TR-069 – A Crash Course

University of New Hampshire
Interoperability Laboratory
2009

Why TR-069?

- TR-069 is the document number of the technical report, defined by the Broadband Forum, that specifies the “CPE WAN Management Protocol”, or..

“CWMP”

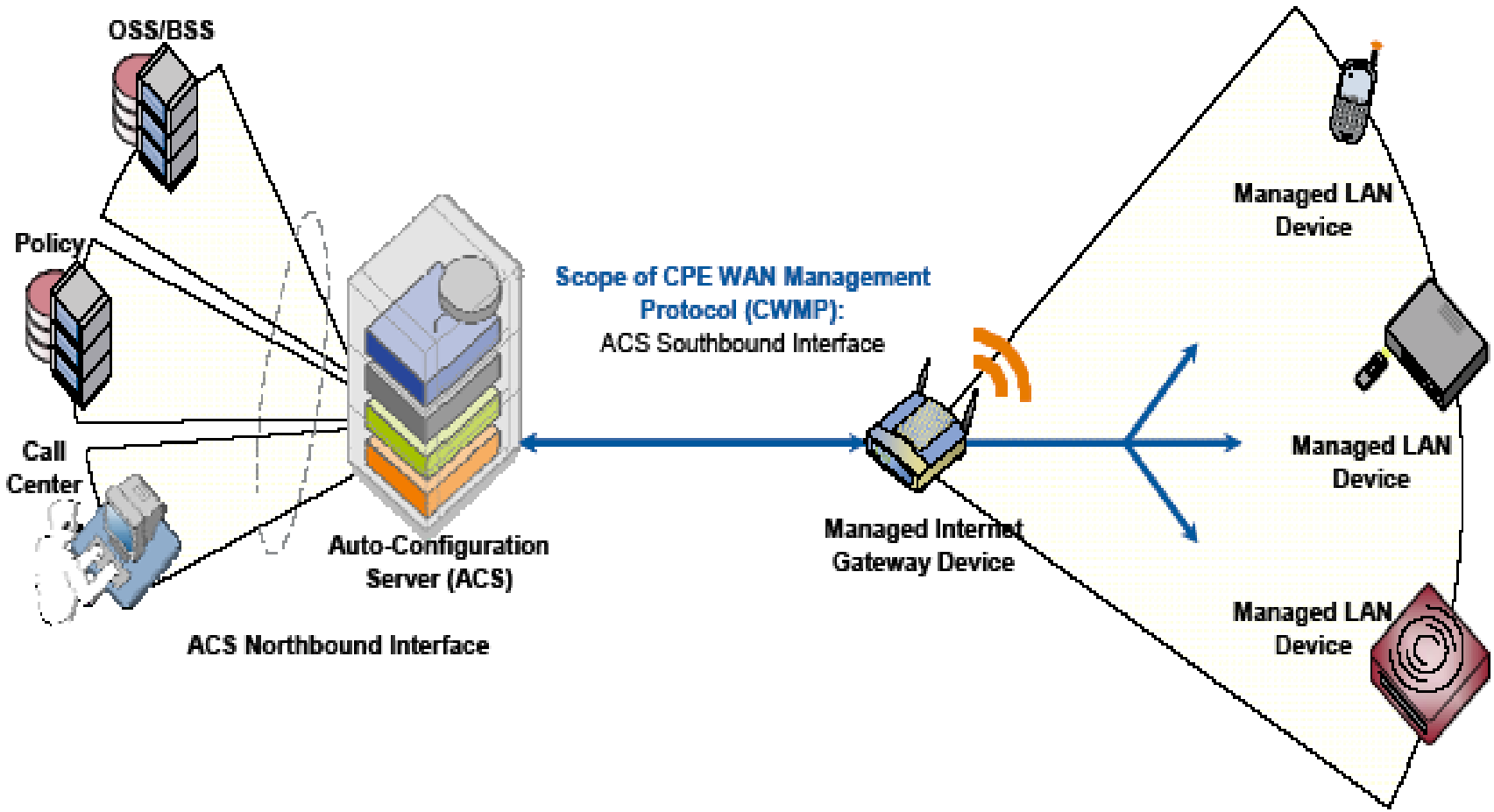
Terms

- XML – The eXtensible Markup Language
- SOAP – The "Simple Object Access Protocol"; an XML based specification for performing application function calls between web app components
- CPE – customer premises equipment, or the device that is being managed; may include Integrated Gateways (IGD), Set-top-boxes (STB), Network Attached Storage (NAS), etc.
- ACS – Auto-Configuration Server, performs the management of the CPE. Generally, operated by an ISP and plugs into their Operational or Billing Support Systems (OSS/BSS)
- Data Model – a set of objects defined for the management of a particular kind of CPE, usually defined in a companion Technical Report by the Broadband Forum
- RPC – Remote Procedure Call. A use of SOAP that allows two applications to make procedure calls on each other.

Documents

- TR-069 – CWMP
  - Currently Amendment 2, which is CWMPv1.1
  - Defines protocol, message structure, session rules, and RPCs
  - Annexes deal with NAT traversal and association of gatways to LAN devices
- TR-106
  - XML Schema definition and common objects for Device Data Models
- TR-098
  - Device Data Model for Internet Gateway Devices
- The Future… a slew of other WTs and PDs
  - Redefining generic device model, adding proxy functions, etc.

Architecture



Protocol Stack

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

XML

- eXtensible Markup Language
- Used to *describe* information
- Through SOAP, allows client/server application transactions through Remote Procedure Calls (RPCs)

XML Schema, Namespaces

- A particular use of XML is described in a "Schema" (.xsd)
- Schemas inherit, like in other languages, through "namespaces"

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- TR-069 (CWMP) data model XML schema definition -->
<!-- $Id: //depot/users/wlupton/cwmp-datamodel/cwmp-datamodel.xsd#30 $ -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:dslforum-org:cwmp:datamodel-0-5
 targetNamespace="urn:dslforum-org:cwmp:datamodel-0-5" elementFormDefault="unqualified"
attributeFormDefault="unqualified">
  <xs:simpleType name="ActiveNotify">
    <xs:annotation>
      <xs:documentation>Parameter active notify support.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:normalizedString">
      <xs:enumeration value="normal"/>
      <xs:enumeration value="forceEnabled"/>
      <xs:enumeration value="forceDefaultEnabled"/>
      <xs:enumeration value="canDeny"/>
    </xs:restriction>
  </xs:simpleType>
```

TR-069 XML Schemas

- RPC Schema (contained in TR-069 document, section A.6)
- CWMP Data Model Schema (TR-106) or cwmp-datamodel.xsd
- The Data Models are xml documents that are “schema-like”, but describe the objects and parameters used for a particular TR-069 use case.

Data Model Example – TR-143 (Performance Test Objects and Parameters)

```
      <parameter name="Interface" access="readWrite">
        <description>IP-layer interface over which the test is to be performed.  The content is the
full hierarchical parameter name of the interface.
The value of this parameter MUST be either a valid interface or an empty string.  An attempt to set
this parameter to a different value MUST be rejected as an invalid parameter value.
If {{empty}} is specified, the CPE MUST use the default routing interface.</description>
        <syntax>
          <string>
            <size maxLength="256"/>
          </string>
        </syntax>
      </parameter>
      <parameter name="UploadURL" access="readWrite">
        <description>The URL as defined in {{bibref|RFC3986}}, for the CPE to Upload to. This
parameter MUST be in the form of a valid HTTP {{bibref|RFC2616}} or FTP {{bibref|RFC959}} URL.
* When using FTP transport, FTP binary transfer MUST be used.
* When using HTTP transport, persistent connections MUST be used and pipelining MUST NOT be used.
* When using HTTP transport the HTTP Authentication MUST NOT be used.</description>
        <syntax>
          <string>
            <size maxLength="256"/>
          </string>
        </syntax>
      </parameter>
```

SOAP

- The Simple Object Access Protocol
- Is just XML, defined in a schema, that describes how to convey information between peers.
- RPC (Remote Procedure Call) is one application of SOAP that:
  - Describes how to convey methods and their arguments to be invoked on the peer
  - Describes how the response should look
  - Describes how to convey fault information
- CWMP uses RPC application of SOAP for faults, call/response format, message ID

Message Structure

```
<soapenv:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cwmp="urn:dslforum-org:cwmp-1-0"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
        <cwmp:ID soapenv:mustUnderstand="1">1323392</cwmp:ID>
    </soapenv:Header>
    <soapenv:Body>
        <cwmp:SetParameterValues>
            <ParameterList soap:arrayType="cwmp:ParameterValueStruct[1]">
                <ParameterValueStruct>
                    <Name>InternetGatewayDevice.ManagementServer.URL</Name>
                    <Value xsi:type="xsd:string">http://someacs.cwmp.org/path/</Value>
                </ParameterValueStruct>
            </ParameterList>
            <ParameterKey>1323392</ParameterKey>
        </cwmp:SetParameterValues>
    </soapenv:Body>
</soapenv:Envelope>
```

The SOAP Body contains the actual RPC call or response, as well as all of the arguments they contain. In this case, we see a SetParameterValues RPC. This is a procedure that exists on the CPE, that is called *by the ACS*, to which the CPE will respond. As you can see, the RPC uses the SOAP definition of an array, further specified by the cwmp namespace to refer to a “ParameterValueStruct”. [1] is the size of the array. The ParameterValueStruct is a name/value pair, and the XML tags represent that.

The SOAP Header, in CWMP, contains transaction information. In this case, it contains the CWMP message ID, which is used by the system to associate calls and responses. Note the use of the namespace – cwmp:ID – it is saying “use the ID data type defined in the cwmp namespace in the envelope attributes, above.”

The SOAP envelope tag. It contains, as attributes, the namespace references that the later tags will draw from. This one draws from the “Schema of Schemas”, the Broadband Forum (when created, it was the DSL Forum, and has been maintained for backwards compatibility) CWMP Schema, and the SOAP schema.

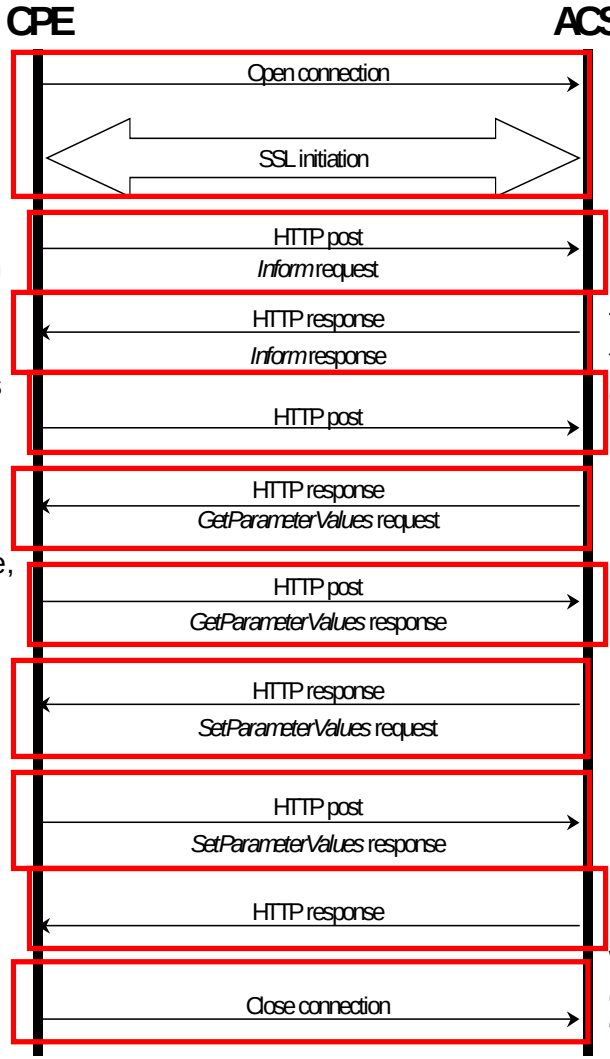
What does a session look like?

Remember, the CPE is always the initiator of the session. It begins EVERY new session by making an Inform RPC on the ACS, with arguments that include the reason for the session and a list of parameters it must inform the ACS about.

Here the CPE indicated that it has no more RPCs to make on the ACS. This is indicated by an empty HTTP post. This post may happen at any time, since the TCP session is still open.

The CPE sends a GetParameterValuesResponse, containing the information the ACS wants. This means the GetParameterValues RPC is over.

Finally, the ACS sends an empty HTTP response to indicate it has no more RPCs to make on the CPE. Since the CPE has already done the same thing, this ends the CWMP session.



The CPE initiates the connection. The CPE is always the initiator, though the ACS can make a "Connection Request" to stimulate the CPE to initiate a connection. The underlying protocols establish a connection. TR-069 is almost always performed over Secure Socket Layer (SSL), unless for testing purposes.

The ACS sends an InformReponse to the CPE for the Inform RPC. This means the Inform RPC is complete.

In this example, the ACS then makes a GetParameterValues RPC on the CPE.

The ACS makes any other Remote Procedure Calls that it needs to make on the CPE during this particular CWMP session. Here, it is a SetParameterValues RPC.

With the session closed, the underlying protocols close their connections and the session is "successfully terminated".

Connection Request

- While the CPE always initiates a session, the ACS can stimulate it to do so
- It does this by issuing a "Connection Request"
- A Connection Request is a simple HTTPGet made on the CPE at an arbitrary URL/port set by the CPE
- The CPE tells the ACS what its CR URL is during the Inform

Authentication

- TR-069 requires the use of HTTP basic, HTTP digest, or Certificate based authentication
- Authentication occurs in both directions
- The CPE authenticates the ACS's Connection Requests
- The ACS authenticates the CPE's session initiation

Remote Procedure Calls List in CWMP v1.1

CPE Methods

- GetRPCMethods
- SetParameterValues
- GetParameterValues
- GetParameterNames
- SetParameterAttributes
- GetParameterAttributes
- AddObject
- DeleteObject
- Reboot
- Download
- Upload
- FactoryReset
- GetQueuedTransfers
- GetAllQueuedTransfers
- ScheduleInform
- SetVouchers
- GetOptions

ACS Methods

- GetRPCMethods
- Inform
- TransferComplete
- AutonomousTransferComplete
- RequestDownload
- Kicked

The Inform

- The Inform RPC is a an RPC made on the ACS by the CPE
- It MUST be called FIRST in every session
- It contains the reason(s) for the session (an Event)
- Contains a list of parameters that are required by the Data Model to be included (“Forced Inform”)
- Contains parameters that the ACS set to be notified upon changes.
- The ACS completes the RPC by sending an InformResponse.

Example Inform

```
<soap:Body>
    <cwmp:Inform>
      <DeviceId>
        <Manufacturer>SomeCompany</Manufacturer>
        <OUI>000A73</OUI>
        <ProductClass>IGD</ProductClass>
        <SerialNumber>123456789</SerialNumber>
      </DeviceId>
      <Event soap-enc:arrayType="cwmp:EventStruct[1]">
        <EventStruct>
          <EventCode>2 PERIODIC</EventCode>
          <CommandKey />
        </EventStruct>
      </Event>
      <MaxEnvelopes>1</MaxEnvelopes>
      <CurrentTime>2008-10-23T01:49:14+00:00</CurrentTime>
      <RetryCount>0</RetryCount>
      <ParameterList soap-enc:arrayType="cwmp:ParameterValueStruct[6]">
        <ParameterValueStruct>
          <Name>Device.DeviceSummary</Name>
          <Value xsi:type="xsd:string" />
        </ParameterValueStruct>
        <ParameterValueStruct>
          <Name>Device.DeviceInfo.HardwareVersion</Name>
          <Value xsi:type="xsd:string">3.1</Value>
        </ParameterValueStruct>
        <ParameterValueStruct>
          <Name>Device.DeviceInfo.SoftwareVersion</Name>
          <Value xsi:type="xsd:string">1.1.5.13</Value>
        </ParameterValueStruct>
        <ParameterValueStruct>
          <Name>Device.ManagementServer.ConnectionRequestURL</Name>
          <Value xsi:type="xsd:string">http://10.0.0.5:1234/</Value>
        </ParameterValueStruct>
        <ParameterValueStruct>
          <Name>Device.ManagementServer.ParameterKey</Name>
          <Value xsi:type="xsd:string">null</Value>
        </ParameterValueStruct>
        <ParameterValueStruct>
          <Name>Device.LAN.IPAddress</Name>
          <Value xsi:type="xsd:string">192.168.1.1</Value>
        </ParameterValueStruct>
      </ParameterList>
    </cwmp:Inform>
  </soap:Body>
```

```
<soapenv:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/en
coding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:cwmp="urn:dslforum-org:cwmp-1-0"
xmlns:soapenv="http://schemas.xmlsoap.org/soap
/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <soapenv:Header>
        <cwmp:ID
soapenv:mustUnderstand="1">1</cwmp:ID>
    </soapenv:Header>
    <soapenv:Body>
        <cwmp:InformResponse>
            <MaxEnvelopes>1</MaxEnvelopes>
        </cwmp:InformResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<Event soap-
enc:arrayType="cwmp:EventStruct[1]">
     <EventStruct>
         <EventCode>2 PERIODIC</EventCode>
       <CommandKey />
     </EventStruct>
</Event>
```

Events

- The Inform "event" array contains a list of one or more pre-defined reasons for the session being initiated.

- "0 BOOTSTRAP"
- "1 BOOT"
- "2 PERIODIC"
- "3 SCHEDULED"
- "4 VALUE CHANGE"
- "5 KICKED"
- "6 CONNECTION REQUEST"
- "7 TRANSFER COMPLETE"
- "8 DIAGNOSTICS COMPLETE"
- "9 REQUEST DOWNLOAD"
- "10 AUTONOMOUS TRANSFER COMPLETE"
- "M Reboot"
- "M Scheduled Inform"
- "M Download"
- "M Upload"
- Vendor Specific Functions and Events…

Why does a CPE start a session?

When it first contacts this ACS	<-- "0 BOOTSTRAP"
When it reboots	<-- "1 BOOT"
When its periodic inform interval is met	<-- "2 PERIODIC"
When an inform was scheduled	<-- "3 SCHEDULED"
When a parameter set for notification has changed	<-- "4 VALUE CHANGED"
When it is asked to by the ACS	<-- "6 CONNECTION REQUEST"
When it has completed an upload or download and is going to call "TransferComplete"	<-- "7 TRANSFER COMPLETE" <-- "10 AUTONOMOUS TRANSFER COMPLETE"
When it has completed diagnostics	<-- "8 DIAGNOSTICS COMPLETE"
The CPE wants to request a download	<-- "9 REQUEST DOWNLOAD"

What about these "M" Events?!

- Events that start with "M" are associated with an RPC recently completed by the CPE
- Usually show up with other events
- Example:
  - "1 BOOT" and "M Reboot"
    - The CPE rebooted, and it was because it completed the "Reboot" RPC
  - "7 TRANSFER COMPLETE" and "M Download"
    - The CPE is going to call "transfer complete", and it was because it finished a download caused by the ACS use of the Download RPC

Stuff the ACS Can Do to the CPE (required)

- Learn what methods it supports
- Learn what objects or parameters exist
- Create or delete an object
- Read or edit a parameter
- Read or edit a parameter's attributes
- Reboot the CPE
- Tell it to download a file or firmware image

GetRPCMethods

- Allows the ACS (or the CPE) to learn the RPCs supported by the CPE (or ACS)
- May contain "vendor extensions" or custom RPCs

```
<soapenv:Body>
    <cwmp:GetRPCMethods/>
</soapenv:Body>
```

ACS Call

CPE Response

```
<soap:Body>
    <cwmp:GetRPCMethodsResponse>
      <MethodList soap-enc:arrayType="xsd:string[12]">
        <string>GetRPCMethods</string>
        <string>SetParameterValues</string>
        <string>GetParameterValues</string>
        <string>GetParameterNames</string>
        <string>SetParameterAttributes</string>
        <string>GetParameterAttributes</string>
        <string>AddObject</string>
        <string>DeleteObject</string>
        <string>Download</string>
        <string>Reboot</string>
        <string>ScheduleInform</string>
        <string>FactoryReset</string>
      </MethodList>
    </cwmp:GetRPCMethodsResponse>
  </soap:Body>
```

Vendor Extensions

- Companies may make custom RPCs, events, objects, and parameters.
- All follow the same format.
- `X_{OUI of Company}_{NameOfNewThing}`
  - For example: `X_012345_MyMethod`

GetParameterNames

- Allows the ACS to learn the objects and parameters that exist on the CPE
- Often used to determine device characteristics
- May be complete or partial path

```
<soapenv:Body>
     <cwmp:GetParameterNames>
     <ParameterPath>Device.LAN.IPAddress
     </ParameterPath>
     <NextLevel>0</NextLevel>
     </cwmp:GetParameterNames>
</soapenv:Body>
```

ACS Call

```
 <soap:Body>
    <cwmp:GetParameterNamesResponse>
<ParameterList soap enc:arrayType="cwmp:ParameterInfoStruct[1]">
        <ParameterInfoStruct>
          <Name>Device.LAN.IPAddress</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
      </ParameterList>
    </cwmp:GetParameterNamesResponse>
  </soap:Body>
```

CPE Response

Complete and Partial Paths

- For any “Get” RPC, and the SetParameterAtrributes RPC
- The “ParameterPath” tag can take a “Complete” or “Partial” path
- A “Complete” path references a single parameter, for example, “Device.LAN.IPAddress”
- A “Partial” path references all objects and parameters under a given “tree”
  - Referenced by parameter path that ends in “.”
  - For example, “Device.LAN.”

Next Level "True" and "False"

- For GetParameterNames, partial paths can be "Next Level True" or "Next Level False"
- If Next Level is "True", the CPE returns only those parameters directly under the ".", without any sub-objects.
- If Next Level is "False", ALL parameters, and sub-objects and their parameters are returned

GPN Examples

```
<soapenv:Body>
    <cwmp:GetParameterNames>
        <ParameterPath>Device.LAN.
        </ParameterPath>
        <NextLevel>1</NextLevel>
    </cwmp:GetParameterNames>
</soapenv:Body>
```

```
<soapenv:Body>
    <cwmp:GetParameterNames>
        <ParameterPath>Device.LAN.
        </ParameterPath>
        <NextLevel>0</NextLevel>
    </cwmp:GetParameterNames>
</soapenv:Body>
```

```
<soap:Body>
    <cwmp:GetParameterNamesResponse>
      <ParameterList soap-
    enc:arrayType="cwmp:ParameterInfoStruct[9]
    ">
        <ParameterInfoStruct>
          <Name>Device.LAN.Stats.</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
    <Name>Device.LAN.IPPingDiagnostics.</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
    <Name>Device.LAN.TraceRouteDiagnostics.</N
    ame>
          <Writable>0</Writable>
</ParameterInfoStruct>
...
        <ParameterInfoStruct>
    <ParameterInfoStruct>
          <Name>Device.LAN.MACAddress</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
      </ParameterList>
    </cwmp:GetParameterNamesResponse>
  </soap:Body>
```

```
 <soap:Body>
    <cwmp:GetParameterNamesResponse>
      <ParameterList soap-
    enc:arrayType="cwmp:ParameterInfoStruct[35
    ]">
        <ParameterInfoStruct>
          <Name>Device.LAN.</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
    <Name>Device.LAN.AddressingType</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
...
          <Name>Device.LAN.Stats.</Name>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
    <Name>Device.LAN.Stats.ConnectionUpTime</N
    ame>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
    <Name>Device.LAN.Stats.TotalBytesSent</Nam
    e>
          <Writable>0</Writable>
        </ParameterInfoStruct>
        <ParameterInfoStruct>
...
    <Name>Device.LAN.IPPingDiagnostics.</Name>
          <Writable>0</Writable>
```

GetParameterValues

- CPE returns name/value pairs listing requested parameters and their current values
- Can be complete, partial, or contain multiple of each in the RPC arguments.

```
<soap:Body>
   <cwmp:GetParameterValuesResponse>
     <ParameterList soap-enc:arrayType="cwmp:ParameterValueStruct[1]">
       <ParameterValueStruct>
         <Name>Device.ManagementServer.URL</Name>
         <Value xsi:type="xsd:string">http://someacs.tr69.com:9999/acs/</Value>
       </ParameterValueStruct>
     </ParameterList>
   </cwmp:GetParameterValuesResponse>
 </soap:Body>
```

SetParameterValues

- Allows ACS to write parameters on CPE
- Always complete path

```
<soap:Body>
   <cwmp:SetParameterValues>
     <ParameterList soap-enc:arrayType="cwmp:ParameterValueStruct[1]">
       <ParameterValueStruct>
         <Name>Device.ManagementServer.URL</Name>
         <Value xsi:type="xsd:string">http://someacs.tr69.com:9999/acs/</Value>
       </ParameterValueStruct>
     </ParameterList>
   </cwmp:SetParameterValuesResponse>
 </soap:Body>
```

```
<soap:Body>
   <cwmp:SetParameterValuesResponse>
           <Status>0</Status>
</cwmp:SetParameterValuesResponse>
  </soap:Body>
```

About the “Status” argument

- SetParameterValues, AddObject, DeleteObject, and Download all use the “Status” argument in the Response
- A Status of “0” means the changes have already been immediately applied.
- A Status of “1” means they will be applied later, possibly after a reboot.
- Source of a LOT of interoperability problems!

Get/Set Parameter Attributes

- Every parameter in CWMP has two XML attributes, Notification and Access List
- Changed and read with GetParameterAttributes and SetParameterAttributes, respectively
- SetParameterAttributes has additional argument stating whether or not each of the two (Notification or Access List) are being asked to be changed from their current state.

Notifications

- Attribute can be Active (2), Passive (1), or None (0)
- Active – CPE must start a session and inform the ACS of a value change as soon as it occurs
- Passive – CPE must inform the ACS of a value change when it next contacts it for whatever other reason
- CPE may deny a parameter being set to Active, but may NOT deny a parameter being set to Passive!
- Inform contains "4 VALUE CHANGE" event code when a notification occurs, and must contain the parameter and value in the Inform RPC.

Add Object/Delete Object

- Allows the ACS to create or delete instances of objects available on the CPE, such as PortMapping entries
- Creates the associated parameters and sub-objects with it.
- Returns an instance number that the CPE then uses to reference the object
- Instance numbers are not standardized and must remain! If something is deleted with an instance number "between" two other instances, the others cannot change!

```
<soapenv:Body>
<cwmp:AddObject>
<ObjectName>InternetGatewayDevice.WANDevice.1.WANConnectionDevic
e.1.WANPPPConnection.1.PortMapping.
</ObjectName>
<ParameterKey>3544041</ParameterKey>
</cwmp:AddObject>
</soapenv:Body>
```

```
<SOAP-ENV:Body SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<cwmp:AddObjectResponse>
<InstanceNumber>4</InstanceNumber>
<Status>0</Status>
</cwmp:AddObjectResponse>
</SOAP-ENV:Body>
```

In this example, the CPE would then have an entry in its database called:

InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnection.1.Portmapping.4.

Reboot

- Very straightforward – ACS asks the CPE to reboot.
- Only for diagnostic or troubleshooting purposes – CPE should reboot on its own if it needs to change configuration or firmware.
- Triggers the “M Reboot” event

Download

- Used by ACS to trigger the CPE to download
  - New Firmware
  - Other files or web content
- Right now, almost always for firmware, which causes some interop problems
- Triggers the "M Download" and "7 TRANSFER COMPLETE" events
- Can be immediate or delayed a time specified by the ACS
- CPE then calls the TransferComplete function on the ACS
  - Only done AFTER firmware has been applied

Other RPCs on the CPE

- Upload – triggers an upload by the CPE to a specified location
- ScheduledInform – schedules a CWMP session for a particular time and triggers the "3 SCHEDULED" event
- FactoryReset
- GetAllQueuedTransfers – returns the current uploads/downloads waiting to complete

Stuff the CPE can do on the ACS

- Send it an Inform and expect a response
- Let it know that a file transfer has completed, either one that was called for by the ACS or an Autonomous Transfer
- Request that a file transfer be started, such as a firmware download

TransferComplete

- RPC called *on the ACS by the CPE* after an upload or download has completed (AND has been applied, if firmware)
- Accompanied by the "7 TRANSFER COMPLETE" *if being called in a new session*
- AutonomousTransferComplete – similar, but used only if the transfer was requested by a source other than the ACS

Faults

- Handled through SOAP
- Listed in section A.5 of TR-069a2
- CPE fault codes are 9000 series
- ACS fault codes are 8000 series
- SetParameterValues faults include the offending parameters!

That's the protocol…
Now onto the Data Models

What's a Data Model?

- A Data Model is a structured representation of the objects, parameters, and their syntax that can be used to "model" a particular use case of TR-069
- Most often, models a particular "kind" of CPE or the services that might exist on one
- The documents have evolved over time and inherit or build on each other

The Documents…

- TR-069 – Specifies CWMP
- TR-098 – Specifies the root data model for Internet Gateway Devices (in Device:1)
- TR-106 – Specifies the schema for device models and device types (used by all other data model documents) – more about device type schema later!
- TR-111 – (was subsumed into Annexes F and G of TR-069 Amendment 2)
- TR-157 – Specifies Component objects for use in data models
- TR-143 – Specifies diagnostics objects for use in data models
- TR-181i1 – Specifies Device:1
- TR-181i2 – Specifies Device:2
- TR-104, TR-135, TR-140, TR-192 – specifies service data models for VoIP, IPTV, NAS, and FAP, respectively

XML Documents and Versioning

- Data models are published on the Broadband Forum website:
  - http://www.broadband-forum.org/cwmp.php
- Document versions are of the format "major-minor-corregendum", for example, tr-098-1-1-0.xml
- Intention is for minor revisions are backwards compatible, major revisions not

Importing models

- Data models import from one another, then add or refine definitions of objects and parameters, creating a new version

```
<dm:document xmlns:dm="urn:broadband-forum-org:cwmp:datamodel-1-0"
             xmlns:dmr="urn:broadband-forum-org:cwmp:datamodel-report-0-1"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="urn:broadband-forum-org:cwmp:datamodel-1-0 cwmp-datamodel.xsd
                                 urn:broadband-forum-org:cwmp:datamodel-report-0-1 cwmp-datamodel-report.xsd"
             spec="urn:broadband-forum-org:tr-098-1-2-0">

  <import file="tr-069-biblio.xml" spec="urn:broadband-forum-org:tr-069-biblio"/>

  <import file="tr-106-1-0-types.xml" spec="urn:broadband-forum-org:tr-106-1-0">
    <dataType name="IPAddress"/>
    <dataType name="MACAddress"/>
  </import>

  <import file="tr-143-1-0.xml" spec="urn:broadband-forum-org:tr-143-1-0">
    <model name="InternetGatewayDevice:1.3"/>
  </import>
```

Here, TR-098-1-2 imports the dataTypes IPAddress and MACAddress from TR-106-1-0-types, and imports the entire InternetGatewayDevice:1.3 from TR-143-1-0

Document and Version Alignment

- The document version and the data model version may be different!
- There is not a direct correlation between a data model and a document. A data model may be changed by multiple documents, tracked through imports
- *Data Model Versions* are denoted as Name:Version, for example InternateGatewayDevice:1.4, or Device:2

```
<import file="tr-143-1-0.xml" spec="urn:broadband-forum-org:tr-143-1-0">
  <model name="InternetGatewayDevice:1.3"/>
</import>
```

For example, though the majority of the InternetGatewayDevice data model is specified in TR-098, updates resulting in version InternetGatewayDevice:1.3 were released in TR-143.

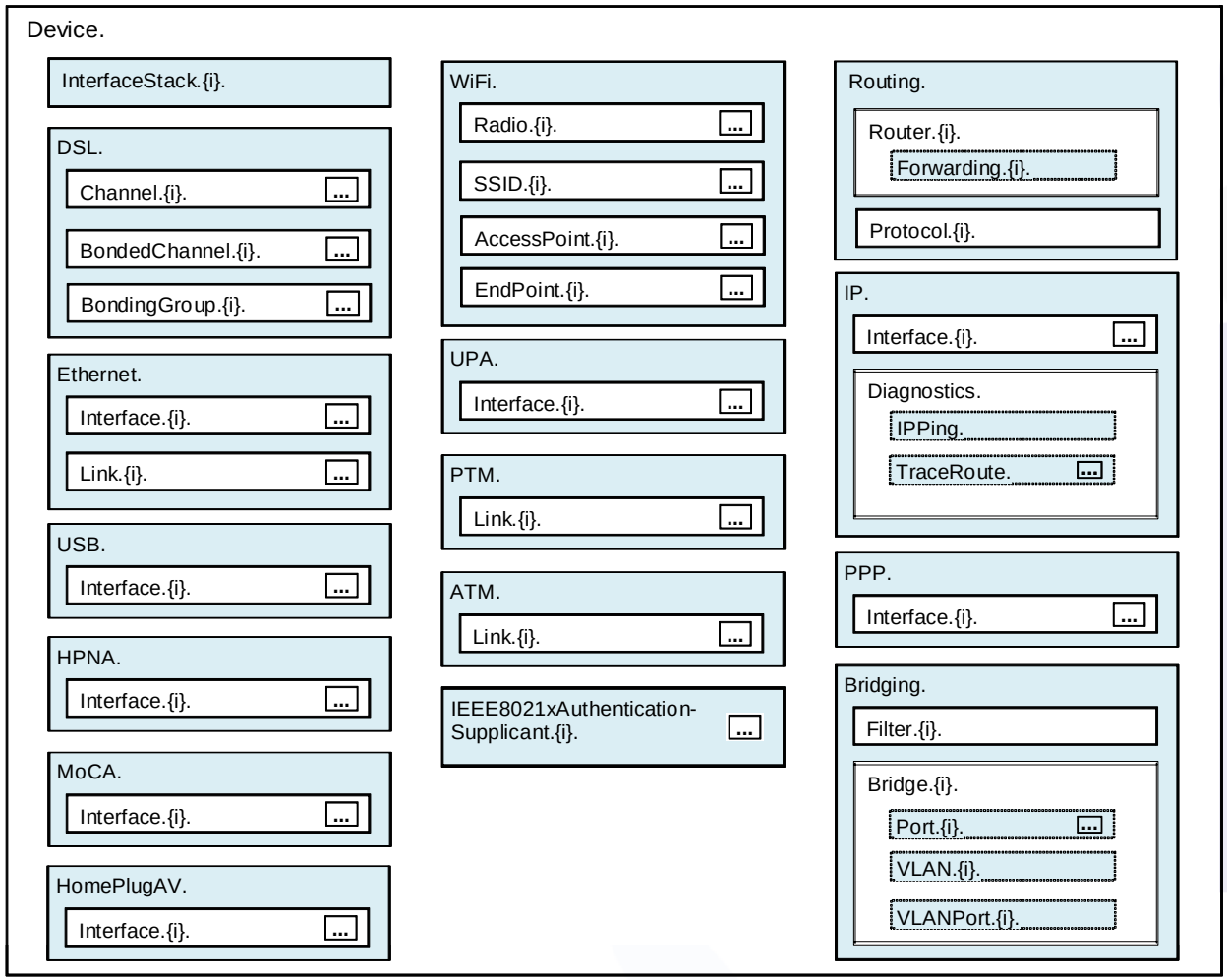
Devices, Components and Services

- At the heart of each data model are devices, components, and services
- Devices
  - The root of the data model. In the first generation of the data models, the this is separated into Device. and InternetGatewayDevice., which are different root models. In TR-181i2 (Device:2), Device. is the root of the data model.
- Components
  - Specified in TR-157
  - Indicate common objects and parameters that can be included in any Device (such as .ManagementServer.) and/or services (such as diagnostic objects specified in TR-143)
- Services
  - Data models that add additional functionality related to the delivery of specific services. These include TR-135 (Set Top Boxes), TR-104 (VoIP), TR-140 (Network Attached Storage), TR-192 (Femto Access Points), etc.

… and Interfaces

- In Device:2, the concept of interfaces and the interface stack are introduced
- An interface is a particular OSI stack component, with its objects and paramters specified within it
- The interface stack is a table containing the links between interfaces – thus any necessary combination of internetworking protocols can be assembled
- The interface stack is read-only, and can be used to learn a device's configuration quickly

Interface Examples



Source: Broadband Forum WT-181i2

Publishing of the Documents

- Descriptive text is published as pdf
- Normative text is published as XML
- Tools exist to trace the xml import tree and generate html tables explaining the data models
- Open source tools exist at:
  - https://tr69xmltool.iol.unh.edu/

Parts of a DM - Objects

- An object is a data structure in the data model. It has attributes and elements.

```
<object name="Device.DeviceInfo." access="readOnly" minEntries="1" maxEntries="1">
      <description>This object contains general device information.</description>
```

- An object can be single instance or multi-instance (remember Add/Delete Object)?
  - Defined by the ”minEntries”/”maxEntries” attributes

Object Attributes

- Name – the name of the object. This can also be "Base", which indicates that it is being extended from an object in one of the imported data models.
- Access – readOnly objects cannot be altered with Add/Delete Object
- minEntries – The minimum number of instances of the object that must exist
- maxEntries – The maximum number of instances that can exist

Object Elements

- Description – contains description and normative text, including any functionality requirements
- Unique Key – specifies the parameter that acts as a unique key for the table (in multi-instance objects)
- Parameter(s) – the meat of an object – parameters have their own sub elements and attributes

What do the curly braces mean?

- In the description element, text enclosed by {{curly braces}} indicates a reference to another object, parameter, or document in the bibliography.
- The reporting tool for BBF XML will change these into hyperlinks

Parts of a DM - Parameters

- A Parameter is a variable of an object in the Data Model

```
<parameter name="Date" access="readOnly">
  <description>Date and time when the content of the current version of this vendor
configuration file was first applied by the CPE.</description>
  <syntax>
    <dateTime/>
  </syntax>
</parameter>
```

```
<parameter name="Username" access="readWrite">
  <description>Username used to authenticate the CPE when making a connection to the ACS using
the CPE WAN Management Protocol.
This username is used only for HTTP-based authentication of the CPE.
Note that on a factory reset of the CPE, the value of this parameter might be reset to its factory
value.  If an ACS modifies the value of this parameter, it SHOULD be prepared to accommodate the
situation that the original value is restored as the result of a factory reset.</description>
  <syntax>
    <string>
      <size maxLength="256"/>
    </string>
  </syntax>
</parameter>
```

Parameter Attributes

- Name – the name of the parameter. This can also be "Base", which indicates that it is being extended from a similar parameter in one of the imported data models.
- Access – readOnly or readWrite. readOnly parameters cannot be affected by SetParameterValues

Parameter Elements

- Description – contains description and normative text, including any functionality requirements
- Syntax – the data type of the parameter and any restrictions on the values
  - Syntaxes are usually self explanatory. They are defined in the Data Model schema

How do we know what a CPE supports?

- The Hard Way – using GetParameterNames multiple times
- Profiles – Defined in each DM, profiles indicate the minimum requirements to support a particular functionality and are conveyed by the CPE in the "DeviceSummary" parameter. This was deprecated, however, in favor of…

The Device Type Schema

- Defined in Annex B of TR-106
- Can be shared beforehand or passed in SupportedDataModel parameter.
- A DT *instance* an *XML document* that is based on the DT *schema* and imports some or all of a particular *Data Model*.

Example Device Type

- Imports Data Model XML, or parts of them

```
<dt:document xmlns:dt="urn:broadband-forum-org:cwmp:devicetype-1-0"
             xmlns:dmr="urn:broadband-forum-org:cwmp:datamodel-report-0-1"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="urn:broadband-forum-org:cwmp:devicetype-1-0 cwmp-devicetype-1-0.xsd
urn:broadband-forum-org:cwmp:datamodel-report-0-1 cwmp-datamodel-report.xsd"
             deviceType="urn:example-com:device-1-0-0">
  <annotation>Auto-generated from Baseline:1, Time:1, TraceRoute:1, WiFiLAN:1, WiFiLAN:2, QoS:1, QoS:2,
Bridging:1 and Bridging:2 profiles.</annotation>
  <import file="tr-098-1-2-0.xml" spec="urn:broadband-forum-org:tr-098-1-2-0">
    <model name="InternetGatewayDevice:1.4"/>
  </import>
```

This Device Type Instance is based on the device type 1-0 schema, and imports the Data Model InternetGatewayDevice:1.4, which is defined in tr-098-1-2-0.xml

Example Device Type, cont.

```
  <model ref="InternetGatewayDevice:1.4">
    <object ref="InternetGatewayDevice." access="readOnly" minEntries="1" maxEntries="1" dmr:version=
"1.0">
      <parameter ref="DeviceSummary" access="readOnly" dmr:version="1.1"/>
      <parameter ref="LANDeviceNumberOfEntries" access="readOnly" dmr:version="1.0"/>
      <parameter ref="WANDeviceNumberOfEntries" access="readOnly" dmr:version="1.0"/>
    </object>
    <object ref="InternetGatewayDevice.DeviceInfo." access="readOnly" minEntries="1" maxEntries="1"
dmr:version="1.0">
      <parameter ref="Manufacturer" access="readOnly" dmr:version="1.0"/>
      <parameter ref="ManufacturerOUI" access="readOnly" dmr:version="1.0"/>
      <parameter ref="ModelName" access="readOnly" dmr:version="1.0"/>
      <parameter ref="Description" access="readOnly" dmr:version="1.0"/>
      <parameter ref="SerialNumber" access="readOnly" dmr:version="1.0"/>
      <parameter ref="HardwareVersion" access="readOnly" dmr:version="1.0"/>
      <parameter ref="SoftwareVersion" access="readOnly" dmr:version="1.0"/>
      <parameter ref="SpecVersion" access="readOnly" dmr:version="1.0"/>
      <parameter ref="ProvisioningCode" access="readWrite" dmr:version="1.0"/>
      <parameter ref="UpTime" access="readOnly" activeNotify="willDeny" dmr:version="1.0"/>
      <parameter ref="DeviceLog" access="readOnly" activeNotify="willDeny" dmr:version="1.0"/>
    </object>
```

The Device Type Instance specifies the parameters in the imported data model that it supports, what their access permissions are, as well as the version. It also indicates (with the “activeNotify” attribute) whether or not the device will deny attempts to set Active Notification on the parameter.

More Resources

- Broadband Forum Website:
  - http://www.broadband-forum.org
- UNH-IOL TR-069 Consortium
  - http://www.iol.unh.edu/services/testing/tr069/
- TR-069 XML Tools Repository
  - http://tr69xmltool.iol.unh.edu/
- W3 Schools XML Primer
  - http://www.w3schools.com/xml/default.asp