



TR-069 Protocol Stack Development

Galla Sobha

School of Computing Science and Engineering
VIT University, Vellore-14
Tamil Nadu, India.
galla.ndu@gmail.com

Swarnalatha. P*

Assistant Professor [Selection Grade]
School of Computing Science and Engineering
VIT University, Vellore-14 TamilNadu, India.
pswarnalatha@vit.ac.in

M.Anbarasi

Assistant Professor,
School of Computer Science & Engineering
VIT University, Vellore-14 TamilNadu, India.
manbarasi@vit.ac.in

Abstract: TR_069 (short for Technical Report 069) is a CPE (client premise equipment) WAN Management Protocol, intended for communication between a CPE and Auto-Configuration Server (ACS). The CPE WAN Management Protocol defines a mechanism that encompasses secure auto-configuration of a CPE, and also incorporates other CPE management functions into a common framework. The project aims in developing the TR_069 protocol stack by implementing various RPC methods involved in the protocol.

TR-069 stack includes twenty RPC methods. At present four RPC methods have been implemented. The scope of project is implementing the RPC's which are not yet implemented. The project also includes the compliance test of the complete TR-069 protocol stack against TR-069 amendment 3.

Keywords: CPE WAN Management protocol, AutoConfigurationServer(ACS), CustomerPremisesEquipment(CPE) device, RemoteProcedureCall, TR69 stack, DataModel.

I. INTRODUCTION

If a Service Provider has to control the CPE devices over Wide Area Network which are from different manufacturers, the service provider has to implement separate mechanism for each type of device, as each CPE vendor creates his own proprietary mechanisms for management, and does not want to expose those mechanisms to its competitors. Because of this reason a common application to manage different CPEs has never been achieved before [1].

The problem of managing different CPEs from various manufacturers is solved using TR069. By utilizing TR-069 standards, a unified abstraction from hardware is achieved. This means the CPE from various manufacturers can be provisioned and managed seamlessly alongside each other. Most leading CPE vendors today offer TR-069 manageability of their devices [2]. For the service providers this is crucial as operational procedures and systems need not change if any new equipment is introduced to deliver their services.

Objective of this project is to develop the TR069 stack by implementing different RPC methods which provides the communication between CPE and ACS server.

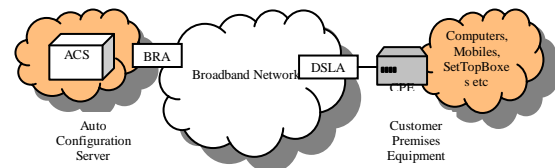


Figure:1 TR069 protocol working

II. COMPONENTS DESCRIPTION

The protocol includes some special and general terminologies for the components of the Management protocol. These constitute the physical parts of the network to be managed and supported by the CWMP. The two main blocks of CWMP are:

CPE: Stands for customer Premise equipment. It corresponds to the network device that has to be managed. It can belong to any class of network devices connected as a part of WAN such as modems, routers, gateways, set-top boxes VoIP-phones etc [1]. A B-NT2 is a special CPE that is capable of being managed by the protocol.

ACS: It is an Auto Configuration Server that is responsible for auto-configuration of the CPE. The ACS also acts as a communication interface for CPE with the rest of management network

III. DETAILED DESIGN

Total TR69 framework is divided into Protocol stack and DataModel. The protocol stack is implemented using the RPC methods. The request from the ACS server is accepted by the CPE protocol stack from where the calls are passed to DataModel. In the DataModel the request by the ACS is handled and served[3].

SOAP messages are carried between a CPE and an ACS using HTTP 1.1, where the CPE acts as the HTTP client and the ACS acts as the HTTP server.

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL/TLS
TCP/IP

Figure: 2. Protocol Stack

A. 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:

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

B. RPC method architecture for communication:

In order to manage a CPE, the ACS periodically needs to know about the values of its various parameters. Remote management of the CPE should also be enabled at the same time. For this to be accomplished, a mechanism for remote invocation of GET and SET methods has to be supported by the technologies of the CWMP. Remote Procedure Call (RPC) is best suited for this type of remote invocation of the managed objects where a specific procedure call happens over a managed network[4]. RPC is used for bi-directional communication between the CPE and ACS. The CWMP defines a set of standard RPC method interfaces that accounts for all the data communication between the CPE and ACS. Some of the overall methods and their brief explanation are as follows:

C. Description of different RPC methods:

Different RPC methods are developed according to the TR-069 amendment standard.

They are

- GetRPCMethods
- GetParameterNames
- SetParameterAttributes

- GetParameterAttributes
- AddObject.
- DeleteObject.

a. GetRPCMethods:

This method should be supported by both ACSserver and CPE. ACS server calls this method to find out the RPCmethods that are supported by the CPE with which it is communicating. CPE can call this method to find out the set of RPC methods supported by ACS in the same way. There are no request arguments for this method. The response of this method call contains the array of RPC method names that are supported by ACS or CPE[5].

b. Get Parameter Names:

This method is used by Auto Configuration Server to find the set of parameters supported by a particular Customer Premises Equipment.

The calling arguments of this function are Parameter Path and Next Level. Parameter Path is the full name or the partial name from the naming hierarchy of the particular CPE. Next level says whether to output the whole hierarchy under the given Parameter Path (i.e. if Next Level is true, output is the whole hierarchy which includes the children and sub children of the Parameter Path) or only the next level (i.e. if Next Level is false, output is only the children of Parameter Path it does not include the sub children).

The response contains the array of names with their Writable values. Writable indicates whether the parameter can be changed by using Set Parameter Values or not.

c. Set Parameter Attributes:

This method is used by the Auto Configuration Server to change the attributes associated with one or more CPE parameters.

The calling arguments of this method are Name, Notification Change, Notification, Access List Change, Access List. Name is the ACS parameter name for which ACS wants to change the attributes. Notification change represents whether ACS wants to change the notification value or not. Notification is the changed value given by ACS.

The notification value one represents passive notification. Passive notification means if the parameter is changed by other than ACS, the change has to be notified and the value has to be written in the next inform message. The next inform message should also include the VALUECHANGE in its event code list. The notification value two represents active notification. Active notification means if the parameter is changed by other than ACS, then it has to be immediately notified to ACS by triggering INFORM message[5]. This inform message should contain the changed value of the parameter and also the VALUECHANGE in its event code list.

Access List Change represents whether the ACS wants to change the Access List or not. Access List is the list of names who have the access to change the parameter attributes.

This method does not contain any response arguments.

d. Get Parameter Attributes:

This method is used by Auto Configuration Server to get the Attributes of one or more CPE parameters.

The calling arguments of this method are the list of parameter names for which ACS wants to know the attribute values.

The response contains Name, Notification, Access List. Name is the ACS parameter name, Notification is either active notification or passive notification. Access List is the list by which the attributes can be changed.

e. AddObject:

Auto Configuration Server uses this method to create a new instance of a Multi-instance object. A multi-instance object is a collection of parameters or objects for which multiple instances are defined.

The calling argument is the path name of the collection of objects for which the new instance has to be created. This

argument does not contain any instance number for the object to be created. Once an instance number is assigned while creating an object, then it cannot be changed and it persists till it is deleted using Delete Object method.

The response contains instance number of the newly created object and status. Status value 0 indicates the object has been created, 1 indicates the object creation has been validated and committed, but not yet applied.

f. Delete Object:

Auto Configuration Server uses this method to remove a particular instance of an object. The calling arguments are the path name of the object instance which includes the instance number and parameter key.

The response contains status code. 0 indicates the object has been deleted, 1 indicates the object deletion has been validated and committed, but not yet applied.

IV. LIST OF RPC METHODS

Table: 1

S.No	RPC Method	Description	To be implemented?
1	GetParameterValues	This method MAY be used by an ACS to obtain the value of one or more CPE Parameters.	No
2	SetParameterValues	This method MAY be used by an ACS to modify the value of one or more CPE Parameters	No
3	GetParameterNames	This method MAY be used by an ACS to discover the Parameters accessible on a particular CPE Parameter.	Yes
4	SetParameterNames	This method MAY be used by an ACS to modify attributes associated with one or more CPE Parameter.	Yes
5	GetParameterAttributes	This method MAY be used by an ACS to read the attributes associated with one or more CPE Parameter.	Yes
6	SetParameterAttributes	This method MAY be used by an ACS to set the attributes associated with one or more CPE Parameters.	Yes
7	AddObject	This method MAY be used by the ACS to create a new instance of a multi-instance object.	Yes
8	DeleteObject	This method is used to remove a particular instance of an object.	Yes
9	Download	This method MAY be used by the ACS to cause the CPE to download a specified file from the designated location.	Yes
10	Reboot	This method causes the CPE to reboot, and calls for use of extreme action.	No
11	GetRPCMethods	This method MAY be used by the ACS to know all the supporting RPC methods.	Yes
12	FactoryReset	This method resets the CPE to its factory default state, and calls for use with extreme action.	No
13	GetQueuedTransfers	This method MAY be used by an ACS to determine the status of previously requested.	Yes
14	SetVouchers	This method MAY be used by an ACS to set one or more option Vouchers in the CPE.	Yes
15	GetOptions	This method MAY be used by an ACS to obtain a list of the options currently set in a CPE.	Yes
16	Upload	This method MAY be used by the ACS to cause the CPE to upload a specified file to the designated location.	Yes
17	ScheduleInform	This method MAY be used by an ACS to request the CPE to schedule a one-time Inform method call some time in future.	No

V. REFERENCES

- [1]. "Research of remote management of RFID reader based on TR069", by Rachidi.H, Karmouch.A, 2nd international conference on Network infrastructure and Digital Content, IEEE-2010.
- [2]. "A framework for self-configuring devices using TR-069", by Rachidi, H., Karmouch, A., International Conference on Multimedia Computing and Systems, IEEE-2011.
- [3]. "Research for remote management and configuration protocol of RFID mobile reader", by Ning Li, Zhongliang Deng, Xiao Liu, Peng Su, international conference on Computer Application and System Modeling, IEEE-2010.

- [4]. “Remote Management of Mobile Devices with Broadband Forum's TR-069” by Hillen.B.A.G, Passchier. I., Matthijssen.E.F, den Hartog.F.T.H, Selgert.F, 13th international conference on Telecommunications Network Strategy and Planning, IEEE-2008.
- [5]. “TR-069 Amendment 3 for CPE WAN management protocol” by Broadband Foru