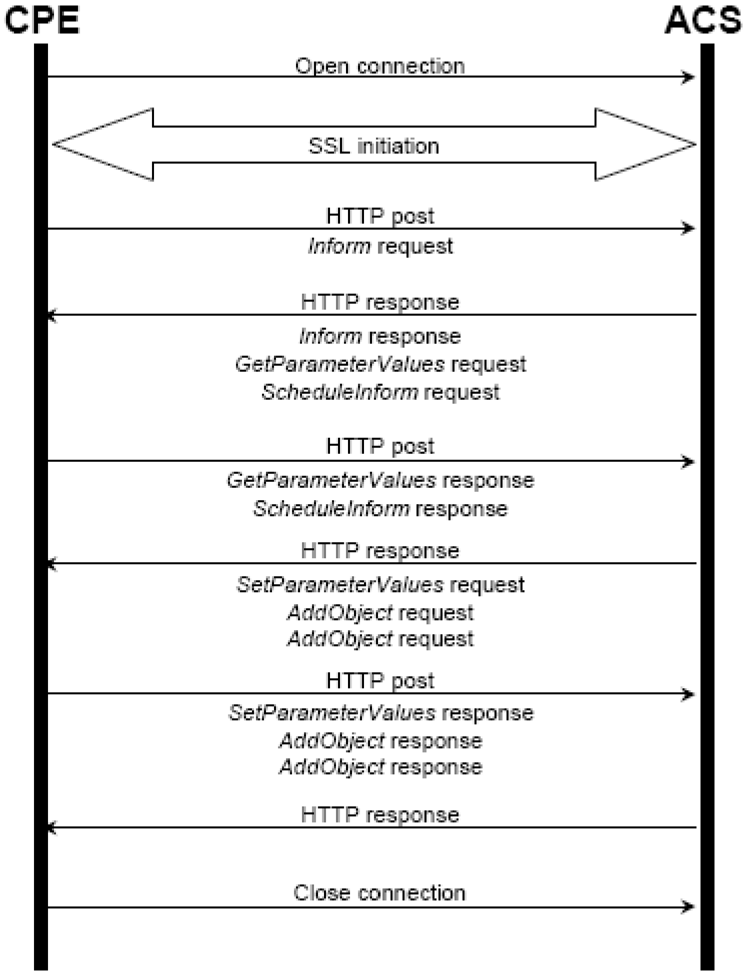
In the example shown in Fig.15, the ACS first reads a set of parameter values, and based on the result, sets some parameter values. In the example show, MaxEnvelopes from both the CPE and ACS equal one, so there is no pipelining of requests from the ACS, nor multiple responses per HTTP post from the CPE.

Fig. 15MaxEnvelopes from both the CPE and ACS equal one



The example in Fig.16 shows a scenario where MaxEnvelopes from both the CPE and ACS are equal to three, allowing the use of message pipelining in both directions. In this example, some additional requests from the ACS are shown.

Fig. 16 MaxEnvelopes from both the CPE and ACS equal three



5.5 Assumption

To use the defined method on the gateway with CPE side on Fig.17 and the system with ACS side on Fig.18.

Fig. 17 Gateway Architecture

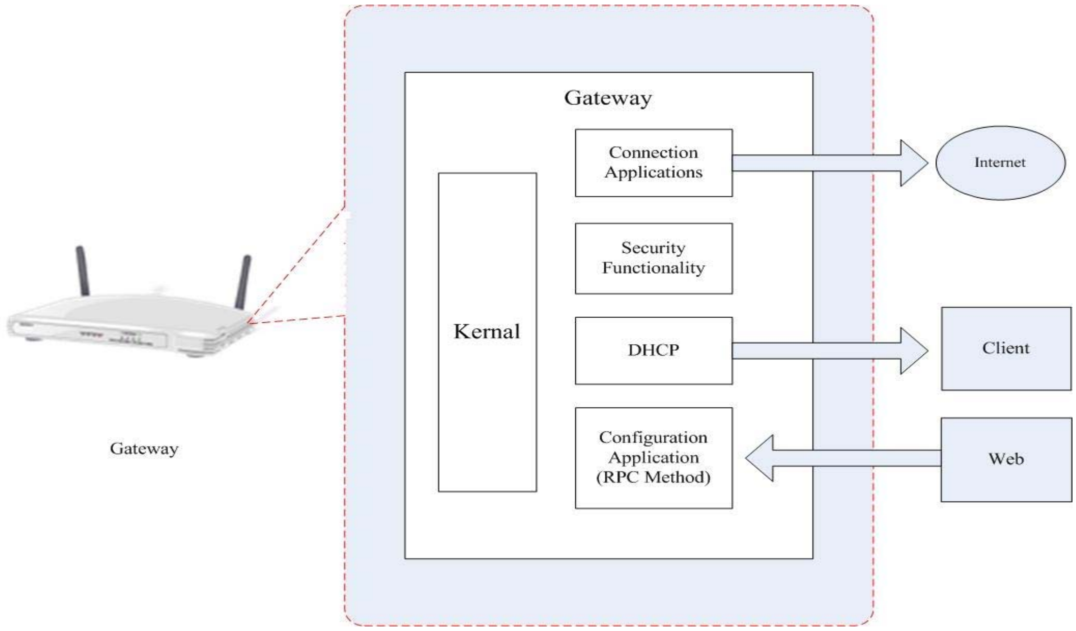


Fig. 18 System Architecture

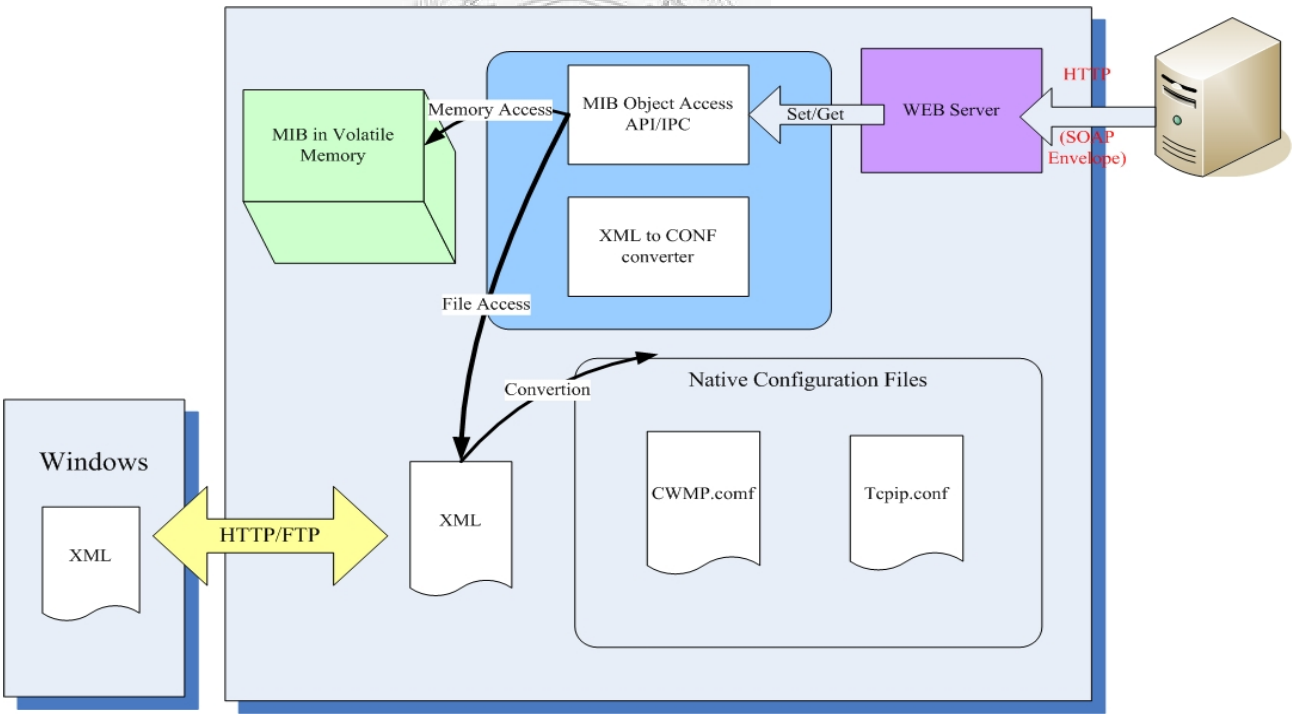
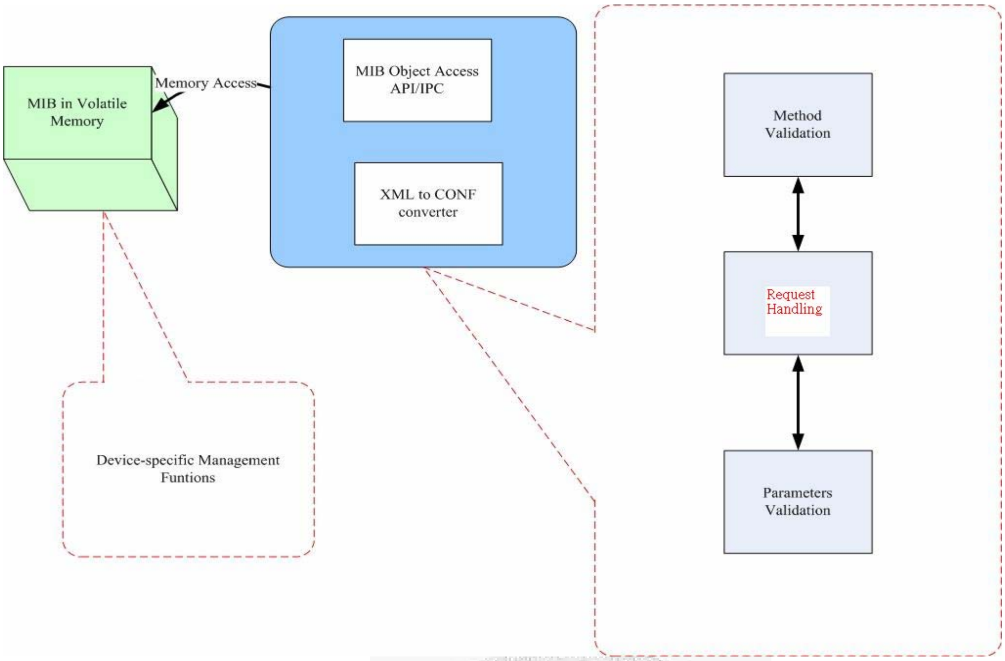


Fig. 19 presents the overall architecture of the automatically generated TR-069 clients. In fact, only a subset of functions needs to be dynamically produced. CPE contains an HTTP client which accepts the management commands (i.e. the RPC methods encapsulated in an HTTP Response) coming from the ACS. It then transfers the connection over to the SOAP module for processing. SOAP module decodes the RPC method and passes the name of the method along with its arguments to the “Request Handling” module. This is the heart of the system, since it decides the management actions to perform depending on the called method. Before proceeds to the management action it checks the RPC method and its arguments using the “Methods Validation” and “Parameters Validation”, respectively; the former module ensures that the called method is supported by the CPE and is encoded correctly inside the SOAP body, while the latter ensures that the arguments are valid CPE parameters. If any of the aforementioned conditions is not valid then the “Request Handling” module ceases any further work and returns an error SOAP response to the ACS. Otherwise, it calls the appropriate management function. Once the management action is completed, the module returns to the ACS a SOAP response indicating the results of the action. Only the “Request Handling” and “Parameters Validation” (dashed) blocks need the CPE parameters information and therefore are dynamically implemented; the former block needs to know for each requested parameter which management function to call, while the latter needs to know all the parameters supported by the CPE

Fig. 19 The detail on system architecture



5.6 Summary

Integration the TR-069 Protocol and understand the all procedures on CPE WAN Management Protocol. Then design the gateway architecture and system architecture.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

TR- 069 has adopted the ripe communication protocol, open target -oriented management information structure, flexibility with power and extendible ability, can meet all kinds of and carry users' equipment control and disposition demand far. Imprison with family network more and more equipment support TR-069, agreement this will replace other various kinds of disposition agreement that equipment have specially, become one layer of the above mainstreams of disposing the way of business of IP.

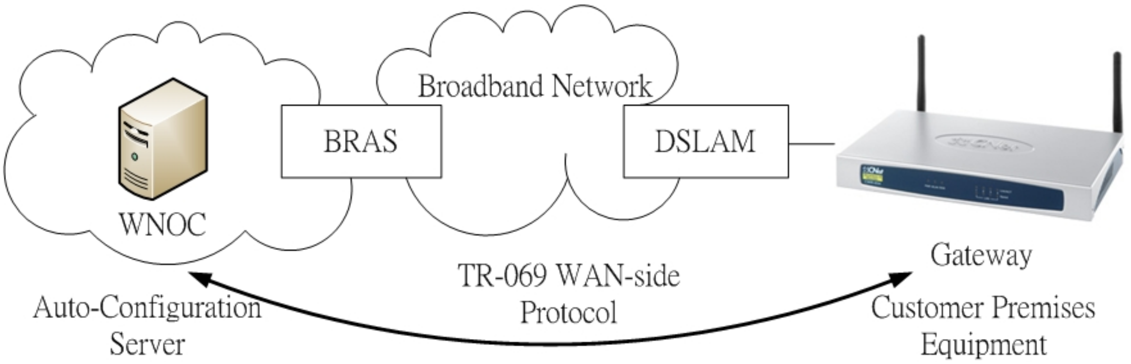
6.2 Future Work

We will implement CPE WAN Management Protocol into our Wireless Network Operation Center(WNOC) and Gateway to achieve the following applications:

- New subscriber activation - When a user selects a service package from a service provider’s web-based ordering system or telephone sales associate, the subscriber’s configuration information is passed from the sales system to the DMS for activation.

- End user moves – When a subscriber moves to a new location, all configuration of that subscriber’s CPE can be automatically sent, eliminating the need for a truck roll or technical support incident.
- CPE replacement or upgrades – The automatic nature of the configuration system lends itself to elegant CPE changes, minimizing the chance that a subscriber must configure their CPE.
- Voice over IP – Changes to a subscribers VoIP profile such as a new phone number, inclusion or deletion of value-add services is done seamlessly. New configuration settings, as well as new features and services are delivered to the subscriber automatically.
- IPTV – Subscriptions to channels and pay-per-view movies may be managed and delivered to the subscriber automatically.
- Home Monitoring – Remotely monitor home and alert subscriber as necessary.

Fig. 20 Future Work



Bibliography

[1] UPnP Forum, “UPnP device architecture 1.0”, December 2003, http://www.upnp.org

[2] TR-046, Auto-Configuration Architecture & Framework, DSL Forum Technical Report.

[3] TR-062, Auto-Configuration for the Connection Between the DSL Broadband Network Termination (B-NT) and the Network using ATM, DSL Forum Technical Report

[4] TR-044, Auto-Configuration for Basic Internet (IP-based) Services, DSL Forum Technical Report

[5] RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt

[6] M-J.Choi, H-M.Choi and J.W.Hong, “XML-based configuration management for IP network devices”, *IEEE Commun. Mag*. vol. 42, no.7, July 2004.

[7] T.Klie, F.Straub, “Integrating SNMP agents with XML-based management systems”, *IEEE Commun. Mag*. vol. 42, no. 7, July 2004.

[8] *Simple Object Access Protocol (SOAP)* http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[9] DSL Forum, “CPE WAN management protocol”, Tech. rep. 069, May 2004.

[10] DSL Forum, “LAN-side CPE configuration”, Tech. rep. 064, May 2004.

[11] DSL Forum, “TR-064 extensions for service differentiation”, Tech. rep.133, September 2005.

[12] RFC 2132, DHCP Options and BOOTP Vendor Extensions, http://www.ietf.org/rfc/rfc2132.txt

[13] DSL Forum, “Data model template for TR-069 enabled devices”, Tech.rep. 106, September 2005.

[14] DSL Forum, “Internet Gateway Device v.1.1 data model”, Tech. rep. 098, September 2005.

[15] DSL Forum, “Provisioning parameters for VoIP CPE”, Tech. rep. 104, September 2005.

[16] DSL Forum, “Data model for a TR-069 enabled STB”, Work. Text. 135,May 2005.

Appendix A: Glossary Acronyms and Abbreviation

ACS Auto-Configuration Server.This is a component in the broadband network responsible for auto-configuration of CPE for advanced services.

B-NT A broadband access CPE device capable of being managed by an ACS.

CPE Customer Premise Equipment. A DSL B-NT is one form of broadband CPE.

Internet Gateway Device A CPE device that is either a B-NT or a broadband router.

Option An optional CPE capability that may only be enabled or disabled using a digitally signed Voucher.

RPC Remote procedure call.

Parameter A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.

Session A contiguous sequence of transactions between a CPE and an ACS.

LAN Local Area Network

Voucher A digitally signed data structure that instructs a particular CPE to enable or disable Options, and characteristics that determine under what conditions the Options persist.

NAT Network Address Translation

SNMP Simple Network Management Protocol

URL Uniform Resource Locator

WAN Wide Area Network

WLAN Wireless Local Area Network