TR069协议介绍

培训目的

学习TR069协议，理解CPE与ACS之间的交互流程

培训对象

研发PON组

培训讲师

方建江

学习重点

1. TR069协议
2. TR069协议栈实现
3. CPE与ACS交互流程
4. RPC方法
5. 自定义参数及事件

Contents

1. Functional Components
2. Architecture
3. Procedures and Requirements
4. RPC Methods
5. Transaction Flow

1. Functional Components

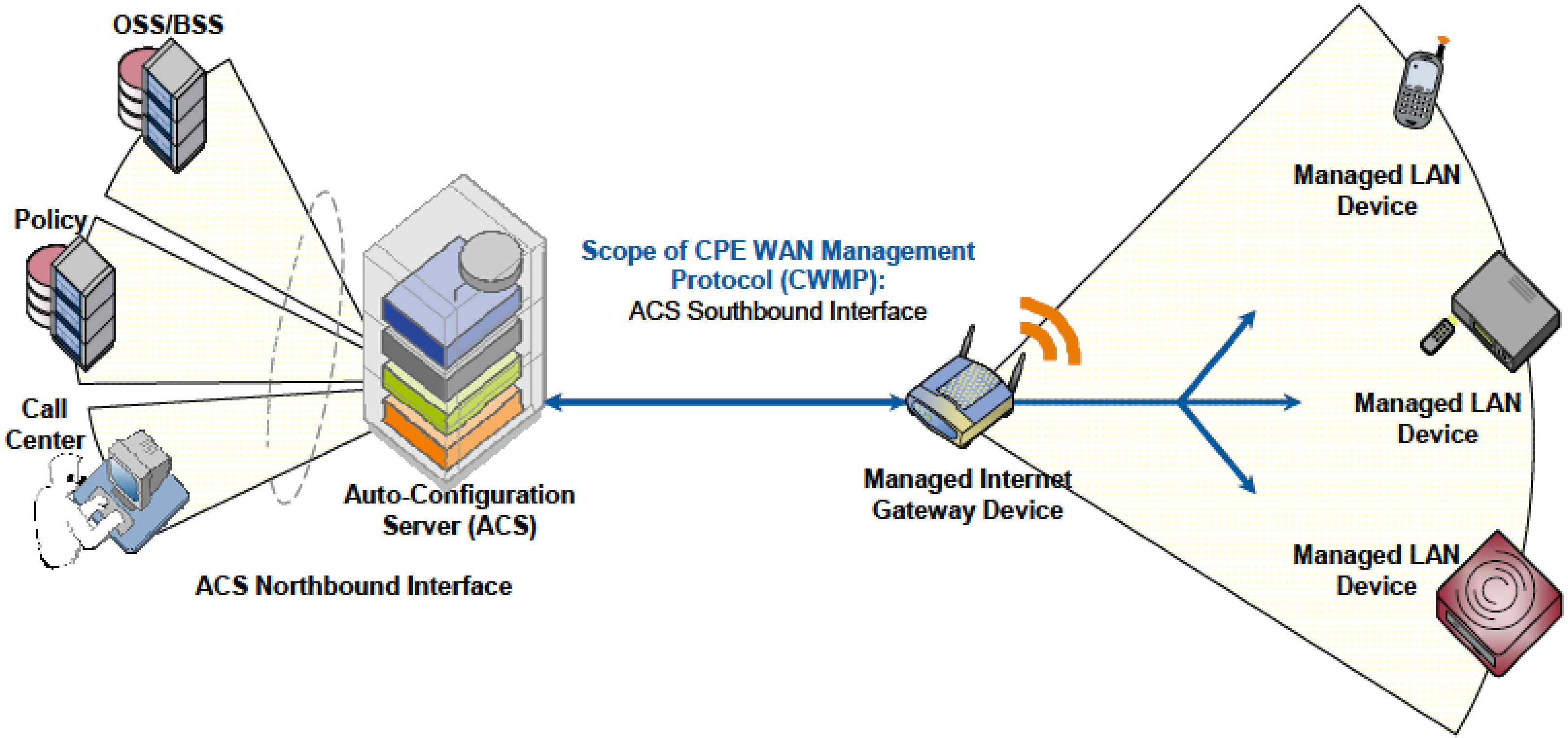
The CPE WAN Management Protocol is intended to support a variety of functionalities to manage a collection of CPE, including the following primary capabilities:

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

Positioning in the End-to-End Architecture

The ACS is a server that resides in the network and manages devices in or at the subscriber premises. The CPE WAN Management Protocol may be used to manage both DSL B-NTs and other types of CPE, including stand-alone routers and LAN-side client devices. It is agnostic to to the specific access medium utilized by the service provider, although it does depend on IP-layer connectivity having been established by the device.

Figure 1 – Positioning in the End-to-End Architecture

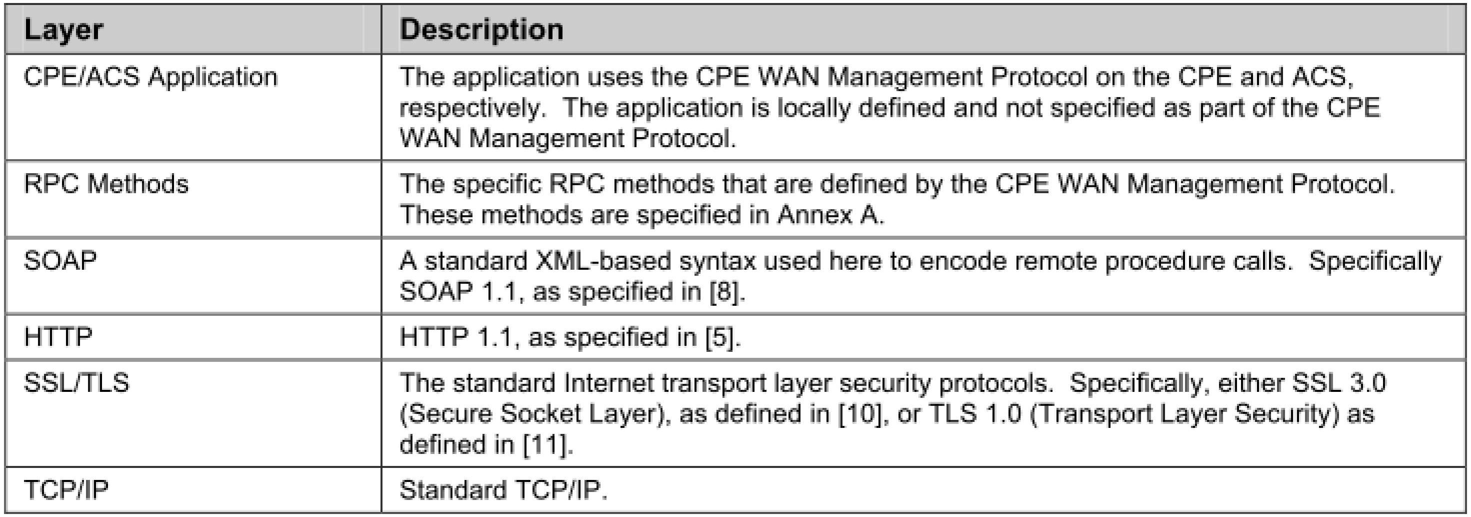


2.Architecture

2.1 Protocol Components

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 2. A brief description of each layer is provided in Table 1. Note that the CPE and ACS must adhere to the requirements of the underlying standard protocols unless otherwise specified.

Table 1 – Protocol layer summary



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. These methods are specified in Annex A.
SOAP	A standard XML-based syntax used here to encode remote procedure calls. Specifically SOAP 1.1, as specified in [8].
HTTP	HTTP 1.1, as specified in [5].
SSL/TLS	The standard Internet transport layer security protocols. Specifically, either SSL 3.0 (Secure Socket Layer), as defined in [10], or TLS 1.0 (Transport Layer Security) as defined in [11].
TCP/IP	Standard TCP/IP.

3. Procedures and Requirements

3.1 Session Procedures

All 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. An Inform message MUST NOT occur more than once during a Session (this limitation does not apply to the potential need to retransmit an Inform request due to an HTTP “401 Unauthorized” status code received as part of the HTTP authentication process, or due to an HTTP 3xx status code received as part of an HTTP redirect).

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

No more than one Session between a CWMP Endpoint and its associated ACS can exist at a time.

3.2 Connection Establishment & Termination

3.2.1 CPE Connection Initiation

The CPE will initiate a Session to the ACS as a result of the conditions listed below:

1. The first time the CPE establishes a connection to the access network on initial installation
2. On power-up or reset
3. Once every PeriodicInformInterval (for example, every 24-hours)
4. When so instructed by the optional ScheduleInform method
5. Whenever the CPE receives a valid Connection Request from an ACS
6. Whenever the URL of the ACS changes
7. Whenever a parameter is modified that is required to initiate an Inform on change.
8. 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).
9. Whenever a download or upload completes (either successfully or unsuccessfully), provided that CPE policy indicates that the ACS needs to be notified of the download or upload completion.
10. Whenever an unsuccessfully terminated session is retried according to the session retry policy

3.2.2 ACS Connection Initiation

- The ACS MAY at any time request that the CPE initiate a connection to the ACS using the Connection Request mechanism. 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 might not be able to access the CPE at all. Annex G defines a mechanism that allows an ACS to contact a CPE connected via a NAT device.

3.2.3 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 and only if the most recent HTTP response from the ACS was empty.

2) The CPE has no further requests to send to the ACS and the CPE has issued an empty HTTP POST to the ACS while HoldRequests is false (which indicates to the ACS that the CPE has no further requests for the remainder of the session). As defined in Table 6, if this condition has not been met but the CPE has no further requests or responses, it MUST send an empty HTTP POST,which will then fulfill this condition.

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.

3.3 Use of SSL/TLS and TCP

- The use of SSL/TLS to transport the CPE WAN Management Protocol is RECOMMENDED, although the protocol MAY be used directly over a TCP connection instead. If SSL/TLS is not used, some aspects of security are sacrificed. Specifically, SSL/TLS provides confidentiality and data integrity, and allows certificate-based authentication in lieu of shared secret-based authentication.

3.4 Use of HTTP

SOAP messages are carried between a CPE and an ACS using HTTP 1.1 [5], where the CPE acts as the HTTP client and the ACS acts as the HTTP server.

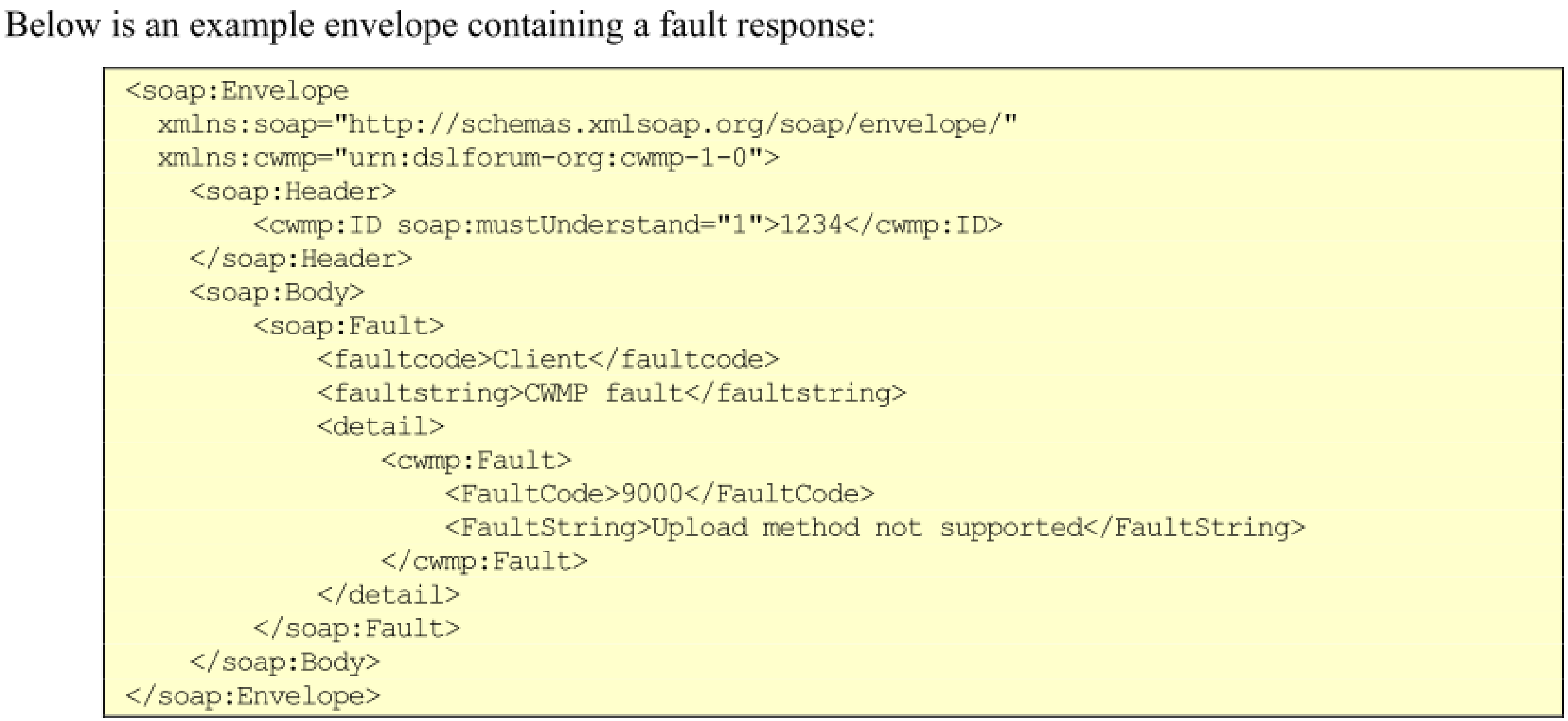
3.4.1 Encoding SOAP over HTTP

- A SOAP request from an ACS to a CPE is sent over an HTTP response, while the CPE's SOAP response to an ACS request is sent over a subsequent HTTP POST.
- When there is a SOAP response in an HTTP Request, or when there is a SOAP Fault response in an HTTP Request, the SOAPAction header in the HTTP Request MUST have no value (with no quotes), indicating that this header provides no information as to the intent of the message. That is, it MUST appear as follows:

  SOAPAction:
- When an HTTP Request or Response contains a SOAP Envelope, the HTTP Content-Type header MUST have a type/subtype of "text/xml".
- An empty HTTP POST MUST NOT contain a SOAPAction header.
- An empty HTTP POST MUST NOT contain a Content-Type header.
- An HTTP response that contains any CPE WAN Management Protocol payload (a SOAP request
- to the CPE, a successful SOAP response to the CPE, or a SOAP fault response containing a Fault element defined in section 3.5) MUST use the HTTP status code 200 (OK).

3.5 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 .



Below is an example envelope containing a fault response:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  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>
```

3.6 RPC Support Requirements

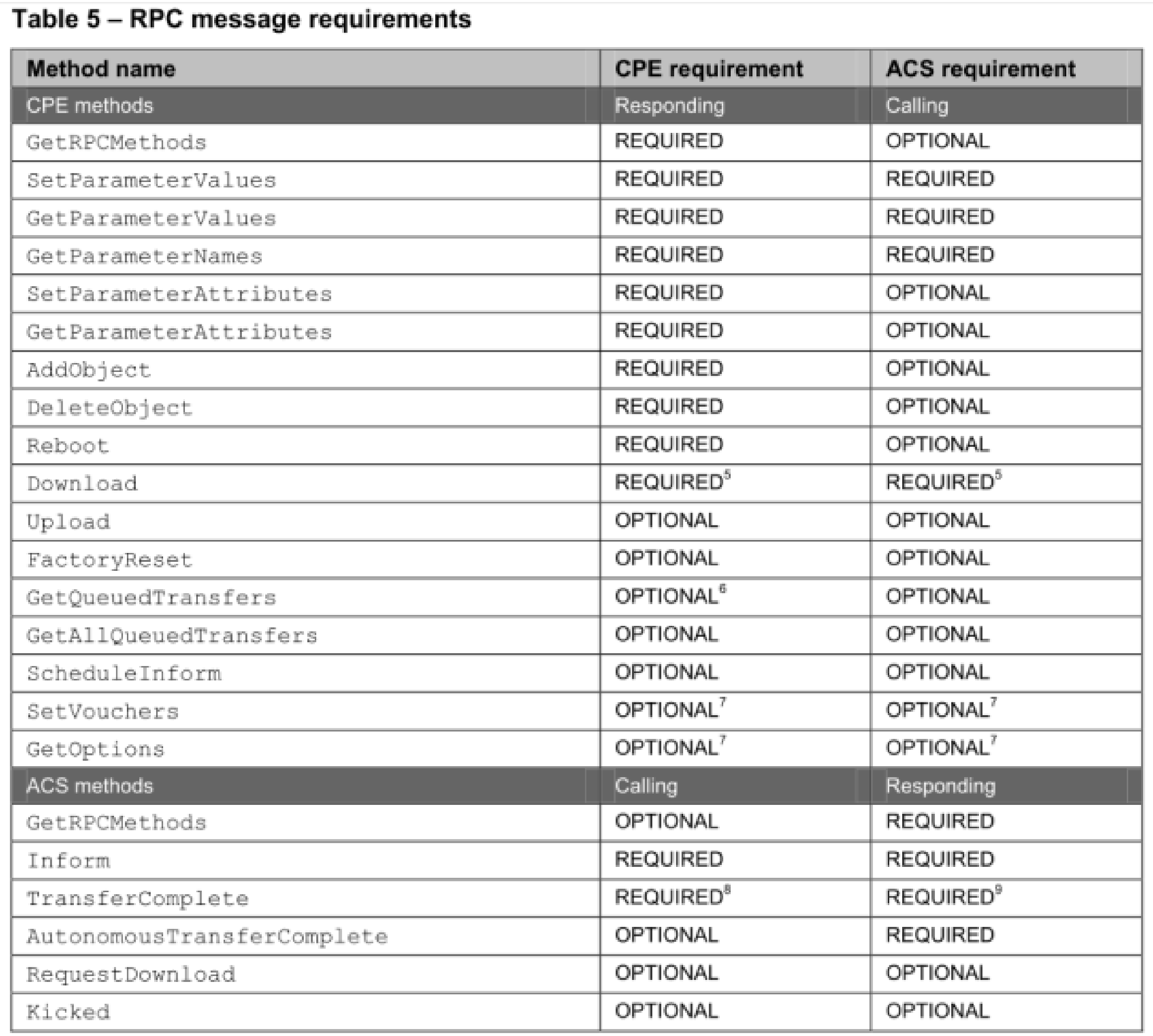


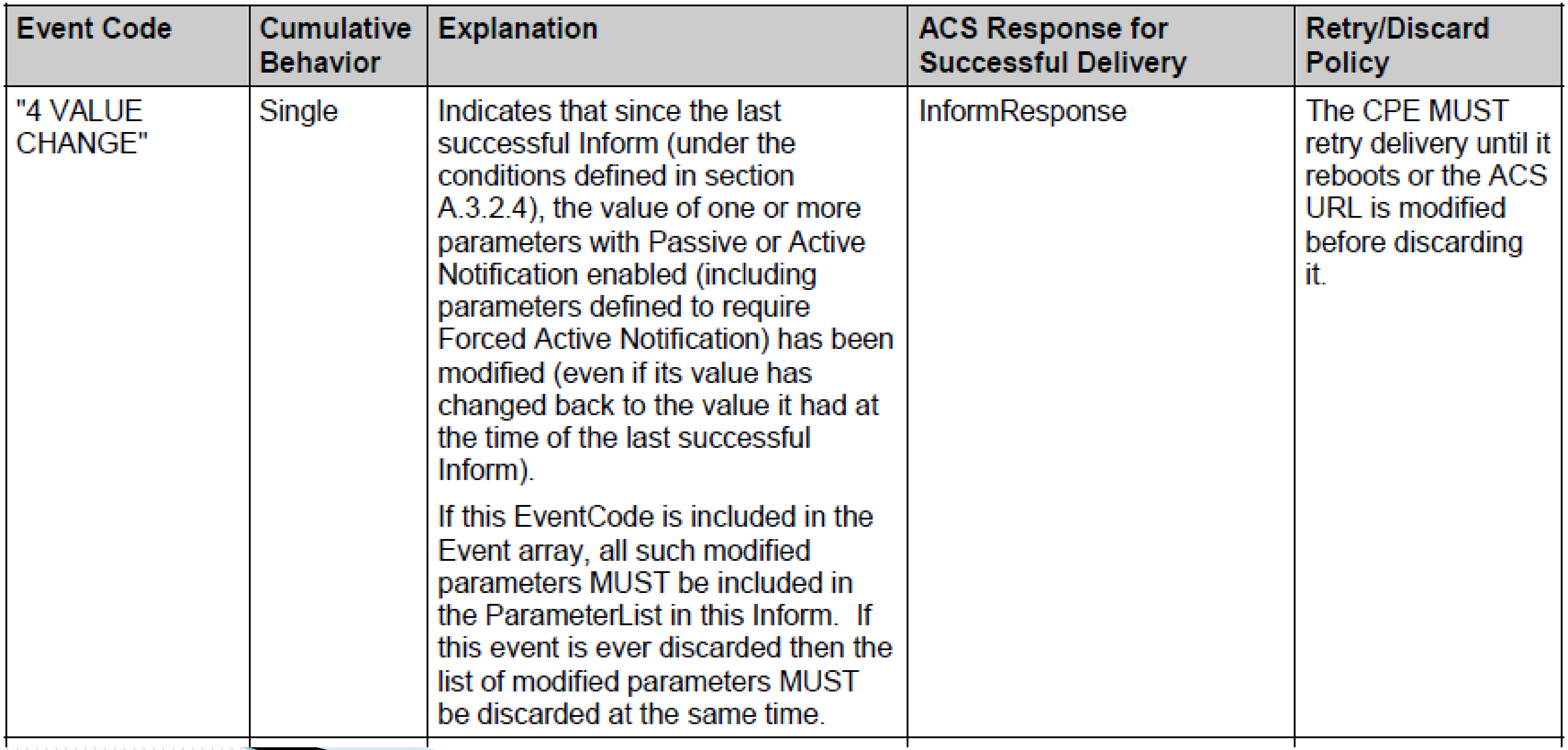
Table 5 – RPC message requirements

Method name	CPE requirement	ACS requirement
CPE methods	Responding	Calling
GetRPCMethods	REQUIRED	OPTIONAL
SetParameterValues	REQUIRED	REQUIRED
GetParameterValues	REQUIRED	REQUIRED
GetParameterNames	REQUIRED	REQUIRED
SetParameterAttributes	REQUIRED	OPTIONAL
GetParameterAttributes	REQUIRED	OPTIONAL
AddObject	REQUIRED	OPTIONAL
DeleteObject	REQUIRED	OPTIONAL
Reboot	REQUIRED	OPTIONAL
Download	REQUIRED[5]	REQUIRED[5]
Upload	OPTIONAL	OPTIONAL
FactoryReset	OPTIONAL	OPTIONAL
GetQueuedTransfers	OPTIONAL[6]	OPTIONAL
GetAllQueuedTransfers	OPTIONAL	OPTIONAL
ScheduleInform	OPTIONAL	OPTIONAL
SetVouchers	OPTIONAL[7]	OPTIONAL[7]
GetOptions	OPTIONAL[7]	OPTIONAL[7]
ACS methods	Calling	Responding
GetRPCMethods	OPTIONAL	REQUIRED
Inform	REQUIRED	REQUIRED
TransferComplete	REQUIRED[8]	REQUIRED[9]
AutonomousTransferComplete	OPTIONAL	REQUIRED
RequestDownload	OPTIONAL	OPTIONAL
Kicked	OPTIONAL	OPTIONAL

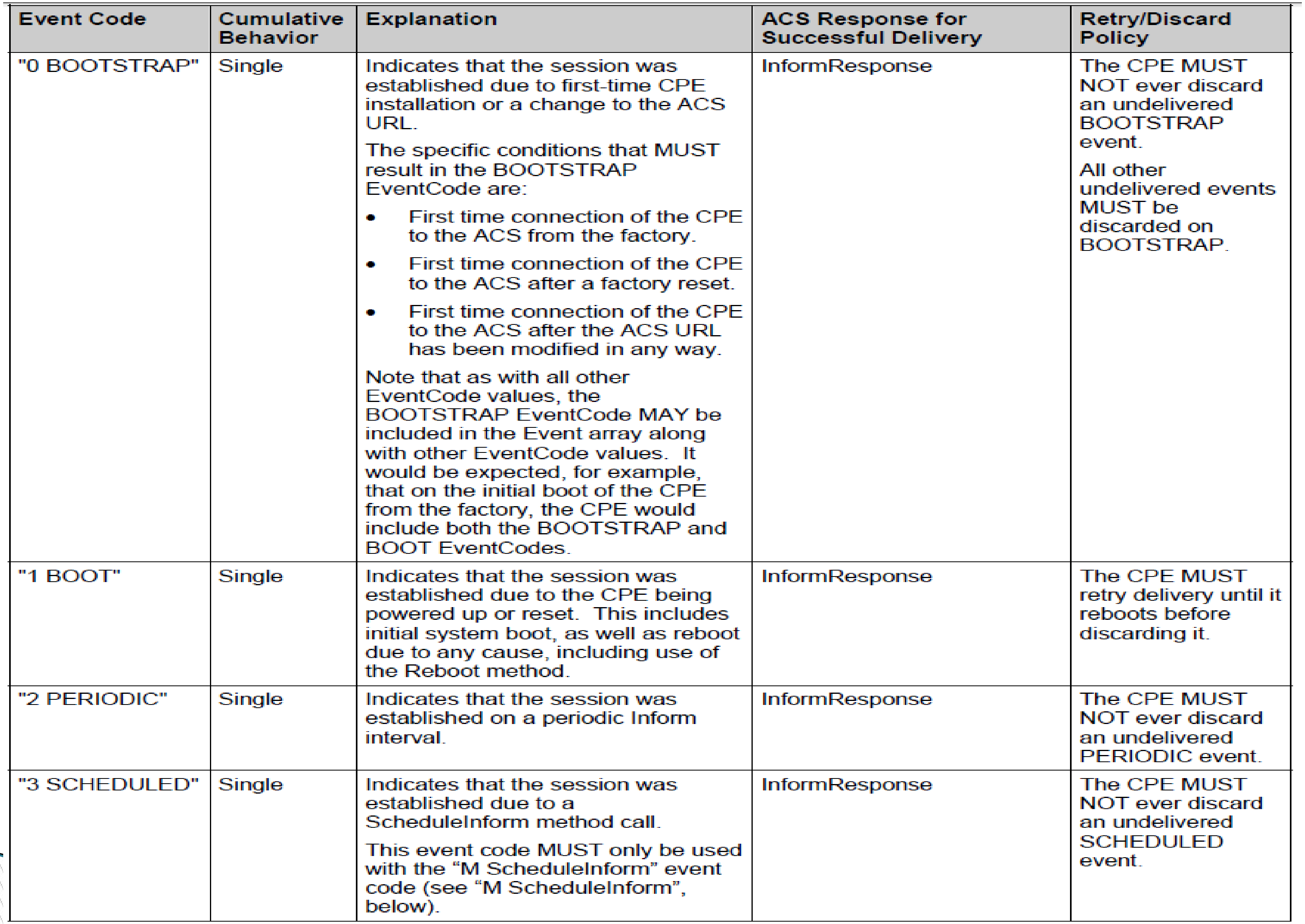
3.7 Events

An event is an indication that something of interest has happened that requires the CPE to notify the ACS via an Inform request, The CPE MUST attempt to deliver every event at least once.

The CPE MUST receive confirmation from the ACS for it to consider an event successfully delivered. Once the CPE has delivered an event successfully.



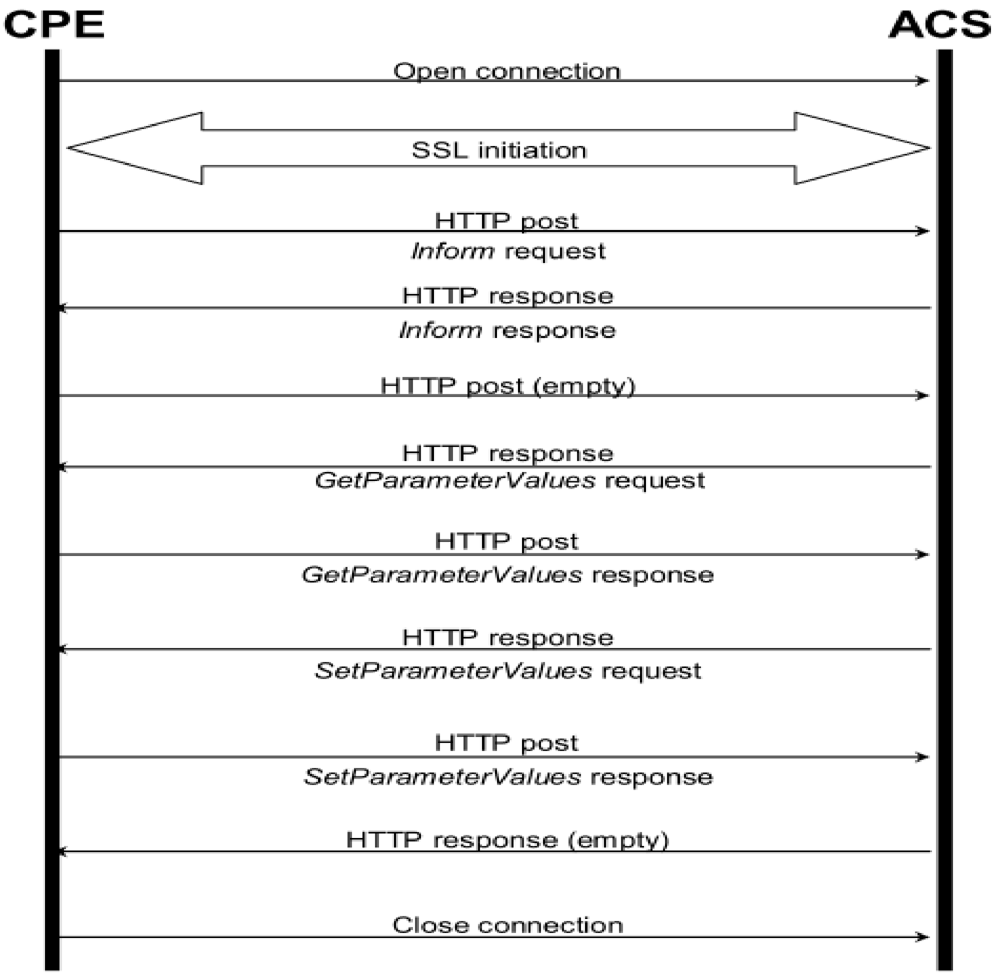
Event Code	Cumulative Behavior	Explanation	ACS Response for Successful Delivery	Retry/Discard Policy
"4 VALUE CHANGE"	Single	Indicates that since the last successful Inform (under the conditions defined in section A.3.2.4), the value of one or more parameters with Passive or Active Notification enabled (including parameters defined to require Forced Active Notification) has been modified (even if its value has changed back to the value it had at the time of the last successful Inform). If this EventCode is included in the Event array, all such modified parameters MUST be included in the ParameterList in this Inform. If this event is ever discarded then the list of modified parameters MUST be discarded at the same time.	InformResponse	The CPE MUST retry delivery until it reboots or the ACS URL is modified before discarding it.



Event Code	Cumulative Behavior	Explanation	ACS Response for Successful Delivery	Retry/Discard Policy
"0 BOOTSTRAP"	Single	Indicates that the session was established due to first-time CPE installation or a change to the ACS URL. The specific conditions that MUST result in the BOOTSTRAP EventCode are: • First time connection of the CPE to the ACS from the factory. • First time connection of the CPE to the ACS after a factory reset. • First time connection of the CPE to the ACS after the ACS URL has been modified in any way. Note that as with all other EventCode values, the BOOTSTRAP EventCode MAY be included in the Event array along with other EventCode values. It would be expected, for example, that on the initial boot of the CPE from the factory, the CPE would include both the BOOTSTRAP and BOOT EventCodes.	InformResponse	The CPE MUST NOT ever discard an undelivered BOOTSTRAP event. All other undelivered events MUST be discarded on BOOTSTRAP.
"1 BOOT"	Single	Indicates that the session was established due to the CPE being powered up or reset. This includes initial system boot, as well as reboot due to any cause, including use of the Reboot method.	InformResponse	The CPE MUST retry delivery until it reboots before discarding it.
"2 PERIODIC"	Single	Indicates that the session was established on a periodic Inform interval.	InformResponse	The CPE MUST NOT ever discard an undelivered PERIODIC event.
"3 SCHEDULED"	Single	Indicates that the session was established due to a ScheduleInform method call. This event code MUST only be used with the “M ScheduleInform” event code (see “M ScheduleInform”, below).	InformResponse	The CPE MUST NOT ever discard an undelivered SCHEDULED event.

3.8 Transaction Flow

Figure 3 – Transaction Session Example



CPE

ACS

Open connection

SSL initiation

HTTP post
Inform request

HTTP response
Inform response

HTTP post (empty)

HTTP response
GetParameterValues request

HTTP post
GetParameterValues response

HTTP response
SetParameterValues request

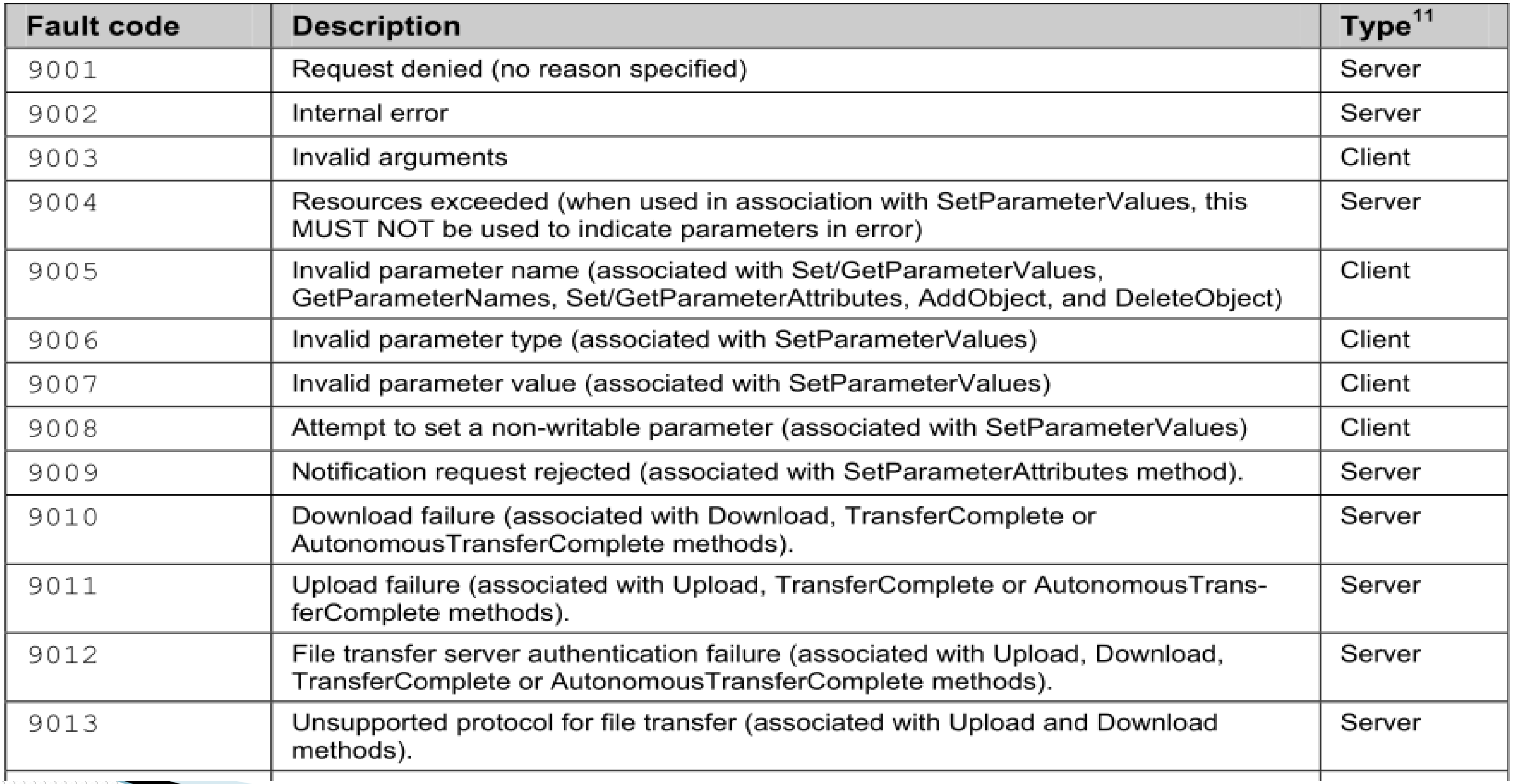
HTTP post
SetParameterValues response

HTTP response (empty)

Close connection

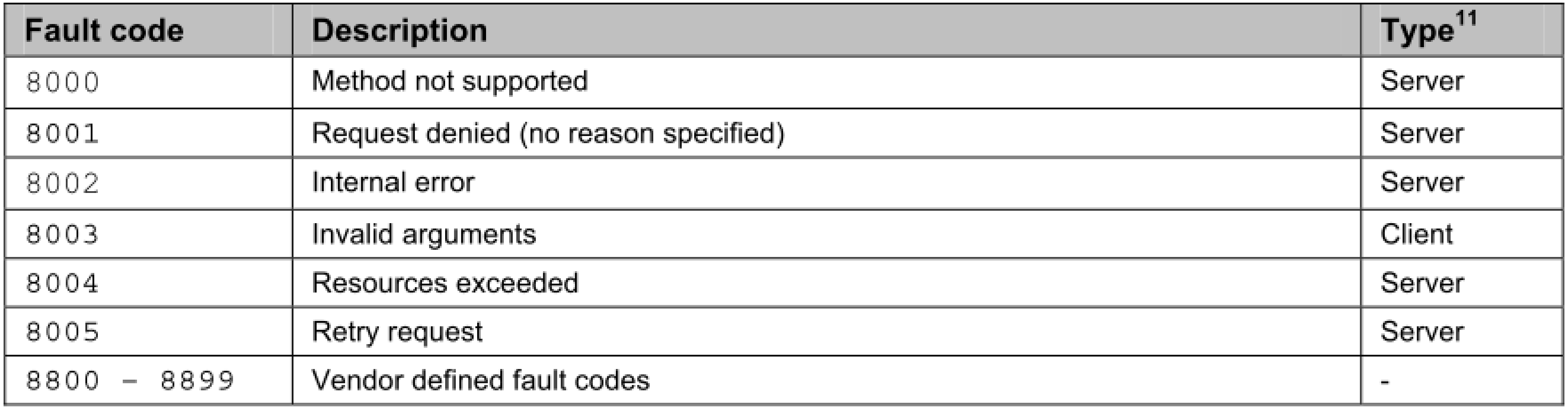
4. Fault Handling

- CPE Fault Codes



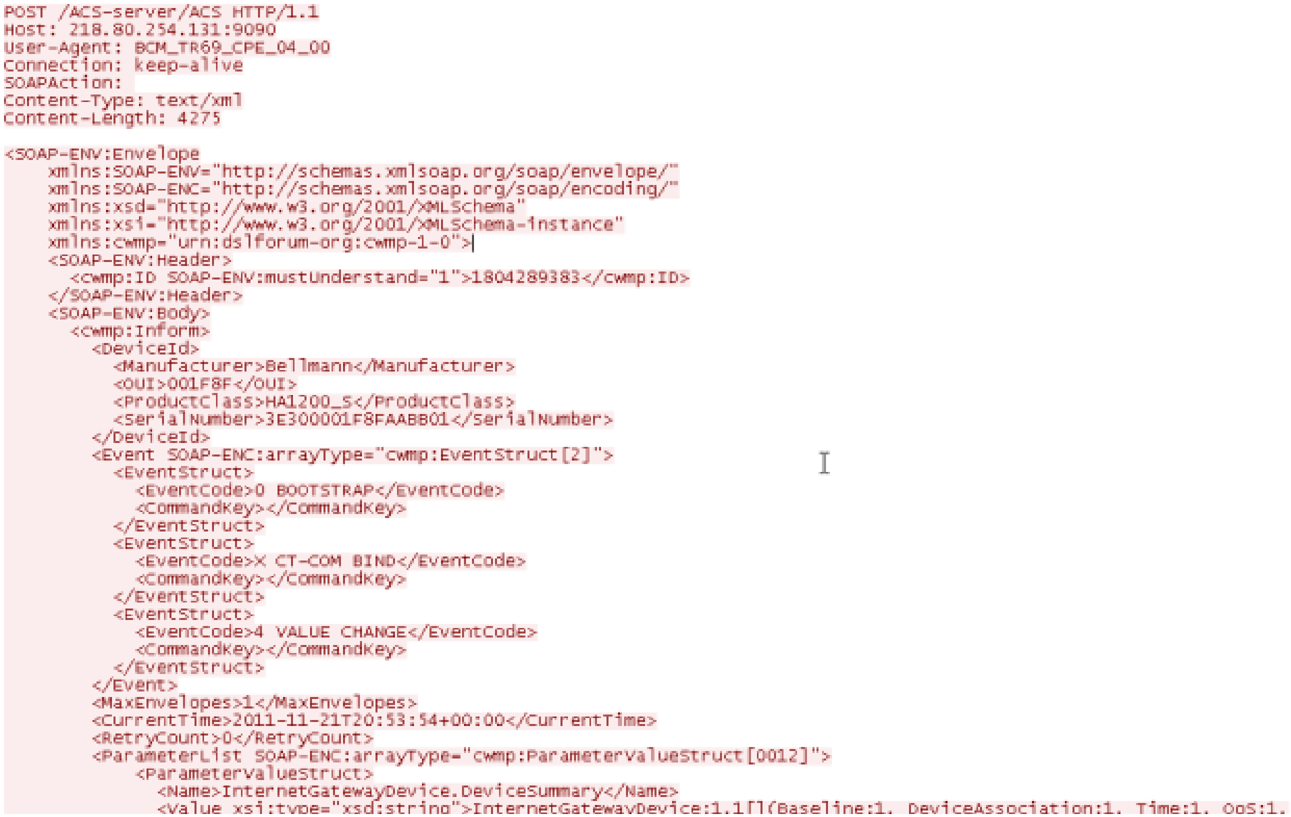
Fault code	Description	Type[11]
9001	Request denied (no reason specified)	Server
9002	Internal error	Server
9003	Invalid arguments	Client
9004	Resources exceeded (when used in association with SetParameterValues, this MUST NOT be used to indicate parameters in error)	Server
9005	Invalid parameter name (associated with Set/GetParameterValues, GetParameterNames, Set/GetParameterAttributes, AddObject, and DeleteObject)	Client
9006	Invalid parameter type (associated with SetParameterValues)	Client
9007	Invalid parameter value (associated with SetParameterValues)	Client
9008	Attempt to set a non-writable parameter (associated with SetParameterValues)	Client
9009	Notification request rejected (associated with SetParameterAttributes method).	Server
9010	Download failure (associated with Download, TransferComplete or AutonomousTransferComplete methods).	Server
9011	Upload failure (associated with Upload, TransferComplete or AutonomousTransferComplete methods).	Server
9012	File transfer server authentication failure (associated with Upload, Download, TransferComplete or AutonomousTransferComplete methods).	Server
9013	Unsupported protocol for file transfer (associated with Upload and Download methods).	Server

- ACS Fault Codes



Fault code	Description	Type[11]
8000	Method not supported	Server
8001	Request denied (no reason specified)	Server
8002	Internal error	Server
8003	Invalid arguments	Client
8004	Resources exceeded	Server
8005	Retry request	Server
8800 – 8899	Vendor defined fault codes	-

Example:



```
POST /ACS-server/ACS HTTP/1.1
Host: 218.80.254.131:9090
User-Agent: BCM_TR69_CPE_04_00
Connection: keep-alive
SOAPAction:
Content-Type: text/xml
Content-Length: 4275

<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
    <SOAP-ENV:Header>
      <cwmp:ID SOAP-ENV:mustUnderstand="1">1804289383</cwmp:ID>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <cwmp:Inform>
        <DeviceId>
          <Manufacturer>Bellmann</Manufacturer>
          <OUI>001F8F</OUI>
          <ProductClass>HA1200_S</ProductClass>
          <SerialNumber>3E300001F8FAABB01</SerialNumber>
        </DeviceId>
        <Event SOAP-ENC:arrayType="cwmp:EventStruct[2]">
          <EventStruct>
            <EventCode>0 BOOTSTRAP</EventCode>
            <CommandKey></CommandKey>
          </EventStruct>
          <EventStruct>
            <EventCode>X CT-COM BIND</EventCode>
            <CommandKey></CommandKey>
          </EventStruct>
          <EventStruct>
            <EventCode>4 VALUE CHANGE</EventCode>
            <CommandKey></CommandKey>
          </EventStruct>
        </Event>
        <MaxEnvelopes>1</MaxEnvelopes>
        <CurrentTime>2011-11-21T20:53:54+00:00</CurrentTime>
        <RetryCount>0</RetryCount>
        <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[0012]">
            <ParameterValueStruct>
              <Name>InternetGatewayDevice.DeviceSummary</Name>
              <Value xsi:type="xsd:string">InternetGatewayDevice:1.1[](Baseline:1, DeviceAssociation:1, Time:1, QoS:1,
```

example_tr69.pcap

Q&A:

1. Single object and multi-instance object
2. Customized dataModel parameters definition

Reference:

http://www.broadband-forum.org/cwmp.php
http://www.broadband-forum.org/cwmp/tr-098-1-3-0.html

Thanks