

**PacketCable™**

## **SMA Provisioning Specification**

**PKT-SP-SMA-PROV-I01-081121**

**ISSUED**

### **Notice**

This PacketCable specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein.

© Copyright 2008 Cable Television Laboratories, Inc.  
All rights reserved.

## Document Status Sheet

|                                   |                                |                      |                             |                   |
|-----------------------------------|--------------------------------|----------------------|-----------------------------|-------------------|
| <b>Document Control Number:</b>   | PKT-SP-SMA-PROV-I01-081121     |                      |                             |                   |
| <b>Document Title:</b>            | SMA Provisioning Specification |                      |                             |                   |
| <b>Revision History:</b>          | I01 - Released 11/21/08        |                      |                             |                   |
| <b>Date:</b>                      | November 21, 2008              |                      |                             |                   |
| <b>Status:</b>                    | <del>Work in Progress</del>    | <del>Draft</del>     | <b>Issued</b>               | <del>Closed</del> |
| <b>Distribution Restrictions:</b> | <del>Author Only</del>         | <del>CL/Member</del> | <del>CL/Member/Vendor</del> | <b>Public</b>     |

### Key to Document Status Codes

- Work in Progress**    An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
  
- Draft**                    A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
  
- Issued**                    A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
  
- Closed**                    A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

### Trademarks

CableLabs®, DOCSIS®, EuroDOCSIS™, eDOCSIS™, M-CMTS™, PacketCable™, EuroPacketCable™, PCMM™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-Card™, DCAS™, tru2way™, and CablePC™ are trademarks of Cable Television Laboratories, Inc.

# Contents

|                |  |           |
|----------------|--|-----------|
| <b>1</b>       | <b>SCOPE</b> .....   | <b>1</b>  |
| 1.1            | Introduction and Purpose.....                                    | 1         |
| 1.2            | Document Overview.....   | 1         |
| 1.3            | Requirements.....  | 1         |
| <b>2</b>       | <b>REFERENCES</b> .....  | <b>3</b>  |
| 2.1            | Normative References .....                                       | 3         |
| 2.2            | Informative References.....                                      | 3         |
| 2.3            | Reference Acquisition .....                                      | 3         |
| <b>3</b>       | <b>TERMS AND DEFINITIONS</b> .....                               | <b>4</b>  |
| <b>4</b>       | <b>ABBREVIATIONS AND ACRONYMS</b> .....                          | <b>5</b>  |
| <b>5</b>       | <b>OVERVIEW</b> .....  | <b>6</b>  |
| 5.1            | SMA gateways and Deployment Scenarios.....                       | 6         |
| 5.2            | PacketCable SMA Provisioning Mechanisms .....                    | 6         |
| 5.3            | IP Network Environments .....                                    | 6         |
| 5.4            | PacketCable SMA Data Models .....                                | 7         |
| <b>6</b>       | <b>PACKETCABLE SMA PROVISIONING</b> .....                        | <b>8</b>  |
| 6.1            | DHCP- and SNMP-based Provisioning.....                           | 8         |
| 6.1.1          | <i>MIB Requirements</i> .....                                    | 9         |
| 6.1.2          | <i>Pre-configuration and Persistence Requirements</i> .....      | 12        |
| 6.2            | RESTful web services based Provisioning.....                     | 12        |
| 6.2.1          | <i>Provisioning Flow</i> .....                                   | 13        |
| 6.2.2          | <i>IP Connectivity</i> .....                                     | 14        |
| 6.2.3          | <i>Pre-configuration and Persistence Requirements</i> .....      | 14        |
| 6.2.4          | <i>Data Model and Configuration Changes</i> .....                | 15        |
| 6.2.5          | <i>Software Download</i> .....                                   | 15        |
| 6.2.6          | <i>Security</i> .....  | 15        |
| 6.3            | Management Event Reporting .....                                 | 16        |
| <b>ANNEX A</b> | <b>SMA DATA MODELS</b> .....                                     | <b>17</b> |
| A.1            | Provisioning and Management Object Model Definitions .....       | 17        |
| A.1.1          | <i>Base Object</i> .....   | 19        |
| A.1.2          | <i>SMASig Object</i> .....                                       | 19        |
| A.1.3          | <i>XMLNamespaceToken</i> .....                                   | 21        |
| A.1.4          | <i>Certificates</i> .....  | 21        |
| A.1.5          | <i>Op Object</i> .....   | 22        |
| A.1.6          | <i>IPv4Cfg Object</i> .....                                      | 22        |
| A.1.7          | <i>IPv6Cfg Object</i> .....                                      | 22        |
| <b>ANNEX B</b> | <b>DHCP- AND SNMP-BASED PROVISIONING DATA MODELS</b> .....       | <b>23</b> |
| B.1            | SNMP MIB Objects from existing MIB Modules Requirements .....    | 23        |
| B.2            | SNMP MIB Module .....  | 24        |
| <b>ANNEX C</b> | <b>RESTFUL WEB SERVICES BASED PROVISIONING DATA MODELS</b> ..... | <b>32</b> |
| C.1            | SMACfgEnvelope Object XML Schema Definition .....                | 32        |
| C.2            | SMASig Object XML Schema Definition.....                         | 32        |

C.3 XMLNamespaces Schema Definition .....33  
 C.4 Certificates Schema Definition .....33  
 C.5 Base Object Schema Definition .....33  
 C.6 Op Object Schema Definition.....34  
 C.7 IPv4Elements Object Schema Definition .....34  
 C.8 IPv6Elements Object Schema Definition .....35  
 C.9 Management Event Mechanism (MEM) XML Schema .....35  
**APPENDIX I ACKNOWLEDGEMENTS .....36**

## Figures

Figure 1 - DHCP- and SNMP-based Provisioning Interfaces .....8  
 Figure 2 - RESTful web services Based Provisioning Interfaces .....12  
 Figure 3 - RESTful web services Based Provisioning Flow .....13  
 Figure 4 - SMA Provisioning and Management Object Model.....18

## Tables

Table 1 - SMA gateway ifTable Requirements .....9  
 Table 2 - ipNetToPhysicalTable MIB Object Details .....10  
 Table 3 - SMA gateway usmUserTable Entry.....10  
 Table 4 - SMA gateway vacmSecurityToGroupTable Entry .....11  
 Table 5 - SMA gateway vacmAccessTable Entry .....11  
 Table 6 - Additional Management Events .....16  
 Table 7 - SMASig Object .....19  
 Table 8 - XMLNamespaceToken Object.....21  
 Table 9 - XMLNamespaceToken Object.....21  
 Table 10 - SMA gateway SNMP MIB Objects Requirements (Existing MIB Modules).....23

# 1 SCOPE

## 1.1 Introduction and Purpose

This specification describes the provisioning and management of PacketCable Security, Monitoring, and Automation (SMA) Gateways. The purpose is to specify the device, network and protocol requirements to configure and manage SMA gateways, along with the associated data element definitions. Provisioning and managing the SMA devices (e.g., controls and sensors) are out of scope of this document.

## 1.2 Document Overview

The document is structured as follows:

- Section 2 - References
- Section 3 - Terms and Definitions
- Section 4 - Abbreviations and Acronyms
- Section 5 - Overview
- Section 6 - PacketCable SMA Provisioning Requirements
- Annex A - PacketCable SMA Provisioning Data Models
- Annex B - DHCP- and SNMP-based Provisioning Data Models
- Annex C - Web Services based Provisioning Data Models

## 1.3 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

|              |   |
|--------------|---|
| "MUST"       | This word means that the item is an absolute requirement of this specification.   |
| "MUST NOT"   | This phrase means that the item is an absolute prohibition of this specification.   |
| "SHOULD"     | This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.   |
| "SHOULD NOT" | This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label. |

"MAY"

This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

## 2 REFERENCES

### 2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

- [RFC 2863] IETF RFC 2863, The Interfaces Group MIB, June 2000.
- [RFC 3418] IETF RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.
- [RFC 4113] IETF RFC 4113, Management Information Base for the User Datagram Protocol (UDP), June 2005.
- [RFC 4293] IETF RFC 4293, Management Information Base for the Internet Protocol (IP), April 2006.
- [RFC 4346] IETF RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, April 2006.
- [RFC 4682] IETF RFC 4682, Multimedia Terminal Adapter (MTA) Management Information Base for PacketCable- and IP-Cablecom-Compliant Devices, December 2006.
- [PKT-PROV-1.5] PacketCable 1.5 Specification, MTA Device Provisioning, PKT-SP-PROV1.5-I03-070412, April 12, 2007, Cable Television Laboratories, Inc.
- [PKT-EUE-PROV] PacketCable 2.0 E-UE Provisioning Specification, PKT-SP-EUE-PROV-I02-080710, July 10, 2008, Cable Television Laboratories, Inc.
- [PKT-EUE-DATA] PacketCable 2.0 E-UE Data Specification, PKT-SP-EUE-DATA-I02-080710, July 10, 2008, Cable Television Laboratories, Inc.
- [PKT-UE-DATA] PacketCable 2.0 UE Data Specification, PKT-SP-UE-DATA-I01-080905, September 5, 2008, Cable Television Laboratories, Inc.
- [PKT-UE-PROV] PacketCable 2.0 UE Provisioning Specification, PKT-SP-UE-PROV-I01-080905, September 5, 2008, Cable Television Laboratories, Inc.

### 2.2 Informative References

This specification uses the following informative references.

- [PKT-TR-SMA] PacketCable Specification, PacketCable SMA Architecture Framework Technical Report, PKT-TR-SMA-ARCH-V01-081121, November 21, 2008, Cable Television Laboratories, Inc.
- [PKT-SP-SMA] PacketCable Specification, Security, Monitoring, and Automation (SMA) Specification, PKT-SP-SMA-I01-081121, November 21, 2008, Cable Television Laboratories, Inc.

### 2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone +1-303-661-9100; Fax +1-303-661-9199; <http://www.cablelabs.com>
- Internet Engineering Task Force (IETF) Secretariat, 4600 Center Oak Plaza, Sterling, VA 20166, Phone +1-571-434-3500, Fax +1-571-434-3535, <http://www.ietf.org>

### 3 TERMS AND DEFINITIONS

This specification uses the following terms:

|                             |  |
|-----------------------------|--|
| <b>Configuration</b>        | Configuration is the process of defining and propagating data to network elements for providing services.  |
| <b>Data Model</b>           | An abstract model that describes representation of data in a system.   |
| <b>Provisioning</b>         | Provisioning refers to the processes involved in the initialization of user attributes and resources to provide services to a User. This involves protocols, methodologies, and interfaces to network elements such as: Order Entry and Workflow Systems that carry out business processes, Operational Support Elements that handle network resources, Application Servers that offer services, and User Equipment that offer services, among others. |
| <b>RESTful web services</b> | Use of REST design principles via HTTP as the protocol.  |

## 4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

|               |  |
|---------------|--|
| <b>CM</b>     | Cable Modem                                      |
| <b>CPE</b>    | Customer Premise Equipment                       |
| <b>DHCP</b>   | Dynamic Host Configuration Protocol              |
| <b>DNS</b>    | Domain Name System                               |
| <b>DOCSIS</b> | Data-Over-Cable Service Interface Specifications |
| <b>eMTA</b>   | Embedded Multimedia Terminal Adapter             |
| <b>eSAFE</b>  | Embedded Service/Application Entity              |
| <b>E-SG</b>   | SMA gateway embedded with a cable modem          |
| <b>eUE</b>    | Embedded User Equipment                          |
| <b>HFC</b>    | Hybrid Fiber-Coax                                |
| <b>HTTP</b>   | Hyper Text Transport Protocol                    |
| <b>IP</b>     | Internet Protocol                                |
| <b>MIB</b>    | Management Information Base                      |
| <b>MSO</b>    | Multiple System Operator                         |
| <b>NAT</b>    | Network Address Translation                      |
| <b>REST</b>   | REpresentational State Transfer                  |
| <b>SG</b>     | SMA gateway                                      |
| <b>SMA</b>    | Security, Monitoring, Automation                 |
| <b>SNMP</b>   | Simple Network Management Protocol               |
| <b>TLS</b>    | Transport Layer Security                         |
| <b>UML</b>    | Unified Modeling Language                        |
| <b>XML</b>    | eXtensible Markup Language                       |

## 5 OVERVIEW

PacketCable SMA is a CableLabs specification effort designed to address the areas of Security, Monitoring, and Automation. PacketCable SMA considers four domains: the home domain, the access domain, the operator domain, and the central station domain. For more information about the SMA architecture, please refer to [PKT-TR-SMA]. The scope of PacketCable SMA is the interface between the SMA gateway within the Home Domain, and the network elements (e.g., event server) within the Operator Domain. The signaling, media, and QoS requirements are addressed by [PKT-SP-SMA]. The network elements and the associated requirements to provision and manage an SMA gateway are specified in this document.

### 5.1 SMA gateways and Deployment Scenarios

The SMA gateway is a Customer Premise Equipment (CPE) device that works in conjunction with network elements in the Operator Domain. There are two types of SMA gateways: those that are embedded with a DOCSIS Cable Modem (E-SG), and those that are standalone. An E-SG always connects via a DOCSIS access network. A standalone SMA gateway may be deployed in two ways: directly connected via a cable modem, or via a home router (acting as a NAT device).

The type of SMA gateway (i.e., embedded or standalone) and deployment (i.e., is it behind a home router or not) affects the provisioning and management mechanisms. For example, SMA gateways that are connected behind home routers may not be able to rely on the Operator's DHCP server for provisioning related information. This document accommodates different SMA gateways and deployment models.

### 5.2 PacketCable SMA Provisioning Mechanisms

To accommodate the different SMA gateway types, and the deployment scenarios, PacketCable SMA presents two different provisioning mechanisms.

One type of provisioning mechanism uses the PacketCable 2.0 E-UE Provisioning Framework as specified in [PKT-EUE-PROV]. The PacketCable 2.0 E-UE Provisioning Framework uses Dynamic Host Configuration Protocol (DHCP) and Simple Network Management Protocol (SNMP) to provision the SMA gateway using basic, hybrid, or secure provisioning flows. SNMP is also used for management.

The other type of provisioning mechanism uses RESTful web services for provisioning. This mechanism is useful for devices that cannot be provisioned using DHCP and SNMP, i.e., those that are behind home routers. This approach reuses the RESTful web services based SMA Signaling interface for provisioning and management.

The choice of provisioning mechanism is dependent on the type of SMA gateway (e.g., whether is it embedded with a DOCSIS Cable Modem or not) and the deployment choices (e.g., a standalone SMA gateway can still be connected directly via a Cable Modem).

### 5.3 IP Network Environments

PacketCable SMA gateways can support IPv4, IPv6, or both. SMA gateways need to make an operational choice during provisioning to pick an IP version. The way this choice is made is dependent on the provisioning mechanism.

## 5.4 PacketCable SMA Data Models

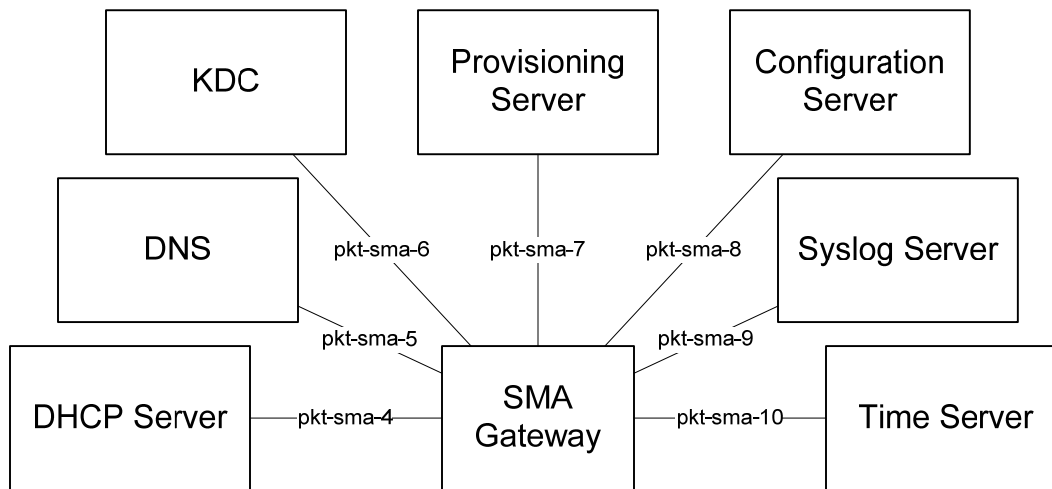
To configure Provisioning, Configuration, and Management of the SMA gateway requires the use of data models. This specification uses UML modeling to develop these data models. The data models are then manifested as SMIV2 MIBs (for DHCP- and SNMP-based provisioning) and XML Schemas (for RESTful web services based provisioning).

## 6 PACKETCABLE SMA PROVISIONING

As indicated in Section 5, this specification specifies two types of provisioning and management mechanisms: DHCP- and SNMP-based and RESTful web services approach. Section 6.1 discusses the DHCP- and SNMP-based provisioning mechanism. Section 6.2 discusses the RESTful web services based provisioning mechanism.

### 6.1 DHCP- and SNMP-based Provisioning

The interfaces required to support this mechanism are depicted in Figure 1.



**Figure 1 - DHCP- and SNMP-based Provisioning Interfaces**

An SMA gateway using the DHCP- and SNMP-based provisioning mechanism **MUST** conform to the provisioning mechanism specified in [PKT-EUE-PROV], with the following clarifications:

- All references to UE and eUE are understood to apply to a conformant SMA gateway (embedded or otherwise), noting exceptions provided within this and dependent sections.
- eDOCSIS and eSAFE requirements are applicable only to the eCM and eSG components of an E-SG.
- All eCM data model requirements apply to an eCM within an E-SG.
- The eUE data model requirements specific to PacketCable 2.0 do not apply.
- When the SMA gateway is not embedded with a DOCSIS CM, the DHCPv4 and DHCPv6 server options for the SMA gateway cannot be retrieved via the CM and are considered to be equivalent to the corresponding wild card entries, 255.255.255.255 (for IPv4) and 0xFF 0xFF 0xFF 0xFF (IPv6), respectively.
- When the SMA gateway is not embedded with a DOCSIS CM, the eSG cannot obtain the time from the CM and has to contact the Time Server on its own; in the secure provisioning flow this needs to happen prior to Kerberos messages.

### 6.1.1 MIB Requirements

The SMA gateway MUST implement all the MIB modules specified in Annex B. The SMA gateway MUST also implement the following MIB modules:

- MIB II system group as defined in [RFC 3418];
- IF MIB as specified in [RFC 2863];
- UDP MIB as specified in [RFC 4113]; and
- IP MIB as specified in [RFC 4293].

The SMA gateway's MIB II sysDescr MIB object MUST conform to the format specified in the DOCSIS specifications.

The SMA gateway MUST implement the row entry specified in Table 1 for the ifTable as specified in [RFC 2863].

**Table 1 - SMA gateway ifTable Requirements**

| ifTable ([RFC2863])       | Row Entry                   |
|---------------------------|-----------------------------|
| ifIndex                   | 1                           |
| ifDescr                   | "DOCSIS Embedded Interface" |
| ifType                    | other(1)                    |
| ifMTU                     | 0                           |
| ifSpeed                   | 0                           |
| ifPhysAddress             | SMA gateway MAC address     |
| ifAdminStatus             | up(1)                       |
| ifOperStatus              | up(1)                       |
| ifLastChange              | per [RFC 2863]              |
| ifInOctets (optional)     | (n) if implemented, else 0  |
| ifInNUCastPkts            | Deprecated                  |
| ifInDiscards              | 0                           |
| ifInErrors                | 0                           |
| ifUnknownProtos           | 0                           |
| ifOutOctets (optional)    | (n) if implemented, else 0  |
| ifOutUCastPkts (optional) | (n) if implemented, else 0  |
| ifOutNUCastPkts           | Deprecated                  |
| ifOutDiscards             | 0                           |
| ifOutErrors               | 0                           |
| ifOutQlen                 | Deprecated                  |
| ifSpecific                | Deprecated                  |

The SMA gateway MUST implement the row specified in Table 2 for the ipNetToPhysicalTable as specified in [RFC 4293].

**Table 2 - ipNetToPhysicalTable MIB Object Details**

| <b>ipNetToPhysicalTable</b>   | <b>SMA gateway</b>    |
|-------------------------------|-----------------------|
| ipNetToPhysicalIfIndex        | 1                     |
| ipNetToPhysicalPhysAddress    | eCM MAC Address       |
| ipNetToPhysicalNetAddressType | ipv4(1) or IPv6(2)    |
| ipNetToPhysicalNetAddress     | eCM IP address        |
| ipNetToPhysicalLastUpdated    | <refer to [RFC 4293]> |
| ipNetToPhysicalType           | status(4)             |
| ipNetToPhysicalState          | <refer to [RFC 4293]> |
| ipNetToPhysicalRowStatus      | 'active'              |

The SMA gateway MUST configure the usmUserTable immediately after receiving the AP REPLY from the Provisioning Server with the entry specified in Table 3.

**Table 3 - SMA gateway usmUserTable Entry**

| <b>usmUserTable ([RFC3414] [IETF STD62])</b> | <b>Row Entry</b>  |
|--|---|
| usmUserEngineID                              | The SNMP local engine ID.   |
| usmUserName                                  | SMA-GW-Prov-xx:xx:xx:xx:xx:xx<br>where xx:xx:xx:xx:xx:xx represents the SMA gateway's MAC address                   |
| usmUserSecurityName                          | SMA-GW-Prov-xx:xx:xx:xx:xx:xx<br>where xx:xx:xx:xx:xx:xx represents the SMA gateway's MAC address                   |
| usmUserCloneFrom                             | 0.0   |
| usmUserAuthProtocol                          | usmHMACACMD5AuthProtocol or usmHMACSHAAuthProtocol  |
| usmUserAuthKeyChange                         | ""  |
| usmUserOwnAuthKeyChange                      | ""  |
| usmUserPrivProtocol                          | usmDESPrivProtocol if privacy is indicated in AP REPLY<br>usmNoPrivProtocol if privacy is not indicated in AP REPLY |
| usmUserPrivKeyChange                         | ""  |
| usmUserOwnPrivKeyChange                      | ""  |
| usmUserPublic                                | ""  |
| usmUserStorageType                           | volatile  |
| usmUserStatus                                | active  |

The SMA gateway MUST configure the vacmSecurityToGroupTable with the entry specified in Table 4.

**Table 4 - SMA gateway vacmSecurityToGroupTable Entry**

| <b>vacmSecurityToGroupTable<br/>([RFC3415])</b> | <b>Row Entry</b>  |
|---|---|
| vacmSecurityModel                               | USM   |
| vacmSecurityName                                | SMA-GW-Prov-xx:xx:xx:xx:xx:xx<br>Where xx:xx:xx:xx:xx:xx represents the SMA gateway's MAC address |
| vacmGroupName                                   | PacketCableFullAccess   |
| vacmSecurityToGroupStorageType                  | volatile  |
| vacmSecurityToGroupStatus                       | Active  |

The SMA gateway MUST configure the vacmAccessTable with the entry specified in Table 5.

**Table 5 - SMA gateway vacmAccessTable Entry**

| <b>vacmAccessTable<br/>([RFC3415])</b> | <b>RowEntry</b>   |
|--|---|
| vacmGroupName                          | PacketCableFullAccess   |
| vacmAccessContextPrefix                | ""  |
| vacmAccessSecurityModel                | USM   |
| vacmAccessSecurityLevel                | authPriv or authNoPriv<br>(depending on whether privacy has been specified) |
| vacmAccessContextMatch                 | Exact   |
| vacmAccessReadViewName                 | ReadOnlyView  |
| vacmAccessWriteViewName                | FullAccessView  |
| vacmAccessNotifyViewName               | NotifyView  |
| vacmAccessStorageType                  | volatile  |
| vacmAccessStatus                       | Active  |

The following requirements are associated with Table 5.

- The SMA gateway's ReadOnlyView MUST consist of the entire MIB tree contained in the SMA gateway.
- The SMA gateway's FullAccessView MUST consist of all the PacketCable-specified MIB modules, the MIB-II system group, and the IF-MIB tree.
- The SMA gateway's FullAccessView MAY include vendor-specific MIBs, VACM, USM, and Notifications MIB.
- The SMA gateway's NotifyView MUST consist of all the PacketCable 2.0 specified MIB modules, the MIB-II system group, and the snmpTrapOID MIB tree.
- The SMA gateway's NotifyView MAY include vendor-specific MIB trees.

The SMA gateway MUST follow the SNMPv2c management requirements as specified in [PKT-PROV-1.5], "SNMPV2C MANAGEMENT REQUIREMENTS", with the following clarifications:

- The requirements applicable to the eMTA apply to the SMA gateway.
- The string (or substring) "mta" is replaced with "sg" in `snmpCommunityIndex`, `snmpCommunitySecurityName`, `snmpCommunityTransportTag`, `snmpTargetAddrName`, `snmpTargetAddrTagList`, `snmpTargetAddrParams`, `vacmSecurityName`, `vacmGroupName`, `vacmAccessReadViewName`, `vacmAccessWriteViewName`, `vacmAccessNotifyViewName`, `vacmViewTreeFamilyViewName`, `snmpTargetParamsName`, `snmpTargetParamsSecurityName`, `snmpNotifyName`, `snmpNotifyTag`, `snmpNotifyFilterProfileName` and `snmpNotifyFilterSubtree`.
- Any references to MIB modules, such as notifications within the `snmpNotifyFilterTable`, applies to the SMA gateway's MIB modules.

### 6.1.2 Pre-configuration and Persistence Requirements

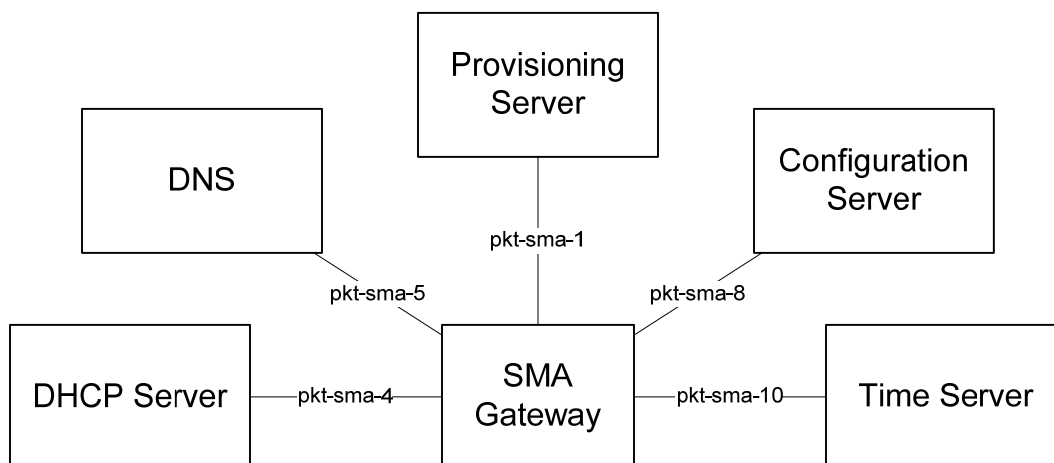
In the DHCP- and SNMP-based provisioning mechanism, the SMA gateway obtains its DHCP from the Operator. This results in configuration request and retrieval. The configuration file includes the event server FQDN (within the URI), among other data elements. This is used for registration and SMA signaling. Thus, there are no pre-configuration requirements required on the SMA gateway.

A disruption (e.g., unsynchronized time) or error (configuration file error) in the configuration process can be disruptive to the critical SMA services. To minimize such errors, the SMA gateway MUST store the URIs (i.e., `EventServerURI`, `GatewayURI` and `SignalingURI`) and the provisioning timer across reboots, unless they can successfully obtain new configuration data. The SMA gateway MAY store the remaining elements from the last successful configuration across resets (until it is overridden). If the SMA gateway fails to obtain the configuration file within the stored provisioning timer, it MUST attempt to register using the previously stored information.

## 6.2 RESTful web services based Provisioning

This section presents the normative requirements when the RESTful web services based provisioning mechanism is used. This mechanism is primarily designed for Standalone SMA gateways that can connect via a home router.

The interfaces required to support this mechanism are indicated in Figure 2.



**Figure 2 - RESTful web services Based Provisioning Interfaces**

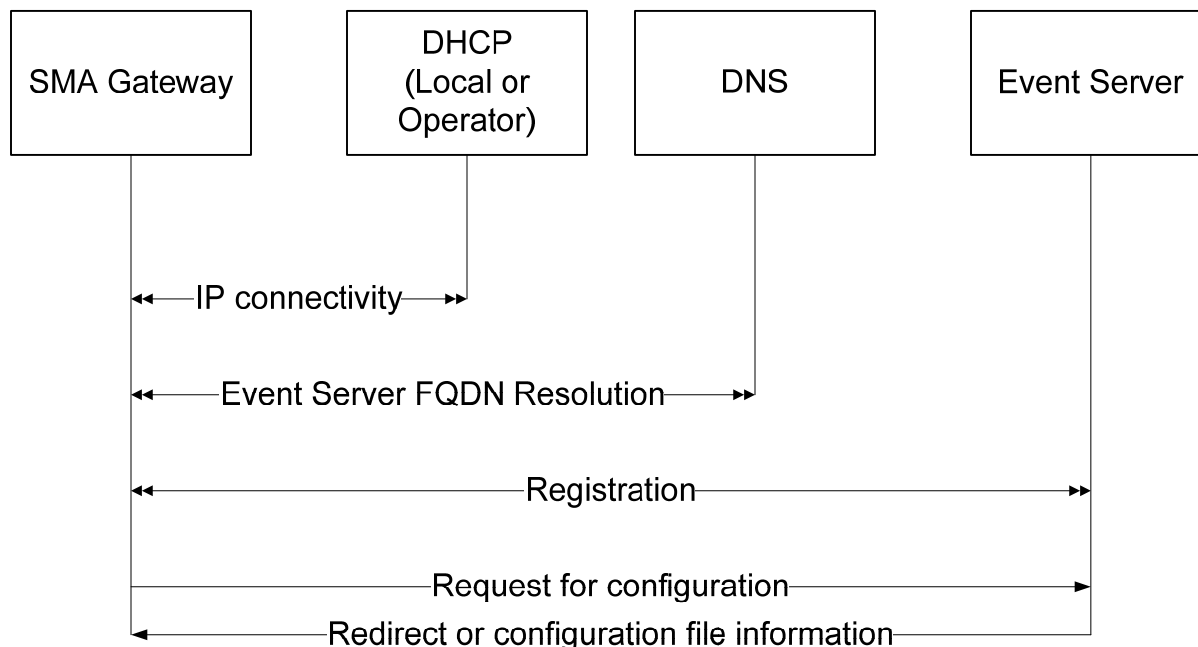
In contrast to the DHCP and SNMP based approach (Figure 1), the SNMP interface (pkt-sma-7) is replaced by pkt-sma-1, the SMA signaling interface. Refer to [PKT-SP-SMA] for more information regarding pkt-sma-1.

There are two options for provisioning:

- The Provisioning Server is co-located on the event server and hence there is only one instance of pkt-sma-1.
- The Provisioning Server is separate from the event server and there are two instances of pkt-sma-1.

### 6.2.1 Provisioning Flow

The provisioning flow for the RESTful web services mechanism is depicted in Figure 3.



**Figure 3 - RESTful web services Based Provisioning Flow**

After the SMA gateway initializes, it obtains IP connectivity (see Section 6.2.2), resolves the event server FQDN (refer to Section 6.2.3 for pre-configuration requirements), and performs registration. After successful registration, the SMA gateway **MUST** send a special event to request for configuration, using the following URI:

`http://<event server FQDN>/ev/configuration`

The vendor model and version information provided in the registration message should allow the event server to provide the correct response. This results in either a redirect request to establish connection with a provisioning server, or the configuration file. If it is redirected, the SMA gateway **MUST** establish an additional signaling path, authenticate as it would with an event server, and re-request configuration.

If the SMA gateway successfully parses all the required data elements, it **MUST** send an event with the following URI:

`http://<event server FQDN>/ev/cfg/success`

If the SMA gateway is unable to utilize the provided configuration (e.g., configuration file errors), it MUST send an event with the following URI:

`http://<event server FQDN>/ev/cfg/failure`

If the SMA gateway can parse all the critical elements, but cannot understand optional elements (e.g., proprietary additions that it does not understand), it MUST send an event with the following URI:

`http://<event server FQDN>/ev/cfg/passwithwarnings`

Within failure and warning events, the SMA Gateway MUST include a list of the data elements and the incorrect values using the SMA Response Schema specified in [PKT-SP-SMA]. Within the failure and warning events the SMA Gateway SHOULD also include comments indicating why the values were deemed erroneous (e.g., "out of range", "internal error").

After it is provisioned the SMA gateway will expect the event server to provide any configuration updates, and manage it. The event server can also direct the SMA gateway to establish a management session with a separate server. This is done by sending an event using the following URI:

`http://<SMA gateway FQDN>/ev/mgmt/session/start`

The data will contain a property-value pair conformant with the SMA Response XML Schema, within the event data body. The property will contain the FQDN of the management server. When the SMA gateway receives an event to connect to a different management server, the SMA gateway MUST adhere to this request. The SMA gateway MUST maintain this connection until it receives a termination event from the management server (or the heartbeat times out). The termination event is provided using the following URI.

`http://<SMA gateway FQDN>/ev/mgmt/session/stop`

The SMA gateway, at a minimum, MUST simultaneously support at least one signaling session, one configuration session (if different from the event server), and one management session. The SMA gateway MUST use the heartbeat specified for the signaling interface to maintain management sessions with a management server.

The SMA gateway MUST support the query of a configuration element by the event server or a management server using the following URI:

`http://<SMA gateway FQDN>/cfg/<data element>`

### **6.2.2 IP Connectivity**

Given that SMA gateways can support IPv4 and IPv6, the SMA gateway MUST adhere to the UE IP connectivity requirements specified in [PKT-UE-PROV] to select and provision in IPv4 or IPv6 operating modes.

### **6.2.3 Pre-configuration and Persistence Requirements**

In the case of the RESTful web services mechanism, unlike the DHCP- and SNMP-based provisioning mechanism, the SMA gateway needs to register prior to configuration. For this, it needs to be aware of the event server FQDN. Thus, a SMA gateway that has never registered MUST be pre-configured with valid event server HTTP URI or FQDN, its own URI or FQDN, and any associated authentication credentials. These URIs or FQDNs can be temporary, and overridden upon initial configuration.

To avoid service disruption, the SMA gateway MUST store the URIs (i.e., EventServerURI, GatewayURI and SignalingURI) and the provisioning timer across reboots, unless they can successfully obtain new configuration data. The SMA gateway MUST retain any data required for normal operation when communication with the

configuration HTTP server can not be established. The SMA gateway SHOULD also store the remaining elements from the last successful configuration, or switch to default values upon reset.

The SMA gateway MUST perform the RESTful web services based provisioning sequence for retrieving its configuration file.

#### 6.2.4 Data Model and Configuration Changes

During normal operation of the SMA gateway, an indication that a configuration change is needed by the SMA gateway is communicated via SMA events. Such events will contain the following URI:

`http://<SMA Gateway FQDN>/ev/cfg/change`

The data within the event will contain the configuration data elements and changes using the SMA Response XML Schema format (property, value pairs).

This event can also direct the SMA Gateway to reload configuration via the following URI and no data within the SMA event body:

`http://<SMA Gateway FQDN>/ev/cfg/reload`

In the case of reload request, the event will use the same format as the configuration change, and contain the Configuration HTTP URI as the property (and a null value). Upon receiving an SMA event to reload configuration, the SMA gateway MUST retrieve the configuration file using the HTTP server URI. If the FQDN within this URI is different from the event server FQDN (i.e., the data element SignalingURI), the SMA gateway MUST establish a separate TLS connection ([RFC 4346]) with the Configuration Server, using the same requirements as for a signaling interface ([PKT-SP-SMA]).

The SMA gateway MUST not disrupt SMA services during the configuration retrieval or during the application of any configuration changes.

Querying dynamic data from the SMA gateway using RESTful web services-based provisioning is accomplished using the SMA signaling between the SMA gateway and the SMA event Server. The SMA gateway MUST properly process all dynamic data queries received over the SMA communications link with the SMA event Server.

Refer to Annex A.1 for the data model describing the provisioning and management items. Annex C describes XML schemas that provides access to all data described in the data model presented in Annex A.1.

SMA gateways that support web services based provisioning MUST implement the XML schema defined in Annex C, and any additional modules specified by the Web Services based provisioning mechanism (e.g., Software Download).

#### 6.2.5 Software Download

The SMA gateway MUST support the Software Download mechanism specified for PacketCable UEs in [PKT-UE-PROV], including the corresponding data model requirements.

#### 6.2.6 Security

Authentication of HTTP requests for retrieving configuration files requires authentication and authorization. Authentication is accomplished using the same mechanisms as those used for SMA signaling, as specified in [PKT-SP-SMA].

### 6.3 Management Event Reporting

The SMA gateway MUST support the management events specified in Table 6, irrespective of the provisioning mechanism used.

**Table 6 - Additional Management Events**

| Event Name | Default Severity for Event | Default Display String                                   | PacketCable Event ID | Comments  |
|------------|----------------------------|--|----------------------|---|
| SMA-EV-1   | critical                   | "Unable to communicate with event server"                | 4000970000           | The SMA gateway was unable to communicate with the event server.      |
| SMA-EV-2   | error                      | "Unable to communicate with SMA Device < SMA device ID>" | 4000970001           | The SMA gateway was unable to communicate with an SMA Device.         |
| SMA-EV-3   | informational              | "New SMA device found"                                   | 4000970002           | The SMA gateway found a new SMA Device.                               |
| SMA-EV-4   | warning                    | "SMA Device reporting trouble <SMA device message>"      | 4000970003           | The SMA gateway was alerted to a problem with one of its SMA Devices. |
| SMA-EV-5   | critical                   | "Backup channel unavailable"                             | 4000970004           | The backup channel is no longer available.                            |
| SMA-EV-6   | informational              | "Backup channel restored"                                | 4000970005           | The backup channel has been restored.                                 |
| SMA-EV-7   | informational              | "Communicating over backup channel"                      | 4000970006           | The backup channel is currently being used for SMA communication.     |
| SMA-EV-8   | informational              | "Communicating over broadband"                           | 4000970007           | The broadband channel is currently being used for SMA communication.  |
| SMA-EV-9   | informational              | "Configuration Changed"                                  | 4000970008           | The alarm system configuration has changed.                           |
| SMA-EV-10  | informational              | "System under test"                                      | 4000970009           | The system is currently undergoing installer test.                    |
| SMA-EV-11  | informational              | "System normal"  | 4000970010           | The system is currently in its normal state.                          |
| SMA-EV-12  | informational              | "Reset initiated"  | 4000970011           | A reset operation has been initiated.                                 |
| SMA-EV-13  | informational              | "System shutdown"  | 4000970012           | A system shutdown operation has been initiated.                       |

SMA gateways implementing DHCP- and SNMP-based provisioning MUST report these events via the Management Event Mechanism. SMA gateways implementing the RESTful web services based provisioning MUST report these events as regular events and use the data model specified in C.9.

When the web services based provisioning mechanism is used, the SMA gateway MUST also support the following events specified in [PKT-EUE-PROV]: UE-SM-1, UE-SM-2, UE-SM-3, UE-SM-4 and UE-SM-5.

## **Annex A SMA Data Models**

### **A.1 Provisioning and Management Object Model Definitions**

This provisioning and management object model definitions are specified in Figure 4. The sub-sections provide a description of the specified objects. Some of the objects are borrowed from [PKT-UE-DATA]. For borrowed objects the SMA gateway **MUST** adhere to the corresponding requirements in [PKT-UE-DATA].

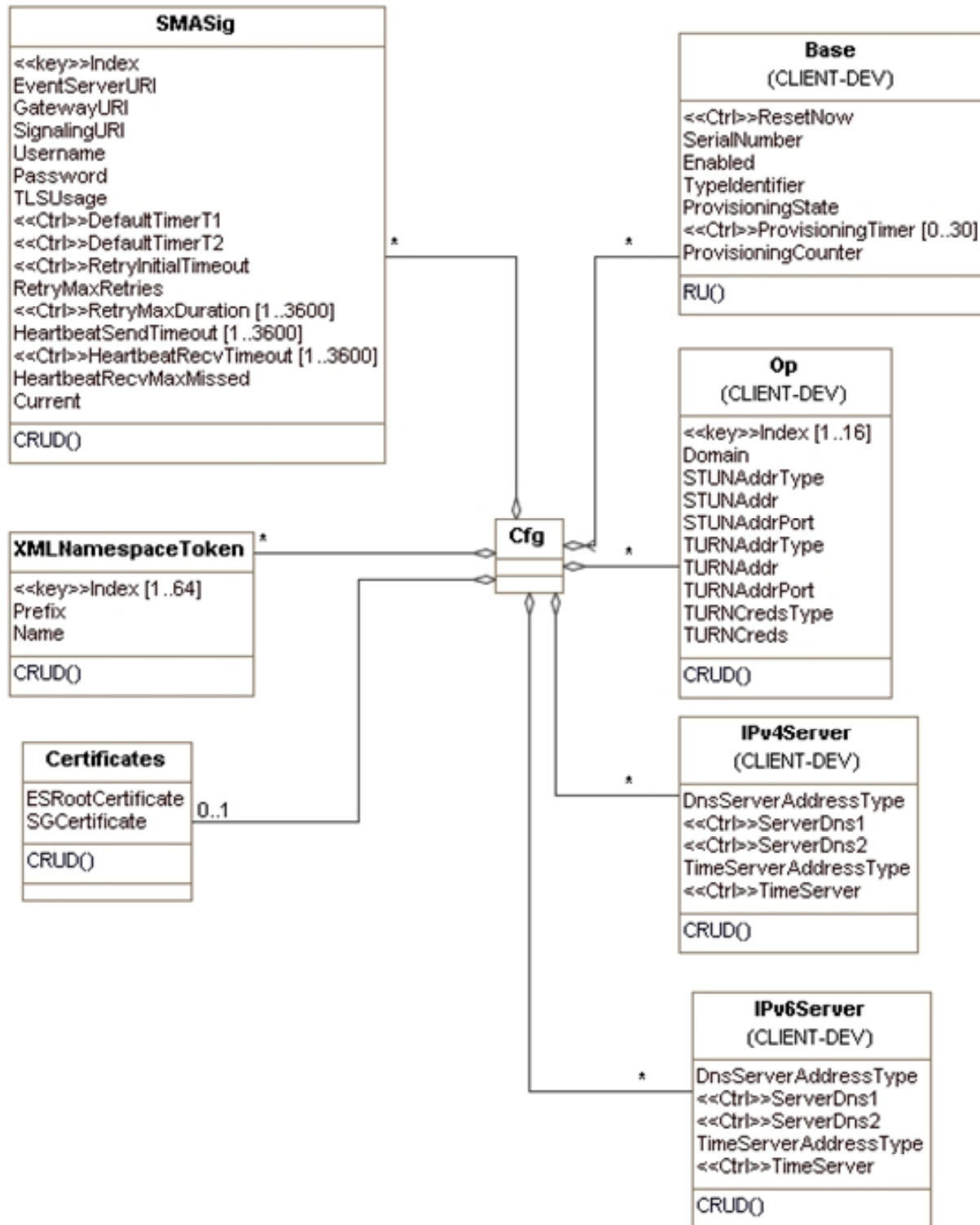


Figure 4 - SMA Provisioning and Management Object Model

### A.1.1 Base Object

This object contains the base attributes and functions for the SMA gateway. Figure 4 represents the specified elements that could be included as attributes in this object. As a note, this object is not constrained to the elements and attributes specified in this specification.

### A.1.2 SMASig Object

This object contains the signaling attributes and functions of the SMA gateway, as specified in Table 7.

**Table 7 - SMASig Object**

| Attribute Name         | Type              | Access | Type Constraints | Units        | Default |
|------------------------|-------------------|--------|------------------|--------------|---------|
| EventServerURI         | AdminString       | CRUD   |                  |              | ""      |
| GatewayURI             | AdminString       | CRUD   |                  |              | ""      |
| SignalingURI           | AdminString       | CRUD   |                  |              | ""      |
| Username               | AdminString       | CRUD   |                  |              | ""      |
| Password               | AdminString       | CRUD   |                  |              | ""      |
| TLSUsage               | TruthValue        | CRUD   |                  |              | 'true'  |
| TimerT1                | unsignedInt       | CRUD   |                  | milliseconds | 1000    |
| TimerT2                | unsignedInt       | CRUD   |                  | milliseconds | 5000    |
| TimerT1ForInstructions | X.509 Certificate | CRUD   |                  | milliseconds | 1000    |
| TimerT2ForInstructions | X.509 Certificate | CRUD   |                  | milliseconds | 5000    |
| RetryInitialTimeout    | unsignedInt       | CRUD   |                  | milliseconds | 200     |
| RetryMaxRetries        | unsignedInt       | CRUD   |                  |              | 3       |
| RetryMaxDuration       | unsignedInt       | CRUD   | 1..3600          | seconds      | 60      |
| HeartbeatSendTimeout   | unsignedInt       | CRUD   | 1..3600          | seconds      | 60      |
| HeartbeatRecvTimeout   | unsignedInt       | CRUD   | 1..3600          | seconds      | 60      |
| HeartbeatRecvMaxMissed | unsignedInt       | CRUD   |                  |              | 5       |

- EventServerURI

This attribute contains the event server URI to be used for communication with the event server, i.e., the actual string that is used in the SMA messages. If the data element 'SignalingURI' is not specified, then this data element is also used for establishing transport connections.

- GatewayURI

This data element contains the SMA gateway URI, i.e., how the SMA gateway will be addressed. The SMA gateway will not respond to requests with URIs that are not a superset of this URI (it will send an error response). The SMA gateway can generate this URI via a pre-configured or acquired FQDN (e.g., http://<SMA gateway FQDN>). However, if the device configuration provides a URI, it overrides any device generated URI.

- SignalingURI

This attribute contains the URI used to setup a transport connection to the SMA event Server. If provided, the host and port portion of this URI is used to establish a transport connection to the event server; even though the Request URI of all SMA requests is set to the contents of the EventServerURI. If this attribute is set to a zero-length string, then the SMA gateway will use the host and port portion of 'EventServerURI' for transport channel establishment (as indicated in the definition of GatewayURI).

- Username

This attribute contains the username used for authentication challenges, whenever applicable.

- Password

This attribute contains the password used for authentication challenges, whenever applicable.

- TLSUsage

This attribute determines whether or not TLS should be used when establishing communication with the event server.

- DefaultTimerT1

This data element contains the default value of timer T1 for received requests, which do not provide a value for T1.

- DefaultTimerT2

This data element contains the default value of timer T2 for received requests, which do not provide a value for T2.

- TimerT1ForInstructions

This data element contains the value of T1 that should be used by the SMA gateway when sending requests to the event server.

- TimerT2ForInstructions

This data element contains the value of T2 that should be used by the SMA gateway when sending requests to the event server.

- RetryInitialTimeout

This data element contains the time duration before sending the initial retry request, following a timeout while communicating with an event server. Future retry attempts are throttled based on exponential retry algorithms.

- RetryMaxRetries

This data element contains the number of times the SMA gateway will try to connect to the SMA event Server before it exhausts the retry attempts.

- RetryMaxDuration

This data element contains the maximum time duration that the SMA gateway should wait before it exhausts the retry attempts. The SMA gateway will adhere to 'RetryMaxDuration' over 'RetryMaxRetries'.

- HeartbeatSendTimeout

This data element contains the time duration that the SMA gateway can wait, without any communication with the event server, prior to sending a Heartbeat message. If there has been no SMA communication with the event server for the time duration indicated by 'HeartbeatSendTimeout', the SMA gateway will send a heartbeat message to the SMA event Server.

- HeartbeatRecvTimeout

This data element contains the time duration that the SMA gateway should tolerate for a response to the heartbeat, from the event server. If the SMA gateway receives no response to a heartbeat for the time duration indicated by HeartbeatRecvTimeout, the SMA gateway will resend the heartbeat message.

- HeartbeatRecvMaxMissed

This data element contains the maximum number of missed heartbeat events (i.e., heartbeat requests that do not receive a response) that can occur before the existing connection to the SMA event Server will be torn down by the SMA gateway. The SMA gateway will then proceed to reinitiate communication via the registration process.

### A.1.3 XMLNamespaceToken

This object contains a mapping of xml namespace prefixes to xml namespace names. It allows messages from the SMA event Server to the SMA gateway to omit the namespace directives in the embedded XML document header, thus reducing packet size.

**Table 8 - XMLNamespaceToken Object**

| Attribute Name | Type        | Access | Type Constraints | Units | Default |
|----------------|-------------|--------|------------------|-------|---------|
| Index          | unsignedInt | Key    |                  |       |         |
| Prefix         | AdminString | CRUD   |                  |       |         |
| Name           | AdminString | CRUD   |                  |       |         |

- Index

A unique value to identify the XML Namespace entry.

- Prefix

A value for the prefix representing the XML namespace.

- Name

A value for the name of the XML namespace.

### A.1.4 Certificates

This objects the certificates that are stored on an SMA gateway and can be used during management.

**Table 9 - XMLNamespaceToken Object**

| Attribute Name    | Type              | Access | Type Constraints | Units | Default |
|-------------------|-------------------|--------|------------------|-------|---------|
| ESRootCertificate | X.509 Certificate | CRUD   |                  |       |         |
| SGCertificate     | X.509 Certificate | CRUD   |                  |       |         |

- ESRootCertificate

This data element contains the event server's Root Certificate that the SMA Gateway will validate during TLS establishment. This is pre-configured and cannot be modified.

- pktcSMAPMSGCertificate

This data element contains the SMA gateway's Certificate that the SMA Gateway will use during TLS establishment. This is pre-configured and cannot be modified.

#### **A.1.5 Op Object**

See [PKT-UE-DATA].

#### **A.1.6 IPv4Cfg Object**

See [PKT-UE-DATA].

#### **A.1.7 IPv6Cfg Object**

See [PKT-UE-DATA].

## Annex B DHCP- and SNMP-based Provisioning Data Models

This section specifies the DHCP- and SNMP-based Provisioning Data Model requirements based on the object model described in Annex A.1.

### B.1 SNMP MIB Objects from existing MIB Modules Requirements

The SMA gateway MUST implement the MIB Objects in Table 10 when supporting the DHCP- and SNMP-based provisioning mode.

**Table 10 - SMA gateway SNMP MIB Objects Requirements (Existing MIB Modules)**

| Object Model | MIB Object                      | Reference      |
|--------------|---------------------------------|----------------|
| Base         | pktMtaDevResetNow               | [RFC 4682]     |
|              | pktcMtaDevSerialNumber          | [RFC 4682]     |
|              | pktcMtaDevEnabled               | [RFC 4682]     |
|              | pktcMtaDevTypeIdentifier        | [RFC 4682]     |
|              | pktcMtaDevProvisioningState     | [RFC 4682]     |
|              | pktcMtaDevProvisioningTimer     | [RFC 4682]     |
|              | pktcMtaDevProvisioningCounter   | [RFC 4682]     |
| IPv4Server   | pktcMtaDevDhcpServerAddressType | [RFC 4682]     |
|              | pktcMtaDevServerDhcp1           | [RFC 4682]     |
|              | pktcMtaDevServerDhcp2           | [RFC 4682]     |
|              | pktcMtaDevDnsServerAddressType  | [RFC 4682]     |
| DNSv4        | pktcMtaDevServerDns1            | [RFC 4682]     |
|              | pktcMtaDevServerDns2            | [RFC 4682]     |
|              | pktcMtaDevTimeServerAddressType | [RFC 4682]     |
|              | pktcMtaDevTimeServer            | [RFC 4682]     |
| IPv6Server   | pktcEUEDhcpv6ServerId1          | [PKT-EUE-DATA] |
|              | pktcEUEDhcpv6ServerId2          | [PKT-EUE-DATA] |
|              | pktcEUEDhcpv6ServerAddressType  | [PKT-EUE-DATA] |
|              | pktcEUEDhcpv6ServerAddress      | [PKT-EUE-DATA] |
| DNSv6        | pktcEUEDnsv6ServerAddressType   | [PKT-EUE-DATA] |
|              | pktcEUEDnsv6ServerAddress1      | [PKT-EUE-DATA] |
|              | pktcEUEDnsv6ServerAddress2      | [PKT-EUE-DATA] |
| Op           | pktcEUEDevOpSTUNAddrType        | [PKT-EUE-DATA] |
|              | pktcEUEDevOpSTUNAddr            | [PKT-EUE-DATA] |
|              | pktcEUEDevOpSTUNAddrPort        | [PKT-EUE-DATA] |
|              | pktcEUEDevOpTURNAddrType        | [PKT-EUE-DATA] |

| Object Model | MIB Object                  | Reference      |
|--------------|-----------------------------|----------------|
|              | pktcEUEDevOpTURNAddr        | [PKT-EUE-DATA] |
|              | pktcEUEDevOpTURNAddrPort    | [PKT-EUE-DATA] |
|              | pktcEUEDevOpTURNCreditsType | [PKT-EUE-DATA] |
|              | pktcEUEDevOpTURNCredits     | [PKT-EUE-DATA] |

## B.2 SNMP MIB Module

The SMA gateway MUST implement the MIB specified in this section if it supports the DHCP- and SNMP-based provisioning mechanism.

```

CL-PKTC-SMA-PROV-MGMT-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32
        FROM SNMPv2-SMI
    OBJECT-GROUP,
    MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB

    TruthValue,
    RowStatus
        FROM SNMPv2-TC

    DocsX509ASN1DEREncodedCertificate
        FROM DOCS-IETF-BPI2-MIB -- RFC 4131
    pktcSMAMibs
        FROM CLAB-DEF-MIB;

pktcSMAPMMib MODULE-IDENTITY
    LAST-UPDATED "200811210000Z" -- November 21, 2008
    ORGANIZATION "Cable Television Laboratories, Inc."
    CONTACT-INFO
        "Broadband Network Services
        Cable Television Laboratories, Inc.
        858 Coal Creek Circle,
        Louisville, CO 80027, USA
        Phone: +1 303-661-9100
        Email: mibs@cablelabs.com

        Acknowledgements:
        Eduardo Cardona, CableLabs - Primary author
        Jerry Mahler, Motorola
        and members of the PacketCable SMA Focus Team."
    DESCRIPTION
        "This MIB module contains the SMA Provisioning data elements."
    REVISION "200811210000Z" -- November 21, 2008
    DESCRIPTION
        "Initial version, published as part of the CableLabs
        PacketCable SMA Provisioning Specification, [PKT-SP-SMA-PROV].
        Copyright 2008, Cable Television Laboratories, Inc.
        All rights reserved."
 ::= { pktcSMAMibs 1 }

```

```

-- Object Definitions
pktcSMAPMNotifications OBJECT IDENTIFIER ::= { pktcSMAPMMib 0 }
pktcSMAPMObjects       OBJECT IDENTIFIER ::= { pktcSMAPMMib 1 }

pktcSMAPMSigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF PktcSMAPMSigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        " This object contains the signaling attributes and functions of the SMA
          gateway."
    ::= {pktcSMAPMObjects 1 }

pktcSMAPMSigEntry  OBJECT-TYPE
    SYNTAX      PktcSMAPMSigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Conceptual row of pktcSMAPMSigTable. The row containing the event
          Server that the SMA gateway was last connected to is persisted across
          resets."
    INDEX {
        pktcSMAPMSigIndex
    }

    ::= {pktcSMAPMSigTable 1 }

PktcSMAPMSigEntry ::= SEQUENCE {
    pktcSMAPMSigIndex
        Unsigned32,
    pktcSMAPMSigEventServerURI
        SnmpAdminString,
    pktcSMAPMSigGatewayURI
        SnmpAdminString,
    pktcSMAPMSigSignalingURI
        SnmpAdminString,
    pktcSMAPMSigUsername
        SnmpAdminString,
    pktcSMAPMSigPassword
        SnmpAdminString,
    pktcSMAPMSigTLSUsage
        TruthValue,
    pktcSMAPMSigDefaultTimerT1
        Unsigned32,
    pktcSMAPMSigDefaultTimerT2
        Unsigned32,
    pktcSMAPMSigTimerT1ForSGInstructions
        Unsigned32,
    pktcSMAPMSigTimerT2ForSGInstructions
        Unsigned32,
    pktcSMAPMSigRetryInitialTimeout
        Unsigned32,
    pktcSMAPMSigRetryMaxRetries
        Unsigned32,
    pktcSMAPMSigRetryMaxDuration
        Unsigned32,
    pktcSMAPMSigHeartbeatSendTimeout
        Unsigned32,
    pktcSMAPMSigHeartbeatRecvTimeout
        Unsigned32,
    pktcSMAPMSigHeartbeatRecvMaxMissed
        Unsigned32,
    pktcSMAPMSigCurrent
        TruthValue,
    pktcSMAPMSigRowStatus
        RowStatus
    }

```

```

pktcSMAPMSigIndex      OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        " This key indicates a unique instance of the SMA gateway protocol
        configuration. An index value of 1 is always required to be configured, when
        provided via a configuration server.
        If other indices are configured, they represent a backup set.

        The SMA gateway will always attempt to establish SMA connectivity
        using the parameters within the first index (i.e., 1). In the case of errors
        it will use any configured backup indices."
    ::= {pktcSMAPMSigEntry 1 }

pktcSMAPMSigEventServerURI  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This attribute contains the event server URI to be used for communication
        with the event server, i.e., the actual string that is used in the SMA
        messages. If the data element 'SignalingURI' is not specified, then this
        data element is also used for establishing transport connections."
    ::= {pktcSMAPMSigEntry 2 }

pktcSMAPMSigGatewayURI    OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This data element contains the SMA gateway URI, i.e., how the SMA gateway
        will be addressed. The SMA gateway will not respond to requests with URIs
        that are not a superset of this URI (it will send an error response). The SMA
        gateway can generate this URI via a pre-configured or acquired FQDN (e.g.,
        http://<SMA gateway FQDN>). However, if the device configuration provides a
        URI, it overrides any device generated URI."
    ::= {pktcSMAPMSigEntry 3 }

pktcSMAPMSigSignalingURI  OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This attribute contains the URI used to setup a transport connection to the
        SMA event Server. If provided, the host and port portion of this URI is used
        to establish a transport connection to the event server; even though the
        Request URI of all SMA requests is set to the contents of the EventServerURI.
        If this attribute is the zero-length string, then the SMA gateway will use
        the host and port portion of 'EventServerURI' for transport channel
        establishment, as indicated in the definition of GatewayURI.."
    ::= {pktcSMAPMSigEntry 4 }

pktcSMAPMSigUsername      OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This attribute contains the username used for authentication challenges,
        whenever applicable."
    ::= {pktcSMAPMSigEntry 5 }

pktcSMAPMSigPassword      OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current

```

```

DESCRIPTION
    " This attribute contains the password used for authentication challenges,
      whenever applicable."
 ::= {pktcSMAPMSigEntry 6 }

pktcSMAPMSigTLUsage OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This attribute determines whether or not TLS should be used when
      establishing communication with the event server."
 ::= {pktcSMAPMSigEntry 7 }

pktcSMAPMSigDefaultTimerT1 OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This data element contains the default value of timer T1 for received
      requests, which do not provide a value for T1"
DEFVAL {1000}
 ::= {pktcSMAPMSigEntry 9 }

pktcSMAPMSigDefaultTimerT2 OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This data element contains the default value of timer T2 for received
      requests, which do not provide a value for T2."
DEFVAL {5000}
 ::= {pktcSMAPMSigEntry 10 }

pktcSMAPMSigTimerT1ForSGInstructions OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This data element contains the value of T1 that should be used by the SMA
      gateway when sending requests to the event server."
DEFVAL {1000}
 ::= {pktcSMAPMSigEntry 11 }

pktcSMAPMSigTimerT2ForSGInstructions OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This data element contains the value of T2 that should be used by the SMA
      gateway when sending requests to the event server."
DEFVAL {5000}
 ::= {pktcSMAPMSigEntry 12 }

pktcSMAPMSigRetryInitialTimeout OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This data element contains the time duration before sending the initial
      retry request, following a timeout while communicating with an event server.
      Future retry attempts are throttled based on exponential retry algorithms."

```

```

DEFVAL {1000}
 ::= {pktcSMAPMSigEntry 13 }

pktcSMAPMSigRetryMaxRetries      OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 " This data element contains the number of times the SMA gateway will try to
   connect to the SMA event Server before it exhausts the retry attempts."
DEFVAL {3}
 ::= {pktcSMAPMSigEntry 14 }

pktcSMAPMSigRetryMaxDuration     OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 " This data element contains the maximum time duration that the SMA gateway
   should wait before it exhausts the retry attempts. The SMA gateway will
   adhere to 'RetryMaxDuration' over 'RetryMaxRetries'."
DEFVAL {60}
 ::= {pktcSMAPMSigEntry 15 }

pktcSMAPMSigHeartbeatSendTimeout OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 " This data element contains the time duration that the SMA gateway can wait,
   without any communication with the event server, prior to sending a
   Heartbeat message. If there has been no SMA communication with the event
   server for the time duration indicated by 'HeartbeatSendTimeout', the SMA
   gateway will send a heartbeat message to the SMA event Server."
DEFVAL {60}
 ::= {pktcSMAPMSigEntry 16 }

pktcSMAPMSigHeartbeatRecvTimeout OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 " This data element contains the time duration that the SMA gateway should
   tolerate for a response to the heartbeat, from the event server. If the SMA
   gateway receives no response to a heartbeat for the time duration indicated by
   this data element, the SMA gateway will resend the heartbeat message."
DEFVAL {60}
 ::= {pktcSMAPMSigEntry 17 }

pktcSMAPMSigHeartbeatRecvMaxMissed OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 " This data element contains the maximum number of missed heartbeat events
   (i.e., heartbeat requests that do not receive a response) that can occur
   before the existing connection to the SMA event Server will be torn down by
   the SMA gateway. The SMA gateway will then proceed to reinitiate
   communication via the registration process."
DEFVAL {5}
 ::= {pktcSMAPMSigEntry 18 }

pktcSMAPMSigCurrent              OBJECT-TYPE
SYNTAX      TruthValue

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
  " This indicates if the current row contains the event server that the SMA
  gateway is currently connected to."
 ::= {pktcSMAPMSigEntry 19 }

pktcSMAPMSigRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
  " The conceptual row creation status of this entry."
 ::= {pktcSMAPMSigEntry 20 }

pktcSMAPMXMLNamespacesTable OBJECT-TYPE
SYNTAX SEQUENCE OF PktcSMAPMXMLNamespacesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "This table contains a mapping of XML Namespace prefixes to XML namespaces.
  This table allows SMA messages to omit the namespace directives in the
  embedded XML, thus reducing packet size."
 ::= {pktcSMAPMObjects 3 }

pktcSMAPMXMLNamespacesEntry OBJECT-TYPE
SYNTAX PktcSMAPMXMLNamespacesEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "The Conceptual row of pktcSMAPMXMLNamespacesTable."
INDEX {
  pktcSMAPMXMLNamespacesIndex
}
 ::= {pktcSMAPMXMLNamespacesTable 1 }

PktcSMAPMXMLNamespacesEntry ::= SEQUENCE {
  pktcSMAPMXMLNamespacesIndex
    Unsigned32,
  pktcSMAPMXMLNamespacesPrefix
    SnmpAdminString,
  pktcSMAPMXMLNamespacesName
    SnmpAdminString,
  pktcSMAPMXMLNamespacesRowStatus
    RowStatus
}

pktcSMAPMXMLNamespacesIndex OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  "A unique value to identify the XML Namespace entries."
 ::= {pktcSMAPMXMLNamespacesEntry 1 }

pktcSMAPMXMLNamespacesPrefix OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-create
STATUS current
DESCRIPTION
  "This contains a XML namespace prefix."
 ::= {pktcSMAPMXMLNamespacesEntry 2 }

pktcSMAPMXMLNamespacesName OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-create
STATUS current

```

```

DESCRIPTION
    "A value for the name of the XML namespace."
 ::= {pktcSMAPMXMLNamespacesEntry 3 }

pktcSMAPMXMLNamespacesRowStatus OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " The conceptual row creation status of this entry."
 ::= {pktcSMAPMXMLNamespacesEntry 4 }

pktcSMAPMESRootCertificate OBJECT-TYPE
SYNTAX      DocsX509ASN1DEREncodedCertificate
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This data element contains the event server's Root Certificate that
      the SMA Gateway will validate during TLS establishment. This is
      pre-configured and cannot be modified."
 ::= {pktcSMAPMObjects 4 }

pktcSMAPMSGCertificate OBJECT-TYPE
SYNTAX      DocsX509ASN1DEREncodedCertificate
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This data element contains the SMA gateway's Certificate that
      the SMA Gateway will use during TLS establishment. This is
      pre-configured and cannot be modified."
 ::= {pktcSMAPMObjects 5 }

-- Conformance Definitions
pktcSMAPMMibConformance OBJECT IDENTIFIER ::= { pktcSMAPMMib 2 }
pktcSMAPMMibCompliances OBJECT IDENTIFIER ::= { pktcSMAPMMibConformance 1 }
pktcSMAPMMibGroups OBJECT IDENTIFIER ::= { pktcSMAPMMibConformance 2 }

pktcSMAPMCompliance MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "The compliance statement for the PacketCable SMA MIB module."
MODULE -- this MODULE
MANDATORY-GROUPS {
    pktcSMAPMGroup
}
 ::= { pktcSMAPMMibCompliances 1 }

pktcSMAPMGroup OBJECT-GROUP
OBJECTS {
    pktcSMAPMSigEventServerURI,
    pktcSMAPMSigGatewayURI,
    pktcSMAPMSigSignalingURI,
    pktcSMAPMSigUsername,
    pktcSMAPMSigPassword,
    pktcSMAPMSigTLSUsage,
    pktcSMAPMSigDefaultTimerT1,
    pktcSMAPMSigDefaultTimerT2,
    pktcSMAPMSigTimerT1ForSGInstructions,
    pktcSMAPMSigTimerT2ForSGInstructions,
    pktcSMAPMSigRetryInitialTimeout,
    pktcSMAPMSigRetryMaxRetries,
    pktcSMAPMSigRetryMaxDuration,
    pktcSMAPMSigHeartbeatSendTimeout,
    pktcSMAPMSigHeartbeatRecvTimeout,
    pktcSMAPMSigHeartbeatRecvMaxMissed,
    pktcSMAPMSigCurrent,

```

```
        pktcSMAPMSigRowStatus,
        pktcSMAPMXMLNamespacesPrefix,
        pktcSMAPMXMLNamespacesName,
        pktcSMAPMXMLNamespacesRowStatus,
        pktcSMAPMESRootCertificate,
        pktcSMAPMSGCertificate
    }
    STATUS          current
    DESCRIPTION
        "Group of objects implemented in the PacketCable SMA MIB module."
    ::= { pktcSMAPMMibGroups 1 }

END
```

## Annex C RESTful Web Services Based Provisioning Data Models

This section presents the Web Service based provisioning XML Schemas, based on the object model described in Annex A.1.

### C.1 SMACfgEnvelope Object XML Schema Definition

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:cfgenv"
xmlns:s1="urn:cablelabs:packetcable:sma:xsd:v1:sig"
xmlns:s2="urn:cablelabs:packetcable:sma:xsd:v1:ns"
xmlns:s3="urn:cablelabs:packetcable:sma:xsd:v1:base"
xmlns:s4="urn:cablelabs:packetcable:sma:xsd:v1:op"
targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:cfgenv"
elementFormDefault="qualified">
  <xsd:import namespace="urn:cablelabs:packetcable:sma:xsd:v1:sig"
schemaLocation="http://www.cablelabs.com/packetcable/sma/xsd/v1/SMASig.xsd"/>
  <xsd:import namespace="urn:cablelabs:packetcable:sma:xsd:v1:ns"
schemaLocation="http://www.cablelabs.com/packetcable/sma/xsd/v1/SMAXMLNS.xsd"/>
  <xsd:import namespace="urn:cablelabs:packetcable:sma:xsd:v1:base"
schemaLocation="http://www.cablelabs.com/packetcable/sma/xsd/v1/SMABase.xsd"/>
  <xsd:import namespace="urn:cablelabs:packetcable:sma:xsd:v1:op"
schemaLocation="http://www.cablelabs.com/packetcable/sma/xsd/v1/SMAOp.xsd"/>
  <xsd:complexType name="SMASigCfg">
    <xsd:sequence>
      <xsd:element name="SMASig" type="s1:SMASig" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="NSMaps" type="s2:NSMaps" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="Base" type="s3:Base"/>
      <xsd:element name="Op" type="s4:Op"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="Config" type="this:SMASigCfg"/>
</xsd:schema>
```

### C.2 SMASig Object XML Schema Definition

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:sig"
elementFormDefault="qualified"
xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:sig">
  <xsd:element name="Index" type="xsd:unsignedInt"/>
  <xsd:element name="EventServerURI" type="xsd:string"/>
  <xsd:element name="GatewayURI" type="xsd:string"/>
  <xsd:element name="SignalingURI" type="xsd:string"/>
  <xsd:element name="Username" type="xsd:string"/>
  <xsd:element name="Password" type="xsd:string"/>
  <xsd:element name="TLSUsage" type="xsd:boolean"/>
  <xsd:element name="DefaultTimerT1" type="xsd:unsignedInt" default="1000"/>
  <xsd:element name="DefaultTimerT2" type="xsd:unsignedInt" default="5000"/>
  <xsd:element name="TimerT1ForSGInstructions" type="xsd:unsignedInt"
default="1000"/>
  <xsd:element name="TimerT2ForSGInstructions" type="xsd:unsignedInt"
default="5000"/>
  <xsd:element name="RetryTimeout" type="xsd:unsignedInt" default="200"/>
  <xsd:element name="RetryMaxRetries" type="xsd:unsignedInt" default="3"/>
  <xsd:element name="RetryMaxDuration" type="xsd:unsignedInt" default="60"/>
  <xsd:element name="HeartbeatSendTimeout" type="xsd:unsignedInt" default="60"/>
  <xsd:element name="HeartbeatRecvTimeout" type="xsd:unsignedInt" default="60"/>
  <xsd:element name="HeartbeatRecvMaxMissed" type="xsd:unsignedInt" default="5"/>
  <xsd:element name="Current" type="xsd:boolean"/>
  <xsd:complexType name="SMASig">
```

```

<xsd:sequence>
  <xsd:element ref="this:Index" />
  <xsd:element ref="this:EventServerURI" />
  <xsd:element ref="this:GatewayURI" />
  <xsd:element ref="this:SignalingURI" />
  <xsd:element ref="this:Username" />
  <xsd:element ref="this:Password" />
  <xsd:element ref="this:TLSUsage" />
  <xsd:element ref="this:DefaultTimerT1" />
  <xsd:element ref="this:DefaultTimerT2" />
  <xsd:element ref="this:TimerT1ForSGInstructions" />
  <xsd:element ref="this:TimerT2ForSGInstructions" />
  <xsd:element ref="this:RetryTimeout" />
  <xsd:element ref="this:RetryMaxRetries" />
  <xsd:element ref="this:RetryMaxDuration" />
  <xsd:element ref="this:HeartbeatSendTimeout" />
  <xsd:element ref="this:HeartbeatRecvTimeout" />
  <xsd:element ref="this:HeartbeatRecvMaxMissed" />
  <xsd:element ref="this:Current" minOccurs="0" maxOccurs="0" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

### C.3 XMLNamespaces Schema Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:ns"
  targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:ns"
  elementFormDefault="qualified">
  <xsd:element name="Index" type="xsd:unsignedInt" />
  <xsd:element name="Prefix" type="xsd:string" />
  <xsd:element name="NameSpace" type="xsd:string" />
  <xsd:complexType name="NSMaps">
    <xsd:sequence>
      <xsd:element ref="this:Index" />
      <xsd:element ref="this:Prefix" />
      <xsd:element ref="this:NameSpace" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

### C.4 Certificates Schema Definition

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:certs"
  elementFormDefault="qualified"
  xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:certs">
  <xsd:element name="ESRootCertificate" type="xsd:hexBinary" />
  <xsd:element name="SGCertificate" type="xsd:hexBinary" />
  <xsd:complexType name="Certs">
    <xsd:sequence>
      <xsd:element ref="this:ESRootCertificate" minOccurs="0" maxOccurs="0" />
      <xsd:element ref="this:SGCertificate" minOccurs="0" maxOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

### C.5 Base Object Schema Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:base"
  elementFormDefault="qualified"
  xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:base">

```

```

<xsd:element name="ResetNow" type="xsd:boolean"/>
<xsd:element name="SerialNumber" type="xsd:string"/>
<xsd:element name="Enabled" type="xsd:boolean"/>
<xsd:element name="TypeIdentifier" type="xsd:string"/>
<xsd:element name="ProvisioningState" type="xsd:string"/>
<xsd:element name="ProvisioningTimer" type="xsd:unsignedInt"/>
<xsd:element name="ProvisioningCounter" type="xsd:unsignedInt"/>

<xsd:complexType name="Base">
  <xsd:sequence>
    <xsd:element ref="this:ResetNow"/>
    <xsd:element ref="this:SerialNumber" minOccurs="0" maxOccurs="0"/>
    <xsd:element ref="this:Enabled" minOccurs="0" maxOccurs="0"/>
    <xsd:element ref="this:TypeIdentifier" minOccurs="0" maxOccurs="0"/>
    <xsd:element ref="this:ProvisioningState" minOccurs="0" maxOccurs="0"/>
    <xsd:element ref="this:ProvisioningTimer"/>
    <xsd:element ref="this:ProvisioningCounter" minOccurs="0" maxOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## C.6 Op Object Schema Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:op"
  elementFormDefault="qualified"
  xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:op">

  <xsd:element name="Index" type="xsd:unsignedInt"/>
  <xsd:element name="Domain" type="xsd:string"/>
  <xsd:element name="STUNAddr" type="xsd:string"/>
  <xsd:element name="STUNAddrPort" type="xsd:unsignedInt"/>
  <xsd:element name="TURNAddr" type="xsd:string"/>
  <xsd:element name="TURNAddrPort" type="xsd:unsignedInt"/>
  <xsd:element name="TURNCreditsType" type="xsd:string"/>
  <xsd:element name="TURNCredits" type="xsd:hexBinary"/>

  <xsd:complexType name="Op">
    <xsd:sequence>
      <xsd:element ref="this:Index"/>
      <xsd:element ref="this:Domain"/>
      <xsd:element ref="this:STUNAddr"/>
      <xsd:element ref="this:STUNAddrPort"/>
      <xsd:element ref="this:TURNAddr"/>
      <xsd:element ref="this:TURNAddrPort"/>
      <xsd:element ref="this:TURNCreditsType"/>
      <xsd:element ref="this:TURNCredits"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

## C.7 IPv4Elements Object Schema Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:ipv6cfg"
  elementFormDefault="qualified"
  xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:ipv6cfg">

  <xsd:element name="v6DNSServer1" type="xsd:string"/>
  <xsd:element name="v6DNSServer2" type="xsd:string"/>
  <xsd:element name="v6TimeServer" type="xsd:string"/>
  <xsd:complexType name="IPv6Server">
    <xsd:sequence>
      <xsd:element ref="this:v6DNSServer1" minOccurs="0" maxOccurs="0"/>

```

```

        <xsd:element ref="this:v6DNSServer2" minOccurs="0" maxOccurs="0"/>
        <xsd:element ref="this:v6TimeServer" minOccurs="0" maxOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## C.8 IPv6Elements Object Schema Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:ipv4cfg"
            elementFormDefault="qualified"
            xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:ipv4cfg">

    <xsd:element name="v4DNSServer1" type="xsd:string"/>
    <xsd:element name="v4DNSServer2" type="xsd:string"/>
    <xsd:element name="v4TimeServer" type="xsd:string"/>
    <xsd:complexType name="IPv6Server">
        <xsd:sequence>
            <xsd:element ref="this:v4DNSServer1" minOccurs="0" maxOccurs="0"/>
            <xsd:element ref="this:v4DNSServer2" minOccurs="0" maxOccurs="0"/>
            <xsd:element ref="this:v4TimeServer" minOccurs="0" maxOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

```

## C.9 Management Event Mechanism (MEM) XML Schema

The Management Event Mechanism XML Schema for use within RESTful web services based provisioning is provided in this section.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:this="urn:cablelabs:packetcable:sma:xsd:v1:mem"
            targetNamespace="urn:cablelabs:packetcable:sma:xsd:v1:mem"
            elementFormDefault="qualified">
    <xsd:element name="mem">
        <xsd:complexType>
            <xsd:attribute name="id" type="xsd:string"/>
            <xsd:attribute name="description" type="xsd:string"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

## Appendix I Acknowledgements

CableLabs wishes to thank the PacketCable SMA vendor focus team participants for various contributions and efforts that led to the development of this specification.

Steve Hughey, Arris Interactive

Chuck Duffy, Cisco Systems

Jerry Mahler, Motorola

Special appreciation is extended to Jerry Mahler as the primary editor of this document.

*Eduardo Cardona, Sumanth Channabasappa, and the PacketCable Architecture Team.*

---