International Telecommunication Union

ITU-T

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

J.167

(11/2005)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

IPCablecom

Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems

ITU-T Recommendation J.167

ITU-T Recommendation J.167

Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems

Summary

This Recommendation describes the IPCablecom MTA device initialization and provisioning process. It is limited to the provisioning of an IPCablecom embedded-MTA device by a single provisioning and network management provider.

Source

ITU-T Recommendation J.167 was approved on 29 November 2005 by ITU-T Study Group 9 (2005-2008) under the ITU-T Recommendation A.8 procedure.

CONTENTS

Page

Page

ITU-T Recommendation J.167

Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems

1 Scope

This Recommendation describes the IPCablecom MTA device initialization and provisioning process. It is limited to the provisioning of an IPCablecom embedded-MTA device by a single provisioning and network management provider.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation J.83 (1997), *Digital multi-programme systems for television, sound and data services for cable distribution*.

– ITU-T Recommendation J.112 Annex B (2004), *Data-over-cable service interface specifications: Radio-frequency interface specification*.

– ITU-T Recommendation J.162 (2005), *Network call signalling protocol for the delivery of time-critical services over cable television networks using cable modems*.

– ITU-T Recommendation J.166 (2005)[*], *IPCablecom Management Information Base (MIB) framework*.

– ITU-T Recommendation J.170 (2005), *IPCablecom security specification*.

– IETF RFC 2131 (1997), *Dynamic Host Configuration Protocol*.

– IETF RFC 2132 (1997), *DHCP Options and BOOTP Vendor Extensions*.

– IETF RFC 2475 (1998), *An Architecture for Differentiated Services*.

– IETF RFC 2616 (1999), *Hypertext Transfer Protocol – HTTP/1.1*.

– IETF RFC 2833 (2000), *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*.

– IETF RFC 2863 (2000), *The Interfaces Group MIB*.

– IETF RFC 3396 (2002), *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*.

– IETF RFC 3410 (2002), *Introduction and Applicability Statements for Internet Standard Management Framework*.

– IETF RFC 3411 (2002), *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*.

[*] Replaces ITU-T Recs J.166 (2001), J.168 (2001), J.169 (2001) and J.176 (2002).

– IETF RFC 3412 (2002), *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP).*

– IETF RFC 3413 (2002), *Simple Network Management Protocol (SNMP) Applications.*

– IETF RFC 3414 (2002), *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3).*

– IETF RFC 3415 (2002), *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP).*

– IETF RFC 3495 (2003), *Dynamic Host Configuration Protocol (DHCP) Option for CableLabs Client Configuration*.

– IETF RFC 3584 (2003), *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework.*

– IETF RFC 3594 (2003), *PacketCable Security Ticket Control Sub-Option for the DHCP CableLabs Client Configuration (CCC) Option*.

– IETF RFC 3617 (2003), *Uniform Resource Identifier (URI) Scheme and Applicability Statement for the Trivial File Transfer Protocol (TFTP).*

3 Terms and definitions

This Recommendation defines the following terms:

3.1 cable modem: A cable modem is a layer-two termination device that terminates the customer end of the J.112 connection.

3.2 IPCablecom: An ITU-T project that includes an architecture and a series of Recommendations that enable the delivery of real-time services (such as telephony) over the cable television networks using cable modems.

4 Abbreviations and conventions

4.1 Abbreviations

This Recommendation uses the following abbreviations:

CM	Cable Modem
CMS	Call Management Server
CPE	Customer Premises Equipment
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IPSec	Internet Protocol Security
MAC	Media Access Control
MTA	Media Terminal Adapter
PSTN	Public Switched Telephone Network
SNMP	Simple Network Management Protocol

TFTP	Trivial File Transfer Protocol
TGS	Ticket Granting Server

4.2 Conventions

It is understood that implementing this Recommendation is optional. If this Recommendation is implemented, the key words "MUST" and "SHALL" as well as "REQUIRED" are to be interpreted as indicating a mandatory aspect of this Recommendation. The key words indicating a certain level of significance of particular requirements that are used throughout this Recommendation are summarized below:

"MUST"	This word or the adjective "REQUIRED" means that the item is an absolute requirement of this Recommendation.
"MUST NOT"	This phrase means that the item is an absolute prohibition of this Recommendation.
"SHOULD"	This word or the adjective "RECOMMENDED" 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 behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
"MAY"	This word or the adjective "OPTIONAL" 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.

5 Introduction

5.1 Service goals

Cable operators are interested in deploying high-speed data communications systems on cable television networks. The intended service enables voice communications, video and data services based on bidirectional transfer of Internet Protocol (IP) traffic, between the cable system headend and customer locations, over an all-coaxial or hybrid-fibre/coax (HFC) cable network, defined by ITU-T Recs J.83 and J.112. This is shown in simplified form in Figure 1.

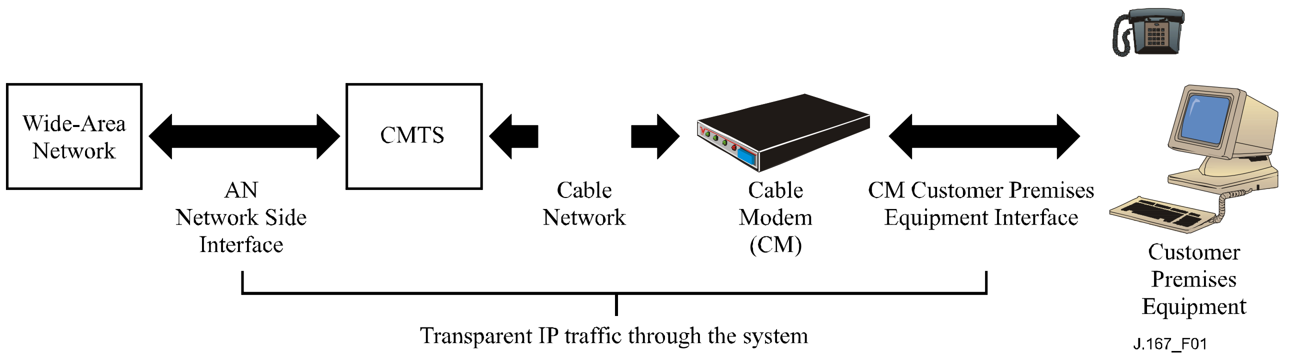

Wide-Area Network
CMTS
AN Network Side Interface
Cable Network
Cable Modem (CM)
CM Customer Premises Equipment Interface
Customer Premises Equipment
Transparent IP traffic through the system
J.167_F01

Figure 1/J.167 – Transparent IP traffic through the data-over-cable system

The transmission path over the cable system is realized at the headend by a cable modem termination system (CMTS), and at each customer location by a cable modem (CM). The intent is for operators to transfer IP traffic transparently between these interfaces.

5.2 Specification goals

Requirements relevant to device provisioning are:

- A single physical device (e.g., embedded-MTA) will be completely provisioned and managed by a single business entity. This provider may establish business relationships with additional providers for services such as data, voice communications and other services.
- An embedded-MTA is an IPCablecom MTA combined with a CM. Both CM and IPCablecom device provisioning steps MUST be performed for this embedded-MTA device to be provisioned. The embedded-MTA MUST have two IP addresses: an IP address for the CM component, and a different IP address for the MTA component. The embedded-MTA MUST have two MAC addresses: one MAC address for the CM component and a different MAC address for the MTA-component. Furthermore, the MTA MUST operate irrespective of whether it is in the same or different subnet as the CM.
- IPCablecom requires a unique FQDN for the MTA-component in the embedded-MTA. This FQDN MUST be included in the DHCP OFFER and DHCP ACK messages to the MTA-component. IPCablecom makes no additional FQDN requirements on the CM component in the embedded-MTA beyond those required by ITU-T Rec. J.112. Mapping of the FQDN to IP address MUST be configured in the network DNS server and be available to the rest of the network.
- IPCablecom embedded-MTA provisioning MUST use DHCP Option 12 and Option 15 to deliver the MTA FQDN to the E-MTA.
- IPCablecom embedded-MTA provisioning MUST support two separate configuration files: ITU-T Rec. J.112-specified configuration file for the CM component, and an IPCablecom-specified configuration file for the MTA component.
- The embedded-MTA is outside the IPCablecom network trust boundary as defined in the IPCablecom architecture J.160.
- IPCablecom MUST support DOCSIS 1.1 (ITU-T Rec. J.112) or DOCSIS 2.0 (ITU-T Rec. J.122) software download as defined in ITU-T Rec. J.112. The DOCSIS 1.1 or DOCSIS 2.0 software download process supports the downloading of a single file to the cable modem or embedded MTA. A single DOCSIS 1.1 or DOCSIS 2.0 software download MUST be used to upgrade code for both DOCSIS and IPCablecom software functions.
- IPCablecom MUST support use of SNMPv2c coexistence for network management operations for devices provisioned under the Basic Flow or the Hybrid Flow and SNMPv3/v2 coexistence for network management operations when the device is provisioned under the Secure Flow.
- IPCablecom embedded-MTA provisioning minimizes the impact to ITU-T Rec. J.112/J.122 devices (CM and CMTS) in the network.
- Standard server solutions (TFTP, SNMP, DNS, etc.) are preferable. It is understood that an application layer may be required on top of these protocols to coordinate IPCablecom embedded-MTA provisioning.
- Where appropriate, the J.112/J.122 management protocols are supported (SNMP, DHCP, TFTP).

5.3 IPCablecom reference architecture

Figure 2 shows the reference architecture for the IPCablecom Network. Refer to the IPCablecom architecture J.160 for more detailed information on this reference architecture.

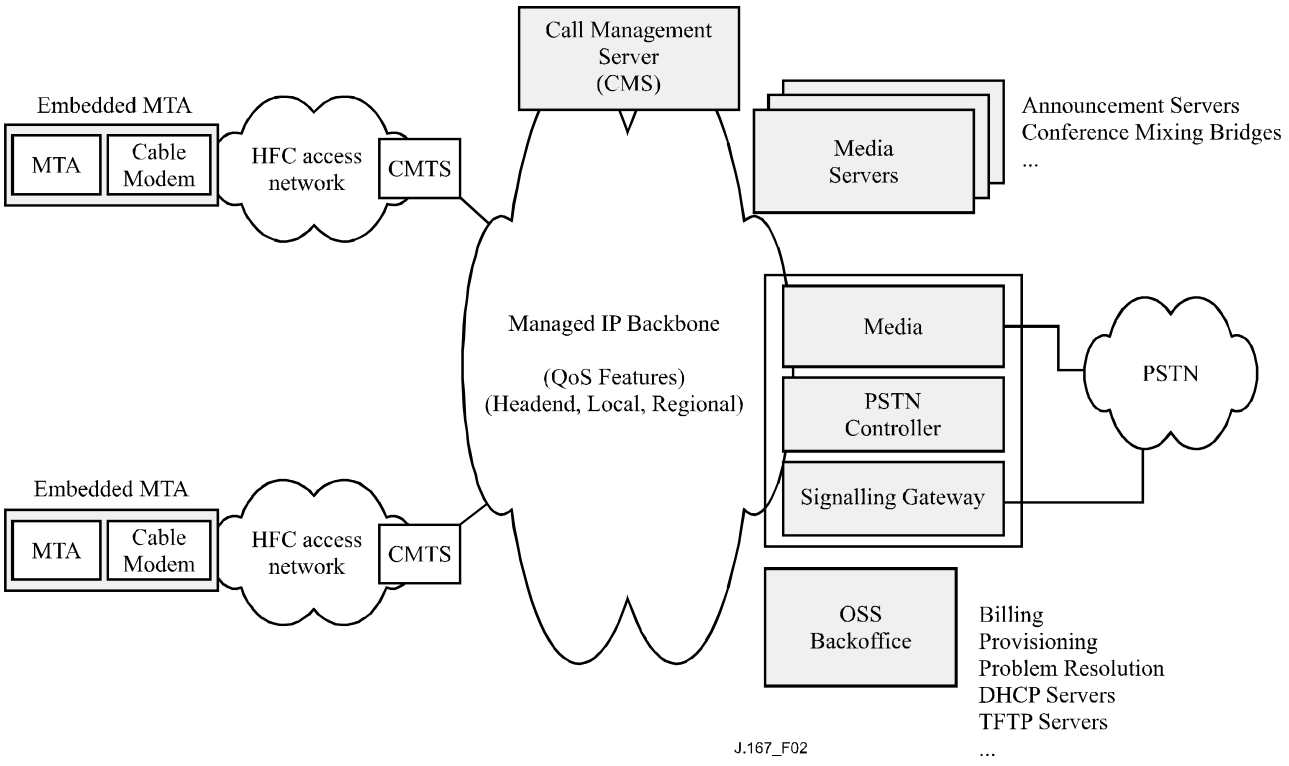


Figure 2/J.167 – IPCablecom network component reference model (partial)

5.4 Components and interfaces

The basic IPCablecom embedded-MTA provisioning reference architecture is shown in Figure 3. This figure represents the components and interfaces discussed in this Recommendation.

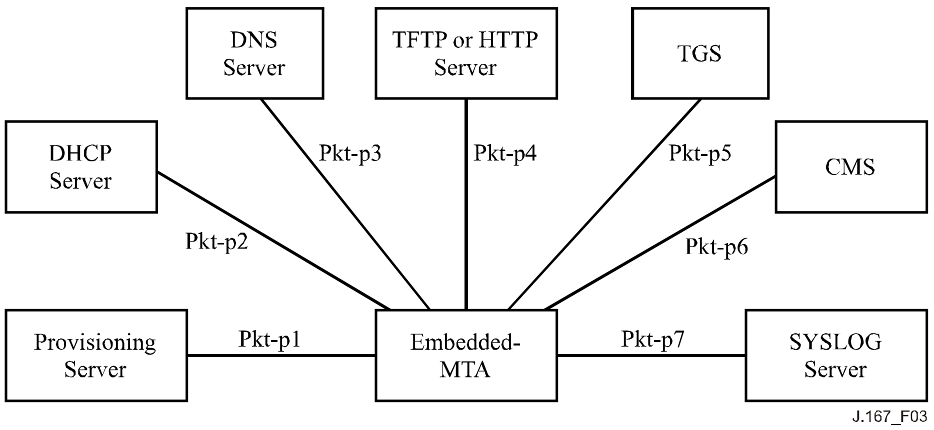


Figure 3/J.167 – IPCablecom provisioning interfaces

5.4.1 MTA

The MTA MUST conform to the following requirements during the provisioning sequence.

5.4.1.1 MTA security requirements

The MTA MUST conform to the following security requirements during the Secure Flow provisioning sequence:

- The MTA device MIB is structured to represent the assignment of MTA endpoints to a CMS. For more information on the security association between an MTA and a CMS, refer to ITU-T Rec. J.170.
- CMS Kerberos Principal Name is not explicitly configured in the MTA endpoints. The MTA MUST be able to determine the CMS Kerberos Principal Name based on the CMS FQDN, as specified in ITU-T Rec. J.170.
- For each unique pair of CMS Kerberos principal Name/Kerberos Realm assigned to an endpoint, the MTA MUST obtain a single Kerberos ticket per ITU-T Rec. J.170.
- If the MTA already has a valid Kerberos ticket for that CMS, the MTA MUST NOT request an additional Kerberos ticket for that CMS. (Unless the expiration time of the current Kerberos ticket ≤ current time + PKINIT Grace Period, in which case the MTA MUST obtain a fresh ticket for the same CMS.)
- In the case that a CMS FQDN maps to multiple IP addresses, the MTA MUST initially establish a pair of IPSec Security Associations with one of the IP addresses returned by the DNS server. The MTA MAY also initially establish IPSec Security Associations with the additional CMS IP addresses. Please see ITU-T Rec. J.170 for more information.
- If the MTA already has a pair of active Security Associations (inbound and outbound) with a particular CMS IP address, the MTA MUST NOT attempt to establish additional Security Associations with the same IP address.

During the provisioning sequence, there are no specific security requirements for the Basic Flow or the Hybrid Flow.

5.4.1.2 MTA SNMP requirements

The MTA MUST conform to the following SNMPv3 requirements during the Secure Flow provisioning sequence:

- MTA SNMPv3 security is separate and distinct from CM SNMPv3 security. USM security information (authentication and privacy keys, and other USM table entries) is set up separately.
- SNMPv3 initialization MUST be completed prior to the provisioning enrolment inform.
- In Secure Flow, the MTA MUST support SNMPv3 and SNMPv2 based device management as defined in RFC 3414 and RFC 3584.

The MTA MUST conform to the following SNMPv2c requirements during the Hybrid Flow or Basic Flow provisioning sequence:

- SNMPv2c initialization MUST be completed immediately after the DHCP phase.

SNMPv2c based device management is defined in RFC 3584.

5.4.2 Provisioning Server

The Provisioning Server is made up of the following components:

- Provisioning Application – The Provisioning Application is responsible for coordinating the embedded-MTA provisioning process. This application has an associated SNMP Entity.
- Provisioning SNMP Entity – The provisioning SNMP entity MUST include a trap/inform handler for provisioning enrolment and the provisioning status traps/informs as well as an SNMP engine for retrieving device capabilities and setting the Configuration filename and

access method. Refer to the IPCablecom MTA MIB J.166 for a description of the MIB accessible MTA attributes.

The interface between the Provisioning Application and the associated SNMP Entity is not specified in IPCablecom and is left to vendor implementation. The interface between the Provisioning Server and the TFTP Server is not specified in IPCablecom and is left to vendor implementation.

5.4.3 MTA to telephony Syslog server

The IPCablecom MTAs MUST implement Management Event Mechanism as per ITU-T Rec. J.172 and incorporate the MEM-MIB as defined in ITU-T Rec. J.166 which includes the support for Syslog server.

The IPCablecom MTAs MUST also implement all the IPCablecom Provisioning Management Events described in Annex A/J.172.

5.4.4 MTA to DHCP server

This interface identifies specific requirements in the DHCP server and the client for IP assignment during the MTA initialization process:

- Both the DHCP server and the embedded-MTA MUST support DHCP option codes 6, 7, 12, 15, 43, 60 and DHCP option code 122 (defined in RFC 2132). Option codes 12 (Host Name) and 15 (Domain Name) MUST form a Fully Qualified Domain Name and MUST be resolvable by the DNS server.
- The DHCP server MUST accept and support broadcast and unicast messages per RFC 3396 from the MTA DHCP client.
- The DHCP server MUST include the MTA's assigned FQDN in the DHCP OFFER and DHCP ACK messages to the MTA-component of the embedded-MTA. Refer to RFC 2131 for details describing the DHCP OFFER message.

5.4.5 MTA to Provisioning Application

This interface identifies specific requirements for the Provisioning Application to satisfy MTA initialization and registration. The Provisioning Application requirements are:

- The MTA MUST generate a correlation ID – an arbitrary value that will be exchanged as part of the device capability data to the Provisioning Application. This value is used as an identifier to correlate related events in the MTA provisioning sequence.
- The Provisioning Application MUST provide the MTA with its MTA configuration data file. The MTA configuration file is specific to the MTA-component of the embedded-MTA and separate from the CM-component's configuration data file.
- The configuration data file format is TLV binary data suitable for transport over the specified TFTP or HTTP access method.
- The Provisioning Application MUST have the capability to configure the MTA with different data and voice service providers.
- The Provisioning Application MUST use only SNMPv3 to provision devices in the Secure Flow. The support of the Basic and Hybrid Flows is optional for the Provisioning Application. If the Basic and Hybrid Flows are supported, the Provisioning Application MUST use only SNMPv2c to provision devices in the Hybrid or Basic Flow.
- The Provisioning Application MUST provide SNMPv3 and SNMPv2 for device management.
- The Provisioning Application MUST support online incremental device/subscriber provisioning using SNMP.

- MTA MUST Specify all of its Capabilities in DHCP Option-60 in accordance with clause 10.
- Provisioning Application MUST NOT assume any Capabilities, which do not have default values. In case capabilities supplied by the MTA are not consistent in format and/or in number and/or in values, the Provisioning Application MUST use the other means to identify the MTA's capabilities (e.g., SNMPv3 if possible).

5.4.6 MTA to CMS

Signalling is the main interface between the MTA and the CMS. Refer to the IPCablecom signalling ITU-T Rec. J.162 for a detailed description of the interface.

- The CMS MUST accept signalling and bearer channel requests from an MTA that has an active security association.
- The CMS MUST NOT accept signalling and bearer channel requests from an MTA that does not have an active security association unless provisioned to do so with information corresponding to the "pktcMtaDevCmslpsecCtrl" MIB Object.

5.4.7 MTA to Security Server (KDC)

The interface between the MTA and the Key Distribution Centre (KDC) MUST conform to the ITU-T J.170 IPCablecom security specification.

AP-REQ/REP exchange backoff and retry mechanism of the Kerberized SNMPv3 key negotiation defined in ITU-T Rec. J.170 is controlled by the values delivered by DHCP Option 122 sub-option 5 (see 8.1.4).

AS-REQ/REP exchange backoff and retry mechanism of the Kerberized SNMPv3 key negotiation defined in J.170 is controlled by the values delivered by DHCP Option 122 sub-option 4 (see 8.1.3) or by the default values of the corresponding MIB objects in the Realm Table if sub-option 4 is not present in the DHCP Option 122.

5.4.8 MTA and configuration data file access

This Recommendation allows for more than one access method to download the configuration data file to the MTA:

- The MTA MUST support the TFTP access method for downloading the MTA configuration data file.
- The MTA MAY support HTTP access method for downloading the MTA configuration data file.
- The Provisioning Server MUST provide the MTA with the URL-encoded TFTP/HTTP server address and configuration filename via a SNMPv3 SET for the Secure Flow. The Provisioning Server MUST provide the MTA with the URL-encoded TFTP/HTTP server address via an SNMPv2c SET if it supports the Hybrid Flow provisioning mode. The Basic Flow does not require an SNMP SET to get the configuration file; the Provisioning Server MUST provide the MTA with the TFTP/HTTP server address in the DHCP "file" and "siaddr" fields if it supports the Basic Flow provisioning mode. For additional information, refer to 7.3.

5.4.9 DOCSIS extensions for MTA provisioning

This Recommendation requires that the following additions to DOCSIS flows for MTA auto-provisioning be supported:

- A new DHCP option code 122 and the associated procedures MUST be implemented in DOCSIS.

6 Provisioning overview

Provisioning is a subset of configuration management control. The provisioning aspects include, but are not limited to, defining configurable data attributes, managing defined attribute values, resource initialization and registration, managing resource software, and configuration data reporting. The resource (also referred to as the managed resource) always refers to the MTA device. Further, the associated subscriber is also referred to as a managed resource.

6.1 Device provisioning

Device provisioning is the process by which an embedded-MTA device is configured to support voice communications service.

Device provisioning involves the MTA obtaining its IP configuration required for basic network connectivity, announcing itself to the network and downloading of its configuration data from its provisioning server.

When the device is provisioned using the "Secure Flow", the MTA device MUST be able to verify the authenticity of the configuration file it downloads from the server. The "Secure Flow" generated configuration file is "signed" and may be "sealed". Refer to ITU-T Rec. J.170 for further information.

Refer to 5.4.1 for provisioning rules related to security associations.

When the device is provisioned using the Basic Flow or the Hybrid Flow, a content integrity verification check MUST be conducted on the configuration file by the MTA. For details, refer to 9.1.

6.2 Endpoint provisioning

Endpoint provisioning is when a provisioned MTA authenticates itself to the CMS and establishes a security association with that server. This allows subsequent call signalling to be protected under the established security association.

The MTA MUST follow the requirements defined in the IPCablecom Security Specification (ITU-T Rec. J.170) for NCS Kerberized Key Management, independently of the provisioning flow (Secure, Hybrid or Basic Flow) the MTA was provisioned with.

6.3 Provisioning state transitions

Figure 4 represents logical device states and the possible transitions across these logical states. This representation is for illustrative purposes only and is not meant to imply a specific implementation. The following MTA state transitions do not specify the number of retry attempts or retry time out values:

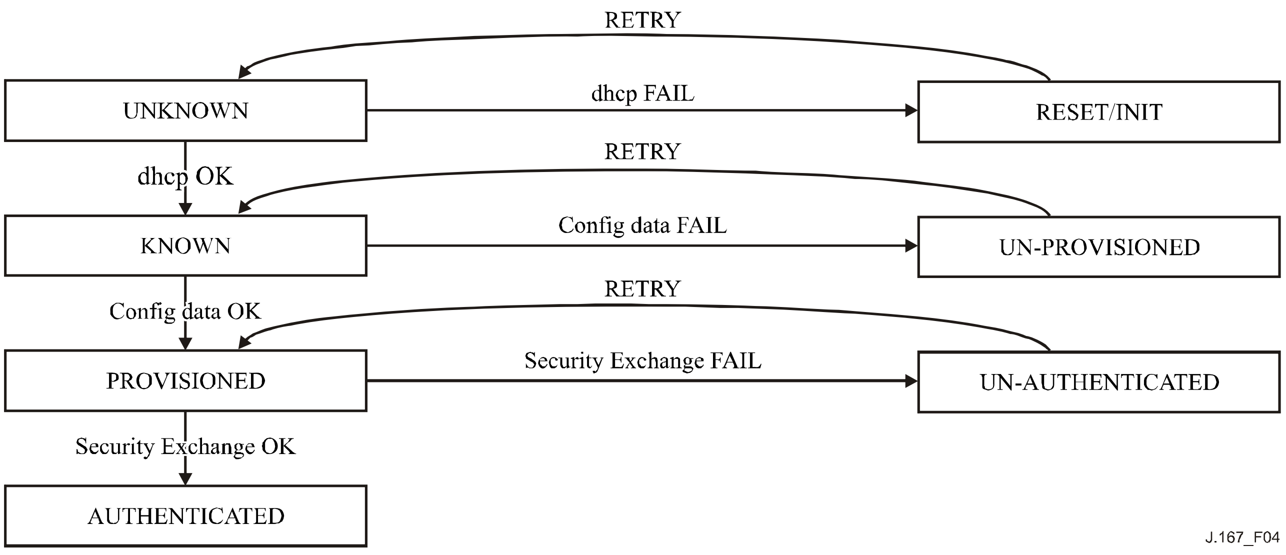


Figure 4/J.167 – Device states and state transitions for secure flow provisioning

6.4 Basic and hybrid flow provisioning state transitions

Figure 5 represents logical device states and the possible transitions across these logical states. This representation is for illustrative purposes only and is not meant to imply a specific implementation. The following MTA state transitions do not specify the number of retry attempts or retry time out values.

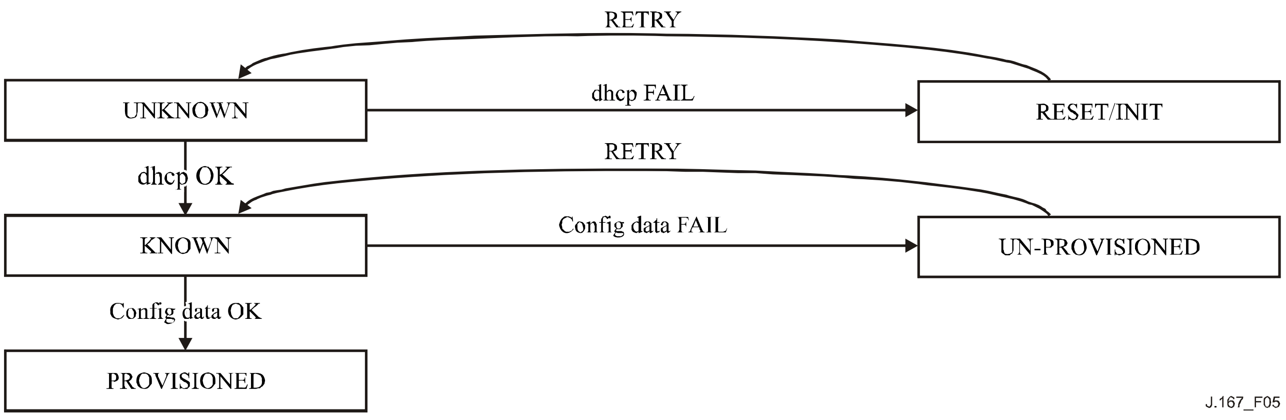


Figure 5/J.167 – Device states and state transitions for basic and hybrid flow provisioning

7 Provisioning flows

An IPCablecom MTA is provisioned via one of three provisioning flows:

- The Secure Flow supports Kerberos mutual authentication between the MTA and the provisioning system, as well as Kerberized SNMPv3 messaging. The Secure Flow MUST be supported by IPCablecom MTAs and the Provisioning Applications.
- The Basic Flows are a simplified DOCSIS-like provisioning flows with no Kerberos or SNMPv3 security and no SNMP enrolment via SNMP INFORM. The Basic Flows SHOULD be supported by IPCablecom MTAs and Provisioning Applications.
- The Hybrid Flows are essentially the Secure flow with the Kerberos message exchanges removed, and SNMPv2c substituted for SNMPv3. The Hybrid Flows SHOULD be supported by IPCablecom MTAs and Provisioning Applications.

Any mention of SNMP in this Recommendation without a specific reference to the SNMP protocol version must be interpreted as follows:

- For the Secure Flow, the MTA MUST support 'SNMPv3 only' for Provisioning and SNMPv3/v2c coexistence for Network Management and/or Monitoring operations. The SNMPv3/v2c coexistence MUST be supported and is configured using the values of TLV-38, or TLV-11 and TLV-64 in the MTA configuration file.
- For the Hybrid or Basic Flows, the MTA MUST support SNMPv2c for Provisioning, Network Management and/or Monitoring operations. The level of SNMPv2c access MUST be supported according to the values of TLV-38 or TLV-11 and TLV-64 in the MTA configuration file.

An MTA can also be configured with additional SNMPv2c targets via its configuration file by using TLV-38 or TLV-11 and TLV-64.

An MTA is commanded to execute a specific flow via the contents of DHCP option 122 sub-option 6, as described in 8.1.5. Each of these flows begin with a common set of flow steps.

7.1 Backoff, retries and time-outs

Backoff mechanisms help the network to throttle device registration during a typical or mass registration condition when the MTA client requests are not serviced within the protocol-specified time-out values. The details of provisioning behaviour under mass-registration is beyond the scope of IPCablecom; however, this clause provides the following recommendations and requirements:

- Throttling of registrations MAY be based on DOCSIS CM registration.
- The MTA MUST follow DHCP (RFC 2131) and HTTP specification time-out and retry mechanisms. It is recommended to follow IETF RFC 3413 for SNMP timeout and retry mechanisms.
- The MTA MUST use an adaptive time-out for TFTP as specified in DOCSIS (ITU-T Rec. J.112/J.122).
- The MTA MUST follow backoff and retry recommendations that are defined in the ITU-T Rec. J.170 security specification for the security message flows.
- All Provisioning Flows (Secure, Basic and Hybrid) are described in 7.2, 7.3 and 7.4.
  – Provisioning timer MUST start immediately after the receipt of DHCP ACK and MUST end with the completion of TFTP/HTTP configuration file response.
  – In case the provisioning timer expires before the completion of TFTP/HTTP configuration file response, the MTA MUST return to MTA1.
  – MTA MUST NOT wait until the Provisioning Timer expires before acting on each Provisioning step's failure condition. For example, in Secure Flow, if step MTA19 fails, the MTA must not wait until the Provisioning Timer expires but must return to MTA1 immediately when the failure condition is discovered.
- In the Secure Provisioning Flow – If a failure occurs in any of the steps related to the PROV_SNMP_ENTITY (MTA13, MTA14, MTA15, MTA19) before the MTA obtains the Device configuration file – and the MTA resolved multiple IP addresses for the PROV_SNMP_ENTITY (FQDN received in Option 122 Sub-option 3), then it MUST retry the steps with all the resolved IP addresses before returning to MTA1, unless directed otherwise by ITU-T Rec. J.170. However, it is to be noted that once the MTA selects a resolved IP address for use in MTA13, it MUST use the same IP address in steps MTA15 and MTA25.
- In the Hybrid Provisioning Flow – If a failure occurs in any of the steps related to the PROV_SNMP_ENTITY (H-MTA15, H-MTA19) before the MTA obtains the Device configuration file – and the MTA resolved multiple IP addresses for the

PROV_SNMP_ENTITY (FQDN received in Option 122 sub-option 3), then it MUST retry the steps with all the resolved IP addresses before returning to MTA1. However, it is to be noted that once the MTA selects a resolved IP address for use in H-MTA15, it MUST use the same IP address in H-MTA25.

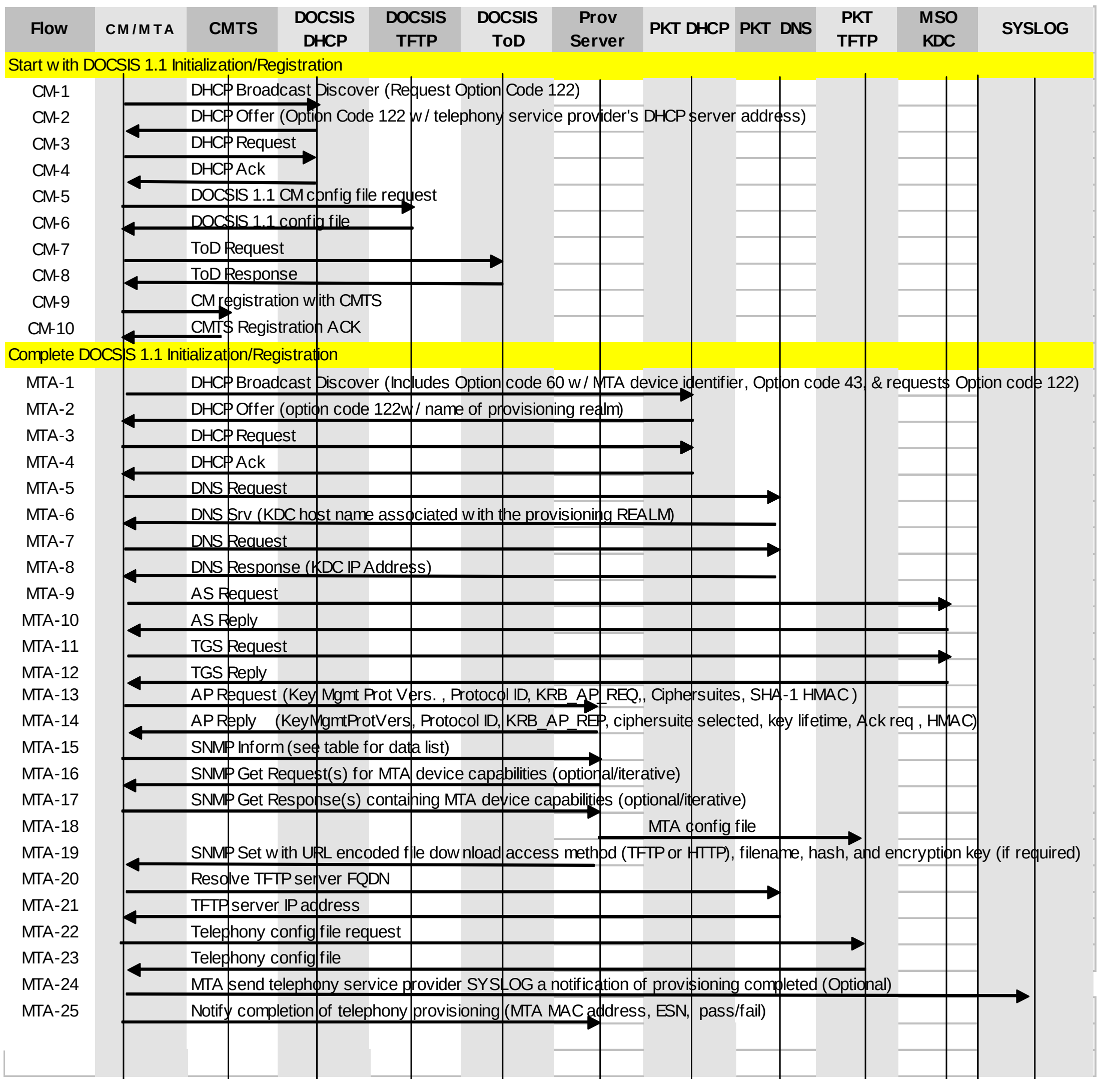
7.2 Embedded-MTA power-on initialization flows

Table 1 shows the mandatory message flow that the embedded-MTA device MUST follow during power-on initialization (unless stated explicitly otherwise). It is understood that these flows do not imply implementation or limit functionality.

Although these flows show the MTA configuration file download from a TFTP Server, the descriptive text details the requirements to support the MTA configuration file download from an HTTP Server.

Note in the flow details below that certain steps may appear to be a loop in the event of a failure. In other words, the step to proceed to if a given step fails, is to retry that step again. However, it is recommended that if the desired number of backoff and retry attempts does not allow the step to successfully complete, the device detecting the failure should generate a failure event notification.

In the flow details below (see Figure 6 and Table 1), the calculation of the Hash and the Encryption/Decryption of the MTA's Configuration File MUST follow the requirements in ITU-T Rec. J.170.



Flow
CM/MTA
CMTS
DOCSIS DHCP
DOCSIS TFTP
DOCSIS ToD
Prov Server
PKT DHCP
PKT DNS
PKT TFTP
MSO KDC
SYSLOG
Start with DOCSIS 1.1 Initialization/Registration
CM-1 DHCP Broadcast Discover (Request Option Code 122)
CM-2 DHCP Offer (Option Code 122 w/ telephony service provider's DHCP server address)
CM-3 DHCP Request
CM-4 DHCP Ack
CM-5 DOCSIS 1.1 CM config file request
CM-6 DOCSIS 1.1 config file
CM-7 ToD Request
CM-8 ToD Response
CM-9 CM registration with CMTS
CM-10 CMTS Registration ACK
Complete DOCSIS 1.1 Initialization/Registration
MTA-1 DHCP Broadcast Discover (Includes Option code 60 w/ MTA device identifier, Option code 43, & requests Option code 122)
MTA-2 DHCP Offer (option code 122w/ name of provisioning realm)
MTA-3 DHCP Request
MTA-4 DHCP Ack
MTA-5 DNS Request
MTA-6 DNS Srv (KDC host name associated with the provisioning REALM)
MTA-7 DNS Request
MTA-8 DNS Response (KDC IP Address)
MTA-9 AS Request
MTA-10 AS Reply
MTA-11 TGS Request
MTA-12 TGS Reply
MTA-13 AP Request (Key Mgmt Prot Vers. , Protocol ID, KRB_AP_REQ,, Ciphersuites, SHA-1 HMAC)
MTA-14 AP Reply (KeyMgmtProtVers, Protocol ID, KRB_AP_REP, ciphersuite selected, key lifetime, Ack req , HMAC)
MTA-15 SNMP Inform (see table for data list)
MTA-16 SNMP Get Request(s) for MTA device capabilities (optional/iterative)
MTA-17 SNMP Get Response(s) containing MTA device capabilities (optional/iterative)
MTA-18 MTA config file
MTA-19 SNMP Set with URL encoded file download access method (TFTP or HTTP), filename, hash, and encryption key (if required)
MTA-20 Resolve TFTP server FQDN
MTA-21 TFTP server IP address
MTA-22 Telephony config file request
MTA-23 Telephony config file
MTA-24 MTA send telephony service provider SYSLOG a notification of provisioning completed (Optional)
MTA-25 Notify completion of telephony provisioning (MTA MAC address, ESN, pass/fail)

Figure 6/J.167 – Embedded-MTA secure power-on initialization flow

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
CM1	The client device begins device registration by having the CM component send a broadcast DHCP discover message. Included in this message is option code 60 (Vendor Specific Option) in the format "docsis1.1:xxxxxxx". This message MUST request Option 122 in Option 55, the request parameter list. The remainder of this message MUST conform to the DHCP discover data as defined in ITU-T Rec. J.112.	Initial MUST Step in Sequence	Per DOCSIS
CM2	The DOCSIS DHCP Server, if it has been configured to support MTA devices, MUST include Option Code 122 with sub-option 1 and, possibly, sub-option 2 as per 8.1. If it is configured to prevent the MTA portion of the device from provisioning, then sub-option 1 in Option Code 122 MUST contain a DHCP server address of value 0.0.0.0. DOCSIS DHCP Servers without any prior knowledge of MTA devices MAY respond with DHCP OFFER without including option 122.	CM2 MUST occur after CM1 Completion	Per DOCSIS
CM3	Upon receiving a DHCP OFFER, the CM MUST check for the requested option 122. If it is not present, then it MUST retry the DHCP DISCOVER process (CM1) exponentially for 3 attempts (e.g., 2, 4, 8 second intervals). Upon failing to receive any DHCP OFFER with option 122 after the exponential retry mechanism, it MUST consider OFFERs without option code 122 and accept one of them as per the DHCP specification RFC 2131. The client device (CM) MUST then send a DHCP REQUEST broadcast message to the DHCP server whose OFFER is accepted as specified in the DHCP specification RFC 2131.	CM3 MUST occur after CM2 Completion	Per DOCSIS
CM4	The DHCP server sends the client device CM component a DHCP ACK message to confirm acceptance of the offered data. Upon receiving the DHCP ACK, the CM MUST check again for option 122. The absence of option 122 in the DHCP ACK message, that was accepted by the CM, implies that it MUST NOT initialize the embedded MTA. The presence of option 122 implies that it MUST initialize the MTA and pass sub-option 1 and, possibly, sub-option 2. If the option content of this DHCP ACK differs with the preceding DHCP OFFER, the option content of this DHCP ACK MUST be treated as authoritative (per RFC 2131).	CM4 MUST occur after CM3 Completion	Per DOCSIS
CM5-CM10	The client device's CM component completes the remainder of the CM specified registration sequence. This includes downloading the CM configuration file, requesting time of day registration and registering with the CMTS.	CM5-CM10 MUST occur after CM4 completion	Per DOCSIS

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA1	DHCP Broadcast Discover. The MTA MUST send a broadcast DHCP DISCOVER message. This message MUST include option code 60 (Vendor Specific Option) in the format "pktc1.0:xxxxxx". The MTA MUST include the DHCP option code 43 in the DHCP DISCOVER message as defined in 8.5. The MTA MUST request in DHCP option 55 the following: 1, 3, 6, 7, 12, 15, and 122 options. If the CM DHCP option code 122 sub-option 1 (passed by the CM to the MTA) contains a DHCP server of value of 0.0.0.0, then the MTA MUST not attempt to provision and MUST remain dormant until it is reinitialized by the CM.	MTA1 MUST NOT occur before completion of CM4	If failure per DHCP protocol, repeat MTA1
MTA2	DHCP OFFER The MTA may receive multiple DHCP OFFERs (during its wait period as per RFC 2131). The following requirements apply to the MTA and/or the Provisioning Applications. 1) The MTA MUST only accept a valid DHCP OFFER message. A valid DHCP OFFER MUST be sent by the primary or secondary DHCP servers returned in DHCP option code 122 sub-options 1 and 2 as obtained by the E-MTA via the CM provisioning step CM4. A valid DHCP OFFER MUST also include the following options: 1, 3, 6, 7, 12, 15, 122 with DHCP option 122 sub-options 3 and 6. DHCP option 122 MAY contain the additional sub-options 4, 5, 7, 8 and 9. 2) If the DHCP option 122 sub-option 6 returned by a valid DHCP server indicates that the Basic or Hybrid flow must be performed, the MTA MUST ignore the DHCP option 122 sub-options 4, 5, 7 and 9 if they are present. 3) If the DHCP option 122 sub-option 6 returned by a valid DHCP server indicates that the Basic Flow must be performed, the Provisioning Server MUST include the configuration file location in the 'siaddr' and 'file' fields in the DHCP responses. 4) If the DHCP option 122 sub-option 6 returned by a valid DHCP server indicates the Secure flow must be performed, the MTA MUST process the DHCP option 122 sub-options 4, 5, 7 and 9. The MTA next applies the following rules to the set of valid DHCP OFFERs: a) The MTA MUST check the value of the DHCP option 122 sub-option 3. If all valid OFFERs contain 0.0.0.0 in DHCP option 122 sub-option 3, then the MTA MUST not further the DHCP process and it MUST shutdown until it is reinitialized. Otherwise, the MTA MUST further restrict its set of valid OFFERs to those with a non-zero value in the DHCP option 122 sub-option 3.	MTA2 MUST occur after MTA1 completion	If failure per DHCP protocol, return to MTA1

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA2	b) The MTA MUST check the value of the DHCP option 122 sub-option 6 for indication of the Secure Flow. If no valid DHCP OFFER message directs the MTA to the Secure flow, the MTA MUST retry the DHCP DISCOVER process (MTA1) exponentially for 3 attempts (e.g., 2, 4, 8 second intervals). Upon failing to receive any valid DHCP OFFER indicating the Secure flow, the MTA MUST select, a valid Hybrid Flow DHCP OFFER, or a valid Basic Flow OFFER in that order. If no valid DHCP OFFER is received, the MTA MUST fail the corresponding provisioning flow step. NOTE – In the case of Secure Flow, if an MTA supports TGTs and receives the DHCP option 122 sub-option 7 set to a FALSE value, it MUST NOT request TGTs. If an MTA supports TGTs and receives the DHCP option 122 sub-option 7 set to a TRUE value, it MUST request TGTs. MTAs that do not support TGTs MUST ignore the DHCP option 122 sub-option 7.		
MTA3	DHCP Broadcast REQUEST Once the MTA has selected a valid DHCP OFFER, the MTA MUST send a DHCP REQUEST broadcast message to accept the DHCP OFFER per RFC 2131.	MTA3 MUST occur after MTA2 completion	If failure per DHCP protocol, return to MTA1
MTA4	DHCP ACK The DHCP server sends a DHCP ACK message to the MTA. The DHCP ACK message MUST include all options and sub-options which had been sent in MTA2 (DHCP OFFER). If the option and sub-option values of this DHCP ACK differ with the preceding DHCP OFFER (MTA2), the option and sub-option values of this DHCP ACK MUST be treated as authoritative (per RFC 2131). If the DHCP ACK is not valid as per the criteria established in MTA2, the MTA MUST fail this step. NOTE – The provisioning flow forks into one of three directions as follows: If the MTA4 DHCP ACK indicates the Basic Flow, the MTA MUST proceed to flow step B-MTA-22 described in 7.3. If the MTA4 DHCP ACK indicates the Hybrid Flow, the MTA MUST proceed to flow step H-MTA-15 described in 7.4. Otherwise, the Secure Flow is indicated and the MTA MUST proceed to step MTA5 below.	MTA4 MUST occur after MTA3 completion	If failure per DHCP protocol, return to MTA1
MTA5	DNS Srv Request The MTA requests the MSO KDC host name for the Kerberos realm.	MTA5 MUST occur after MTA4 completion	MTA1
MTA6	DNS Srv Reply Returns the MSO KDC host name associated with the provisioning REALM.	MTA6 MUST occur after MTA5 completion	MTA1

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA7	DNS Request The MTA now requests the IP Address of the MSO KDC.	MTA7 MUST occur after MTA6 completion	MTA1
MTA8	DNS Reply The DNS Server returns the IP Address of the MSO KDC.	MTA8 MUST occur after MTA7 completion	MTA1
MTA9	AS Request The AS Request message is sent to the MSO KDC to request a Kerberos ticket.	If MTA9 occurs, it MUST occur after MTA8 completion.	MTA1 The failure conditions are defined by the Security Specification J.170
MTA10	AS Reply The AS Reply Message is received from the MSO KDC containing the Kerberos ticket. NOTE 1 – The KDC must map the MTA MAC address to the FQDN before sending the AS Reply. NOTE 2 – Flows MTA11-MTA12 are optional in some cases, please refer to the Security Specification (J.170). NOTE 3 – SNMPv3 entity (FQDN) MUST be resolved to an IP address anywhere during flows MTA5 to MTA12. NOTE 4 – If an IP address is provided in the Additional information field of the DNS-SRV response (MTA6), MTA MAY use the same and skip the flows MTA7 and MTA8. NOTE 5 – If the MTA has valid provisioning application server ticket saved in NVRAM, then it MUST skip the flows MTA5 to MTA12 in successive MTA resets (flows MTA1 to MTA25).	MTA10 MUST occur after MTA9 completion	MTA1
MTA11	TGS Request If MTA obtained TGT in MTA10, the TGS Request message is sent to the MSO KDC.	If MTA11 occurs, it MUST occur after MTA10 completion	MTA1
MTA12	TGS Reply The TGS Reply message is received from the MSO KDC.	MTA12 MUST occur after MTA11 completion	MTA1
MTA13	AP Request The AP Request message is sent to the Provisioning Server to request the keying information for SNMPv3.	MTA13 MUST occur after MTA12 or MTA10 completion	MTA1 The failure conditions are defined by the Security Specification J.170

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA14	AP Reply The AP Reply message is received from the Provisioning Server containing the keying information for SNMPv3. NOTE – The SNMPv3 keys must be established before the next step using the information in the AP Reply.	MTA14 MUST occur after MTA13 completion	MTA1
MTA15	SNMP Enrollment INFORM The MTA MUST send an SNMPv3 Enrollment INFORM to the PROV_SNMP_ENTITY (specified in the DHCP option 122 sub-option 3). The SNMP INFORM MUST contain a "PktcMtaDevProvisioningEnrolment" object as defined in ITU-T Rec. J.166. The PROV_SNMP_ENTITY notifies the Provisioning Application that the MTA has entered the management domain. NOTE – The provisioning server can reset the MTA at this point in the flows. The MTA is part of the security domain and MUST respond to management requests, the SNMP INFORM of MTA15 is the indicator, see 5.4.1.2.	MTA15 MUST occur after MTA14 completion	If failure per SNMP protocol, return to MTA1. SNMP server MUST send response to SNMP-INFORM.
MTA16	SNMPv3 GET Request (Optional). If any additional MTA device capabilities are needed by the PROV_APP, the PROV_APP requests these from the MTA via SNMPv3 Get Requests. This is done by having the PROV_APP send the PROV_SNMP_ENTITY a "get request" Iterative: The PROV_SNMP_ENTITY sends the MTA one or more SNMPv3 GET requests to obtain any needed MTA capability information. The Provisioning Application may use a GETBulk request to obtain several pieces of information in a single message.	MTA16 is optional, can occur after MTA15 completion	N/A
MTA17	SNMPv3 GET Response Iterative: MTA sends the PROV_SNMP_ENTITY a response for each GET Request. After all the Gets, or the GetBulk, finish, the PROV_SNMP_ENTITY sends the requested data to the PROV_APP.	MTA17 MUST occur after MTA16 completion if MTA16 is performed	N/A
MTA18	This Protocol is not defined by IPCablecom. The PROV_APP MAY use the information from MTA16 and MTA17 to determine the contents of the MTA Configuration Data file. Mechanisms for sending, storing and, possibly, creating the configuration file are outlined in MTA19.	MTA18 SHOULD occur after MTA15 completion unless MTA16 is performed, then it SHOULD be after MTA17 has completed	N/A

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA19	SNMPv3 SET The PROV_APP MAY create the configuration file at this point, or send a predefined one. A hash MUST be run on the contents of the configuration file. The configuration file MAY be encrypted. The hash and the encryption key (if the configuration file is encrypted) MUST be sent to the MTA. The PROV_APP MUST store the configuration file on the appropriate TFTP server. The PROV_APP then instructs the PROV_SNMP_ENTITY to send an SNMP SET message to the MTA containing the following variables (defined in ITU-T Rec. J.166). pktcMtaDevConfigFile pktcMtaDevProvConfigHash and pktcMtaDevProvConfigKey (This MUST NOT be included if the MTA configuration file is unencrypted). NOTES: 1) In the case of file download using the HTTP access method, the filename MUST be URL-encoded with a URL format compliant with RFC 2616 with exception stated below in Note 3. 2) In the case of file download using the TFTP access method, the filename MUST be URL-encoded with a URL format compliant with RFC 3617 with exception stated below in Note 3. 3) MTA MUST accept IPv4 addresses embedded in URL encoded format with or without square brackets.	MTA19 MUST occur after MTA18 completion	If failure per SNMP protocol, return to MTA1
MTA20	DNS Request If the URL-encoded access method contains a FQDN instead of an IPv4 address, the MTA MUST use the service provider network's DNS server to resolve the FQDN into an IPv4 address of either the TFTP Server or the HTTP Server.	MTA20 MUST occur after MTA19 completion if FQDN is used	If failure per DNS protocol, return to MTA1
MTA21	DNS Reply DNS Response: DNS server returns the IP address against MTA20 DNS request.	MTA21 MUST occur after MTA20 completion if FQDN is used	If failure per DNS protocol, return to MTA1
MTA22	TFTP/HTTP Configuration file Request The MTA MUST perform either the TFTP or HTTP protocol exchange, as specified in step S-MTA-19, to download its configuration file. For specific details of each protocol, see RFC 3415, RFC 3412.	MTA22 MUST occur: After MTA19 if DNS resolution is not required. After MTA21 if DNS resolution is required.	If failure per TFTP or HTTP protocols, return to MTA1

Table 1/J.167 – Embedded-MTA power-on initialization flow description

Flow	Embedded-MTA power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
MTA23	TFTP/HTTP Configuration file Request The TFTP/HTTP Server MUST send the requested configuration file to the MTA. Specific details of each protocol are found in RFC 3415 and RFC 3412. The hash of the downloaded configuration file is calculated by the MTA and compared to the value received in step MTA19. If the hashes do not match, the MTA MUST fail this step. If encrypted, the configuration file MUST be decrypted. Refer to 9.1 for MTA configuration file contents.	MTA23 MUST occur after MTA22 completion	If the configuration file download failed per TFTP or HTTP protocols, return to MTA1. Otherwise, proceed to MTA24 or MTA25, and send the failed response if the MTA configuration file itself is in error.
MTA24	SYSLOG Notification If a SYSLOG server is configured and enabled as part of the Provisioning Process (Refer to step MTA2 for DHCP Options and ITU-T Recs J.172 and J.166 for configuration using the MEM-MIB), then the MTA MUST send the voice service provider's SYSLOG a "provisioning complete" event indicating the status of the provisioning operation. This notification will include the pass-fail result of the provisioning operation. The general format of this notification is as defined in 5.4.3.	MTA24 MUST occur after MTA23 completion if SYSLOG is configured	The MTA MAY retry this step before proceeding to MTA25.
MTA25	SNMP INFORM The MTA MUST send the PROV_SNMP_ENTITY (specified in DHCP option 122 sub-option 3) an SNMP INFORM containing a "provisioning complete" notification. The receipt of the inform is acknowledged by the response message as defined in RFC 3414. The SNMP INFORM MUST contain a "PktcMtaDevProvisioningStatus" object as defined in ITU-T Rec. J.166. NOTES: 1) At this stage, the MTA device provisioning data is sufficient to provide any minimal services as determined by the service provider (e.g., 611). 2) Depending on the TLV-38 configuration, there might be multiple SNMP INFORMs sent to the configured SNMP Management stations.	MTA25 MUST occur after MTA24 if SYSLOG is used, otherwise MUST occur after MTA23 completion	MTA MAY generate a Provisioning Failure event notification to the Service Provider's Fault Management server. Provisioning process stops; Manual interaction required. SNMP server MUST send response to SNMP-INFORM.

7.3 Embedded-MTA power-on initialization flow (Basic flow)

The Basic MTA provisioning flow is very similar to the DOCSIS CM provisioning flow. See Figure 7 and Table 2.

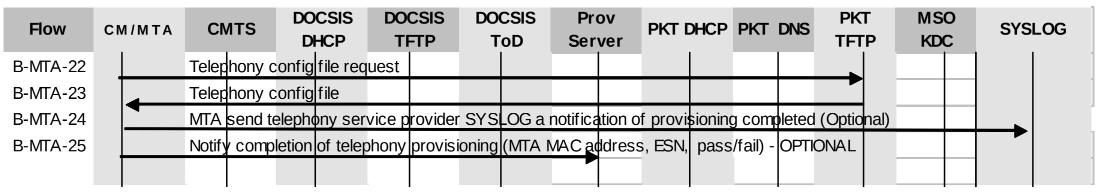

Flow
CM/MTA
CMTS
DOCSIS DHCP
DOCSIS TFTP
DOCSIS ToD
Prov Server
PKT DHCP
PKT DNS
PKT TFTP
MSO KDC
SYSLOG
B-MTA-22
Telephony config file request
B-MTA-23
Telephony config file
B-MTA-24
MTA send telephony service provider SYSLOG a notification of provisioning completed (Optional)
B-MTA-25
Notify completion of telephony provisioning (MTA MAC address, ESN, pass/fail) - OPTIONAL

Figure 7/J.167 – Embedded-MTA basic power-on initialization flow

Table 2/J.167 – Embedded-MTA basic power-on initialization flow description

Flow	**Embedded-MTA basic power-on initialization flow description**	**Normal flow sequencing**	**MUST proceed to here if this step fails**
	NOTE – The FQDN provided in the DHCP ACK in DHCP option 122 sub-option 3 (Provisioning Entity Address) MUST be resolved to an IP address before step B-MTA-22.		
B-MTA-22	TFTP Configuration File Request The MTA MUST perform a TFTP protocol exchange to download its configuration file. The 'siaddr' and 'file' fields of the DHCP ACK are used to locate the configuration file. Specific details of the TFTP protocol can be found in RFC 3415.	B-MTA-22 MUST occur after MTA4.	If failure per TFTP protocol, return to MTA1.
B-MTA-23	TFTP Configuration File Response The TFTP server MUST send the requested configuration file to the MTA. Specific details of the TFTP protocol can be found in RFC 3415. The downloaded configuration file MUST contain the MIB object 'pktcMtaDevConfigHash'. The MTA MUST calculate the hash of the downloaded configuration file per 9.1 and compare this value to the value contained in the 'pktcMtaDevConfigHash' object. If these values do not match, this step MUST fail. Refer to 9.1 for MTA configuration file contents.	B-MTA-23 MUST occur after B-MTA-22	If the configuration file download failed per TFTP protocols, return to MTA1. Otherwise, proceed to B-MTA-24 and send the failed response if the MTA configuration file itself is in error.

Table 2/J.167 – Embedded-MTA basic power-on initialization flow description

Flow	Embedded-MTA basic power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
B-MTA-24	SYSLOG Notification If a SYSLOG server is configured and enabled as part of the Provisioning Process (refer to step MTA2 for DHCP Options and ITU-T Recs J.172 and J.166 for configuration using the MEM-MIB), then the MTA MUST send the voice service provider's SYSLOG a "provisioning complete" event indicating the status of the provisioning operation. The general format of this notification is as defined in 5.4.3.	B-MTA-24 MUST occur after B-MTA-23 completion if SYSLOG is configured	The MTA MAY retry this step before proceeding to B-MTA-25.
B-MTA-25	SNMPv2C Provisioning Status INFORM (optional) If commanded by DHCP option 122 sub-option 6, the MTA MUST send the PROV_SNMP_ENTITY (specified in DHCP option 122 sub-option 3) an SNMP INFORM containing a "provisioning complete" notification. The receipt of the SNMP INFORM is acknowledged. The SNMP INFORM MUST contain a "PktcMtaDevProvisioningStatus" object as defined in ITU-T Rec. J.166. The SNMPv2c community name used in the status SNMP INFORM MUST have a value "public" (taken without the quotation mark). NOTES: 1) At this stage, the MTA device provisioning data is sufficient to provide any minimal services as determined by the service provider (e.g., 611). 2) Depending on the TLV-38 configuration value pairs, there might be multiple SNMP INFORMs sent to the configured SNMP Management stations.	B-MTA-25 is optional, MAY occur after B-MTA-24 if SYSLOG is used, otherwise MAY occur after B-MTA-23 completion	Provisioning process stops; Manual interaction required. SNMP server MUST send response to SNMP-INFORM

7.4 Embedded-MTA power-on initialization flow (Hybrid flow)

The Hybrid Provisioning Flow (Hybrid Flow) is essentially the Secure Flow with Kerberos exchanges removed and SNMPv2c substituted for SNMPv3. The SNMPv2c community name, used in the SNMP INFORM messages sent by the MTA in steps H-MTA-15 and H-MTA-25 below, MUST have a value "public" (taken without the quotation mark). See Figure 8 and Table 3.

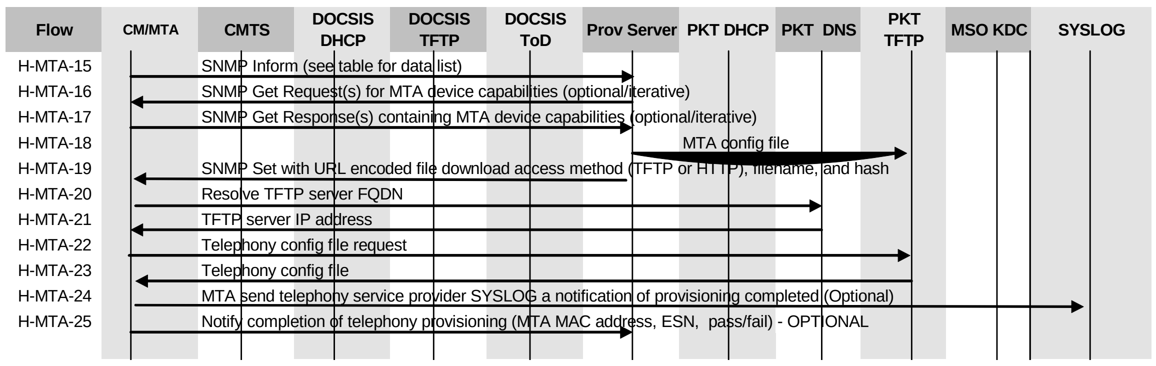

Flow
CM/MTA
CMTS
DOCSIS DHCP
DOCSIS TFTP
DOCSIS ToD
Prov Server
PKT DHCP
PKT DNS
PKT TFTP
MSO KDC
SYSLOG
H-MTA-15
SNMP Inform (see table for data list)
H-MTA-16
SNMP Get Request(s) for MTA device capabilities (optional/iterative)
H-MTA-17
SNMP Get Response(s) containing MTA device capabilities (optional/iterative)
H-MTA-18
MTA config file
H-MTA-19
SNMP Set with URL encoded file download access method (TFTP or HTTP), filename, and hash
H-MTA-20
Resolve TFTP server FQDN
H-MTA-21
TFTP server IP address
H-MTA-22
Telephony config file request
H-MTA-23
Telephony config file
H-MTA-24
MTA send telephony service provider SYSLOG a notification of provisioning completed (Optional)
H-MTA-25
Notify completion of telephony provisioning (MTA MAC address, ESN, pass/fail) - OPTIONAL

Figure 8/J.167 – Embedded-MTA hybrid power-on initialization flow

Table 3/J.167 – Embedded-MTA hybrid power-on initialization flow description

Flow	**Embedded-MTA hybrid power-on initialization flow description**	**Normal flow sequencing**	**MUST proceed to here if this step fails**
	NOTE – The FQDN provided in the DHCP ACK in DHCP option 122 sub-option 3 (Provisioning Entity Address) MUST be resolved to an IP address before step H-MTA-15.		
H-MTA-15	SNMPv2c Enrolment INFORM The MTA MUST send a SNMPv2c Enrolment INFORM to PROV_SNMP_ENTITY (specified in the DHCP option 122 sub-option 3). The SNMP INFORM MUST contain a 'PktcMtaDevProvisioningEnrollment' object as defined in ITU-T Rec. J.166. The PROV_SNMP_ENTITY notifies the PROV_APP that the MTA has entered the management domain.	H-MTA-15 MUST occur after MTA4 completion	If failure per SNMP protocol, return to MTA1. SNMP server MUST send response to SNMP-INFORM.
H-MTA-16	SNMPv2c GET Request (optional) The Provisioning Application may request additional MTA device capabilities from the MTA via SNMPv2c GET requests. This is done by having the Provisioning Application send the PROV_SNMP_ENTITY an SNMP GET request. Iterative: The PROV_SNMP_ENTITY sends the MTA one or more SNMPv2c GET requests to obtain any needed MTA capability information. The Provisioning Application may use a GETBulk request to obtain several pieces of information in a single message.	H-MTA-16 is optional, can occur after H-MTA-15 completion	N/A

Table 3/J.167 – Embedded-MTA hybrid power-on initialization flow description

Flow	Embedded-MTA hybrid power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
H-MTA-17	SNMPv2c GET Response (optional) Iterative: MTA sends the PROV_SNMP_ENTITY a Get Response for each Get Request. After all the Gets, or the GetBulk, finish, the PROV_SNMP_ENTITY sends the requested data to the Provisioning Application.	H-MTA-17 MUST occur after H-MTA-16 completion if H-MTA-16 is performed	N/A
H-MTA-18	This protocol is not defined by IPCablecom. The Provisioning Application MAY use the information from H-MTA-15, -16, and -17 to determine the contents of the MTA configuration data file. Mechanisms for sending, storing and, possibly, creating the configuration file are outlined in H-MTA-19.	H-MTA-18 SHOULD occur after H-MTA-15 completion unless H_MTA-16 is performed, then it SHOULD be after H-MTA-17 has completed	N/A
H-MTA-19	SNMPv2c Configuration File Set The Provisioning Application MAY create the configuration file at this point, or send a predefined one. The Provisioning Application MUST calculate SHA-1 hash on the contents of the configuration file. The Provisioning Application MUST store the configuration file on the appropriate TFTP server. The Provisioning Application then instructs the PROV_SNMP_ENTITY to send an SNMPv2c SET message to the MTA, containing the following varbindings (defined in ITU-T Rec. J.166): pktcMtaDevConfigFile pktcMtaDevProvConfigHash Unlike the Secure Flow, the pktcMtaDevProvConfigKey MIB object MUST NOT be included. If the pktcMtaDevProvConfigKey MIB object is included, the MTA MUST return an 'inconsistent value' error (Refer to RFC 3413 for more information regarding SNMP SET Responses).	H-MTA-19 MUST occur after H-MTA-18 completion	If failure per SNMP protocol, return to MTA1

Table 3/J.167 – Embedded-MTA hybrid power-on initialization flow description

Flow	Embedded-MTA hybrid power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
	NOTES: 1) In the case of file download using the HTTP access method, the filename MUST be URL-encoded with a URL format compliant with RFC 2616 with the exception stated below in Note 3. 2) In the case of file download using the TFTP access method, the filename MUST be URL-encoded with a URL format compliant with RFC 3617 with the exception stated below in Note 3. 3) MTA MUST accept IPv4 addresses embedded in URL encoded format with or without square brackets.		
H-MTA-20	DNS Request (optional) If the URL-encoded access method contains a FQDN instead of an IPv4 address, the MTA MUST use the service provider network's DNS server to resolve the FQDN into an IPv4 address of either the TFTP Server or the HTTP Server.	H-MTA-20 MUST occur after H-MTA-19 completion if FQDN is used	If failure per DNS protocol, return to MTA1
H-MTA-21	DNS Reply (optional) DNS Response: DNS server returns the IP address against H-MTA-20 DNS request.	H-MTA-21 MUST occur after H-MTA-20 completion if FQDN is used	If failure per DNS protocol, return to MTA1
H-MTA-22	TFTP/HTTP Configuration file Request The MTA MUST perform either the TFTP or HTTP protocol exchange, as specified in step H-MTA-19, to download its configuration file. For specific details of each protocol, see RFCs 3415 and 3412.	H-MTA-22 MUST occur after H-MTA-19 unless FQDN is specified then MUST be after H-MTA-21.	If failure per TFTP or HTTP protocols, return to MTA1.

Table 3/J.167 – Embedded-MTA hybrid power-on initialization flow description

Flow	Embedded-MTA hybrid power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
H-MTA-23	TFTP/HTTP Configuration file Response TFTP/HTTP server MUST send the requested configuration file to the MTA. Specific details of each protocol are found in RFCs 3415 and 3412. The hash of the downloaded configuration file is calculated by the MTA and compared to the value received in step H-MTA-19. If the hashes do not match, this step MUST fail. Refer to 9.1 for MTA configuration file contents.	H-MTA-23 MUST occur after H-MTA-22	If the configuration file download failed per TFTP or HTTP protocols, return to MTA1. Otherwise, proceed to MTA24 or MTA25, and send the failed response if the MTA configuration file itself is in error.
H-MTA-24	SYSLOG Notification If a SYSLOG server is configured and enabled as part of the Provisioning Process (Refer to step MTA2 for DHCP Options and ITU-T Recs J.172 and J.166 for configuration using the MEM-MIB), then the MTA MUST send the voice service provider's SYSLOG a "provisioning complete" event indicating the status of the provisioning operation. This notification will include the pass-fail result of the provisioning operation. The general format of this notification is as defined in 5.4.3.	H-MTA-24 MUST occur after H-MTA-23 completion if SYSLOG is configured.	The MTA MAY retry this step before proceeding to H-MTA-25.

Table 3/J.167 – Embedded-MTA hybrid power-on initialization flow description

Flow	Embedded-MTA hybrid power-on initialization flow description	Normal flow sequencing	MUST proceed to here if this step fails
H-MTA-25	SNMPv2c Provisioning Status Inform (optional) If commanded by DHCP 122 sub-option 6, the MTA MUST send the PROV_SNMP_ENTITY (specified in DHCP option 122 sub-option 3) a SNMPv2C Provisioning Status INFORM containing a "provisioning complete" notification. The receipt of the inform is acknowledged. The inform MUST contain a 'PktcMtaDevProvisioningStatus' object as defined in ITU-T Rec. J.166. NOTES: 1) At this stage, the MTA device provisioning data is sufficient to provide any minimal services as determined by the service provider (e.g., 611). 2) Depending on the TLV-38 configuration, there might be multiple SNMPv2c INFORM sent to the configured SNMP Management stations.	H-MTA-25 is optional. It MAY occur after H-MTA-24 if SYSLOG is used, otherwise it MAY occur after H-MTA-23 completion	Provisioning process stops; Manual interaction required. SNMP server MUST send response to SNMP-INFORM

7.5 Endpoint provisioning completion notifications

After the MTA has been provisioned successfully regardless of the selected provisioning flow, the MTA will set up the necessary security association for the related CMS configured realms (KDCs). The MTA NCS signalling software will initiate the establishment of the IPSec security association to the configured CMS clusters. Event notifications are triggered if security associations cannot be established (based on ITU-T Rec. J.170).

With the selected Basic, Hybrid, or Secure flow complete, and after any required security associations are established, the MTA NCS signalling software determines whether a signalling path can be setup with an RSIP message and the associated ACK. Coming from a link down situation, the MTA will send an SNMP Link Up Trap when the RSIP has been properly acknowledged. This indicates that the endpoint is provisioned. If the same CMS is used for multiple endpoints, a SNMP link up message will be sent for each associated endpoint. If not all endpoints use the same CMS, the same process needs to be repeated for each endpoint needing a different configured CMS.

7.6 Post initialization incremental provisioning

This clause describes the flows allowing the Provisioning Application to perform incremental provisioning of individual voice communications endpoints after the MTA has been initialized. Post-initialization incremental provisioning MAY involve communication with a Customer Service Representative (CSR).

7.6.1 Synchronization of provisioning attributes with configuration file

Incremental provisioning includes adding, deleting and modifying subscriber services on one or more endpoints of the embedded-MTA. Services on an MTA endpoint MUST be modified using SNMP via the MTA MIB (ITU-T Rec. J.166). The back office applications SHOULD support a "flow-through" provisioning mechanism that synchronizes all device provisioning information on the embedded-MTA with the appropriate back office databases and servers. Synchronization is required in the event that provisioning information needs to be recovered in order to reinitialize the device. Although the details of the back office synchronization are beyond the scope of this Recommendation, it is expected that, at a minimum, the following information be updated: customer records and the MTA configuration file on the TFTP or HTTP server.

7.6.2 Adding/Enabling telephone services on an MTA endpoint

The Telephony Services may be added and/or enabled on an MTA endpoint. Telephony Services may be added to MTA endpoints that have not been previously provisioned.

Whenever such an MTA endpoint is added/enabled:

- The MTA MUST have been provisioned with the 'device level' configuration data via the configuration file (as described in 9.1.1).
- The authorized SNMP Management Station MUST provision all required configuration attributes as described in 9.1.3, 9.1.4 and 9.1.5 using SNMP SET operations to update the provisioning attributes on the device for the specific telephony port being enabled.

Telephony Services may be enabled for MTA endpoints with services provisioned, but disabled (refer to 7.6.3 and 9.1.1 for more details). To enable previously disabled telephony services on the MTA endpoint, an authorized SNMP Management Station MUST use appropriate SNMP SET operations to achieve both of the following:

- Ensure that the row status MIB Object (pktcNcsEndPntConfigStatus) for the row corresponding to the endpoint is set to a value of "active (1)" (modify it appropriately if it is set to any other value).
- Ensure that the value of "ifAdminStatus" corresponding to the endpoint being enabled has a value of "up(2)" (modify it appropriately if it is set to any other value).

When an endpoint is provisioned or enabled, the MTA MUST perform the following steps (not necessarily in this order):

- Follow the procedures described in 7.1.1.2.5 of the Security Specification (J.170).
- Modify the "ifOperStatus" MIB Object according to 7.7.

If "pktcMtaDevEnabled" MIB Object is set to "true (1)", the MTA MUST follow the above steps for all configured endpoints.

It is to be noted that, given the nature of the MIB Object controlling the absence or presence of IPSec Security Associations with a Call Management Server, Endpoint Provisioning cannot be used to change the IPSec status (refer to Annex B/J.166 for more information). Thus, enabling new services with a Call Management Server whose status has not been indicated earlier (via the configuration file) will result in IPSec being enabled, upon assignment to an endpoint.

As an example of enabling telephony services on an endpoint, consider the case where a subscriber has requested service on an endpoint that has not been previously provisioned.

NOTE – This example assumes the service provider's account creation process has been completed, and shows only the components critical for the flows. For instance, account creation and billing database creation are assumed to be available and integrated in the back office application suite.

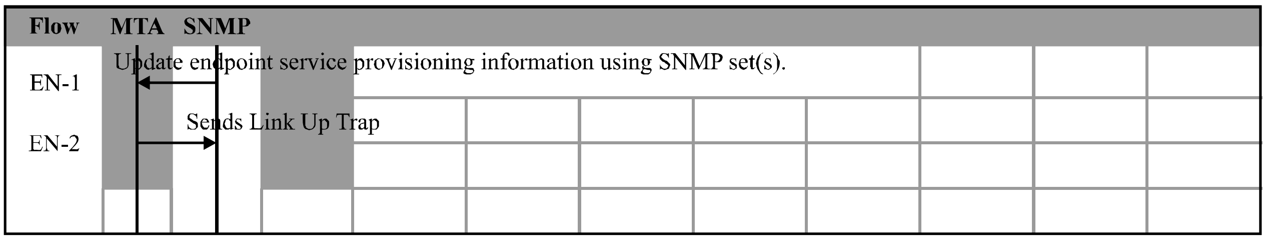

Flow
MTA
SNMP
EN-1
EN-2
Update endpoint service provisioning information using SNMP set(s).
Sends Link Up Trap
J.167_F09

Figure 9/J.167 – Enabling services on an MTA endpoint

Table 4/J.167 – Enabling services on an MTA endpoint flow description

Flow	Enabling services on an MTA endpoint flow description	Normal flow sequencing
EN-1	Authorized SNMP Management Station performs required SNMP SET operations to add services on the MTA Endpoint.	If Endpoint configuration is desired, EN-1 MUST occur after successful completion of power-on initialization flow.
EN-2	The MTA MUST send a Link Up trap to the configured SNMP Management Stations. Refer to 7.7 and the IF-MIB (RFC 2863) for more information.	EN-2 MUST occur after EN-1

7.6.3 Deleting/Disabling telephony services on an MTA endpoint

Provisioned and enabled Telephony Services can be disabled (taken out of service) or deleted if required using SNMP via the MTA MIB (ITU-T Rec. J.166) and the Signaling MIB (ITU-T Rec. J.166) on a per-endpoint basis.

Whenever a telephony service is desired to be deleted on an endpoint, the authorized SNMP Management Station MUST delete appropriate configuration attributes described in 9.1.3, 9.1.4 and 9.1.5 using SNMP SET operations for the corresponding endpoint.

To disable the services on an MTA endpoint, an authorized SNMP Management Station MUST use SNMP SET operations to accomplish one or more of the following conditions:

- For the particular endpoint, modify the row status object to a value other than "active (1)" in "pktcNcsEndPntConfigTable".
- Modify the value of "ifAdminStatus" to "down (2)", for the particular endpoint.

If the endpoint is being deleted or disabled while a call is in progress, the MTA MUST:

- Shutdown all media sessions if present.
- Shutdown NCS signalling by following the Restart in Progress procedures in the IPCablecom NCS Specification (ITU-T Rec. J.162).
- Set the pktcNcsEndPntStatusError MIB Object for the particular endpoint to the "disconnected (3)" state.

If "pktcMtaDevEnabled" MIB Object is set to "false (2)", the MTA MUST follow the above procedure for all configured endpoints.

As an example of disabling telephony services on an endpoint, consider the case where a subscriber has requested disabling telephony services on a previously configured endpoint.

NOTE 1 – It is assumed that the service provider's account update process has been completed and shows only the applications critical to MTA operation.

NOTE 2 – This example assumes the service provider's account update process has been completed and shows only the applications critical to MTA operation.

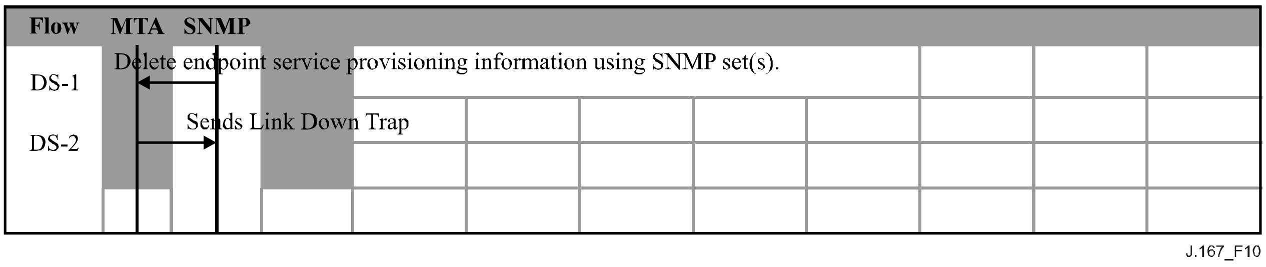

Flow
MTA
SNMP
DS-1
Delete endpoint service provisioning information using SNMP set(s).
DS-2
Sends Link Down Trap
J.167_F10

Figure 10/J.167 – Disabling services on an MTA endpoint

Table 5/J.167 – Disabling services on an MTA endpoint flow description

Flow	Disabling services on an MTA endpoint flow description	Normal flow sequencing
DS-1	Authorized SNMP Management Station performs required SNMP SET operations to disable services on the MTA Endpoint.	DS-1 MUST occur after the endpoint is brought to enabled state either immediately after the initial provisioning or after the per-endpoint incremental provisioning.
DS-2	The MTA MUST send a Link Down trap to the configured SNMP Management Stations. Refer to 7.7 and the IF-MIB (RFC 2863) for more information.	DS-2 MUST occur after DS-1

7.6.4 Modifying telephone services on an MTA endpoint

Telephony Services may be modified on a currently provisioned 'MTA Endpoint'. This is accomplished using SNMP via the MTA MIB (ITU-T Rec. J.166), and the Signalling MIB (ITU-T Rec. J.166) on a per-endpoint basis. If such a modification to an endpoint changes the CMS association (pktcNcsEndPntConfigCallAgentId) and/or the port (pktcNcsEndPntConfigCallAgentUdpPort), the endpoint is treated as being taken out of service (as per 7.6.3), followed by placing the endpoint back in service (as per 7.6.2).

The MTA MUST also follow the procedures described in 7.1.1.2.5 of the Security Specification (J.170).

It is to be noted that:

- Modification to call service features requires modifications in the CMS, not in the MTA.
- Modification to service level parameters related to the eCM component of the eMTA may require rebooting of the E-MTA.

7.7 Reflecting the State of the Endpoint Interface in the ifTable

The operational state of each 'MTA Endpoint' is reflected in the "ifOperStatus" MIB Object of the MTA. This is influenced by the following conditions:

- The corresponding administrative status for the endpoint, reflected in the "ifAdminStatus" table.
- The state of the telephony service assigned to the corresponding endpoint.

- The presence or absence of the IPSec security associations on the corresponding endpoint, provided IPSec is enabled (i.e., the MIB Object "pktcMtaDevCmsIpsecCtrl" set to a value of "true(1)" for that endpoint).

Whenever an MTA reinitializes (following a reboot or a reset), it MUST immediately set the "ifAdminStatus" entries corresponding to all available physical endpoints to a value of 'up (1)'. However, entries in the configuration file or the SNMP Management station can change this status. The MTA MUST further reflect the above conditions in the operational status of each endpoint as explained below.

For each entry corresponding to an endpoint in the "ifTable" MIB, the MTA MUST set the "ifOperStatus" to a value of:

- "down(2)", if the corresponding endpoint is disabled or deleted, or the corresponding "ifAdminStatus" is set to a value of "down(2)";
- "up(1)", if the corresponding "ifAdminStatus" has a value of "up(1)", the telephony services have been added/enabled for the particular endpoint, and IPSec is disabled with the assigned Call Management Server;
- "up(1)", if the corresponding "ifAdminStatus" has a value of "up(1)", the telephony services have been added/enabled for the particular endpoint, IPSec is enabled for the assigned Call Management Server, and the IPSec Security association has been established;
- "dormant(3)", if the corresponding "ifAdminStatus" has a value of "up(1)", the telephony services have been added/enabled for the particular endpoint, IPSec is enabled for the assigned Call Management Server, but the IPSec Security association has not been established.

Further, the MTA MUST not set the 'ifOperStatus' to a value of 'dormant(3)' for endpoints on which IPSec is disabled. Refer to ITU-T Rec. J.166 for more details on enabling/disabling IPSec, 7.6.2 for more details on adding/enabling endpoints, and 7.6.3 for more details on deleting/disabling endpoints.

The MTA MUST be able to enable or disable the 'Link Up Trap' and 'Link Down Trap' by using the "ifLinkUpDownTrapEnable" MIB Object (Refer to the IF-MIB for more details).

7.8 Provisioning of the signalling communication path between the MTA and CMS

All issues related to the creation and handling of the NCS Service Flows are considered to be resolved by the DOCSIS means and are out of the scope of this Recommendation.

7.9 MTA replacement

IPCablecom has no requirement to specify MTA replacement procedures. However, the provisioning sequence flows detailed within this Recommendation provide sufficient coverage and flexibility to support replacement. In fact, the initialization sequence for a replacement MTA could be the same as the original MTA's first time initialization. Back office procedures related to migration of subscriber profiles from one MTA to another are specific to individual service provider's network operations. As a result of this wide variance, discussion of these back office procedures are beyond the scope of this Recommendation.

7.10 Temporary signal loss

If the eCM (in an E-MTA) resets due to any Rf condition (for example Temporary Rf loss), then the associated IPCablecom eMTA MUST also reset.

NOTE – This will impact calls in progress.

7.11 MTA hard reboot/soft reset scenarios

Hard Reboot is defined as a 'power cycle' of the entire eMTA device. Soft Reset is defined as an 'SNMP reset' of the MTA portion of an eMTA, an SNMP reset of the eCM (resulting in the reset of the associated eMTA) or an Rf condition that results in a reset of the eCM (resulting in the reset of the associated MTA).

The MTA part of an eMTA MUST NOT differentiate between a 'Hard Reboot' and a 'Soft Reset'. To be more specific, the MTA MUST have the same initialization parameters (for example, SNMP tables) and follow any requirements regarding persistent information (e.g., NVRAM ticket storage) the same way in either scenario.

8 DHCP options

DHCP is used to obtain IPv4 addresses for both the CM and the MTA. The CM and MTA requirements for DHCP Option Codes 122 and 60 are detailed in 8.1 and 8.2. If total number of octets in any DHCP option exceeds 255 octets, then the MTA MUST follow RFC 3396 to split the DHCP message into multiple sub-messages.

8.1 DHCP option 122: Client configuration option

DHCP option code 122 is the RFCed replacement for the former option 177 (which was intended as a temporary code). CM and MTA MUST NOT request option 177 in their DHCP DISCOVER or REQUEST message in option 55 (parameter request list). In the case that a CM or MTA requests both options 122 and 177:

- The provisioning server MUST respond with DHCP option 122.
- The provisioning server MUST NOT respond with DHCP option 177.
- CM and MTA MUST treat DHCP option 122 as authoritative.

DHCP option code 122 is used in both the CM and MTA DHCP OFFER/ACK messages to provide the addresses of valid IPCablecom network servers and various device configuration data.

Full details of DHCP option 122 encoding can be found in RFC 3495 and RFC 3594.

The following clauses provide additional semantic details of each sub-option in DHCP option 122.

Table 6/J.167 – Server options

Option	Sub-option	Description and comments	Sub-option required or optional	Default value
122	1	Service Provider's Primary DHCP Server Address Required by CM only.	Required	N/A
	2	Service Provider's Secondary DHCP Server Address Optional requirement for CM.	Optional	Empty String
	3	Service Provider's Provisioning Entity Address	Required	N/A
	4	AS-REQ/REP Exchange Backoff and Retry for SNMPv3 Key Management	Optional	As per the following MIB Objects: "pktcMtaDevRealmUnsolicitedKeyNomTimeout", "pktcMtaDevRealmUnsolicitedKeyMaxTimeout", "pktcMtaDevRealmUnsolicitedKeyMaxRetries"
	5	AP-REQ/REP Kerberized Provisioning Backoff and Retry	Optional	As per the following MIB Objects: "pktcMtaDevProvUnsolicitedKeyNomTimeout" "pktcMtaDevProvUnsolicitedKeyMaxTimeout" "pktcMtaDevProvUnsolicitedKeyMaxRetries"
	6	Kerberos Realm of SNMP Entity	Required	N/A
	7	Ticket Granting Server Usage	Optional	N/A – if MTA does not implement TGT.0 – otherwise.
	8	Provisioning Timer	Optional	As per "pktcMtaDevProvisioningTimer" MIB Object (10 minutes)
	9	Security Ticket Invalidation	Optional	0 – apply normal ticket invalidation rules per J.170

MTA MUST be able to retrieve and process the data from all sub-options in the above table. Provisioning Server MUST supply to the MTA all "required" sub-options and MAY supply all "optional" sub-options.

If an "optional" sub-option is not supplied by the Provisioning Server, the MTA MUST use the default value of the sub-option.

If the "required" sub-option is not supplied by the Provisioning Server, the MTA MUST reject the corresponding DHCP OFFER/ACK.

If the sub-option contains wrong (invalid) value, the MTA MUST:

- reject the corresponding DHCP OFFER/ACK in case of "required" sub-option;
- use the default value in case of "optional" sub-option. For any sub-option with multiple parameters (e.g., Option 122 sub-option 4 or Option 122 sub-option 5), the MTA MUST apply the corresponding default value only to the parameter (or parameters) that contains the wrong value.

An MTA MUST ignore any other sub-option in Option 122 except those listed in the above table.

8.1.1 Service Provider's DHCP Address (sub-option 2)

The Service Provider's DHCP Server Address identifies the DHCP servers that a DHCP OFFER will be accepted from in order to obtain an MTA-unique IP address for a given service provider's network administrative domain.

The encoding of these sub-options is defined in RFC 3495.

Sub-option 1 MUST be included in the DHCP OFFER/ACK to the CM and it indicates the Primary DHCP server's IP address. The value contained in sub-option 1 MUST be a valid IP address, a value of 255.255.255.255 or a value of 0.0.0.0. The value contained in sub-option 2 MUST be a valid IP address.

The MTA MUST follow the logic in Table 7 when defining its DHCP strategy regardless of the Provisioning Flow used:

Table 7/J.167 – Service Provider's DHCP Address (sub-option 2)

Value of sub-option 1	Value of sub-option 2	
	Valid IP – DHCP Server is Responding	Valid IP – DHCP is not responding
Valid IP – DHCP Server is Responding	MTA MUST accept DHCP OFFERs coming only from the IP Address in the sub-option 1.	MTA MUST accept DHCP OFFERs coming only from the IP Address in the sub-option 1.
Valid IP – DHCP is NOT responding	MTA MUST try exponentially at least three times before accepting the DHCP OFFER coming from the DHCP Server pointed out by sub-option 2.	MTA MUST return to MTA1 step.
255.255.255.255	MTA MUST select the OFFERs according to the logic of RFC 2131. Value in the sub-option 2 MUST be ignored.	MTA MUST select the OFFERs according to the logic of RFC 2131. Value in the sub-option 2 MUST be ignored.
0.0.0.0	MTA MUST stop all provisioning attempts as well as all other activities.	MTA MUST stop all provisioning attempts as well as all other activities.

8.1.2 Service Provider's Provisioning Entity Address (sub-option 3)

The Service Provider's Provisioning Entity Address is the network address of the provisioning server for a given voice service provider's network administrative domain.

The encoding of this sub-option is defined in RFC 3495. This address MUST be configured as an FQDN only.

An FQDN value of 0.0.0.0 in sub-option 3 of a valid MTA DHCP OFFER/ACK specifies that the MTA MUST shutdown and not try to provision unless it is reinitialized by the CM. This is explained in step MTA2 of the provisioning flow process of 7.2.

The Service Provider's provisioning Entity Address component MUST be capable of accepting SNMP traps.

Sub-option 3 MUST be included in the DHCP OFFER to the MTA.

8.1.3 AS-REQ/REP exchange backoff and retry for SNMPv3 key management (sub-option 4)

The MTA MUST use the DHCP option 122 sub-option 4, if supplied in Secure Flow only. AS-REQ/REP exchange backoff and retry mechanism of the Kerberized SNMPv3 key negotiation defined in ITU-T Rec. J.170 is controlled by the values delivered in this sub-option or by the default values of the corresponding MIB objects in the Realm Table if this sub-option is not present in the DHCP Option 122.

The encoding of this sub-option is defined in RFC 3495.

The sub-option's nominal timeout value corresponds to the pktcMtaDevRealmUnsolicitedKeyNomTimeout MIB object in the pktcMtaDevRealmTable.

The sub-option's maximum timeout value corresponds to the pktcMtaDevRealmUnsolicitedKeyMaxTimeout MIB object in the pktcMtaDevRealmTable.

The sub-option's max retry count corresponds to the pktcMtaDevRealmUnsolicitedKeyMaxRetries MIB object in the pktcMtaDevRealmTable.

An MTA MUST be able to retrieve the above parameters from this sub-option, if they are supplied by the Provisioning Server.

Provisioning Server MAY provision an MTA with the above parameters using this sub-option.

If any of the values defined in this sub-option are "FFFFFFFF" (hexadecimal), then the default value of the corresponding column from the Realm Table MUST be used.

8.1.4 AP-REQ/REP kerberized provisioning backoff and retry (sub-option 5)

The MTA MUST use the DHCP option 122 sub-option 5, if supplied in Secure Flow only. AP-REQ/REP backoff and retry mechanism of the Kerberized SNMPv3 key negotiation defined in security Specification J.170 is controlled by the values delivered by this sub-option.

The encoding of this sub-option is defined in RFC 3495.

The sub-option's nominal timeout value corresponds to the pktcMtaDevProvUnsolicitedKeyNomTimeout MIB object.

The sub-option's maximum timeout value corresponds to the pktcMtaDevProvUnsolicitedKeyMaxTimeout MIB object.

The sub-option's max retry count corresponds to the pktcMtaDevProvUnsolicitedKeyMaxRetries MIB object.

An MTA MUST be able to retrieve the above parameters from this sub-option, if they are supplied by the Provisioning Server.

Provisioning Server MAY provision an MTA with the above parameters using this sub-option.

If any of the values defined in this sub-option are "FFFFFFFF" (hexadecimal), then the default value of the corresponding MIB Object MUST be used.

8.1.5 Kerberos realm of SNMP Entity (sub-option 6)

In conjunction with the Provisioning Entity Address, the Kerberos Realm is used as a means of contacting a SNMP Entity in the provisioning realm. The realm name is used to perform a DNS SRV lookup for the realm's KDC.

The DHCP option 122 sub-option 6 MUST be included in the DHCP OFFER to the MTA. For the Secure Flow, the DHCP option 122 sub-option 6 MUST only contain the realm name in the format of FQDN (type=0 as per RFC 3495).

The MTA MUST select the corresponding Provisioning Flow as per Table 8 (the DHCP option 122 sub-option 6 content comparison is case-sensitive and MUST be in all capital letters).

Table 8/J.167 – MTA device provisioning flow selection

Content of the DHCP option 122 sub-option 6	MTA device provisioning flow selection
BASIC.1	If the DHCP option 122 sub-option 6 value is BASIC.1, the MTA MUST execute the Basic flow without the provisioning complete SNMP INFORM.
BASIC.2	If the DHCP option 122 sub-option 6 value is BASIC.2, the MTA MUST execute the Basic flow with the provisioning complete SNMP INFORM.
HYBRID.1	If the DHCP option 122 sub-option 6 value is HYBRID.1, the MTA MUST execute the Hybrid flow without the provisioning complete SNMP INFORM.
HYBRID.2	If the DHCP option 122 sub-option 6 value is HYBRID.2, the MTA MUST execute the Hybrid flow with the provisioning complete SNMP INFORM.

The MTA MUST use the Secure Flow if any other value is provided in the DHCP option 122 sub-option 6. For Secure Flow, the encoding of the DHCP option 122 sub-option 6 is defined in RFC 3495.

8.1.5.1 SNMPv3 Key Establishment

The SNMPv3 Key Establishment is applicable for Secure Flow only. The AP Request/AP Reply described in Figure 6, the accompanying flow description and the security specification are used by the MTA in the initial provisioning phase to establish keys with the SNMPv3 USM User "MTA-Prov-xx:xx:xx:xx:xx:xx", where xx:xx:xx:xx:xx:xx represents the MAC address of the MTA and MUST be uppercase. The MTA MUST instantiate this user in the USM MIB described in RFC 3414, with the ability to be keyed using the IPCablecom Kerberized key management method described in the security specification. SNMPv3 authentication is required and privacy is optional. For the list of allowed SNMPv3 authentication and privacy algorithms, see ITU-T Rec. J.170.

Additionally, the usmUserSecurityName MUST be set to the string "MTA-Prov-xx:xx:xx:xx:xx:xx" (quotation marks not included), where xx:xx:xx:xx:xx:xx represents the MAC address of the MTA and MUST be uppercase. This ensures a unique usmUserSecurityName is created for each MTA.

The MTA must first obtain a service ticket for the provisioning realm as described in step MTA9. USM key management is performed over UDP, as specified in ITU-T Rec. J.170. The SNMPv3 keys are established prior to any SNMPv3 communication and therefore SNMPv3 messages MUST be authenticated at all times (with privacy being optional). The MTA MUST use the USM user created above in the initial INFORM.

8.1.6 Ticket Granting Server Usage (sub-option 7)

The MTA MUST use the DHCP option 122 sub-option 7 if supplied for the provisioning kerberized key management in Secure Flow only. This sub-option contains a Boolean, which when true, indicates that the MTA SHOULD get its TGT (ticket granting ticket).

Sub-option 7 MAY be included in the DHCP OFFER/ACK to the MTA.

The encoding of this sub-option is defined in RFC 3495.

8.1.7 Provisioning timer (sub-option 8)

Sub-option 8 defines the value to be used for the provisioning timer. Sub-option 8 MAY be included in the DHCP OFFER/ACK to the MTA.

The encoding of this sub-option is defined in RFC 3495.

8.1.8 Security ticket invalidation (sub-option 9)

Sub-option 9 contains a bit mask that directs the MTA to invalidate specific application server security tickets. Sub-option 9 MAY be included in the DHCP OFFER/ACK to the MTA. The encoding of this sub-option is defined in RFC 3594.

8.2 DHCP Option 60: Vendor client identifier

Option code 60 contains a string identifying Capabilities of the MTA. The MTA MUST send the following ASCII Coded String in DHCP Option code 60: "pktc1.0:xxxxxx", where xxxxxx MUST be an ASCII representation of the hexadecimal encoding of the MTA TLV Encoded Capabilities, as defined in clause 10.

8.3 DHCP options 12 and 15

MTA FQDN MUST be sent to the E-MTA in option 12 and option 15. Option 12 MUST contain "Host Name" part of the FQDN, and option 15 MUST contain "Domain Name" part of the FQDN.

For example, if MTA FQDN is "mta1.pclab.com", then option 12 must contain "mta1" and option 15 must contain "pclab.com".

8.4 DHCP option 6

DHCP option 6 MUST be used to provide the MTA with its list of DNS server addresses. Option 6 MUST contain at least one DNS server address. Option 6 MAY contain a secondary DNS server address. If this option contains more than two DNS servers, the MTA MUST use the first two addresses.

8.5 DHCP option 43

The MTA MUST send the DHCP option 43 in the DHCP DISCOVER and DHCP REQUEST for the Secure, Hybrid and Basic Flows.

DHCP option 43 contains the number of sub-options defined to provide the MTA device specific information to the back-office systems. The DHCP option 43 sub-options 1 through 10, 31 and 32 are specified by IPCablecom, sub-options 11-30 are reserved for the IPCable2Home Recommendations (J.19x series), sub-options 33 through 50 are reserved for IPCablecom, sub-options 51 through 127 are reserved for future standardized use, and sub-options 128 and above are reserved for vendor use. The IPCablecom DHCP option 43 sub-options MUST be present in the format of "Encapsulated vendor-specific extensions" (RFC 2132).

Table 9 contains the sub-options of the DHCP Option 43, which the MTA MUST use. The MTA MUST send all required sub-options listed in the table below unless explicitly stated otherwise. If the total number of octets in all DHCP option 43 sub-options exceeds 255 octets, the MTA MUST follow RFC 3396 to split the option into multiple smaller options.

Table 9/J.167 – DHCP option 43 syntax

MTA DHCP option 43 sub-options	Required/ Not used in option 43	Value	Description
Sub-option 1	Not Used		The request sub-option vector is a list of sub-options (within option 43) to be returned to client by the server upon reply to the request. None defined. The DHCP option 43 sub-option 1 MUST NOT be used by the MTA, and if present, it MUST be ignored by the Provisioning Server
Sub-option 2	R	<DevType>	The sub-option 2 contains the device type of the component making the DHCP request. The MTA MUST send the DHCP option 43 sub-option 2. For IPCablecom MTAs, the allowable device types are: – "EMTA" – for E-MTAs – "SMTA" – for S-MTAs
Sub-option 3	Not Used		The sub-option 3 contains a colon separated list of all components in the eDOCSIS device. It is used by the eDOCSIS eCM device. The DHCP option 43 sub-option 3 MUST NOT be sent by the MTA, and if present, it MUST be ignored by the Provisioning Server.
Sub-option 4	R	<device serial number>	The sub-option 4 contains the device serial number represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 4. The DHCP option 43 sub-option 4 value MUST be identical to the value of the pktcMtaDevSerialNumber MIB Object.
Sub-option 5	R	<Hardware version>	The sub-option 5 contains the hardware version number represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 5. The DHCP option 43 sub-option 5 MUST be identical to the value of the Hardware version number as in <Hardware version> field in the MIB II object sysDescr.
Sub-option 6	R	<Software version>	The sub-option 6 contains the software version number represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 6. The DHCP option 43 sub-option 6 value MUST be identical to the value of the pktcMtaDevSwCurrentVers MIB object.

Table 9/J.167 – DHCP option 43 syntax

MTA DHCP option 43 sub-options	Required/ Not used in option 43	Value	Description
Sub-option 7	R	<Boot ROM Version>	The sub-option 7 contains the Boot ROM Version represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 7. The DHCP option 43 sub-option 7 value MUST be identical to the <Boot ROM version> field in MIB II object sysDescr.
Sub-option 8	R	<OUI>	The sub-option 8 contains the Organizational Unique Identifier (OUI) represented as a hexadecimal-encoded 3-byte octet string. It MAY match the OUI in the MTA MAC address. The MTA MUST send the DHCP option 43 sub-option 8. If omitted, the Provisioning Server SHOULD use the MTA MAC address as the MTA OUI.
Sub-option 9	R	<Model Number>	The sub-option 9 contains the MTA Device Model Number represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 9. The DHCP option 43 sub-option 9 value MUST be identical to <Model Number> field in the MIB-II object sysDescr.
Sub-option 10	R	<Vendor Name>	The sub-option 10 contains the Vendor Name represented as an ASCII string. The MTA MUST send the DHCP option 43 sub-option 10. The DHCP option 43 sub-option 10 value MUST be identical to <Vendor Name> field in the MIB-II object sysDescr.
Sub-options 11-30			Reserved for CableHome
Sub-option 31	R	<MTA MAC Address>	The sub-option 31 contains the MTA MAC Address encoded as a 6-byte octet string. The MTA MUST send the DHCP option 43 sub-option 31. The DHCP option 43 sub-option 31 value MUST be identical to the content of the pktcMtaDevMacAddress MIB object.

Table 9/J.167 – DHCP option 43 syntax

MTA DHCP option 43 sub-options	Required/ Not used in option 43	Value	Description
Sub-option 32	R	<Correlation ID>	The sub-option 32 contains the Correlation ID number encoded as 4-byte INTEGER in the network order. The MTA MUST send the DHCP option 43 sub-option 32. The DHCP option 43 sub-option 32 value MUST be identical to the content of the pktcMtaDevCorrelationId MIB object.
Sub-options 33-50			Reserved for IPCablecom.
Sub-options 51 to 127			Reserved for CableLabs.
Sub-options 128 to 254			Reserved for vendors.

8.6 DHCP option 1

DHCP option 1 is defined in RFC 2132.

8.7 DHCP option 3

DHCP option 3 is defined in RFC 2132.

9 MTA provisionable attributes

This clause includes the list of attributes and their associated properties used in device provisioning. All of the provisionable attributes specified in this clause MAY be updated via the MTA configuration data file, or on a per-attribute basis using SNMP.

IPCablecom requires that an MTA configuration data file MUST be provided to all embedded-MTAs during the registration sequence. Endpoint voice services do not have to be enabled at the time of device initialization. MTA device level configuration data MUST be provisioned during initialization. These items are contained in 9.1.1.

The MTA configuration data URL generated by the Provisioning Application MUST be less than 255 bytes in length and cannot be NULL. Since this filename is provided to the MTA by the Provisioning Application during the registration sequence, it is not necessary to specify a file-naming convention.

9.1 MTA configuration file

This clause explains the format and contents of an MTA configuration file. This file contains a series of "type length and value" (TLV) parameters. Each TLV parameter in the configuration file describes an MTA or endpoint attribute. The configuration data file includes TLVs that have read-write, read only, and no MIB access. Unless specifically indicated, all MIB-accessible configuration file parameters MUST be defined using DOCSIS TLV type 11, the IPCablecom type 64, or IPCablecom TLV type 38. TLV-64 is an IPCablecom defined TLV where the length value is 2 bytes long instead of the 1 byte for DOCSIS TLV type 11. The TLV type 64 MUST be used when the length is greater than 254 bytes. If desired, vendor-specific information may be added to the configuration file using the vendor-specific TLV-43. This TLV has been specified by the DOCSIS specification (ITU-T Rec. J.112). Vendors MUST NOT provision vendor-specific information using TLV type 11 or 64. TLV-38 is an

IPCablecom defined TLV, analogous to TLV-38 used by DOCSIS and IPCable2Home. The MTA MUST be able to process the TLVs given in Table 10:

Table 10/J.167 – MTA configuration file

Type	Length	Value
11	n, where n is 1 byte	variable binding
64	m, where m is 2 bytes	variable binding
38	n, where n is 1 byte	Composite (Contains sub TLVs)
254	1 byte	0xFE for beginning of the file and 0xFF for the end of the file
NOTE – The use of TLV type 11 rather than TLV-64 is recommended wherever possible.		

In the future, new TLVs introduced in IPCablecom must have a "length field" size of 2 bytes.

The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP Set request.

The MTA configuration file MUST start with the "telephony configuration file start" tag and MUST end with the "telephony configuration file end" tag. These tags enable the MTA TLV parameters to be distinguished from DOCSIS TLV parameters. These tags also provide deterministic indications for start and stop of the MTA configuration file.

The MTA configuration file MUST contain the attributes identified as "required" in the Device Level Configuration Data table, which appears in 9.1.1; failing which, the MTA MUST reject the configuration file and take the necessary steps as defined in 7.2 (failure of step MTA23 due to 'Configuration file error'). The MTA configuration file MAY contain any of the non-required attributes which appear in the Device Level Configuration Data table. If the configuration file does not contain required attributes, it MUST be rejected. The MTA configuration file MUST be sent to the embedded-MTA every time this device is powered on.

The Device Level Service Data MAY be sent to the MTA as part of the MTA configuration file or it MAY be sent to the MTA using SNMP. If included in the configuration file, it MUST contain all of attributes identified as 'required' in the Device Level service data, if any. The MTA configuration file MAY additionally contain any of the non-required attributes that appear in the Device Level Service Data table.

If voice services are required on the MTA on any endpoint, the following MUST be done:

1) pktcMtaDevEnabled MUST be set to TRUE;
2) Per endpoint configuration data MUST be supplied either through the MTA configuration file (during provisioning) or through endpoint provisioning (using SNMP) in the post-provisioning phase.

The Endpoint details, when included MUST contain the attributes identified as "required" in the Per-Endpoint Configuration Data table, which appears in 9.1.3. The MTA configuration file MAY contain any of the non-required attributes which appear in the Per-Endpoint Configuration Data table in 9.1.3. The Per-Endpoint Configuration Data MUST be sent to the MTA when voice communications service is activated.

It is to be noted that the Device Level Service Data and Per-Endpoint Configuration Data MAY also be sent to the MTA via incremental provisioning, using SNMP. The MTA MUST support incremental provisioning.

The MTA MUST be able to process all TLV-11 and TLV-64 values with variable bindings containing all MIB objects defined in ITU-T Rec. J.166 unless stated otherwise.

The Device Level Configuration data parameter 'pktcMtaDevEnabled' is used to actually enable or disable voice services on an MTA.

Refer to 7.6.1 for a discussion concerning synchronization of provisioning attributes with back office systems.

For the Secure and Hybrid Provisioning Flows, the MTA MUST authenticate the configuration file according to IPCablecom Security Specification J.170; the MTA MUST reject the configuration file if the configuration file authentication fails and take the necessary steps as defined in 7.2 for the Secure Flow and 7.4 for the Hybrid Flow. If the configuration file contains the MIB object 'pktcMtaDevProvConfigHash' in the Secure Flow or the Hybrid Flow, the MTA MUST ignore the value of this MIB object and proceed with further processing of the configuration file and report passWithWarnings and populate the Error OID table (pktcMtaDevErrorOidsTable).

For the Basic Flow, the Provisioning Server and the MTA MUST support the configuration file data verification process as described below:

1) When the Provisioning Server creates a new MTA Configuration File or modifies an existing one, to be served for an MTA intended to go through the Basic Flow, it MUST calculate a SHA-1 hash value of the contents of the entire MTA Configuration File including start and end markers, taken as a byte string.

2) The Provisioning Server MUST add the hash value, calculated in Step 1 to the MTA Configuration File as a TLV-11 triplet corresponding to the 'pktcMtaDevProvConfigHash' MIB Object. The Provisioning Server MUST insert the TLV-11 triplet before the Configuration file end-marker. The Provisioning Server MUST NOT change the order of the TLVs in the configuration file after the hash has been calculated. The MTA Configuration File is then made available to the MTA through the appropriate TFTP/HTTP server.

3) Upon receiving the configuration file, the MTA MUST do the following: If the MIB object 'pktcMtaDevProvConfigHash' is absent, the MTA MUST reject the configuration file and MUST report 'failOtherReason'.

If the MIB object 'pktcMtaDevProvConfigHash' is present, then the MTA MUST:

a) Calculate SHA-1 over the contents of the file without TLV-11 triplet containing the pktcMtaDevProvConfigHash' and MUST populate the calculated value into pktcMtaDevProvConfigHash MIB object. The MTA must maintain the order of the TLVs for the hash calculation to be correct.

b) If the computed hash and the value of the 'pktcMtaDevProvConfigHash' MIB object are the same, the MTA Configuration File integrity is verified and the MTA MUST accept the configuration file for further processing; otherwise, the MTA MUST reject the Configuration File and the MTA MUST report 'failOtherReason'.

The MTA must also check for errors in the configuration file. As described above, errors in any of the mandatory parameters MUST be treated as an error in the configuration file and appropriate steps taken (failure of step MTA23 due to 'Configuration file error').

If there are errors in the non-required OIDs then the MTA MUST accept the configuration file, but report the same in the status (MTA25).

If the Configuration file contains per-cms data and per-endpoint parameters related to CMSs which are not associated to endpoints, an MTA MUST NOT establish SAs till an endpoint gets associated with that particular CMS (either using SNMP or via NCS redirection).

The MTA MUST report the state of the configuration file it received in the 'Provisioning complete Inform' (step MTA25 in the provisioning process) as given below:

- If the configuration file could be parsed successfully and the MTA is able to reflect the same in its MIB, it must return: 'pass'.
- If the configuration file was in error due to incorrect values in the mandatory parameters, the MTA MUST reject the configuration file and return: 'failConfigFileError'.
- It MUST also populate 'pktcMtaDevErrorOidsTable' with the parameter containing the incorrect value and MAY also populate it with other OID errors/warnings if it parsed the file completely.
- If the configuration file had proper values for all the mandatory parameters but has errors in any of the optional parameters (this includes any vendor specific OIDs which are incorrect or not known to the MTA) it must return: 'passWithWarnings'.
- It MUST also populate 'pktcMtaDevErrorOidsTable' with a list of all the parameters which were rejected and the reason for the same. The MTA MUST also use the default values for all such parameters, unless they were overridden by some other means like DHCP, in which case it must use the overridden values.
- If the configuration file is proper, but the MTA cannot reflect the same in its MIB (For ex: Too many entries leading to memory exhaustion), it MUST accept details related to the CMSs associated with the endpoints and return: 'passWithIncompleteParsing'.
- It MUST also populate 'pktcMtaDevErrorOidsTable' with a list of all the parameters which cannot be reflected in the MIB.
- If the configuration file cannot be parsed due to an internal error it must return 'failureInternalError'. It SHOULD try to populate 'pktcMtaDevErrorOidsTable' for parameters which lead to failure.
- If the MTA cannot accept the configuration file for any other reason than the ones stated above, it must return 'failureOtherReason'. It SHOULD try to populate 'pktcMtaDevErrorOidsTable' for parameters, which lead to the failure.

The MTA Configuration File MUST contain Per-Realm Configuration Data. In the case of the Secure Provisioning Flow, per-Realm Configuration Data MUST contain at least the data for the Provisioning Realm that is identified in DHCP Option 122, sub-option 6.

In the case of the Secure Provisioning Flow, after receiving the MTA Configuration File, an MTA MUST validate the following:

- "pktcMtaDevRealmName" MIB Object of the Realm Table MUST be the same as the Realm Name supplied to the MTA in DHCP Option 122, sub-option 6.
- "pktcMtaDevRealmOrgName" MIB Object of the Realm Table MUST be the same as the "Organization Name" attribute in the Service Provider Certificate.
- Encryption and Authentication of the MTA Configuration File as per ITU-T Rec. J.170.

An MTA MUST treat any of the above validation failures as failure of the MTA23 Provisioning Flow and the MTA MUST discard the Configuration File.

If the MTA encounters a vendor-specific TLV-43 with a vendor ID that the MTA does not recognize as its own, the MTA must ignore the TLV-43 and the MTA MUST continue to process the configuration file. If the MTA detects the presence of an unrecognized TLV (TLV type other than TLV-11, TLV-43, TLV-64, TLV-38, or TLV-254), the MTA MUST ignore the TLV assuming the length field of the unrecognized TLV is 2 bytes and proceed with further processing. The MTA MUST report a provisioning state of passWithWarnings and populate the error OID table (pktcMtaDevErrorOidsTable) if it detects the presence of an unrecognized TLV. If the MTA encounters an unrecognized variable binding in a TLV-11 or TLV-64, it MUST ignore this binding, MUST report a provisioning state of passWithWarnings and populate the error OID table (pktcMtaDevErrorOidsTable). It is strongly recommended for the vendors to give serious considerations to backward compatibility issues when modifying existing or introducing new sub-TLVs for TLV-43.

The MTA MUST attempt to accept configuration file that contains valid set of per-realm and per-CMS configuration data identified in 9.1.4 and 9.1.5, even if the MTA endpoints are not associated with the CMS in the per-CMS configuration data.

IPCablecom MIB objects in MTA-MIB (ITU-T Rec. J.166), Signaling-MIB (ITU-T Rec. J.166) and Event-MIB (ITU-T Rec. J.166) of type RowStatus MUST NOT be included in the MTA configuration file. If any IPCablecom MIB objects (MTA MIB, Signaling MIB and Event MIB) of type 'RowStatus' are included in the configuration file, the MTA MUST ignore the value supplied in any RowStatus object, report a 'passWithWarnings' and populate the MIB table 'pktcMtaDevErrorOidsTable' appropriately. Regardless of the action taken by the MTA, it MUST properly populate the Error OIDs table with the Row status OID. Non IPCablecom MIB objects type Row status can be present or absent in the MTA configuration file and MTA MUST process these objects according to the corresponding RFCs for the particular MIB objects (for example SNMPv2c table).

IPCablecom MIB object pktcEnMtaDevMltplGrantsPerInterval if included in the configuration file is set to enable Multiple Grants per Interval (MGPI) functionality and if the MTA does not support this functionality, then the MTA MUST ignore the object and report 'PassWithWarnings' and populate the ErrorOidsTable.

9.1.1 Device level configuration data

Refer to the MTA MIB (ITU-T Rec. J.166) for more detailed information concerning these attributes and their default values (see Table 11).

- The MTA Manufacturer Certificate validates the MTA Device Certificate.

Table 11/J.167 – Device level configuration

Attribute	Syntax	Configuration access	SNMP access	MIB file	Object	Comments
Telephony Config File Start	Integer	W, required	None	N/A	N/A	Type Length Value 254 1 1 The MTA config file MUST start with this attribute.
Telephony Config File End	Integer	W, required	None	N/A	N/A	Type Length Value 254 1 255 This MUST be the last attribute in the MTA config file.
Telephony MTA Admin State	ENUM	W, required	R/W	MTA Device MIB	pktcMtaDev Enabled	Used to enable/disable all telephony ports on the MTA. Applies to the MTA side of the embedded-MTA or the entire stand-alone MTA. Allows blanket management of all telephony ports (external interfaces) on the device. The state of the MTA is controlled by this MIB Object. For more information about this object, refer to the MTA MIB (J.166).
Realm Organization Name	String	W, Required (Secure Provisioning Flow) W, Optional (Basic and Hybrid Provisioning Flows)	R/W	MTA Device MIB	pktcMtaDev RealmOrg Name	The value of the X.500 name organization name attribute in the subject name of the service provider certificate.
Solicited Key Timeout	Integer	W, optional	R/W	N/A	pktcMtaDev ProvSolicite dKey Timeout	This timeout applies only when the Provisioning Server initiated key management (with a Wake Up message) for SNMPv3. It is the period during which the MTA will save a nonce (inside the sequence number field) from the sent out AP Request and wait for the matching AP Reply from the Provisioning Server. Since there is a default value, this is optional.
Reset Kerberos ticket information	Integer 32	W, optional	R/W	MTA Device MIB	pktcMtaDev ResetKrb Tickets	Security Specification (J.170) allows the Kerberos tickets associated with any of the application server (Provisioning Server or CMS) to be stored in the MTA NVRAM until ticket expiry. In order to control the invalidation of the tickets stored in NVRAM, this MIB attribute is used to communicate the required action to the MTA. Upon receiving this attribute in the config file, an MTA MUST take the specified action. Refer to J.166 for more information.

9.1.2 Device level service data

Refer to the MTA MIB (ITU-T Rec. J.166), the NCS MIB (ITU-T Rec. J.166), the NCS Call Signalling specification (ITU-T Rec. J.162) and RFC 2475 for more detailed information concerning these attributes and their default values (see Table 12).

Table 12/J.167 – Device level service

Attribute	Syntax	Configu-ration access	SNMP access	MIB file	Object	pktcDevEvSyslog comments
NCS Default Call Signalling TOS	Integer	W, optional	R/W	MTA Signalling MIB	pktcSigDef CallSigTos	The default value used in the IP header for setting the TOS value for NCS call signalling.
NCS Default Media Stream TOS	Integer	W, optional	R/W	MTA Signalling MIB	pktcSigDef MediaStream Tos	The default value used in the IP header for setting the TOS value for NCS media stream packets.
MTA UDP receive port used for NCS	Integer (1025..65 535)	W, optional	R/O	MTA Signalling MIB	pktcSigDef NcsReceive UdpPort	This object contains the MTA User Datagram Protocol Receive Port that is used for NCS call signalling. This object should only be changed by the configuration file.
NCS TOS Format Selector	ENUM	W, optional	R/W	MTA Signalling MIB	pktcSigTos FormatSele ctor	The format of the default NCS signalling and media TOS values. Allowed values are "IPv4 TOS octet" or "DSCP codepoint". Refer to RFC 2475.
R0 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R0Cadence	User-defined field where each bit represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence. 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R6 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R6Cadence	User-defined bit field where each bit represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence. 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R7 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R7Cadence	User-defined bit field where each bit represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence. 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R1 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R1Cadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.

Table 12/J.167 – Device level service

Attribute	Syntax	Configu-ration access	SNMP access	MIB file	Object	pktcDevEvSyslog comments
R2 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R2Cadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R3 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R3Cadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R4 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R4Cadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
R5 cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev R5Cadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
Rg cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev RgCadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1= active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.

Table 12/J.167 – Device level service

Attribute	Syntax	Configu-ration access	SNMP access	MIB file	Object	pktcDevEvSyslog comments
Rt cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev RtCadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.
Rs cadence	Bit-field	W, optional	R/W	MTA Signalling MIB	pktcSigDev RsCadence	User-defined field where each bit (least significant bit) represents a duration of 100 ms (6 s total) 1 = active ringing, 0 = silence 64 bits are used for representation; MSB 60 bits for ring cadence. Bit 61 is used to represent repeatable (when set to ZERO) and non-repeatable (when set to ONE). Other three bits are reserved for future use, and currently set to 000.

9.1.3 Per-endpoint configuration data

Refer to the NCS MIB ITU-T (Rec. J.166), the NCS specification ITU-T Rec. J.162, the security specification ITU-T Rec. J.170 and the MTA MIB (ITU-T Rec. J.166) for more detailed information concerning these attributes and their default values (see Table 13).

- MTA sends KDC the MTA/CMS certificate, MTA's FQDN, CMS-ID. The KDC returns the MTA a "Kerberos Ticket" that says "this MTA is assigned to this CMS".
- The Telephony Service Provider Certificate validates the MTA Telephony Certificate.
- If two different endpoints share the same Kerberos Realm and same CMS FQDN, then these four attributes MUST be identical: PKINIT grace period, KDC name list, MTA telephony certificate, telephony service provider certificate.

Table 13/J.167 – Per-endpoint configuration

Attribute	Syntax	Configuration access	SNMP access	MIB file	Object	Comments
Port Admin State	ENUM	W, optional	R/W	IF-MIB (RFC 2863)	ifAdmin Status	The administrative state of the port the operator can access to either enable or disable service to the port. The administrative state can be used to disable access to the user port without de-provisioning the subscriber. Allowed values for this attribute are: up(1) or down(2). For SNMP access ifAdminStatus is found in the ifTable of MIB-II.

Table 13/J.167 – Per-endpoint configuration

Attribute	Syntax	Configuration access	SNMP access	MIB file	Object	Comments
Call Management Server Name	String	W, required	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig CallAgentId	This attribute is a string indicating the name of the CMS assigned to the endpoint. The call agent name after the character '@' MUST be a fully qualified domain name and MUST have a corresponding conceptual row in the pktcMtaDevCmsTable. DNS support is assumed to support multiple CMSs as described in the NCS spec.
Call Management Server UDP Port	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig CallAgentUd pPort	UDP port for the CMS.
Partial dial time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigP artialDialTO	Time-out value in seconds for partial dial time-out.
Critical dial time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigC riticalDialTO	Time-out value in seconds for critical dial time-out.
Busy tone time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig BusyToneTO	Time-out value in seconds for busy tone.
Dial tone time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigDial ToneTO	Time-out value in seconds for dialtone.
Message waiting time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig MessageWai tingTO	Time-out value in seconds for message waiting.
Off-hook warning time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigOff HookWarn ToneTO	Time-out value in seconds for off-hook warning.
Ringing time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigRin gingTO	Time-out value in seconds for ringing.
Ringback time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigRin gBackTO	Time-out value in seconds for ringback.
Reorder tone time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigReo rderToneTO	Time-out value in seconds for reorder tone.
Stutter dial time-out	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigStutt erDialToneTO	Time-out value in seconds for stutter dial tone.
TS Max	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfigTS Max	Contains the maximum time in seconds since the sending of the initial datagram.
Max1	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig Max1	The suspicious error threshold for each endpoint retransmission.
Max2	Integer	W	R/W	MTA Signalling MIB	pktcNcsEnd PntConfig Max2	The disconnect error threshold per endpoint retransmission.

Table 13/J.167 – Per-endpoint configuration

Attribute	Syntax	Configuration access	SNMP access	MIB file	Object	Comments
Max1 Queue Enable	Enum	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigMax1QEnable	Enables/disables the Max1 DNS query operation when Max1 expires.
Max2 Queue Enable	Enum	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigMax2QEnable	Enables/disables the Max2 DNS query operation when Max2 expires.
MWD	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigMWD	Number of seconds to wait to restart after a restart is received.
Tdinit	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigTdinit	Number of seconds to wait after a disconnect.
TDMin	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigTdmin	Minimum number of seconds to wait after a disconnect.
TDMax	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigTdmax	Maximum number of seconds to wait after a disconnect.
RTO Max	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigRtoMax	Maximum number of seconds for the retransmission timer.
RTO Init	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigRtoInit	Initial value for the retransmission timer.
Long Duration Keepalive	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigLongDurationKeepAlive	Time-out in minutes for sending long duration call notification messages.
Thist	Integer	W	R/W	MTA Signalling MIB	pktcNcsEndPntConfigThist	The time-out period in seconds before no response is declared.
Call Waiting Max Reps	Integer	W, optional	R/W	MTA Signalling MIB	pktcNcsEndPntConfigCallWaitingMaxRep	This object contains the maximum number of repetitions of the call waiting that the MTA will play from a single CMS request. A value of zero (0) will be used when the CMS invokes any play repetition.
Call Waiting Delay	Integer	W, optional	R/W	IF-MIB (RFC 2863)	pktcNcsEndPntConfigCallWaitingDelay	This object contains the delay between repetitions of the call waiting that the MTA will play from a single CMS request.

9.1.4 Per-realm configuration data

Refer to the MTA MIB (J.166) for more detailed information concerning these attributes and their default values. Refer to the security Recommendation (J.170) for more information on the use of Kerberos realms. There MUST be at least one conceptual row in the pktcMtaDevRealmTable to establish service upon completion of configuration. These configuration parameters are optional in the config file, but if included the config file MUST contain at least one Realm name to permit proper instantiation of the table. There may be more than one set of entries if multiple realms are supported.

Table 14/J.167 – Per-realm configuration data

Attribute	Syntax	Access	SNMP access	MIB file	Object	Comments
Pkinit Grace Period	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevRealmPkinitGracePeriod	For the purpose of IPSec key management with a CMS, the MTA MUST obtain a new Kerberos ticket (with a PKINIT exchange), many minutes before the old ticket expires. The minimum allowable value is 15 mins. The default is 30 mins. This parameter MAY also be used with other Kerberized applications.
TGS Grace Period	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevRealmTgsGracePeriod	When the MTA implementation uses TGS Request/TGS Reply Kerberos messages for the purpose of IPSec key management with the CMS, the MTA MUST obtain a new service ticket for the CMS (with a TGS request), many minutes before the old ticket expires. The minimum allowable value is 1 min. The default is 10 mins. This parameter MAY also be used with other Kerberized applications.
Realm Org Name	Integer	W, required	R/W	MTA Device MIB	pktcMtaDevRealmOrgName	The value of the X.500 organization name attribute in the subject name of the Service provider certificate.
Unsolicited Keying max Timeout	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevRealmUnsolicitedKeyMaxTimeout	This timeout applies only when the MTA initiated key management. The maximum timeout is the value which may not be exceeded in the exponential backoff algorithm.
Unsolicited Keying Nominal Timeout	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevRealmUnsolicitedKeyNomTimeout	This timeout applies only when the MTA initiated key management. Typically this is the average roundtrip time between the MTA and the KDC.
Unsolicited Keying Max Retries	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevRealmUnsolicitedKeyMaxRetries	This is the maximum number of retries before the MTA gives up attempting to establish a Security Association.

9.1.5 Per-CMS configuration data

Refer to the MTA MIB (J.166) for more detailed information concerning these attributes and their default values. There MUST be at least one conceptual row in the pktcDevCmsTable to establish service upon completion of configuration. These configuration parameters are optional in the config file, but if included the config file MUST identify at least one CMS and its corresponding Kerberos Realm Name. There may be more than one set of entries if multiple CMSs are supported.

As per ITU-T Rec. J.170, the IPSec signalling security must be controlled by the Operator depending on the deployment and operational conditions. As the IPSec Security Association is established between the MTA and the CMS, the IPSec enabling/disabling control should also be on per CMS basis. Enabling/Disabling of the IPSec Signalling Security MUST be defined only by the information in the MTA's Configuration File when the file is being downloaded, and the enable/disable toggling MUST be done only as a result of the MTA Reset.

For more details on the MIB Object controlling the IPSec enabling/disabling, refer to the MTA MIB (J.166).

Table 15/J.167 – Per-CMS configuration data

Attribute	Syntax	Access	SNMP access	MIB file	Object	Comments
Kerberos Realm Name	String	W, required (Note)	R/W	MTA Device MIB	pktcMtaDevCmsKerbRealmName	The name for the associated Kerberos Realm. This is the corresponding Kerberos Realm Name in the Per Realm Configuration Data.
CMS Maximum Clock Skew	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevCmsMaxClockSkew	This is the maximum allowable clock skew between the MTA and CMS.
CMS Solicited Key Timeout	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevCmsSolicitedKeyTimeout	This timeout applies only when the CMS initiated key management (with a Wake Up or Rekey message). It is the period during which the MTA will save a nonce (inside the sequence number field) from the sent out AP Request and wait for the matching AP Reply from the CMS.
Unsolicited Key Max Timeout	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevCmsUnsolicitedKeyMaxTimeout	This timeout applies only when the MTA initiated key management. The maximum timeout is the value which may not be exceeded in the exponential backoff algorithm.
Unsolicited Key Nominal Timeout	Integer	W. optional	R/W	MTA Device MIB	pktcMtaDevCmsUnsolicitedKeyNomTimeout	This timeout applies only when the MTA initiated key management. Typically this is the average roundtrip time between the MTA and the CMS.
Unsolicited Key Max Retries	Integer	W, optional	R/W	MTA Device MIB	pktcMtaDevCmsUnsolicitedKeyMaxRetries	This is the maximum number of retries before the MTA gives up attempting to establish a security association.
IPSec Control	Integer	W, optional	R/O	MTA Device MIB	pktcMtaDevCmsIpsecCtrl	IPSec Control for each CMS: controls the IPSec establishment and IPSec related Key Management.
NOTE – If any data from the Per-CMS Data Table is included in the config file, this entry MUST be included.						

9.1.6 Exclusion of MIB objects in configuration file

The following MIB objects MUST NOT be sent in the configuration file since the values of these objects can be either set only by the MTA or by DHCP options during provisioning. If an MTA receives the following MIB objects in its configuration file, the MTA MUST ignore the object and report "passWithWarnings" and populate the Error OIDs Table.

- PktcMtaDevSnmpEntity
- PktcMtaDevProvKerbRealmName
- PktcMtaDevFqdn
- PktcMtaDevSerialNumber
- PktcMtaDevMacAddress
- PktcMtaDevEndPntCount
- PktcMtaDevTypeIdentifier
- PktcEnNcsEndPntQuarantineState
- PktcEnNcsEndPntHookState
- pktcEnEndPntInfoTable
- pktcDevEventDescrEnterprise
- pktcDevEventDescrFacility
- pktcDevEventDescrText
- pktcDevEvLogIndex
- pktcDevEvLogTime
- pktcDevEvLogLevel
- pktcDevEvLogId
- pktcDevEvLogText
- pktcDevEvLogEndpointName
- pktcDevEvLogType
- pktcDevEvLogTargetInfo
- pktcDevEvLogCorrelationId
- pktcMtaDevProvConfigKey

NOTE – For Syslog entries, specifically the MIB Objects "pktcDevEvSyslogAddressType" and "pktcDevEvSyslogAddress", the MTA MUST validate the 'type' provided (or stored) with the provided (or stored) 'Syslog Address' – if they are inconsistent, it MUST ignore any such entries in the configuration file, report a 'passWithWarnings' and populate the Error OIDs Table.

10 MTA device capabilities

MTA Capabilities string is supplied to the Provisioning Server in Option code 60 (Vendor Class Identifier) – to allow the Back-Office to differentiate between MTAs during the Provisioning Process. Use of Capabilities information by the Provisioning Application is optional.

Capabilities string is encoded as an ASCII string containing the Capabilities information in Type/Length/Value (TLV) Format.

For example, the ASCII encoding of the first two TLVs (IPCablecom Version 1.0 and Number of Telephony Endpoints = 2) of an MTA would be 05nn0101020102. Note that many more TLVs are required for IPCablecom MTA, and the field "nn" will contain the length of all the TLVs. This example shows only two TLVs for simplicity.

The "value" field describes the capabilities of a particular modem, i.e., implementation dependent limits on the particular features or number of features, which the modem can support. It is composed from a number of encapsulated TLV fields. The encapsulated sub-types define the specific capabilities for the MTA. Note that the sub-type fields defined are only valid within the encapsulated capabilities configuration setting string.

Type	Length	Value
5	n	

The set of possible encapsulated fields is described below.

MTA MUST Send Capabilities String in option 60 of the DHCP DISCOVER request.

10.1 IPCablecom version

This TLV MUST be supplied in the Capabilities String.

Type	Length	Value	Comment	Default value
5.1	1	0	PacketCable 1.0	NONE
		1	PacketCable 1.5	

10.2 Number of telephony endpoints

This TLV of sub-type 5.2 (Number of telephony Endpoints) MUST be supplied in the Capabilities String.

Type	Length	Value	Comment	Default
5.2	1	n	Number of endpoints	NONE

10.3 TGT support

Type	Length	Value	Comment	Default value
5.3	1	0	0: No	0
		1	1: Yes	

10.4 HTTP download file access method support

Type	Length	Value	Comment	Default value
5.4	1	0	0: No	0
		1	1: Yes	

10.5 MTA24 event SYSLOG notification support

Type	Length	Value	Comment	Default value
5.5	1	0	0: No	1
		1	1: Yes	

10.6 NCS service flow support

Type	Length	Value	Comment	Default value
5.6	1	Undefined	Reserved	Undefined

Sub-type 5.6, which was previously used to indicate support for NCS Service Flow functionality, is currently undefined and reserved for future usage.

10.7 Primary line support

Type	Length	Value	Comment	Default value
5.7	1	0	0: No	0
		1	1: Yes	

10.8 Vendor specific TLV Type(s)

This TLV can be supplied in the Capabilities String if an MTA requires a specific processing of the Vendor Specific TLV Type(s).

Type	Length	Value	Comment	Default value
5.8	n	{seq-of-bytes}	One type per byte	43

Sub-type 5.8 which was previously used to indicate vendor specific TLV support by MTAs is currently obsolete, and the sub-type (5.8) is reserved for future usage. This MUST not be used by MTAs.

10.9 NVRAM ticket/Ticket information storage support

Type	Length	Value	Comment	Default value
5.9	1	0	0: No	1
		1	1: Yes	

10.10 Provisioning event reporting support

Type	Length	Value	Comment	Default value
5.10	1	0	0: No	1
		1	1: Yes	

10.11 Supported CODEC(s)

This TLV MUST be supplied in the Capabilities String.

Type	Length	Value	Comment	Default value
5.11	n	{seq-of-bytes}	One ID per byte	NONE

CODEC ID is the value represented by the Enumerated Type of "PktcCodecType" TEXTUAL CONVENTION in MTA MIB:

1: other;

2: unknown;

3: G.729;

4: reserved;

5: G.729E;

6: PCMU;

7: G.726-32;

8: G.728;

9: PCMA;

10: G.726-16;

11: G.726-24;

12: G.726-40;

13:	iLBC;
14:	BV16;
15:	telephone-event.

Telephone-event represents RFC 2833 DTMF events. For more information, refer to ITU-T Rec. J.161.

10.12 Silence suppression support

Type	Length	Value	Comment	Default value
5.12	1	0	0: No	0
		1	1: Yes	

10.13 Echo cancellation support

Type	Length	Value	Comment	Default value
5.13	1	0	0: No	0
		1	1: Yes	

10.14 RSVP support

Type	Length	Value	Comment	Default value
5.14	1	Undefined	Reserved for future Usage	Undefined

Sub-Type 5.14 which was previously used to indicate RSVP support is currently undefined and reserved for future usage.

10.15 UGS-AD support

Type	Length	Value	Comment	Default value
5.15	1	0	0: No	0
		1	1: Yes	

10.16 MTA's "ifIndex" starting number in "ifTable"

This TLV contains the value of the "ifIndex" for the first MTA Telephony Interface in "ifTable" MIB Table. The TLV MUST be supplied in the Capabilities String.

Type	Length	Value	Comment	Default value
5.16	1	n	first MTA Interface	9

10.17 Provisioning flow logging support

This capability is set to a corresponding value depending on the support of the logging capability of the Provisioning Flow (as per 5.4.3).

Type	Length	Value	Comment	Default value
5.17	1	0	0: No	0
		1	1: Yes	

10.18 Supported provisioning flows

An MTA MUST include this TLV of sub-type 5.18 (Supported Provisioning flows) in the Capabilities String. This TLV indicates the provisioning flows the MTA supports (Basic, Hybrid and Secure). It contains a bitmask indicating all the provisioning flows supported by the MTA.

Type	Length	Value	Comment	Default value
5.18	2	{bit-mask}	See below	NONE

The Value field is an unsigned 16-bit integer encoded in network byte order. Each bit represents a specific provisioning flow. If a bit is set to 1, the MTA supports the corresponding flow. If a bit is set to 0 (zero), the MTA does not support the flow.

Bit assignments:

Bit 0 – Secure Flow (Full Secure Provisioning Flow)

Bit 1 – Hybrid Flow

Bit 2 – Basic Flow

The MTA MUST set all unused bits in the bitmask to 0. The MTA MUST set bit 0 in the TLV to 1 to indicate that it supports the Secure Flow. The MTA MUST set bits 1 and 2 in the TLV to indicate whether it supports the Basic and Hybrid Flows. An example: if an MTA supports Secure and Basic Provisioning Flows, the integer value of the mask is 0x0005, and the capability will be encoded in option 60 as the following sequence of octets (in HEX notation): 12 02 00 05.

To provide backward compatibility prior to the introduction of the Basic & Hybrid Flows, the absence of this TLV indicates that the MTA only supports the Secure Flow.

10.19 T38 version support

An MTA MUST include this TLV of sub-type 5.19 (T38 Version Support) in the Capabilities String. This TLV indicates the version of T.38 the MTA supports. For more details, refer to ITU-T Rec. J.161.

Type	Length	Value	Comment	Default value
5.19	1	0	0: Unsupported:	1
		1	1: Version Zero	
		2	2: Version One	
		3	3: Version Two	
		4	4: Version Three	

10.20 T38 error correction support

An MTA MUST include this TLV of sub-type 5.20 (T38 Error Correction Support) in the Capabilities String. This TLV indicates the type of error correction the MTA supports for T.38. For more details, refer to ITU-T Rec. J.161.

Type	Length	Value	Comment	Default value
5.20	1	0	0: None	1
		1	1: Redundancy	
		2	2: FEC	

If you support FEC, it means you also support Redundancy. For more information, refer to ITU-T Rec. J.161.

10.21 RFC 2833 DTMF support

An MTA MUST include this TLV of sub-type 5.21 (RFC 2833 DTMF Support) in the Capabilities String. This TLV indicates the support for RFC 2833 DTMF relay. For more details, refer to ITU-T Rec. J.161.

Type	Length	Value	Comment	Default value
5.21	1	0	0: No	1
		1	1: Yes	

10.22 Voice metrics support

An MTA MUST include this TLV of sub-type 5.22 (Voice Metrics Support) in the Capabilities String. This TLV indicates the support for voice metrics as defined in RFC 3611.

Type	Length	Value	Comment	Default value
5.22	1	0	0: No	1
		1	1: Yes	

10.23 Device MIB support

An MTA MUST include this TLV of sub-type 5.23 (Device MIB support) in the Capabilities String. This TLV indicates the various MIBs supported by the MTA.

Type	Length	Value	Comment	Default value
5.23	n	{seq-of-bytes}	MIB Support encoded as 'length-value' pairs	NONE

The 'length-value' pairs are defined as follows:

[L1] [OCTET-1] [OCTET-2][OCTET-3] …[OCTET-L1],

[L2] [OCTET-1] [OCTET-2][OCTET-3] …[OCTET-L2]

(And other Length-Value pairs as deemed appropriate)

Where:

'L1' and 'L2' denote lengths.

The first OCTET (OCTET-1) always represents the MIB issuing organization (Example, CableLabs, IETF, etc.).

The remaining OCTETS are always placed in network-byte order to form a bit string where each bit represents a particular MIB. Setting a bit (to a value of 1) indicates support for the representative MIB and unsetting a bit (to a value of 0) indicates absence of support for the representative MIB.

MTAs MUST NOT use any 'reserved assignments' unless defined by IPCablecom or assigned as 'vendor specific'.

10.23.1 Issuing organization assignments

OCTET-1 of the 'length-value' pair indicates the MIB issuing organization and the assignments are as follows:

Assignment	Organization indicator
0	CableLabs
1	IETF
2-9	*reserved*
10-63	*vendor-specific*

NOTE – The higher order two bits of OCTET-1 are reserved allowing for 64 possibilities.

10.23.2 Representing CableLabs MIBs

For CableLabs issued MIBs (OCTET-1 = 0), the bit mask is defined as follows:

Bit 0	PacketCable 1.5 MTA MIB.
Bit 1	PacketCable 1.5 Signalling MIB.
Bit 2	PacketCable 1.5 Management Event MIB.
Bit 3	PacketCable 1.5 MTA Extension MIB.
Bit 4	PacketCable 1.5 Signalling Extension MIB.
Bit 5	PacketCable 1.5 MEM Extension MIB.
Bit 6	*reserved*
Bit 7	*reserved*

Where the bits are placed as follows:

7	6	5	4	3	2	1	0

Given only one octet is used currently for the bit mask, the length for this length-value pair MUST be two (one each for the Organization Indicator and the bit mask, respectively).

10.23.3 Representing IETF MIBs

For MIBs represented as IETF RFCs (OCTET-1 = 1), the bit mask is defined as follows:

Bit 0	MTA MIB.
Bit 1	Signalling MIB.
Bit 2	Management Event MIB.
Bit 3	*reserved*
Bit 4	*reserved*
Bit 5	*reserved*
Bit 6	*reserved*
Bit 7	*reserved*

Given only one octet is used currently for the bit mask, the length for this length-value pair MUST be two (One each for the Organization Indicator and the bit mask, respectively).

Example

For an MTA that supports all defined IETF MIBs (MTA, Signalling and MEM) and all defined IPCablecom 1.5 extension MIBs (MTA extension, Signalling extension and MEM extension), this sub-option would be encoded (in Hex) as follows (Taken as a snapshot of option 60):

…	…	17	06	02	00	38	02	00	07	...	...

NOTE – As of this writing, none of the proposed IETF drafts have received the status of RFC and this reference has been used only as an example.

10.24 Multiple grants per interval support

An MTA MUST include this TLV of sub-type 5.24 (Multiple Grants Per Interval Support) in the Capabilities String. This TLV indicates the support for Multiple Grants per interval. For more details, refer to ITU-T Rec. J.163.

Type	Length	Value	Comment	Default value
5.24	1	0	0: No	0
		1	1: Yes	

11 TLV-38 SNMP notification receiver specification

This IPCablecom TLV-38 specifies one or more Network Management Stations that must receive notifications from the MTA (MTA25 or H-MTA-25 or B-MTA-25 and post-provisioning, if required). If TLV-38 and its sub-TLVs defined in this clause contain incorrect value in 'Length' field, the MTA MUST reject the configuration file and report a "failConfigFile" error. If TLV-38 contains sub-types with wrong values, then the MTA MUST follow the requirements specified below in each sub-TLV.

In addition, if the MTA encounters unknown sub-TLVs within TLV-38, it MUST:

- Assume the length field size of 1 byte for the sub-TLV;
- Ignore the sub-TLV and continue with further processing; and
- Report a provisioning state of passWithWarnings and populate Error Oid Table.

Type	Length	Value
38	N	Composite (contains sub-TLVs)

Unless specified or configured otherwise, the MTA MUST send the notifications to the default provisioning system (defined in DHCP option 122 sub-option 3).

11.1 Sub-TLVs of TLV-38

11.1.1 SNMP notification receiver IP address

This sub-TLV specifies the IP address of the notification receiver.

Type	Length	Value
38.1	4	4 bytes of an IPv4 address in network byte order

If TLV-38 is present in the configuration file and the sub-TLV 38.1 is absent, the MTA MUST ignore TLV-38 and proceed with further processing of the configuration file and MUST report a provisioning state of passWithWarnings and populate the error OID table (pktcMtaDevErrorOidsTable).

11.1.2 SNMP notification receiver UDP port number

This sub-TLV specifies the Port number on the notification receiver to receive the notifications.

Type	Length	Value
38.2	2	UDP Port Number

If TLV-38 is present and the sub-TLV 38.2 is absent, then a default value of 162 MUST be used.

11.1.3 SNMP Notification Receiver Type

This sub-TLV specifies the SNMP Notification Receiver Type; it is the type of SNMP notifications the MTA MUST send to the associated SNMP Notification Receiver.

Type	Length	Value
38.3	2	1: SNMPv1 trap in an SNMPv1 packet
		2: SNMPv2c trap in an SNMPv2c packet
		3: SNMP INFORM in an SNMPv2c packet
		4: SNMP trap in an SNMPv3 packet
		5: SNMP INFORM in a SNMPv3 packet

If TLV-38 is present in the configuration file but sub-TLV 38.3 is absent, the MTA MUST ignore the entire TLV-38 and proceed with further processing of the configuration file and MUST report passWithWarnings and populate the Error OID table (pktcMtaDevErrorOidsTable). The MTA and Provisioning Server MUST support notification type values 2 and 3, and MAY support notification type values 1, 4 or 5 from the above table. If an unsupported or invalid notification type value is received, the MTA MUST ignore the entire TLV-38 that contains this entry and MUST report passWithWarnings and populate the error OID table (pktcMtaDevErrorOidsTable). If the notification types of 4 or 5 are used in the Basic or Hybrid provisioning flows, SNMPv3 communication is assumed to be implemented as per SNMPv3 recommendations and is outside the scope of this Recommendation.

11.1.4 SNMP Notification Receiver Timeout

This sub-TLV specifies the wait time before a retry is attempted when the sender of an SNMP INFORM fails to receive an acknowledgement. Note that the number of retries is defined in sub-TLV 38.5.

Type	Length	Value
38.4	2	Time in milliseconds

If TLV-38 is present in the configuration file and the sub-TLV 38.4 is absent, the MTA MUST assume a default value of 15000 milliseconds. This corresponds to the default value of 1500 hundredths of a second defined for the snmpTargetAddrTimeout MIB object (see RFC 3413).

11.1.5 SNMP Notification Receiver Retries

This sub-TLV specifies the maximum number of times the MTA MUST retry sending an SNMP INFORM message if an acknowledgement is not received. Note that the wait time before each retry is defined by sub-TLV 38.4.

Type	Length	Value
38.5	2	Number of retries

If not present, the MTA MUST use a default value of 3. The maximum number of retries that can be specified is 255.

11.1.6 SNMP Notification Receiver Filtering Parameters

This sub-TLV specifies the filtering scheme for notifications and contains the root OID of the MIB sub-tree that defines the notifications to be sent to the Notification Receiver. The MTA MUST filter notifications being sent to the SNMP manager specified in sub-TLV 38.1 using the information provided. If this sub-TLV is not present, the MTA MUST use the default OID value for the 'iso' root.

Type	Length	Value
38.6	n	Filter OID (ASN.1 formatted Object Identifier)

The encoding of this TLV value field starts with the ASN.1 Universal type 6 (Object Identifier) followed by the ASN.1 length field and is terminated by the ASN.1 encoded object identifier component.

11.1.7 SNMPv3 Notification Receiver Security Name

This sub-TLV specifies the SNMPv3 Security Name to use when sending an SNMPv3 Notification. This sub-TLV is only being used if MTA supports sub-TLV 38.3 (Notification Receiver Type) types 4 and 5. The MTA MUST ignore this sub-TLV 38.7 if a Notification Receiver Type (sub-TLV 38.3) other than 4 or 5 is received in the configuration file.

The following requirements apply to MTAs that support Notification Receiver Type values of 4 or 5 in sub-TLV 38.3:

- If this sub-TLV 38.7 is omitted, then the SNMPv3 Notifications MUST be sent in the noAuthNoPriv security level using the security name "@mtaconfig".
- If this sub-TLV is included, the MTA verifies that the value of the Security Name exists for the MTA local authoritative SNMP engine and creates an entry to further associate with the notification receiver authoritative engine (using the security levels and keys from the existing Security Name). If the Security Name of this sub-TLV does not exist for the local engine, the entire TLV-38 MUST be ignored and the MTA MUST report passWithWarnings and populate the Error OID table (pktcMtaDevErrorOidsTable) for the entire TLV-38 and associated sub-TLVs that are ignored.

Type	Length	Value
38.7	2-26	Security Name

11.2 Mapping of TLV fields into SNMP tables

The following clauses detail the MTA configuration file TLV-38 "PacketCable SNMP Notification Receiver" mapping onto SNMP functional tables.

Upon receiving each TLV-38 value, the MTA MUST make entries to the following tables in order to cause the desired SNMP TRAP or INFORM transmission: snmpNotifyTable, snmpTargetAddrTable, snmpTargetAddrExtTable, snmpTargetParamsTable, snmpNotifyFilterProfileTable, snmpNotifyFilterTable, snmpCommunityTable, usmUserTable, vacmSecurityToGroupTable, vacmAccessTable, and vacmViewTreeFamilyTable. An MTA MUST support a minimum of ten TLV-38 elements in a configuration file.

11.2.1 Mapping of TLV fields into created SNMP table rows

The tables in this clause show how the fields from the Configuration file TLV element (the tags in angle brackets <>) are placed into the SNMP tables.

The correspondence between the tags and the sub-TLVs themselves is as shown below:

<IP Address>	TLV 38.1
<Port>	TLV 38.2
<Trap type>	TLV 38.3
<Timeout>	TLV 38.4
<Retries>	TLV 38.5
<Filter OID>	TLV 38.6
<Security Name>	TLV 38.7

The creation of rows with column values or indices containing the suffix "n" in the tables below indicates that these entries are created with the (n – 1)th TLV-38 found in the MTA configuration file.

11.2.1.1 snmpNotifyTable

If TLV-38 elements are present and irrespective of the number of elements, the MTA MUST create two rows with fixed values as described in Table 16.

Table 16/J.167 – snmpNotifyTable

snmpNotifyTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**First row**	**Second row**
Column Name (* = Part of Index)	Column Value	Column Value
* snmpNotifyName	"@mtaconfig_inform"	"@mtaconfig_trap"
snmpNotifyTag	"@mtaconfig_inform"	"@mtaconfig_trap"
snmpNotifyType	inform (2)	trap (1)
snmpNotifyStorageType	Volatile	Volatile
snmpNotifyRowStatus	active (1)	active (1)

11.2.1.2 snmpTargetAddrTable

For each TLV-38 element in the configuration file, the MTA MUST create one row according to Table 17.

Table 17/J.167 – snmpTargetAddrTable

snmpTargetAddrTable (RFC 3413, SNMP-TARGET-MIB)	**New row**
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
snmpTargetAddrTDomain	snmpUDPDomain = snmpDomains.1
snmpTargetAddrTAddress (IP Address and UDP Port of the Notification Receiver)	OCTET STRING (6) Octets 1-4: <IP Address> Octets 5-6: <Port>
snmpTargetAddrTimeout	<Timeout> from the TLV

Table 17/J.167 – snmpTargetAddrTable

snmpTargetAddrTable (RFC 3413, SNMP-TARGET-MIB)	New row
snmpTargetAddrRetryCount	<Retries> from the TLV
snmpTargetAddrTagList	If <Trap type> = 2 "@mtaconfig_trap" Else If <Trap type> = 3 "@mtaconfig_inform"
snmpTargetAddrParams	"@mtaconfig_n" (Same as snmpTargetAddrName value)
snmpTargetAddrStorageType	Volatile
snmpTargetAddrRowStatus	active (1)

11.2.1.3 snmpTargetAddrExtTable

For each TLV-38 element in the configuration file, the MTA MUST create one row according to Table 18.

Table 18/J.167 – snmpTargetAddrExtTable

snmpTargetAddrExtTable (RFC 3584, SNMP-COMMUNITY-MIB)	New row
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
snmpTargetAddrTMask	<Zero length octet string>
snmpTargetAddrMMS	0

11.2.1.4 snmpTargetParamsTable

For each TLV-38 element in the configuration file, the MTA MUST create one row according to Table 19.

Table 19/J.167 – snmpTargetParamsTable

snmpTargetParamsTable (RFC 3413, SNMP-TARGET-MIB)	New row
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
snmpTargetParamsMPModel SYNTAX: snmpMessageProcessingModel	SNMPv2c (1)
snmpTargetParamsSecurityModel SYNTAX: snmpSecurityModel	SNMPv2c (2) NOTE – The mapping of SNMP protocol types to value here is different from

Table 19/J.167 – snmpTargetParamsTable

snmpTargetParamsTable (RFC 3413, SNMP-TARGET-MIB)	**New row**
	snmpTargetParamsMPModel
snmpTargetParamsSecurityName	"@mtaconfig"
snmpTargetParamsSecurityLevel	NoAuthNoPriv
snmpTargetParamsStorageType	Volatile
snmpTargetParamsRowStatus	active (1)

11.2.1.5 snmpNotifyFilterProfileTable

For each TLV-38 element in the configuration file with non-zero value of TLV-38 sub-type 6, the MTA MUST create one row according to Table 20.

Table 20/J.167 – snmpNotifyFilterProfileTable

snmpNotifyFilterProfileTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**New row**
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
snmpNotifyFilterProfileName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
snmpNotifyFilterProfileStorageType	volatile
snmpNotifyFilterProfileRowStatus	active (1)

11.2.1.6 snmpNotifyFilterTable

For each TLV-38 element in the configuration file with non-zero value of TLV-38 sub-type 6, the MTA MUST create one row according to Table 21.

Table 21/J.167 – snmpNotifyFilterTable

snmpNotifyFilterTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**New row**
Column Name (* = Part of Index)	Column Value
* snmpNotifyFilterProfileName	"@mtaconfig_n" Where n ranges from 0 to m – 1 where m is the number of notification receiver TLV elements in the Configuration file
* snmpNotifyFilterSubtree	<Filter OID> from the TLV
snmpNotifyFilterMask	<Zero Length Octet String>
snmpNotifyFilterType	included (1)
snmpNotifyFilterStorageType	Volatile
snmpNotifyFilterRowStatus	active (1)

11.2.1.7 snmpCommunityTable

If TLV-38 elements are present and irrespective of the number of elements, the MTA MUST create one row with fixed values as described in Table 22.

Table 22/J.167 – snmpCommunityTable

snmpCommunityTable (RFC 3584, SNMP-COMMUNITY-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	"@mtaconfig"
snmpCommunityName	"public"
snmpCommunitySecurityName	"@mtaconfig"
snmpCommunityContextEngineID	<The engineID of the MTA>
snmpCommunityContextName	<Zero length octet string>
snmpCommunityTransportTag	<Zero length octet string>
snmpCommunityStorageType	Volatile
snmpCommunityStatus	active (1)

11.2.1.8 usmUserTable

The usmUserTable is defined in RFC 3414. The entries in the table specify the user name on the remote notification receiver to which notification is to be sent. Rows in usmUserTable are created in two different ways when <Notification Receiver Type> (sub-TLV 38.3) values 4 and 5 are supported by the MTA and is included in TLV-38.

- If <Security Name> (TLV-38.7) is not included, irrespective of the number of TLV-38 elements in the configuration file, the MTA MUST create one entry row with fixed values as described by the first column ("Static row") in Table 23.
- If <Security Name> (TLV-38.7) is included, the MTA MUST create additional entry rows as described by the second column ("Other Rows") in Table 23. In this case, the creation of additional rows in usmUserTable occurs each time the engine ID of a notification receiver needs to be discovered (see RFC 3414 for more details).

Table 23/J.167 – usmUserTable

usmUserTable (RFC 3414, SNMP-USER-BASED-SM-MIB)	**Static row Case 1**	**Other rows Case 2**
Column Name (* = Part of Index)	Column Value	Column Value
* usmUserEngineID	0x00, create a new row on each time the EngineID of the Authoritative Notification Receiver is discovered.	0x00, create a new row on each time the EngineID of the Authoritative Notification Receiver is discovered.
usmUserName	"@mtaconfig".	When other rows are created, this is replaced with the <Security Name> field from the TLV element.
usmUserSecurityName	"@mtaconfig"	When other rows are created, this is replaced with the <Security Name> field from the TLV element.

Table 23/J.167 – usmUserTable

usmUserTable (RFC 3414, SNMP-USER-BASED-SM-MIB)	Static row Case 1	Other rows Case 2
usmUserCloneFrom	<ignore> (zerodotZero) This row is not created by cloning mechanism	<ignore> (zerodotZero) This row is not created by cloning mechanism.
usmUserAuthProtocol	None (usmNoAuthProtocol)	When other rows are created, this is replaced with none (usmNoAuthProtocol), or MD5 (usmHMACMD5AuthProtocol), or SHA (usmHMACSHAAuthProtocol) depending on the security level of the SNMPv3 user.
usmUserAuthKeyChange	Empty	Empty
usmUserOwnAuthKeyChange	Empty	Empty
usmUserPrivProtocol	Case 1: none (usmNoPrivProtocol)	When other rows are created this is replaced with none (usmNoPrivProtocol) or DES (usmDESPrivProtocol), depending on the security level of the SNMPv3 user.
usmUserPrivKeyChange	Empty	Empty
usmUserOwnPrivKeyChange	Empty	Empty
usmUserPublic	Empty	Empty
usmUserStorageType	volatile (2)	volatile (2)
usmUserStatus	active (1)	Active (1)

11.2.1.9 vacmSecurityToGroupTable

If TLV-38 elements are present and irrespective of the number of elements, the MTA MUST create "Second Row" column and MAY create "First Row" or "Third Row" columns with fixed values as described in Table 24. MTA MUST populate "Second Row" and "Third Row" columns for Secure Provisioning Flow only.

Table 24/J.167 – vacmSecurityToGroupTable

vacmSecurityToGroupTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	First row	Second row	Third row
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmSecurityModel	SNMPV1 (1)	SNMPV2c (2)	SNMPUSM (3)
* vacmSecurityName	"@mtaconfig"	"@mtaconfig"	"@mtaconfig"
vacmGroupName	"@mtaconfigV1"	"@mtaconfigV2"	"@mtaconfigUSM"
vacmSecurityToGroupStorageType	volatile (2)	volatile (2)	volatile (2)
vacmSecurityToGroupStatus	Active (1)	active (1)	active (1)

11.2.1.10 VacmAccessTable

If TLV-38 elements are present and irrespective of the number of elements, the MTA MUST create "Second Row" column and MAY create "First Row" or "Third Row" columns with fixed values as described in Table 25. MTA MUST populate "Second Row" and "Third Row" columns for Secure Provisioning Flow only.

Table 25/J.167 – vacmAccessTable

vacmAccessTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	**First row**	**Second row**	**Third row**
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmGroupName	"@mtaconfigV1"	"@mtaconfigV2"	"@mtaconfigUSM"
* vacmAccessContextPrefix	Empty	Empty	Empty
* vacmAccessSecurityModel	SNMPv1 (1)	SNMPv2c (2)	USM (3)
* vacmAccessSecurityLevel	noAuthNoPriv (1)	noAuthNoPriv (1)	noAuthNoPriv (1)
vacmAccessContextMatch	exact (1)	exact (1)	exact (1)
vacmAccessReadViewName	Empty	Empty	Empty
vacmAccessWriteViewName	Empty	Empty	Empty
vacmAccessNotifyViewName	"@mtaconfig"	"@mtaconfig"	"@mtaconfig"
vacmAccessStorageType	volatile (2)	volatile (2)	volatile (2)
vacmAccessStatus	active (1)	active (1)	active (1)

11.2.1.11 vacmViewTreeFamilyTable

If TLV-38 elements are present and irrespective of the number of elements, the entry below as defined in Table 26 MUST be created. Note that this entry is already created at MTA initialization.

Table 26/J.167 – vacmViewTreeFamilyTable

vacmViewTreeFamilyTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	"@mtaconfig"
* vacmViewTreeFamilySubtree	1.3
vacmViewTreeFamilyMask	<Default from MIB>
vacmViewTreeFamilyType	included (1)
vacmViewTreeFamilyStorageType	Volatile
vacmViewTreeFamilyStatus	active (1)

11.3 TLV-38 and TLV-11 configuration example

This clause presents configuration examples for the generation of TLV-38 and TLV-11 for the purpose of SNMP framework configuration based on the framework model and message processing defined in RFC 3410, RFC 3411 and RFC 3412.

11.3.1 TLV-38 example

This clause is informative. The example below presents the usability of TLV-38. One of the objectives of this clause is to illustrate the usage of @mtaConfig_n. The following assumptions are made:

- MTA ignores entries with <trap type> 1 and supports <trap type> 2, 3, 4 and 5.
- MTA already via a configuration process has an entry with usmUserName and usmUserSecurityName which is 'mtaUser' and another entry set for 'superUser'. For simplification, no VACM entries associated to this profile are included.

Table 27 contains the Configuration File elements. Empty cells means use default values when applicable.

Table 27/J.167 – Example configuration file elements

Sub-TLV					
TLV-38 order in the configuration file	**TLV-38 Number 1**	**TLV-38 Number 2**	**TLV-38 Number 3**	**TLV-38 Number 4**	**TLV-38 Number 5**
SNMP Notification Receiver IP Address	10.0.5.9	10.0.5.9	10.0.4.9	10.0.4.9	10.0.8.9
SNMPv2c Notification Receiver UDP Port Number		162		57000	
SNMPv2c Notification Receiver Trap Type	2	3	1	4	5
SNMPv2c Notification Receiver Timeout	1500		2000		
SNMPv2c Notification Receiver Retries	3	1	2		
Notification Receiver Filtering Parameters	org	pktcMtaDevProv isioningStatus	mib-2	pktcMtaMib	pktcMtaDevPr ovisioning Status
Notification Receiver Security Name		notused		SuperUser	mtaUser
@mta@config_n	0	1	2	3	4

11.3.2 Content of the SNMP framework tables after processing of the above example TLV-38s

Based on the above assumptions and the contents of TLV-38 specified in previous clauses, this clause illustrates the tables the MTA should create. The MTA ignores TLV-38 number 1 (notification type = 1), therefore @mtaconfig_2 entries do not exist; the Security Name in TLV n = 2 is ignored.

Table 28/J.167 – snmpCommunityTable

Index	[@mtaconfig]
Name	"public"
SecurityName	@mtaconfig
ContextEngineID	<MTA ENGINEID>
ContextName	""
TransportTag	""
StorageType	volatile
Status	active

Table 29/J.167 – snmpTargetAddrExtTable

Index	[@mtaconfig_0]	[@mtaconfig_1]	[@mtaconfig_2]	[@mtaconfig_3]	[@mtaconfig_4]	[@mtaconfig_5]
TMask	""	""	""	""	""	""
MMS	0	0	0	0	0	0

Table 30/J.167 – usmUserTable

Index	[0x00][@ mtaconfig]	[<local-EngineID>] [mtaUser]	[<local-EngineID>] [superUser]	[0x00/<Notif-recv-EngineID>] [mtaUser]	[0x00/<Notif-recv-EngineID>] [superUser]
SecurityName	@mtaconfig	MtaUser	superUser	mtaUser	superUser
CloneFrom	ZeroDotZero	ZeroDotZero	zeroDotZero	zeroDotZero	zeroDotZero
AuthProtocol	usmNoAuthProtocol	usmNoAuthProtocol	usmHMACMD5AuthProtocol	usmNoAuthProtocol	usmHMACMD5Auth Protocol
AuthKeyChange	""	""	""	""	""
OwnAuthKeyChange	""	""	""	""	""
PrivProtocol	usmNoPrivProtocol	usmNoPrivProtocol	usmDESPrivProtocol	usmNoPrivProtocol	usmDESPrivProtocol
PrivKeyChange	""	""	""	""	""
OwnPrivKeyChange	""	""	""	""	""
Public	""	""	""	""	""
StorageType	Volatile	Volatile	Volatile	Volatile	Volatile
Status	active	active	active	active	active

Table 31/J.167 – vacmContextTable

Index
VacmContextName

Table 32/J.167 – vacmSecurityToGroupTable

Index	[1][@mtaconfig]	[2][@mtaconfig]	[3][@mtaconfig]
GroupName	@mtaconfigV1	@mtaconfigV2	@mtaconfigUSM
SecurityToGroupStorageType	Volatile	Volatile	Volatile
SecurityToGroupStatus	active	active	active

Table 33/J.167 – vacmAccessTable

Index	[@mtaconfigV1][] [1][noAuthNoPriv]	[@mtaconfigV2][] [2][noAuthNoPriv]	[@mtaconfigUSM][][3] [noAuthNoPriv]
ContextMatch	exact	exact	exact
ReadViewName	""	""	""
WriteViewName	""	""	""
NotifyViewName	@mtaconfig	@mtaconfig	@mtaconfig
StorageType	Volatile	Volatile	Volatile
Status	active	active	active

Table 34/J.167 – vacmViewTreeFamilyTable

Index	[@mtaconfig][org]
Mask	""
Type	Included
StorageType	Volatile
Status	Active

Table 35/J.167 – snmpNotifyTable

Index	[@mtaconfig_inform]	[@mtaconfig_trap]
Tag	@mtaconfig_inform	@mtaconfig_trap
Type	Inform	Trap
StorageType	Volatile	Volatile
RowStatus	Active	Active

Table 36/J.167 – snmpTargetAddrTable

Index	[@mtaconfig_0]	[@mtaconfig_1]	[@mtaconfig_3]	[@mtaconfig_4]
TDomain	snmpUDPDomain	snmpUDPDomain	snmpUDPDomain	snmpUDPDomain
TAddress	"0A 00 05 09 00 82"	"0A 00 05 09 00 82"	"0A 00 04 09 DE A8"	"0A 00 08 09 00 82"
Timeout	1500	1500	1500	1500
RetryCount	3	1	3	3
TagList	@mtaconfig_trap	@mtaconfig_inform	@mtaconfig_trap	@mtaconfig_inform
Params	@mtaconfig_0	@mtaconfig_1	@mtaconfig_3	@mtaconfig_4
StorageType	Volatile	Volatile	Volatile	Volatile
RowStatus	active	active	active	active

Table 37/J.167 – snmpTargetParamsTable

Index	[@mtaconfig_0]	[@mtaconfig_1]	[@mtaconfig_3]	[@mtaconfig_4]
MPModel	1	1	3	3
SecurityModel	2	2	3	3
SecurityName	@mtaconfig	@mtaconfig	@mtaconfig	@mtaconfig
SecurityLevel	noAuthNoPriv	noAuthNoPriv	noAuthNoPriv	NoAuthNoPriv
StorageType	Volatile	Volatile	Volatile	Volatile
RowStatus	active	active	active	active

Table 38/J.167 – snmpNotifyFilterProfileTable

Index	[@mtaconfig_0]	[@mtaconfig_1]	[@mtaconfig_3]	[@mtaconfig_4]
Name	[@mtaconfig_0]	[@mtaconfig_1]	[@mtaconfig_3]	[@mtaconfig_4]
StorType	Volatile	Volatile	Volatile	Volatile
RowStatus	active	active	active	active

Table 39/J.167 – snmpNotifyFilterTable

Index	[@mtaconfig_0] [org]	[@mtaconfig_1] [pktcMtaProvision-ingStatus]	[@mtaconfig_3] [PktcMtaMib]	[@mtaconfig_4] [pktcMtaProvision-ingStatus]
Mask	""	""	""	""
Type	Included	Included	Included	Included
StorageType	Volatile	Volatile	Volatile	Volatile
RowStatus	Active	Active	Active	Active

12 SNMPv2c management requirements

The management of an MTA device using SNMPv2c can be configured if required by an operator by setting the proper coexistence tables (using TLV-11) in the MTA configuration file or via post-provisioning management.

- In the Basic and Hybrid Flows, the MTA MUST configure the tables described in 12.1 and 12.2 after MTA4 to provide SNMPv2c read/write access to the default management system (provisioning entity provided in DHCP option 122 sub-option 3).
- In the Secure Flow, the MTA MUST configure the tables in 12.2 if the configuration file contains TLV-11 varbindings with the data of snmpCommunityTable. Additionally, in order to restrict SNMP access to the MTA in the inbound direction, the configuration file may also contain TLV-11 varbindings for snmpTargetAddrTable and/or snmpTargetAddrExtTab.

Appendix I provides an example template for operators to enable SNMPv2c management.

12.1 SNMPV2c coexistence mode tables content created by MTA after MTA4 for hybrid and basic flows

See Tables 40 to 42.

Table 40/J.167 – snmpCommunityTable Content

snmpCommunityTable (RFC 3584, SNMP-COMMUNITY-MIB)	**Read write access**
Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	"@mtaprov"
snmpCommunityName	"private"
snmpCommunitySecurityName	"@mtaprov"
snmpCommunityContextEngineID	<The engineID of the MTA>
snmpCommunityContextName	Empty
snmpCommunityTransportTag	"@mtaprovTag"
snmpCommunityStorageType	Volatile (2)
snmpCommunityStatus	active(1)

Table 41/J.167 – snmpTargetAddrTable Content

snmpTargetAddrTable (RFC 3413, SNMP-TARGET-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@mtaprov"
snmpTargetAddrTDomain	snmpUDPDomain = snmpDomains.1
snmpTargetAddrTAddress (IP Address non-Authoritative SNMP entity)	OCTET STRING (6) Octets 1-4: <IP address of SNMP Entity derived from 122.3> Octets 5-6: any 2 byte port value
snmpTargetAddrTimeout	Ignore, <use default>
snmpTargetAddrRetryCount	ignore, <use default>
snmpTargetAddrTagList	"@mtaprovTag"
snmpTargetAddrParams	"@mtaprov"
snmpTargetAddrStorageType	volatile (2)
snmpTargetAddrRowStatus	active(1)

Table 42/J.167 – snmpTargetAddrExtTable Content

snmpTargetAddrExtTable (RFC 3584, SNMP-COMMUNITY-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@mtaprov"
snmpTargetAddrTMask	FFFFFFFF:0000
snmpTargetAddrMMS	0

12.2 SNMP default entries for SNMPv2 Access

Tables 43 to 49 MUST be created by the MTA during the SNMP agent initialization to configure SNMPv2 access.

Table 43/J.167 – vacmSecurityToGroupTable default entries

vacmSecurityToGroupTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	**First row**	**Second row**	**Third row**
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmSecurityModel	SNMPv2c (2)	SNMPv2c (2)	SNMPv2c (2)
* vacmSecurityName	"@mtaprov"	"admin"	"operator"
vacmGroupName	"@mtaprov"	"admin"	"operator"
vacmSecurityToGroupStorageType	permanent (4)	permanent (4)	permanent (4)
vacmSecurityToGroupStatus	active (1)	active (1)	active (1)

Table 44/J.167 – vacmAccessTable default entries

vacmAccessTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	**First row**	**Second row**	**Third row**
Column Name (* = Part of Index)	Column Value	Column Value	Column Value
* vacmGroupName	"@mtaprov"	"admin"	"operator"
* vacmAccessContextPrefix	Empty	Empty	Empty
* vacmAccessSecurityModel	SNMPv2 (2)	SNMPv2 (2)	SNMPv2 (2)
* vacmAccessSecurityLevel	noAuthNoPriv (1)	noAuthNoPriv (1)	noAuthNoPriv (1)
vacmAccessContextMatch	exact (1)	exact (1)	exact (1)
vacmAccessReadViewName	"@mtaconfig"	"@mtaconfig"	"@mtaconfig"
vacmAccessWriteViewName	"@mtaconfig"	"@mtaconfig"	Empty
vacmAccessNotifyViewName	"@mtaconfig"	Empty	Empty
vacmAccessStorageType	permanent (4)	permanent (4)	permanent (4)
vacmAccessStatus	active (1)	active (1)	active (1)

Table 45/J.167 – vacmViewTreeFamilyTable default entry

vacmViewTreeFamilyTable (RFC 3415, SNMP-VIEW-BASED-ACM-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	@mtaconfig
vacmViewTreeFamilySubtree	1.3
vacmViewTreeFamilyMask	Empty <default from MIB>
vacmViewTreeFamilyType	included (1)
vacmViewTreeFamilyStorageType	volatile (2)
vacmViewTreeFamilyStatus	active (1)

Note that this entry is also created by default for the purpose of TLV-38 processing. It means only one default entry is needed in the MTA to define SNMPv2 management and TLV-38 configuration.

Table 46/J.167 – snmpTargetParamsTable default entry

snmpTargetParamsTable (RFC 3413, SNMP-TARGET-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@mtaprov"
snmpTargetParamsMPModel	1
snmpTargetParamsSecurityModel	2
snmpTargetParamsSecurityName	"@mtaprov"
snmpTargetParamsSecurityLevel	noAuthNoPriv
snmpTargetParamsStorageType	permanent (4)
snmpTargetParamsRowStatus	active (1)

Table 47/J.167 – snmpNotifyTable default entry

snmpNotifyTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpNotifyName	"@mtaprov"
snmpNotifyTag	"@mtaprovTag"
snmpNotifyType	inform (2)
snmpNotifyStorageType	permanent (4)
snmpNotifyRowStatus	active (1)

Table 48/J.167 – snmpNotifyFilterProfileTable default entry

snmpNotifyFilterProfileTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**First row**
Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@mtaprov"
snmpNotifyFilterProfileName	"@mtaprov"
snmpNotifyFilterProfileStorageType	permanent (4)
snmpNotifyFilterProfileRowStatus	active (1)

Table 49/J.167 – snmpNotifyFilterTable default entry

snmpNotifyFilterTable (RFC 3413, SNMP-NOTIFICATION-MIB)	**First row**	**Second row**
Column Name (* = Part of Index)	Column Value	Column Value
* snmpNotifyFilterProfileName	"@mtaprov"	"@mtaprov"
* snmpNotifyFilterSubtree	pktcMtaNotification	snmpTraps
snmpNotifyFilterMask	Empty	Empty
snmpNotifyFilterType	included (1)	included (1)
snmpNotifyFilterStorageType	permanent (4)	permanent (4)
snmpNotifyFilterRowStatus	active (1)	active (1)

13 Service interruption impact reporting and other enhanced features support

13.1 eDocsis requirements support

The IPCablecom eMTA is considered an eSAFE device under eDOCSIS and MUST adhere to relevant clauses of the eDOCSIS specification defined in ITU-T Rec. J.126. In addition to common requirements, the specification has certain requirements that are contingent upon the definition in the corresponding eSAFE specification. This clause deals with those additional requirements that are deemed required by the IPCablecom Specification for implementation.

The requirements can be grouped as:

- Impact Analysis and Reporting requirements.
- eSAFE reboot directives.

13.1.1 Impact analysis and reporting requirements

As specified in ITU-T Rec. J.126, the eCM has the ability to report 'Service Interruption Impact' for each eSAFE device, if in fact the data service was interrupted at the time of the query. This clause deals with the impact levels and the reporting mechanism. It is to be noted that the IPCablecom eMTA is typically associated with multiple services (Voice, Fax) and multiple instances of each service (on each configured endpoint) and hence the eMTA MUST report the highest possible impact across services/endpoints.

13.1.1.1 Impact analysis

A service on an endpoint is considered impacted when an endpoint is 'active' and the data service is interrupted. The 'active' condition is defined as the states offHook(3) and onHookPlusNCSActivity (2) as defined in pktcNcsEndPntHookState. (Refer to ITU-T Rec. J.126 for more information.)

13.1.1.2 Supported impact levels and reporting

In IPCablecom, any interruption to an 'active' service (even potentially) MUST be considered as 'High Impact' and everything else considered 'Low Impact'.

Thus, impacts MUST be reported by the MTA as follows:

- High Impact – If any of the endpoints associated with an MTA are 'Active', then the impact MUST be reported as 'High Impact'.
- Low Impact – If all of the endpoints associated with an MTA that are capable of providing service are not 'active', then the impact MUST be reported as 'Low Impact'.

13.2 IPCablecom extension MIB

IPCablecom extension MIB has been defined for all the new MIBs that are part of IPCablecom 1.5. For more information, see ITU-T Rec. J.166. The extensions are in the areas of MTA MIB and Signalling MIB.

13.2.1 MTA MIB Extension

The IPCablecom MTA MIB Extension is defined in ITU-T Rec. J.166. This provides the additional functionality for controlling new functionality like Multiple Grants Per Interval (MGPI) on the endpoint.

13.2.2 Signalling MIB Extension

The IPCablecom Signalling MIB Extension is defined in ITU-T Rec. J.166. This provides additional control and reporting functionality for endpoints in the areas of DTMF relay, Quarantine handling, Hookstate, etc.

13.3 Battery backup MIBs

The E-MTA is an embedded device with the Cable Modem. Since telephony is a high availability service, battery backup is very essential. In order to service and maintain the battery modules, a set of MIBs have been defined in an ITU-T draft Recommendation, J.bb. E-MTA devices that provide battery backup functionality MUST support the MIBs defined in an ITU-T draft Recommendation, J.bb.

13.4 Syslog MIBs

In order to maintain granularity for the syslog service, a set of MIBs have been defined in ITU-T Rec. J.166. These MIBs help the operator in troubleshooting the syslog service and also obtain a higher level of control over the syslog messages.

13.5 Foreign potential detection

Detecting foreign Potential is very important for providing telephony service. A MIB "pktcEnEndPntInfoTable" has been defined in ITU-T Rec. J.166 to report any such detection. E-MTA devices SHOULD implement this functionality.

Appendix I

SNMPv2c coexistence configuration example – Template for service providers

The operators can use the template defined in this appendix to enable SNMPv2c management (default entries defined in 12.2 are reused in the example). Note that service providers are not restricted to use this template. See Tables I.1 to I.3.

Table I.1/J.167 – snmpCommunityTable template for basic and hybrid flows configuration file

snmpCommunityTable (RFC 3584, SNMP-COMMUNITY-MIB)	**Read write access**	**Read only access**
Column Name (* = Part of Index)	Column Value	Column Value
* snmpCommunityIndex	"admin"	"operator" or <any>
snmpCommunityName	<SNMP Community Name>	<SNMP Community Name>
snmpCommunitySecurityName	"admin"	"operator"
snmpCommunityContextEngineID	<The engineID of the MTA>	<The engineID of the MTA>
snmpCommunityContextName	Empty	Empty
snmpCommunityTransportTag	"adminTag"	"operatorTag"
snmpCommunityStorageType	volatile (2)	volatile (2)
snmpCommunityStatus	createAndGo (4)	createAndGo (4)

Table I.2/J.167 – snmpTargetAddrTable template for basic and hybrid flows configuration file

snmpTargetAddrTable (RFC 3413 – SNMP-TARGET-MIB)	**First row**	**Second row**
Column Name (* = Part of Index)	Column Value	Column Value
* snmpTargetAddrName	"admin"	"operator"
snmpTargetAddrTDomain	snmpUDPDomain = snmpDomains.1	snmpUDPDomain = snmpDomains.1
snmpTargetAddrTAddress (IP Address non-Authoritative SNMP entity)	OCTET STRING (6) Octets 1-4: <SNMP Mgmt Station IPv4 Address> Octets 5-6: <0x0000>	OCTET STRING (6) Octets 1-4: <SNMP Mgmt Station IPv4 Address> Octets 5-6: <0x0000>
snmpTargetAddrTimeout	Ignore, <use default>	Ignore, <use default>
snmpTargetAddrRetryCount	Ignore, <use default>	Ignore, <use default>
snmpTargetAddrTagList	"adminTag"	"operatorTag"
snmpTargetAddrParams	Empty	Empty
snmpTargetAddrStorageType	volatile (2)	volatile (2)
snmpTargetAddrRowStatus	createAndGo (4)	createAndGo (4)

Table I.3/J.167 – snmpTargetAddrExtTable template for basic and hybrid flows configuration file

snmpTargetAddrExtTable (RFC 3584, SNMP-COMMUNITY-MIB)	**First row**	**Second row**
Column Name (* = Part of Index)	Column Value	Column Value
* snmpTargetAddrName	"admin"	"operator"
snmpTargetAddrTMask	OCTET STRING (6) Octets 1-4: <SNMP Mgmt Station Sub Net Mask> Octets 5-6: <0x0000>	OCTET STRING (6) Octets 1-4: <SNMP Mgmt Station Sub Net Mask> Octets 5-6: <0x0000>
snmpTargetAddrMMS	0	0

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2006