

IPCDN
Internet Draft
Document: draft-ietf-ipcdn-pktc-signaling-
03.txt

Gordon Beacham
Motorola, Inc

Satish Kumar
Texas Instruments
Sumanth
Channabasappa
Alopa Networks
Inc
February 2004

Expires: July 2004

Network-Based Call Signaling (NCS) Signaling MIB for PacketCable and
IPCablecom Multimedia Terminal Adapters (MTAs)

Status of this Memo

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it provides a common data and format representation for PacketCable and IPCablecom compliant Multimedia Terminal Adapter devices.

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2. The set of objects are consistent with the SNMP framework and existing SNMP standards.

Table of Contents

- 1. The Internet-Standard Management Framework.....2
- 2. Terminology.....2
 - 2.1 MTA.....3
 - 2.2 Endpoint.....3
 - 2.3 L Line Package.....3
 - 2.4 E Line Package.....3
- 3. Overview.....3
 - 3.1 Structure of the MIB.....4
 - 3.2 pktcSigDevConfigObjects.....4
 - 3.3 pktcSigNotification.....6
 - 3.4 pktcSigConformance.....6
- 4. Definitions.....7
- 5. Acknowledgments.....54
- 6. Normative References.....55
- 7. Informative References.....55
- 8. Security Considerations.....57
- 9. Intellectual Property.....58
- 10. Authors' Addresses.....59
- 11. Full Copyright Statement.....59

1. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL", when used in the guidelines in this memo, are to be interpreted as described in RFC 2119 [RFC2119].

The terms "MIB module" and "information module" are used interchangeably in this memo. As used here, both terms refer to any of the three types of information modules defined in Section 3 of RFC 2578 [RFC2578].

2.1 MTA

A Multimedia Terminal Adapter is a PacketCable or IPCablecom compliant device providing telephony services over a cable or hybrid system used to deliver video signals to a community. It contains an interface to endpoints, a network interface, CODECs, and all signaling and encapsulation functions required for Voice over IP transport, call signaling, and Quality of Service signaling. A MTA can be an embedded or a standalone device. An Embedded MTA (E-MTA) is a MTA device containing an embedded DOCSIS Cable Modem. A Standalone MTA (S-MTA) is a MTA device separated from the DOCSIS Cable Modem by non-DOCSIS MAC interface (e.g., Ethernet, USB).

2.2 Endpoint

An endpoint or MTA endpoint is a standard RJ-11 telephony physical port located on the MTA and used for attaching the telephone device to the MTA.

2.3 L Line Package

The L line package refers to the core signaling functionality as defined by PacketCable and IPCablecom. An MTA provides all L package elements, however the operator determines their application.

2.4 E Line Package

The E line package refers to extensions, over and above the core L package, defined in support of international requirements. E line package elements are optional, vary from country to country, and are set by operator or regulatory requirements.

3. Overview

This MIB provides a set of objects required for Signaling PacketCable and IPCablecom compliant Multimedia Terminal Adapter (MTA) devices. MTA devices include one or more endpoints (e.g., telephone ports) which receive signaling information to establish ring cadence, and codecs used for providing telephony service.

Telephone systems are typically very complex and often have a wide distribution. It is therefore important for management systems to support MTAs from multiple vendors at the same time, including those from multiple countries. This MIB module provides objects suitable for managing signaling for MTA devices in the widest possible variety of marketplaces.

Several normative and informative references are used to help define NCS Signaling MIB objects. As a convention, in the object REFERENCE

PacketCable/IPCablecom NCS Signaling MIB

February 2004

clauses, we use the PacketCable reference. IPCablecom compliant MTA devices MUST use the equivalent IPCablecom references.

3.1 Structure of the MIB

This MIB is identified by pktcSigMib and is structured in three groups:

- Signaling information that control device and endpoint configuration objects (pktcSigMibObjects)
- Signaling Notification object, that notifies the status (pktcSigNotification)
- Signaling Conformance has mandatory signaling objects (pktcSigConformance)

Each group of objects is explained in detail.

3.2 pktcSigDevConfigObjects

pktcSigDevEchoCancellation - This object identifies the capability of echo cancellation on the device.

pktcSigDevSilenceSuppression - This object specifies if the device is capable of silence suppression (Voice Activity Detection).

pktcSigDevR0Cadence - this object specifies ring cadence 0.

pktcSigDevR6Cadence - this object specifies ring cadence 6.

pktcSigDevR7Cadence - this object specifies ring cadence 7.

pktcSigDefCallSigDscp - this object specifies the default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for call signaling.

pktcSigDefMediaStreamDscp - this object specifies the default value used in the IP header for setting the Differentiated Services Code Point (DSCP) value for media stream packets.

pktcSigCapabilityTable - this table specifies list of supported signaling types, versions and vendor extensions for MTA.

pktcSigDefNcsReceiveUdpPort - this object contains the MTA User Datagram Protocol (UDP) receive port that is being used for NCS call signaling.

pktcSigServiceClassNameUS - this object contains a string indicating the Service Class Name to create an Upstream Service Flow for NCS.

pktcSigServiceClassNameDS - this object contains a string indicating the Service Class Name to create an Downstream Service Flow for NCS.

pktcSigServiceClassNameMask - this object contains a value for the Call Signaling Network Mask.

pktcSigNcsServiceFlowState - this object contains a status value of the Call Signaling Service Flow.

pktcSigDevR1Cadence - this object specifies ring cadence 1.

pktcSigDevR2Cadence - this object specifies ring cadence 2.

pktcSigDevR3Cadence - this object specifies ring cadence 3.

pktcSigDevR4Cadence - this object specifies ring cadence 4.

pktcSigDevR5Cadence - this object specifies ring cadence 5.

pktcSigDevRgCadence - this object specifies ring cadence rg.

pktcSigDevRsCadence - this object specifies ring cadence rs.

pktcSigDevCIDMode - this international object selects various modes of caller id in common use.

pktcSigDevCIDFskAfterRing - this international object sets the delay between the end of first ringing and the transmission of caller id information.

pktcSigDevCIDFskAfterDTAS - this international object sets the delay between the dual-tone alert signal and the transmission of caller id information.

pktcSigDevCIDFskAfterRPAS - this international object sets the delay

between the ring pulse alert signal and the transmission of caller id information.

pktcSigDevCIDRingAfterFSK - this international object sets the delay between the transmission of caller id information and the first ringing pattern.

pktcSigDevCIDDTASAfterLR - this international object sets the delay between the end of a line reversal and the dual-tone alert signal.

pktcSigDevVmwiMode - this object selects various modes of visual message waiting indicator service in common use.

pktcSigDevVmwiFskAfterDTAS - this international object sets the delay between the dual-tone alert signal and the transmission of visual message waiting information.

pktcSigDevVmwiFskAfterRPAS - this international object sets the delay between the ring pulse alert signal and the transmission of visual message waiting information.

pktcSigDevVmwiDTASAfterLR - this international object sets the delay between the end of a line reversal and the dual-tone alert signal for visual message waiting information.

pktcSigDevRingCadenceTable - this international object provides a flexible structure within which to specify a variety of ring cadences.

pktcSigDevStandardRingCadence - this international object specifies the standard ring assumed by the MTA.

pktcSigDevRingSplashCadence - this international object specifies the standard splash ring assumed by the MTA.

pktcSigDevToneTable - this international object specifies a flexible structure within which to specify all of the tones used in the MTA.

pktcNcsEndPntConfigTable - this table describes the PacketCable NCS EndPoint configuration. The number of entries in this table represents the number of provisioned endpoints.

pktcSigEndPntConfigTable - this table describes the PacketCable EndPoint selected signaling type. The number of entries in this

table represents the number of provisioned endpoints.

3.3 pktcSigNotification

pktcSigNotification - this object is used for signaling notification and reserved for future use.

3.4 pktcSigConformance

pktcSigCompliances - this table has one object that has compliance statements for devices that implement Signaling on the MTA.

pktcSigGroups - this table contains group of objects for the common portion of the PacketCable NCS and Signaling MIB.

pktcInternationalGroup - this table extends this MIB Module by establishing a set of objects designed to support operations over the widest range of marketplaces as is possible.

4. Definitions

```
PKTC-IETF-SIG-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY,  
    OBJECT-TYPE,  
    Integer32,  
    Unsigned32,  
    mib-2  
        FROM SNMPv2-SMI  
    InetAddressType,  
    InetAddress,  
    InetPortNumber  
        FROM INET-ADDRESS-MIB  
    TEXTUAL-CONVENTION,  
    RowStatus,  
    TruthValue  
        FROM SNMPv2-TC  
    OBJECT-GROUP,  
    MODULE-COMPLIANCE  
        FROM SNMPv2-CONF  
    SnmpAdminString  
        FROM SNMP-FRAMEWORK-MIB  
    ifIndex
```

FROM IF-MIB
Dscp
FROM DIFFSERV-DSCP-TC;

pktcSigMib MODULE-IDENTITY

LAST-UPDATED "200402180000Z" -- February 18, 2004
ORGANIZATION "IETF IPCDN Working Group"
CONTACT-INFO

"Satish Kumar Mudugere Eswaraiah
Texas Instruments India (P) Ltd.,
Golf view, Wind Tunnel Road
Murugesh Palya
Bangalore - 560 017, INDIA
Phone: +91 80 5269451
Email: satish.kumar@ti.com

Sumanth Channabasappa
Alopa Networks Inc
248, McCaslin Blvd #101,
Louisville, Colorado - 80027

Beacham/Kumar/Channabasappa Expires - July 2004

[Page 7]

PacketCable/IPCablecom NCS Signaling MIB

February 2004

U.S.A.
Phone: +1 303 604 6595
Email: Sumanth@alopa.com

Gordon Beacham
Motorola BCS/DCG
6450 Sequence Drive, Bldg. 1
San Diego, CA 92121, USA
Phone: +1 858-404-2335
E-Mail: gordon.beacham@motorola.com

IETF IPCDN Working Group

General Discussion: ipcdn@ietf.org
Subscribe: <http://www.ietf.org/mailman/listinfo/ipcdn>
Archive: <ftp://ftp.ietf.org/ietf-mail-archive/ipcdn>
Co-Chair: Jean-Francois Mule, jf.mule@cablelabs.com
Co-Chair: Richard Woundy, Richard_Woundy@cable.comcast.com

DESCRIPTION

"This MIB module supplies the basic management object for the PacketCable and IPCablecom Signaling protocols. This version of the MIB includes common signaling and Network Call Signaling (NCS) related signaling objects.

Copyright (C) The Internet Society (2004). This version of this MIB module is part of RFC yyyy; see the RFC itself for full legal notices."

-- RFC Ed: replace yyyy with actual RFC number and remove this note

```

REVISION                "200402180000Z"

DESCRIPTION
    "Initial version, published as RFC yyyy."
-- RFC Ed: replace yyyy with actual RFC number and remove this note

 ::= { mib-2 XXX }
-- RFC Ed: replace XXX with IANA-assigned number and remove this
-- note

-- Textual Conventions

TenthdBm ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d-1"
    STATUS      current
    DESCRIPTION
        "This data type represents power levels that are normally
        expressed in dBm. Units are in tenths of a dBm;
        for example, -13.5 dBm will be represented as -135."

```

```

SYNTAX      Integer32

PktcCodecType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        " Textual Convention defines various types of CODECs that
        MAY be supported. The list of CODECs MUST be consistent
        with the Codec RTP MAP Parameters Table in the PacketCable
        CODEC specification. In-line embedded comments below
        contain the Literal Codec Name for each CODEC. The Literal
        Codec Name is the second column of the table with CODEC
        RTP Map Parameters. Literal Codec Name Column contains the
        CODEC name used in LCD of the NCS messages CRCX/MDCX and
        is also used to identify the CODEC in the CMS
        Provisioning Specification. RTP Map Parameter Column of
        the Table contains the string used in the media attribute
        line (a=) of the SDP parameters in NCS messages."
    SYNTAX INTEGER {
        other      (1),

```

```

        unknown (2),
        g729 (3), -- G729
        reserved (4), -- reserved for future use
        g729E (5), -- G729E
        pcmu (6), -- PCMU
        g726at32 (7), -- G726-32
        g728 (8), -- G728
        pcma (9), -- PCMA
        g726at16 (10), -- G726-16
        g726at24 (11), -- G726-24
        g726at40 (12) -- G726-40
    }

```

PktcRingCadence ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

" This object represents a ring cadence and repeatable characteristics in bit string format. The ring cadence representation starts with the first 1 in the pattern (the leading 0s in the MSB are padding and are to be ignored). Each bit represents 100ms of tone; 1 is tone, 0 is no tone. 64 bits MUST be used for cadence representation. The LSB 4 bits of this object are used for representing repeatable characteristics. 0000 means repeatable, and 1000 means non repeatable. During SNMP SET operations, 64 bits MUST be used, otherwise the MTA MUST reject the value. As an example, the hex representation of a ring cadence of 0.5 secs on; 4 secs off; repeatable would be: 0x0001F0000000000000."

SYNTAX BITS {

```

        interval1 (0),
        interval2 (1),
        interval3 (2),
        interval4 (3),
        interval5 (4),
        interval6 (5),
        interval7 (6),
        interval8 (7),
        interval9 (8),
        interval10 (9),
        interval11 (10),
        interval12 (11),
        interval13 (12),
        interval14 (13),

```

interval15 (14),
interval16 (15),
interval17 (16),
interval18 (17),
interval19 (18),
interval20 (19),
interval21 (20),
interval22 (21),
interval23 (22),
interval24 (23),
interval25 (24),
interval26 (25),
interval27 (26),
interval28 (27),
interval29 (28),
interval30 (29),
interval31 (30),
interval32 (31),
interval33 (32),
interval34 (33),
interval35 (34),
interval36 (35),
interval37 (36),
interval38 (37),
interval39 (38),
interval40 (39),
interval41 (40),
interval42 (41),
interval43 (42),
interval44 (43),
interval45 (44),
interval46 (45),
interval47 (46),
interval48 (47),
interval49 (48),

interval50 (49),
interval51 (50),
interval52 (51),
interval53 (52),
interval54 (53),
interval55 (54),
interval56 (55),
interval57 (56),
interval58 (57),

```

        interval59 (58),
        interval60 (59),
        repeat1    (60),
        repeat2    (61),
        repeat3    (62),
        repeat4    (63)
    }

PktcSigType ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        " This object lists the various types of signaling that may
        be supported.
        ncs - Network call signaling is a derivation of MGCP
        (Media Gateway Control Protocol) defined for
        IPCablecom/PacketCable MTAs. "
    SYNTAX INTEGER {
        other(1),
        reserved(2), -- reserved for future use
        ncs(3)
    }

pktcSigMibObjects OBJECT IDENTIFIER ::= { pktcSigMib 1 }
pktcSigDevConfigObjects OBJECT IDENTIFIER ::=
    { pktcSigMibObjects 1 }
pktcNcsEndPntConfigObjects OBJECT IDENTIFIER ::=
    { pktcSigMibObjects 2 }
pktcSigEndPntConfigObjects OBJECT IDENTIFIER ::=
    { pktcSigMibObjects 3 }
--
-- The codec table (pktcSigDevCodecTable) defines all combinations
-- of codecs supported by the Multimedia Terminal Adapter (MTA).
--
pktcSigDevCodecTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PktcSigDevCodecEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        " This table describes the MTA supported codec types. An MTA
        MUST populate this table with all possible combinations of

```

codecs it supports for simultaneous operation. For example, an MTA with two endpoints may be designed with a particular DSP and memory architecture that allows it to support the following fixed combinations of codecs for

simultaneous operation:

Codec Type	Maximum Number of Simultaneous Codecs
PCMA	3
PCMA	2
PCMU	1
PCMA	1
PCMU	2
PCMU	3
PCMA	1
G729	1
G729	2
PCMU	1
G729	1

Based on this example, the entries in the codec table would be:

CodecComboIndex	pktcSigDevCodecType	pktcSigDevCodecMax
1	pcma	3
2	pcma	2
2	pcmu	1
3	pcma	1
3	pcmu	2
4	pcmu	3
5	pcma	1
5	g729	1
6	g729	2
7	pcmu	1
7	g729	1

An operator querying this table is able to determine all Possible codec combinations the MTA is capable of simultaneously supporting."

```
::= { pktcSigDevConfigObjects 1 }
```

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Each entry represents the maximum number of active
    connections with a particular codec the MTA is capable of
    supporting. Each row is indexed by a composite key
    consisting of a number enumerating the particular codec
    combination and the codec type."
INDEX { pktcSigDevCodecComboIndex, pktcSigDevCodeType }
 ::= { pktcSigDevCodecTable 1 }

PktcSigDevCodecEntry ::= SEQUENCE {
    pktcSigDevCodecComboIndex Unsigned32,
    pktcSigDevCodeType PktcCodecType,
    pktcSigDevCodecMax Unsigned32
}

pktcSigDevCodecComboIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..255)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        " The index value which enumerates a particular codec
        combination in the pktcSigDevCodecTable."
    ::= { pktcSigDevCodecEntry 1 }

pktcSigDevCodeType OBJECT-TYPE
    SYNTAX PktcCodecType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " A codec type supported by this MTA."
    ::= { pktcSigDevCodecEntry 2 }

pktcSigDevCodecMax OBJECT-TYPE
    SYNTAX Unsigned32(1..255)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " The maximum number of simultaneous sessions of a
        particular codec that the MTA can support."
    ::= { pktcSigDevCodecEntry 3 }

--
-- These are the common signaling related definitions that affect

```

```

-- the entire MTA device.
--

pktcSigDevEchoCancellation OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object specifies if the device is capable of echo
          cancellation."
    ::= { pktcSigDevConfigObjects 2 }

pktcSigDevSilenceSuppression OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        " This object specifies if the device is capable of
          silence suppression (Voice Activity Detection)."
    ::= { pktcSigDevConfigObjects 3 }

pktcSigDevCallerIdSigProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
                    fsk (1),
                    dtmf (2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object identifies the subscriber line protocol used
          for signaling on-hook caller id information."
    REFERENCE
        "EN 300 659-1 Specification"
    DEFVAL { fsk }
    ::= { pktcSigDevConfigObjects 4 }

--
-- In the United States Ring Cadences 0, 6, and 7 are custom ring
-- cadences definable by the system administrator. The following
-- three objects are used for these definitions.
--

pktcSigDevR0Cadence OBJECT-TYPE
    SYNTAX      PkctcRingCadence
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This object specifies ring cadence 0 (a user defined
          field) where each bit represents a duration of 100

```

```

        milliseconds (6 seconds total), except the LSB 4 bits
        which are used to represent repeatable characteristics."
    DEFVAL {{ interval1, interval2, interval3, interval4, interval5,
              interval6, interval7, interval8, interval9,
              interval10, interval11, interval12, interval13,
              interval14, interval15, interval16, interval17,
              interval18, interval19, interval20 }}
    -- 111111111111111111111000000000000000000000000000000000000000000000000000
    -- 00000
    ::= { pktcSigDevConfigObjects 5 }

```

```

pktcSigDevR6Cadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence 6 (a user defined
          field) where each bit represents a duration of 100
          milliseconds (6 seconds total), except the LSB 4 bits
          which are used to represent repeatable characteristics."
    DEFVAL {{ interval1, interval2, interval3, interval4, interval5,
              interval6, interval7, interval8, interval9,
              interval10, interval11, interval12, interval13,
              interval14, interval15, interval16, interval17,
              interval18, interval19, interval20 }}
    -- 111111111111111111111000000000000000000000000000000000000000000000000000
    -- 00000
    ::= { pktcSigDevConfigObjects 6 }

```

```

pktcSigDevR7Cadence      OBJECT-TYPE
    SYNTAX                PktcRingCadence
    MAX-ACCESS             read-write
    STATUS                 current
    DESCRIPTION
        " This object specifies ring cadence 7 (a user defined
          field) where each bit represents a duration of 100
          milliseconds (6 seconds total), except the LSB 4 bits
          which are used to represent repeatable characteristics."
    DEFVAL {{ interval1, interval2, interval3, interval4, interval5,
              interval6, interval7, interval8, interval9,
              interval10, interval11, interval12, interval13,
              interval14, interval15, interval16, interval17,
              interval18, interval19, interval20 }}
    -- 111111111111111111111000000000000000000000000000000000000000000000000000
    -- 00000
    ::= { pktcSigDevConfigObjects 7 }

```

```

pktcSigDefCallSigDscp    OBJECT-TYPE
    SYNTAX                Dscp -- RFC 3289: DIFFSERV-DSCP-TC

```

```
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    " The default value used in the IP header for setting the
      Differentiated Services Code Point (DSCP) value for call
      signaling."
DEFVAL { 0 }
 ::= { pktcSigDevConfigObjects 8 }

pktcSigDefMediaStreamDscp OBJECT-TYPE
SYNTAX Dscp -- RFC 3289: DIFFSERV-DSCP-TC
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    " The default value used in the IP header for setting
      the Differentiated Services Code Point (DSCP) value for
      media stream packets."
DEFVAL { 0 }
 ::= { pktcSigDevConfigObjects 9 }

--
-- pktcSigCapabilityTable - This table defines the valid signaling
-- types supported by this MTA.
--

pktcSigCapabilityTable OBJECT-TYPE
SYNTAX SEQUENCE OF PktcSigCapabilityEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    " This table describes the signaling types supported by this
      MTA."
 ::= { pktcSigDevConfigObjects 10 }

pktcSigCapabilityEntry OBJECT-TYPE
SYNTAX PktcSigCapabilityEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    " Entries in pktcMtaDevSigCapabilityTable - List of
      supported signaling types, versions and vendor extensions
      for this MTA. Each entry in the list provides for one
      signaling type and version combination. If the device
      supports multiple versions of the same signaling type it
      will require multiple entries."
INDEX { pktcSignalingIndex }
 ::= { pktcSigCapabilityTable 1 }

PktcSigCapabilityEntry ::= SEQUENCE {
```

PacketCable/IPCablecom NCS Signaling MIB

February 2004

```

pktcSignalingIndex      Unsigned32,
pktcSignalingType       PktcSigType,
pktcSignalingVersion     SnmpAdminString,
pktcSignalingVendorExtension SnmpAdminString
}

```

```

pktcSignalingIndex      OBJECT-TYPE
    SYNTAX                Unsigned32 (1..255)
    MAX-ACCESS             not-accessible
    STATUS                 current
    DESCRIPTION
        " The index value which uniquely identifies an entry in the
          pktcSigCapabilityTable."
    ::= { pktcSigCapabilityEntry 1 }

```

```

pktcSignalingType       OBJECT-TYPE
    SYNTAX                PktcSigType
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        " This object identifies the type of signaling used. This
          value has to be associated with a single signaling
          version."
    ::= { pktcSigCapabilityEntry 2 }

```

```

pktcSignalingVersion     OBJECT-TYPE
    SYNTAX                SnmpAdminString
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        " Provides the version of the signaling type - reference
          pktcSignalingType. Examples would be 1.0 or 2.33 etc."
    ::= { pktcSigCapabilityEntry 3 }

```

```

pktcSignalingVendorExtension OBJECT-TYPE
    SYNTAX                SnmpAdminString
    MAX-ACCESS             read-only
    STATUS                 current
    DESCRIPTION
        " The vendor extension allows vendors to provide a list of
          additional capabilities, vendors can decide how to encode
          these extensions, although space separated text is
          suggested."
    ::= { pktcSigCapabilityEntry 4 }

```

```
pktcSigDefNcsReceiveUdpPort OBJECT-TYPE
    SYNTAX      InetPortNumber (1025..65535)
    MAX-ACCESS  read-only
    STATUS      current
```

Beacham/Kumar/Channabasappa Expires - July 2004

[Page 17]

PacketCable/IPCablecom NCS Signaling MIB

February 2004

DESCRIPTION

" This object contains the MTA User Datagram Protocol (UDP) receive port that is being used for NCS call signaling. This object should only be changed by the configuration file."

REFERENCE

"PacketCable NCS Specification"

DEFVAL { 2427 }

::= { pktcSigDevConfigObjects 11 }

```
pktcSigServiceClassNameUS OBJECT-TYPE
```

```
SYNTAX      SnmpAdminString (SIZE (0..15))
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

DESCRIPTION

" This object contains a string indicating the Service Class Name to create an Upstream Service Flow for NCS. If the object has an empty string value then the NCS SF is not created and the best effort primary SF is used for NCS data. If this object is set to a non-empty (non-zero length) string, the MTA MUST create the NCS SF if it does not currently exist and the pktcSigServiceClassNameMask object has a non-zero value. If this object is subsequently set to an empty (zero-length) string, the MTA MUST delete the NCS SF if it exists. Setting this object to a different value does not cause the US SF to be re-created. The string MUST contain printable ASCII characters. The length of the string does not include a terminating zero. The MTA MUST append a terminating zero when the MTA creates the service flow."

DEFVAL { "" }

::= { pktcSigDevConfigObjects 12 }

```
pktcSigServiceClassNameDS OBJECT-TYPE
```

```
SYNTAX      SnmpAdminString (SIZE (0..15))
```

```
MAX-ACCESS  read-write
```

```
STATUS      current
```

DESCRIPTION

" This object contains a string indicating the Service

Class Name to create a Downstream Service Flow for NCS. If the object has an empty string value then the NCS SF is not created and the best effort primary SF is used for NCS data. If this object is set to a non-empty (non-zero length) string, the MTA MUST create the NCS SF if it does not currently exist and the pktcSigServiceClassNameMask object has a non-zero value. If this object is subsequently set to an empty (zero-length) string, the MTA MUST delete the NCS SF if it exists. Setting this object to a different value does not cause the DS SF to

be re-created. The string MUST contain printable ASCII characters. The length of the string does not include a terminating zero. The MTA MUST append a terminating zero when the MTA creates the service flow."

```
DEFVAL { "" }
 ::= { pktcSigDevConfigObjects 13 }
```

pktcSigServiceClassNameMask OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object contains a value used for the NCS Call Signaling classifier mask. If this object is set to a zero value, the MTA MUST delete both NCS SFs. When this object is set to a non-zero value, the MTA MUST create the NCS SF for which the corresponding MIB object (pktcSigServiceClassNameUS or pktcSigServiceClassNameDS) has a non-empty value, if the NCS SF does not already exist."

```
DEFVAL { 0 }
 ::= { pktcSigDevConfigObjects 14 }
```

pktcSigNcsServiceFlowState OBJECT-TYPE

```
SYNTAX INTEGER {
    notactive (1),
    active (2),
    error (3)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object contains a status value of the Call Signaling Service Flow.


```

        f25Hz(2),
        f33Point33Hz(3),
        f50Hz(4),
        f15Hz(5),
        f16Hz(6),
        f22Hz(7),
        f23Hz(8),
        f45Hz(9)
    }
    UNITS "Hertz"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " This object must only be set via the configuration file
        during the provisioning process. The power ring frequency
        is the frequency at which the sinusoidal voltage must
        travel down the twisted pair to make terminal equipment
        ring. Different countries define different electrical
        characteristics to make terminal equipment ring."
    REFERENCE
        "EN 300 001 contains a list of frequency ranges that
        are defined for each country."
    DEFVAL { f20Hz }
    ::= { pktcSigDevConfigObjects 23 }

```

```

pktcSigPulseSignalTable OBJECT-TYPE
    SYNTAX SEQUENCE OF PktcSigPulseSignalEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION

```

```

    " The Pulse signal table defines the pulse signal operation.
    There are nine types of international pulse signals,
    with each signal having a set of provisionable parameters.
    The values of the MIB objects in this table take effect
    only if these parameters are not defined via signaling, in
    which case the latter determines the values of the
    parameters. This object is required for the E line
    package. Signals defined in this table are triggered using
    the E line package.
    Objects in this table do not persist across MTA reboots."
    REFERENCE
        "TS 101 909-4 Specification"
    ::= { pktcSigDevConfigObjects 24 }

```

```

pktcSigPulseSignalEntry    OBJECT-TYPE
    SYNTAX                  PktcSigPulseSignalEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        " This object defines the set of parameters associated with
          each particular value of pktcSigPulseSignalType. Each
          entry in the pktcSigPulseSignalTable is indexed by the
          pktcSigPulseSignalType object."
    INDEX { pktcSigPulseSignalIndex }
    ::= { pktcSigPulseSignalTable 1 }

PktcSigPulseSignalEntry ::= SEQUENCE {
    pktcSigPulseSignalType          INTEGER,
    pktcSigPulseSignalFrequency    INTEGER,
    pktcSigPulseSignalDbLevel      TenthdBm,
    pktcSigPulseSignalDuration     Unsigned32,
    pktcSigPulseSignalPulseInterval Unsigned32,
    pktcSigPulseSignalRepeatCount  Unsigned32
}

pktcSigPulseSignalType    OBJECT-TYPE
    SYNTAX                  INTEGER
                           {
                                initialRing(1),
                                pulseLoopClose(2),
                                pulseLoopOpen(3),
                                enableMeterPulse(4),
                                meterPulseBurst(5),
                                pulseNoBattery(6),
                                pulseNormalPolarity(7),
                                pulseReducedBattery(8),
                                pulseReversePolarity(9)
                           }
    MAX-ACCESS              not-accessible

```

```

STATUS                  current
DESCRIPTION
    " There are nine types of international pulse signals."
REFERENCE
    "EN 300 324-1 Specification"

 ::= { pktcSigPulseSignalEntry 1 }

pktcSigPulseSignalFrequency    OBJECT-TYPE

```

```

SYNTAX      INTEGER {
                twentyfive (1),
                twelvethousand(2),
                sixteenthousand(3)
            }
UNITS       "Hertz"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is only applicable to the initialRing,
      enableMeterPulse, and meterPulseBurst signal type. This
      object identifies the frequency of the generated signal.
      The following table defines the default values for this
      object depending on signal type:
      pktcSigPulseSignalType      Default
      initialRing                  25
      enableMeterPulse             16000
      meterPulseBurst              16000
      The value of twentyfive must only be used for the
      initialRing signal type. The values of twelvethousand and
      sixteenthousand must only be used for enableMeterPulse and
      meterPulseBurst signal types."
REFERENCE
    "EN 300 001 Specification"
    ::= { pktcSigPulseSignalEntry 2}

```

```

pktcSigPulseSignalDbLevel      OBJECT-TYPE
SYNTAX      TenthdBm (-250..152)
UNITS       "dbm"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is only applicable to the enableMeterPulse and
      meterPulseBurst signal types. This is the decibel level
      for each frequency at which tones could be generated at
      the a and b terminals (TE connection point). "
REFERENCE
    "EN 300 001 Specification"
DEFVAL { -135 }
::={pktcSigPulseSignalEntry 3 }

```

```

pktcSigPulseSignalDuration      OBJECT-TYPE
SYNTAX      Unsigned32 (0..5000)
UNITS       "Milliseconds"

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION

" This object specifies the pulse duration for each signal type. In addition, the MTA must accept the values in the incremental steps specific for each signal type. The following table defines the default values and the incremental steps for this object depending on the signal type.

pktcSigPulseSignaltype	Default (ms)	Increment (ms)
initialRing	200	50
pulseLoopClose	200	10
pulseLoopOpen	200	10
enableMeterPulse	150	10
meterPulseBurst	150	10
pulseNoBattery	200	10
pulseNormalPolarity	200	10
pulseReducedBattery	200	10
pulseReversePolarity	200	10"

REFERENCE

"EN 300 324-1 Specification"
 ::= {pktcSigPulseSignalEntry 4 }

pktcSigPulseSignalPulseInterval OBJECT-TYPE

SYNTAX Unsigned32 (0..5000)
UNITS "Milliseconds"
MAX-ACCESS read-write
STATUS current

DESCRIPTION

" This object specifies the repeat interval, or the period for each signal type. In addition, the MTA must accept the values in the incremental steps specific for each signal type. The following table defines the default values and the incremental steps for this object depending on the signal type.

pktcSigPulseSignaltype	Default (ms)	Increment (ms)
initialRing	200	50
pulseLoopClose	1000	10
pulseLoopOpen	1000	10
enableMeterPulse	1000	10
meterPulseBurst	1000	10
pulseNoBattery	1000	10
pulseNormalPolarity	1000	10
pulseReducedBattery	1000	10
pulseReversePolarity	1000	10"

REFERENCE

"EN 300 324-1 Specification"
 ::= { pktcSigPulseSignalEntry 5 }

pktcSigPulseSignalRepeatCount OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object specifies how many times to repeat a pulse. This object is not used by the enableMeterPulse signal type and as such must have a value of zero. The following table defines the default values and the valid ranges for this object depending on the signal type.

pktcSigPulseSignaltype	Default	Range
initialRing	1	1-5
pulseLoopClose	1	1-50
pulseLoopOpen	1	1-50
enableMeterPulse	0	Not Used
meterPulseBurst	1	1-50
pulseNoBattery	1	1-50
pulseNormalPolarity	1	1-50
pulseReducedBattery	1	1-50
pulseReversePolarity	1	1-50"

::={ pktcSigPulseSignalEntry 6 }

pktcSigDevCIDMode OBJECT-TYPE

SYNTAX INTEGER {
 duringRingingETS(1),
 dtAsETS(2),
 rpAsETS(3),
 lrAsETS(4)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object is used for international systems only. For on-hook Caller ID, pktcSigDevCIDMode selects the method of Caller ID. For the duringRingingETS method, the FSK containing the Caller ID information is sent between the first and second ring pattern. For the dtAsETS, rpAsETS, and lrAsETS methods, the FSK containing the Caller ID information is sent before the first ring pattern. For the dtAsETS method, the FSK is sent after the Dual Tone Alert Signal. For the rpAsETS method, the FSK is sent after a Ring Pulse. For the lrAsETS method, the Line Reversal occurs first, then the Dual Tone Alert Signal, and finally the FSK is sent."

```
DEFVAL { dtAsETS }
 ::= { pktcSigDevConfigObjects 25 }
```

```
pktcSigDevCIDFskAfterRing    OBJECT-TYPE
SYNTAX      Unsigned32 (50..2000)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is used for international systems only. This
    object specifies the delay between the end of first
    ringing pattern and the start of the transmission of the
    FSK containing the Caller ID information. It is only used
    when pktcSigDevCIDMode is duringRingingETS.
    pktcSigDevCIDMode      pktcSigDevCIDFskAfterRing
    duringringingETS      550 ms
    dtAsETS                not used
    rpAsETS                not used
    lrAsETS                not used"
REFERENCE
    "EN 300 659-1 Specification"
DEFVAL { 550 }
 ::= { pktcSigDevConfigObjects 26 }
```

```
pktcSigDevCIDFskAfterDTAS    OBJECT-TYPE
SYNTAX      Unsigned32 (45..500)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is used for international systems only. This
    object specifies the delay between the end of the Dual
    Tone Alert Signal (DT-AS) and the start of the
    transmission of the FSK containing the Caller ID
    information. This object is only used when
    pktcSigDevCIDMode is dtAsETS or lrAsETS.
    pktcSigDevCIDMode      pktcSigDevCIDFskAfterDTAS
    duringringingETS      not used
    dtAsETS                50 ms
    rpAsETS                not used
    lrAsETS                50 ms"
REFERENCE
    "EN 300 659-1 Specification"
DEFVAL { 50 }
 ::= { pktcSigDevConfigObjects 27 }
```

```
pktcSigDevCIDFskAfterRPAS    OBJECT-TYPE
SYNTAX      Unsigned32 (500..800)
UNITS       "Milliseconds"
```

```

MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    " This object is used for international systems only. This
      object specifies the delay between the end of the Ring
      Pulse Alert Signal (RP-AS) and the start of the
      transmission of the FSK containing the Caller ID
      information. This object is only used when
      pktcSigDevCIDMode is rpAsETS.
      pktcSigDevCIDMode      pktcSigDevCIDFskAfterRPAS
      duringringingETS      not used
      dtAsETS                not used
      rpAsETS                650 ms
      lrAsETS                not used"
REFERENCE
    "EN 300 659-1 Specification"
DEFVAL { 650 }
 ::= {pktcSigDevConfigObjects 28 }

```

```

pktcSigDevCIDRingAfterFSK    OBJECT-TYPE
SYNTAX      Unsigned32 (50..500)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is used for international systems only. This
      object specifies the delay between the end of the complete
      transmission of the FSK containing the Caller ID
      information and the start of the first ring pattern. It is
      only used when pktcSigDevCIDMode is dtAsETS, rpAsETS or
      lrAsETS.
      pktcSigDevCIDMode      pktcSigDevCIDFskAfterFSK
      duringringingETS      not used
      dtAsETS                250 ms
      rpAsETS                250 ms
      lrAsETS                250 ms"
REFERENCE
    "EN 300 659-1 Specification"
DEFVAL { 250 }
 ::= {pktcSigDevConfigObjects 29 }

```

```

pktcSigDevCIDDTASAfterLR    OBJECT-TYPE
SYNTAX      Unsigned32 (50..655)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This object is used for international systems only. This
      object specifies the delay between the end of the Line

```

PacketCable/IPCablecom NCS Signaling MIB

February 2004

Reversal and the start of the Dual Tone Alert Signal (DT-AS). This object is only used when pktcSigDevCIDMode is lrAsETS.

pktcSigDevCIDMode	pktcSigDevCIDFskAfterLR
duringringingETS	not used
dtAsETS	not used
rpAsETS	not used
lrAsETS	250 ms"

REFERENCE

"EN 300 659-1 Specification"

DEFVAL { 250 }

::= {pktcSigDevConfigObjects 31 }

pktcSigDevVmwMode OBJECT-TYPE

SYNTAX INTEGER {
 dtAsETS(1),
 rpAsETS(2),
 lrAsETS(3),
 osi(4)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" For visual message waiting indicator (VMWI), pktcSigDevVmwMode selects the alerting signal method. For the dtAsETS, rpAsETS, lrAsETS, and OSI methods, the FSK containing the VMWI information is sent after an alerting signal. For the dtAsETS method, the FSK is sent after the Dual Tone Alert Signal. For the rpAsETS method, the FSK is sent after a Ring Pulse. For the lrAsETS method, the Line Reversal occurs first, then the Dual Tone Alert Signal, and finally the FSK is sent. For the OSI method, the FSK is sent after the Open Switching Interval."

DEFVAL { dtAsETS }

::= {pktcSigDevConfigObjects 32 }

pktcSigDevVmwFskAfterDTAS OBJECT-TYPE

SYNTAX Unsigned32 (45..500)

UNITS "Milliseconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This object is used for international systems only. This

object specifies the delay between the end of the Dual Tone Alert Signal (DT-AS) and the start of the transmission of the FSK containing the VMWI information. This object is only used when pktcSigDevVmwiMode is

```
dtAsETS or lrAsETS.
pktcSigDevVmwiMode      pktcSigDevVmwiFskAfterDTAS
dtAsETS                  50 ms
rpAsETS                  not used
lrAsETS                  50 ms"
REFERENCE
  "EN 300 659-1 Specification"
DEFVAL { 50 }
 ::= {pktcSigDevConfigObjects 33 }
```

```
pktcSigDevVmwiFskAfterRPAS  OBJECT-TYPE
SYNTAX      Unsigned32 (500..800)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  " This object is used for international systems only. This
  object specifies the delay between the end of the Ring
  Pulse Alert Signal (RP-AS) and the start of the
  transmission of the FSK containing the VMWI information.
  This object is only used when pktcSigDevVmwiMode is
  rpAsETS.
  pktcSigDevVmwiMode      pktcSigDevVmwiFskAfterRPAS
  dtAsETS                  not used
  rpAsETS                  650 ms
  lrAsETS                  not used "
REFERENCE
  "EN 300 659-1 Specification"
DEFVAL { 650 }
 ::= {pktcSigDevConfigObjects 34 }
```

```
pktcSigDevVmwiDTASAfterLR  OBJECT-TYPE
SYNTAX      Unsigned32 (50..655)
UNITS       "Milliseconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
  " This object is used for international systems only. This
  object specifies the delay between the end of the Line
```

Reversal and the start of the Dual Tone Alert Signal (DT-AS) for VMWI information. This object is only used when pktcSigDevVmwiMode is lrAsETS.

pktcSigDevVmwiMode	pktcSigDevVmwiDTASAfterLR
dtAsETS	not used
rpAsETS	not used
lrAsETS	250 ms "

REFERENCE

"EN 300 659-1 Specification"

DEFVAL { 250 }
 ::= { pktcSigDevConfigObjects 35 }

pktcSigDevRingCadenceTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcSigDevRingCadenceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" In V5.2, Cadence rings are defined by the telco governing body for each country. The MTA must be able to support various ranges of cadence patterns and cadence periods. The MTA will be able to support country specific provisioning of the cadence and idle period. There will be at most 3 on/off transitions per cadence period. Each cadence pattern will be assigned a unique value ranging from 1-128 (inclusive) corresponding to the value of x plus one, where x is the value sent in the cr(x) signal requested per the appropriate NCS message, and defined in the E package. The MTA will derive the cadence periods from the ring Cadence table entry as provisioned by the customer. The MTA is allowed to provide appropriate default values for each of the ring cadences. This table only needs to be supported when the MTA implements the NCS E package. Objects in this table do not persist across MTA reboots."

REFERENCE

"TS 101 909-4 Specification"

::= { pktcSigDevConfigObjects 36 }

pktcSigDevRingCadenceEntry OBJECT-TYPE

SYNTAX PktcSigDevRingCadenceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" Unique value ranging from 1 to 128 that corresponds to the

value sent by the LE (plus one) based on country specific cadences, one row per cadence cycle. In any given system implementation for a particular country, it is anticipated that a small number of ring cadences will be in use. Thus, this table most likely will not be populated to its full 128-row size."

```
INDEX { pktcSigDevRingCadenceIndex }
 ::= { pktcSigDevRingCadenceTable 1 }
```

```
PktcSigDevRingCadenceEntry ::= SEQUENCE {
    pktcSigDevRingCadenceIndex    Unsigned32,
    pktcSigDevRingCadence         OCTET STRING
}
```

```
pktcSigDevRingCadenceIndex    OBJECT-TYPE
    SYNTAX      Unsigned32 (1..128)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        " The table index equivalent to the country specific cadence
          (1 - 128). "
    ::= { pktcSigDevRingCadenceEntry 1 }
```

```
pktcSigDevRingCadence        OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4..36))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the Ring Cadence Octet String. The first two
          octets of the bit string represent the length in bits of
          the duration of the cadence. The third octet is used to
          represent repeatable characteristics. 00000000 means
          repeatable, and 10000000 means non repeatable. Each Bit
          after the third octet represents 50 ms and 1 represents
          ring and 0 represents silent. The first bit of the fourth
          octet is the first bit of the ring cadence. A total of 264
          Bits can be set to represent 13200 ms of total cadence
          cycle. This object is required for the E line package. "
    ::= { pktcSigDevRingCadenceEntry 2 }
```

```
pktcSigDevStandardRingCadence OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(4..36))
    MAX-ACCESS  read-write
    STATUS      current
```

DESCRIPTION

" This is the Ring Cadence Octet String for the standard ring. The first two octets of the bit string represent the length in bits of the duration of the cadence. The third octet is used to represent repeatable characteristics. 00000000 means repeatable, and 10000000 means non repeatable. Each Bit after the third octet represents 50 ms and 1 represents ring and 0 represents silent. The first bit of the fourth octet is the first bit of the ring cadence. A total of 264 Bits can be set to represent 13200 ms of cadence cycle. The MTA MUST provide a default value for this object in accordance with published specifications for the country of operation. This object is required for the E line package."

REFERENCE

"TR 101 183 Specification"
 ::= { pktcSigDevConfigObjects 37 }

pktcSigDevRingSplashCadence OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(4..36))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

" This is the Ring Cadence Octet String for splash ring. The first two octets of the bit string represent the length in bits of the duration of the cadence. The third octet is used to represent repeatable characteristics. 00000000 means repeatable, and 10000000 means non repeatable. Each Bit after the third octet represents 50 ms and 1 represents ring and 0 represents silent. The first bit of the fourth octet is the first bit of the ring cadence. A total of 264 Bits can be set to represent 13200 ms of cadence cycle. The MTA MUST provide a default value for this object in accordance with published specifications for the country of operation. This object is required for the E line package."

REFERENCE

"TR 101 183 Specification"
 ::= { pktcSigDevConfigObjects 38 }

pktcSigDevToneTable OBJECT-TYPE

SYNTAX SEQUENCE OF PktcSigDevToneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" The Tone Table defines the various tone operations. Any definition of the tones callWaiting1-4 in this table should just contain the audible tone itself and NOT contain the delay between tones or the tone repeat count. The delay between tones or the repeat count is controlled by the MIB objects pktcNcsEndPntConfigCallWaitingDelay, and pktcNcsEndPntConfigCallWaitingMaxRep. The MTA MUST make sure that, after the provisioning cycle, the table is fully populated (i.e., for each possible index, an entry MUST be defined) using reasonable defaults for each row that was not defined by the provisioning information. Objects in this table do not persist across MTA reboots."

REFERENCE

"NCS Specification, TS 101 909-4 Specification"
 ::= { pktcSigDevConfigObjects 39 }

pktcSigDevToneEntry OBJECT-TYPE

SYNTAX PktcSigDevToneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" Unique value ranging from 1 to 21 that will correspond to the different tone types that are being supported by the

device. These tones can be provisioned based on country specific needs."

INDEX { pktcSigDevToneType }

::= { pktcSigDevToneTable 1 }

PktcSigDevToneEntry ::= SEQUENCE {

pktcSigDevToneType	INTEGER,
pktcSigDevToneDbLevel	TenthdBm,
pktcSigDevToneFreqType	INTEGER,
pktcSigDevToneNumFrequencies	Unsigned32,
pktcSigDevToneFirstFrequency	Unsigned32,
pktcSigDevToneSecondFrequency	Unsigned32,
pktcSigDevToneThirdFrequency	Unsigned32,
pktcSigDevToneFourthFrequency	Unsigned32,
pktcSigDevToneNumOnOffTimes	Unsigned32,
pktcSigDevToneFirstToneOn	Unsigned32,
pktcSigDevToneFirstToneOff	Unsigned32,
pktcSigDevToneSecondToneOn	Unsigned32,
pktcSigDevToneSecondToneOff	Unsigned32,
pktcSigDevToneThirdToneOn	Unsigned32,

```

pktcSigDevToneThirdToneOff      Unsigned32,
pktcSigDevToneFourthToneOn     Unsigned32,
pktcSigDevToneFourthToneOff    Unsigned32,
pktcSigDevToneWholeToneRepeatCount Unsigned32,
pktcSigDevToneSteady           TruthValue
}

```

```

pktcSigDevToneType      OBJECT-TYPE
SYNTAX                  INTEGER {
                        busy(1),
                        confirmation(2),
                        dial(3),
                        messageWaiting(4),
                        offHookWarning(5),
                        ringBack(6),
                        reOrder(7),
                        stutterdial(8),
                        callWaiting1(9),
                        callWaiting2(10),
                        callWaiting3(11),
                        callWaiting4(12),
                        alertingSignal(13),
                        specialDial(14),
                        specialInfo(15),
                        release(16),
                        congestion(17),
                        userDefined1(18),
                        userDefined2(19),
                        userDefined3(20),

```

```

                        userDefined4(21)
                        }
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    " This object defines the type of tone being accessed. The
      alertingSignal, specialDial, specialInfo, release,
      congestion and userDefined1-4 tone types are triggered
      using the E line package."
 ::= {pktcSigDevToneEntry 1 }

```

```

pktcSigDevToneDbLevel      OBJECT-TYPE
SYNTAX                    TENTHdBm (-600..40)
UNITS                     "dbm"
MAX-ACCESS                read-write

```

```

STATUS          current
DESCRIPTION
    " This is the decibel level for each frequency at which
      tones could be generated at the a and b terminals (TE
      connection point). "
DEFVAL { -40 }
 ::= { pktcSigDevToneEntry 2 }

pktcSigDevToneFreqType OBJECT-TYPE
SYNTAX          INTEGER {
                allFrequencies (1),
                singleFrequencySequence (2),
                dualFrequencySequence (3),
                allFrequenciesModulated (4)
                }
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    " This object describes how the frequencies are applied.
      allFrequencies indicates all frequencies specified by
      pktcSigDevToneNumFrequencies are mixed to form a single
      tone. The tone is then applied in sequence using the
      number of on/off times specified in
      pktcSigDevToneNumOnOffTimes. SingleFrequencySequence
      indicates all frequencies specified by
      pktcSigDevToneNumFrequencies are applied in sequence
      using the corresponding frequency number on/off time
      (e.g., pktcSigDevToneFirstFrequency uses
      pktcSigDevToneFirstToneOn and pktcSigDevToneFirstToneOff,
      pktcSigDevToneSecondFrequency uses
      pktcSigDevToneSecondToneOn and
      pktcSigDevToneSecondToneOff). For this tone type
      pktcSigDevToneNumFrequencies MUST equal
      pktcSigDevToneNumOnOffTimes. DualFrequencySequence

```

indicates two pairs of frequencies are added to form two sequenced tones. The first and second frequency are added to form tone one and are applied using pktcSigDevToneFirstToneOn and pktcSigDevToneFirstToneOff. The third and fourth frequency are added to form tone two and are applied using pktcSigDevToneSecondToneOn and pktcSigDevToneSecondToneOff. For this tone type pktcSigDevToneNumFrequencies MUST equal 4 and pktcSigDevToneNumOnOffTimes MUST equal 2. allFrequenciesModulated indicates all frequencies

specified by pktcSigDevToneNumFrequencies are modulated to form a single tone. The tone is then applied in sequence using the number of on/off times specified in pktcSigDevToneNumOnOffTimes. For allFrequenciesModulated, the pktcSigDevToneNumFrequencies MUST equal 2, and the frequency in pktcSigDevToneFirstFrequency modulates the frequency in pktcSigDevToneSecondFrequency."

REFERENCE

"ITU-T E.180"

DEFVAL { allFrequencies }
::={ pktcSigDevToneEntry 3 }

pktcSigDevToneNumFrequencies OBJECT-TYPE
SYNTAX Unsigned32 (1..4)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 " Specifies the number of frequencies specified in the table entry."
DEFVAL { 1 }
::={pktcSigDevToneEntry 4}

pktcSigDevToneFirstFrequency OBJECT-TYPE
SYNTAX Unsigned32 (0..4000)
UNITS "Hertz"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 " This is the first frequency at which the tones could be generated in a multiple frequency tone."
::={pktcSigDevToneEntry 5}

pktcSigDevToneSecondFrequency OBJECT-TYPE
SYNTAX Unsigned32 (0..4000)
UNITS "Hertz"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 " This is the second frequency at which the tones could be

generated in a multiple frequency tone."
::={pktcSigDevToneEntry 6}

pktcSigDevToneThirdFrequency OBJECT-TYPE
SYNTAX Unsigned32 (0..4000)

```

UNITS          "Hertz"
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    " This is the third frequency at which the tones could be
    generated."
 ::= {pktcSigDevToneEntry 7}

pktcSigDevToneFourthFrequency  OBJECT-TYPE
SYNTAX          Unsigned32 (0..4000)
UNITS           "Hertz"
MAX-ACCESS     read-write
STATUS         current
DESCRIPTION
    " This is the fourth frequency at which the tones could be
    generated."
 ::= {pktcSigDevToneEntry 8}

pktcSigDevToneNumOnOffTimes  OBJECT-TYPE
SYNTAX          Unsigned32 (1..4)
MAX-ACCESS     read-write
STATUS         current
DESCRIPTION
    " Specifies the number of on/off times specified in the
    table entry."
DEFVAL { 1 }
 ::= {pktcSigDevToneEntry 9}

pktcSigDevToneFirstToneOn    OBJECT-TYPE
SYNTAX          Unsigned32 (0..5000)
UNITS           "Milliseconds"
MAX-ACCESS     read-write
STATUS         current
DESCRIPTION
    " This is the first tone interval."
 ::= { pktcSigDevToneEntry 10}

pktcSigDevToneFirstToneOff    OBJECT-TYPE
SYNTAX          Unsigned32 (0..5000)
UNITS           "Milliseconds"
MAX-ACCESS     read-write
STATUS         current
DESCRIPTION
    " This is the first idle interval."

```

```

::={ pktcSigDevToneEntry 11 }

pktcSigDevToneSecondToneOn    OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the second tone interval."
    ::= { pktcSigDevToneEntry 12 }

pktcSigDevToneSecondToneOff    OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the second idle interval."
    ::= { pktcSigDevToneEntry 13 }

pktcSigDevToneThirdToneOn     OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the third tone interval."
    ::= { pktcSigDevToneEntry 14 }

pktcSigDevToneThirdToneOff    OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the third idle interval."
    ::= { pktcSigDevToneEntry 15 }

pktcSigDevToneFourthToneOn    OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This is the fourth tone interval."
    ::= { pktcSigDevToneEntry 16 }

pktcSigDevToneFourthToneOff    OBJECT-TYPE
    SYNTAX      Unsigned32 (0..5000)

```

```

UNITS          "Milliseconds"
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION
    " This is the Fourth idle interval."
 ::= { pktcSigDevToneEntry 17 }

pktcSigDevToneWholeToneRepeatCount      OBJECT-TYPE
SYNTAX      Unsigned32 (0..5000)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This is the repeat count, which signifies how many times
      to repeat the entire on-off sequence."
 ::= { pktcSigDevToneEntry 18 }

pktcSigDevToneSteady      OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    " This is the steady tone. Device must playout the on-off
      sequence for pktcSigDevToneWholeRepeatCount times and
      then apply the last tone forever."
 ::= { pktcSigDevToneEntry 19 }

--
-- The NCS End Point Config Table is used to define attributes that
-- are specific to connection EndPoints.
--

pktcNcsEndPntConfigTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF PktcNcsEndPntConfigEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    " This table describes the information pertaining to each
      endpoint of the MTA. All entries in this table represent
      the provisioned endpoints provisioned with the information
      required by the MTA to maintain the NCS signaling protocol
      communication with the CMS. Each endpoint can be assigned
      to a its own CMS. If the specific endpoint does not have
      the corresponding CMS information in this table, the
      endpoint is considered as not provisioned with voice
      services. Objects in this table do not persist across
      MTA reboots."
 ::= { pktcNcsEndPntConfigObjects 1 }

pktcNcsEndPntConfigEntry  OBJECT-TYPE

```

```

SYNTAX          PktcNcsEndPntConfigEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     " Entries in pktcNcsEndPntConfigTable û Each entry
                 represents the required signaling parameters for the
                 specific endpoint provisioned with voice services."
INDEX { ifIndex }
 ::= { pktcNcsEndPntConfigTable 1 }

```

```

PktcNcsEndPntConfigEntry ::= SEQUENCE {
    pktcNcsEndPntConfigCallAgentId          SnmpAdminString,
    pktcNcsEndPntConfigCallAgentUdpPort    InetPortNumber,
    pktcNcsEndPntConfigPartialDialTO       Unsigned32,
    pktcNcsEndPntConfigCriticalDialTO      Unsigned32,
    pktcNcsEndPntConfigBusyToneTO          Unsigned32,
    pktcNcsEndPntConfigDialToneTO          Unsigned32,
    pktcNcsEndPntConfigMessageWaitingTO    Unsigned32,
    pktcNcsEndPntConfigOffHookWarnToneTO   Unsigned32,
    pktcNcsEndPntConfigRingingTO           Unsigned32,
    pktcNcsEndPntConfigRingBackTO          Unsigned32,
    pktcNcsEndPntConfigReorderToneTO       Unsigned32,
    pktcNcsEndPntConfigStutterDialToneTO   Unsigned32,
    pktcNcsEndPntConfigTSMMax              Unsigned32,
    pktcNcsEndPntConfigMax1                Unsigned32,
    pktcNcsEndPntConfigMax2                Unsigned32,
    pktcNcsEndPntConfigMax1QEnable         TruthValue,
    pktcNcsEndPntConfigMax2QEnable         TruthValue,
    pktcNcsEndPntConfigMWD                 Unsigned32,
    pktcNcsEndPntConfigTdinit              Unsigned32,
    pktcNcsEndPntConfigTdmin               Unsigned32,
    pktcNcsEndPntConfigTdmax               Unsigned32,
    pktcNcsEndPntConfigRtoMax              Unsigned32,
    pktcNcsEndPntConfigRtoInit             Unsigned32,
    pktcNcsEndPntConfigLongDurationKeepAlive Unsigned32,
    pktcNcsEndPntConfigThist               Unsigned32,
    pktcNcsEndPntConfigStatus              RowStatus,
    pktcNcsEndPntConfigCallWaitingMaxRep   Unsigned32,
    pktcNcsEndPntConfigCallWaitingDelay    Unsigned32,
    pktcNcsEndPntStatusCallIpAddressType   InetAddressType,
    pktcNcsEndPntStatusCallIpAddress       InetAddress,
    pktcNcsEndPntStatusError               INTEGER,
    pktcNcsEndPntConfigMinHookFlash         Unsigned32,
    pktcNcsEndPntConfigMaxHookFlash        Unsigned32,
    pktcNcsEndPntConfigPulseDialInterdigitTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMinMakeTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMaxMakeTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMinBreakTime Unsigned32,
    pktcNcsEndPntConfigPulseDialMaxBreakTime Unsigned32
}

```

PacketCable/IPCablecom NCS Signaling MIB

February 2004

}

```

pktcNcsEndPntConfigCallAgentId      OBJECT-TYPE
    SYNTAX      SnmpAdminString(SIZE (3..255))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains a string indicating the call agent
          name. (e.g.,: ca@example.com). The call agent name after
          the @ character must be a fully qualified domain name and
          have a corresponding pktcMtaDevCmsFqdn entry in the
          pktcMtaDevCmsTable. The object pktcMtaDevCmsFqdn is
          defined in the PacketCable MIBMTA Specification."
    ::= { pktcNcsEndPntConfigEntry 1 }

pktcNcsEndPntConfigCallAgentUdpPort  OBJECT-TYPE
    SYNTAX      InetPortNumber (1025..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the call agent User Datagram Protocol
          (UDP) receive port that is being used for this instance of
          call signaling, i.e. the default port on which the call
          agent will receive NCS signaling from the endpoint."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL     { 2727 }
    ::= { pktcNcsEndPntConfigEntry 2 }

pktcNcsEndPntConfigPartialDialTO     OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains maximum value of the partial dial
          time out."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL     { 16 }
    ::= { pktcNcsEndPntConfigEntry 3 }

pktcNcsEndPntConfigCriticalDialTO    OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create

```

STATUS current
DESCRIPTION
" This object contains the maximum value of the critical

dial time out."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 4 }
 ::= { pktcNcsEndPntConfigEntry 4 }

pktcNcsEndPntConfigBusyToneTO OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the timeout value for busy tone."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 30 }
 ::= { pktcNcsEndPntConfigEntry 5 }

pktcNcsEndPntConfigDialToneTO OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the timeout value for dial tone."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 16 }
 ::= { pktcNcsEndPntConfigEntry 6 }

pktcNcsEndPntConfigMessageWaitingTO OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the timeout value for message waiting
indicator."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 16 }

```
::= { pktcNcsEndPntConfigEntry 7 }
```

```
pktcNcsEndPntConfigOffHookWarnToneTO      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "seconds"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
    " This object contains timeout value for off hook Warning
      tone."
```

```
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 0 }
::= { pktcNcsEndPntConfigEntry 8 }
```

```
pktcNcsEndPntConfigRingingTO      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "seconds"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
    " This object contains the timeout value for ringing."
```

```
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 180 }
::= { pktcNcsEndPntConfigEntry 9 }
```

```
pktcNcsEndPntConfigRingBackTO     OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "seconds"
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
```

```
    " This object contains the timeout value for ring back."
```

```
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 180 }
::= { pktcNcsEndPntConfigEntry 10 }
```

```
pktcNcsEndPntConfigReorderToneTO  OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "seconds"
    MAX-ACCESS   read-create
    STATUS       current
```

DESCRIPTION
" This object contains the timeout value for reorder tone."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 30 }
::= { pktcNcsEndPntConfigEntry 11 }

pktcNcsEndPntConfigStutterDialToneTO OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current
DESCRIPTION

Beacham/Kumar/Channabasappa Expires - July 2004

[Page 43]

PacketCable/IPCablecom NCS Signaling MIB

February 2004

" This object contains timeout value for stutter dial tone."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 16 }
::= { pktcNcsEndPntConfigEntry 12 }

pktcNcsEndPntConfigTSMMax OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION

" This object contains the max time in seconds since the
sending of the initial datagram."

REFERENCE
"PacketCable NCS Specification"
DEFVAL { 20 }
::= { pktcNcsEndPntConfigEntry 13 }

pktcNcsEndPntConfigMax1 OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION

" This object contains the suspicious error threshold for
signaling messages."

REFERENCE
"PacketCable NCS Specification"
DEFVAL { 5 }
::= { pktcNcsEndPntConfigEntry 14 }

pktcNcsEndPntConfigMax2 OBJECT-TYPE

```

SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This object contains the disconnect error threshold for
      signaling messages."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 7 }
 ::= { pktcNcsEndPntConfigEntry 15 }

```

```

pktcNcsEndPntConfigMax1QEnable      OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This object enables/disables the Max1 Domain Name Server
      (DNS) query operation when Max1 expires."

```

```

DEFVAL { true }
 ::= { pktcNcsEndPntConfigEntry 16 }

```

```

pktcNcsEndPntConfigMax2QEnable      OBJECT-TYPE
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " This object enables/disables the Max2 DNS query operation
      when Max2 expires."
DEFVAL { true }
 ::= { pktcNcsEndPntConfigEntry 17 }

```

```

pktcNcsEndPntConfigMWD              OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    " Maximum Waiting Delay (MWD) contains the maximum number of
      seconds a MTA waits after a restart. When this timer
      expires the MTA initiates the restart procedure."
REFERENCE
    "PacketCable NCS Specification"
DEFVAL { 600 }
 ::= { pktcNcsEndPntConfigEntry 18 }

```

```

pktcNcsEndPntConfigTdinit      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the initial number of seconds the MTA
          waits after a disconnect. When this timer expires the MTA
          initiates the disconnected procedure."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 15 }
    ::= { pktcNcsEndPntConfigEntry 19 }

```

```

pktcNcsEndPntConfigTdmin      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the minimum number of seconds the MTA
          waits after a disconnect. When this timer expires the MTA

```

```

        initiates the disconnected procedure."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 15 }
    ::= { pktcNcsEndPntConfigEntry 20 }

```

```

pktcNcsEndPntConfigTdmax      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the maximum number of seconds the MTA
          waits after a disconnect. When this timer expires the MTA
          initiates the disconnected procedure.
          "
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 600 }
    ::= { pktcNcsEndPntConfigEntry 21 }

```

```

pktcNcsEndPntConfigRtoMax      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the maximum number of seconds for the
          retransmission timer. When this timer expires the MTA
          retransmits the message."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 4 }
    ::= { pktcNcsEndPntConfigEntry 22 }

```

```

pktcNcsEndPntConfigRtoInit     OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "milliseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " This object contains the initial number of seconds for the
          retransmission timer."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 200 }
    ::= { pktcNcsEndPntConfigEntry 23 }

```

```

pktcNcsEndPntConfigLongDurationKeepAlive      OBJECT-TYPE

```

```

    SYNTAX      Unsigned32
    UNITS       "minutes"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        " Specifies a timeout value in minutes for sending long
          duration call notification message."
    REFERENCE
        "PacketCable NCS Specification"
    DEFVAL { 60 }
    ::= { pktcNcsEndPntConfigEntry 24 }

```

```

pktcNcsEndPntConfigThist      OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create

```

STATUS current
DESCRIPTION
" Timeout period in seconds before no response is declared."
REFERENCE
"PacketCable NCS Specification"
DEFVAL { 30 }
::= { pktcNcsEndPntConfigEntry 25 }

pktcNcsEndPntConfigStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the Row Status associated with the
pktsNcsEndPntTable. "
::= { pktcNcsEndPntConfigEntry 26 }

pktcNcsEndPntConfigCallWaitingMaxRep OBJECT-TYPE
SYNTAX Unsigned32 (0..10)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the maximum number of repetitions
of the call waiting tone that the MTA will play from a
single CMS request. A value of zero (0) can be used if
the CMS is to control the repetitions of the call waiting
tone."
DEFVAL { 1 }
::= { pktcNcsEndPntConfigEntry 27 }

pktcNcsEndPntConfigCallWaitingDelay OBJECT-TYPE
SYNTAX Unsigned32 (1..100)
UNITS "seconds"

MAX-ACCESS read-create
STATUS current
DESCRIPTION
" This object contains the delay between repetitions of the
call waiting tone that the MTA will play from a single CMS
request."
DEFVAL { 10 }
::= { pktcNcsEndPntConfigEntry 28 }

pktcNcsEndPntStatusCallIpAddressType OBJECT-TYPE
SYNTAX InetAddressType

MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This object contains the type of Internet address of the
CMS currently being used for this endpoint."
 ::= { pktcNcsEndPntConfigEntry 29 }

pktcNcsEndPntStatusCallIpAddress OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This object contains the Internet address of the CMS
currently being used for this endpoint. This Internet
address is used to create the appropriate security
association. The type of this IP address is determined by
the value of the pktcNcsEndPntStatusCallIpAddressType
object."
 ::= { pktcNcsEndPntConfigEntry 30 }

pktcNcsEndPntStatusError OBJECT-TYPE
SYNTAX INTEGER {
operational (1),
noSecurityAssociation (2),
disconnected (3)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
" This object contains the error status for this interface.
The operational status indicates that all operations
necessary to put the line in service have occurred, and the
CMS has acknowledged the RSIP message successfully. If
pktcMtaDevCmsIpsecCtrl is enabled for the associated Call
Agent, the noSecurityAssociation status indicates that no
Security Association (SA) yet exists for this endpoint.
Otherwise, the state is unused. The disconnected status
indicates one of the following two:

If pktcMtaDevCmsIpsecCtrl is disabled, then no security
association is involved with this endpoint. The NCS
signaling software is in process of establishing the NCS
signaling link via an RSIP exchange.
Otherwise, pktcMtaDevCmsIpsecCtrl is enabled, the security
Association has been established, and the NCS signaling

```
software is in process of establishing the NCS signaling
link via an RSIP exchange."
 ::= { pktcNcsEndPntConfigEntry 31 }
```

```
pktcNcsEndPntConfigMinHookFlash    OBJECT-TYPE
SYNTAX      Unsigned32 (20..1000)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the minimum time a line needs to be on hook for a
    valid hook flash. The value of this object MUST be
    greater than the value of
    pktcNcsEndPntConfigPulseMaxBreakTime. This object must
    only be set via the configuration file during the
    provisioning process."
DEFVAL { 300 }
 ::= { pktcNcsEndPntConfigEntry 32 }
```

```
pktcNcsEndPntConfigMaxHookFlash    OBJECT-TYPE
SYNTAX      Unsigned32 (20..500)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the maximum time a line needs to be on hook for a
    valid hook flash. This object must only be set via the
    configuration file during the provisioning process."
DEFVAL { 500 }
 ::= { pktcNcsEndPntConfigEntry 33 }
```

```
pktcNcsEndPntConfigPulseDialInterdigitTime    OBJECT-TYPE
SYNTAX      Unsigned32 (100..1500)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the pulse dial inter-digit timeout. This object
    must only be set via the configuration file during the
    provisioning process."
DEFVAL { 100 }
 ::= { pktcNcsEndPntConfigEntry 34 }
```

```
pktcNcsEndPntConfigPulseDialMinMakeTime    OBJECT-TYPE
```

```

SYNTAX      Unsigned32 (20..200)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the minimum make pulse width for the dial pulse.
      This object must only be set via the configuration file
      during the provisioning process."
DEFVAL { 25 }
 ::= { pktcNcsEndPntConfigEntry 35 }

pktcNcsEndPntConfigPulseDialMaxMakeTime      OBJECT-TYPE
SYNTAX      Unsigned32 (20..200)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the maximum make pulse width for the dial pulse.
      This object must only be set via the configuration file
      during the provisioning process."
DEFVAL { 55 }
 ::= { pktcNcsEndPntConfigEntry 36 }

pktcNcsEndPntConfigPulseDialMinBreakTime     OBJECT-TYPE
SYNTAX      Unsigned32 (20..200)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the minimum break pulse width for the dial pulse.
      This object must only be set via the configuration file
      during the provisioning process."
DEFVAL { 45 }
 ::= { pktcNcsEndPntConfigEntry 37 }

pktcNcsEndPntConfigPulseDialMaxBreakTime     OBJECT-TYPE
SYNTAX      Unsigned32 (20..200)
UNITS       "Milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    " This is the maximum break pulse width for the dial pulse.
      This object must only be set via the configuration file
      during the provisioning process."
DEFVAL { 75 }
 ::= { pktcNcsEndPntConfigEntry 38 }

```

```
--
-- notification group is for future extension.
--
pktcSigNotification OBJECT IDENTIFIER ::= { pktcSigMib 0 }
pktcSigConformance OBJECT IDENTIFIER ::= { pktcSigMib 3 }
pktcSigCompliances OBJECT IDENTIFIER ::= { pktcSigConformance 1 }
pktcSigGroups OBJECT IDENTIFIER ::= { pktcSigConformance 2 }

--
-- compliance statements
--

pktcSigBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        " The compliance statement for devices that implement
          Signaling on the MTA."

MODULE -- pktcSigMib

--
-- unconditionally mandatory groups
--

MANDATORY-GROUPS {
    pktcSigGroup
}

GROUP pktcNcsGroup
    DESCRIPTION
        " This group is mandatory for any MTA implementing NCS
          signaling"

GROUP pktcInternationalGroup
    DESCRIPTION
        " This group is mandatory for any MTA implementing
          international telephony features. In such cases, it is
          left to manufacturers to determine whether to support both
          PacketCable and IPCablecom objects in the same MTA."
    ::= { pktcSigCompliances 1 }

--
-- units of conformance
--

pktcSigGroup OBJECT-GROUP
    OBJECTS {
        pktcSigDevCodecType,
        pktcSigDevCodecMax,
```

```

pktcSigDevEchoCancellation,
pktcSigDevSilenceSuppression,
pktcSigDevR0Cadence,
pktcSigDevR6Cadence,
pktcSigDevR7Cadence,
pktcSigDefCallSigDscp,
pktcSigDefMediaStreamDscp,
pktcSigDevVmwMode,
pktcSignalingType,
pktcSignalingVersion,
pktcSignalingVendorExtension,
pktcSigDefNcsReceiveUdpPort,
pktcSigServiceClassNameUS,
pktcSigServiceClassNameDS,
pktcSigServiceClassNameMask,
pktcSigNcsServiceFlowState,
pktcSigDevR1Cadence,
pktcSigDevR2Cadence,
pktcSigDevR3Cadence,
pktcSigDevR4Cadence,
pktcSigDevR5Cadence,
pktcSigDevRgCadence,
pktcSigDevRsCadence

}
STATUS current
DESCRIPTION
    "Group of objects for the common portion of the
    PacketCable Signaling MIB."
 ::= { pktcSigGroups 1 }

```

```

pktcNcsGroup OBJECT-GROUP
OBJECTS {
pktcNcsEndPntConfigCallAgentId,
pktcNcsEndPntConfigCallAgentUdpPort,
pktcNcsEndPntConfigPartialDialTO,
pktcNcsEndPntConfigCriticalDialTO,
pktcNcsEndPntConfigBusyToneTO,
pktcNcsEndPntConfigDialToneTO,
pktcNcsEndPntConfigMessageWaitingTO,
pktcNcsEndPntConfigOffHookWarnToneTO,
pktcNcsEndPntConfigRingingTO,
pktcNcsEndPntConfigRingBackTO,
pktcNcsEndPntConfigReorderToneTO,
pktcNcsEndPntConfigStutterDialToneTO,
pktcNcsEndPntConfigTSMMax,
pktcNcsEndPntConfigMax1,
pktcNcsEndPntConfigMax2,
pktcNcsEndPntConfigMax1QEnable,

```

```

pktcNcsEndPntConfigMax2QEnable,
pktcNcsEndPntConfigMWD,
pktcNcsEndPntConfigTdinit,
pktcNcsEndPntConfigTdmin,
pktcNcsEndPntConfigTdmax,
pktcNcsEndPntConfigRtoMax,
pktcNcsEndPntConfigRtoInit,
pktcNcsEndPntConfigLongDurationKeepAlive,
pktcNcsEndPntConfigThist,
pktcNcsEndPntConfigStatus,
pktcNcsEndPntConfigCallWaitingMaxRep,
pktcNcsEndPntConfigCallWaitingDelay,
pktcNcsEndPntStatusCallIpAddressType,
pktcNcsEndPntStatusCallIpAddress,
pktcNcsEndPntStatusError
}
STATUS current
DESCRIPTION
    "Group of objects for the NCS portion of the PacketCable
    Signaling MIB. This is mandatory for NCS signaling."
 ::= { pktcSigGroups 2 }

```

```

pktcInternationalGroup    OBJECT-GROUP
    OBJECTS {
        pktcNcsEndPntConfigMinHookFlash,
        pktcNcsEndPntConfigMaxHookFlash,
        pktcNcsEndPntConfigPulseDialInterdigitTime,
        pktcNcsEndPntConfigPulseDialMinMakeTime,
        pktcNcsEndPntConfigPulseDialMaxMakeTime,
        pktcNcsEndPntConfigPulseDialMinBreakTime,
        pktcNcsEndPntConfigPulseDialMaxBreakTime,
        pktcSigDevRingCadence,
        pktcSigDevStandardRingCadence,
        pktcSigDevRingSplashCadence,
        pktcSigDevCallerIdSigProtocol,
        pktcSigDevCIDMode,
        pktcSigDevCIDFskAfterRing,
        pktcSigDevCIDFskAfterDTAS,
        pktcSigDevCIDFskAfterRPAS,
        pktcSigDevCIDRingAfterFSK,
        pktcSigDevCIDDTASAfterLR,
        pktcSigDevVmwiFskAfterDTAS,
        pktcSigDevVmwiFskAfterRPAS,
        pktcSigDevVmwiDTASAfterLR,
        pktcSigPowerRingFrequency,
        pktcSigPulseSignalType,
        pktcSigPulseSignalFrequency,
    }

```

```
pktcSigPulseSignalDbLevel,  
pktcSigPulseSignalDuration,
```

PacketCable/IPCablecom NCS Signaling MIB

February 2004

```
pktcPulseSignalPulseInterval,  
pktcSigPulseSignalRepeatCount,  
pktcSigDevToneDbLevel,  
pktcSigDevToneFreqType,  
pktcSigDevToneNumFrequencies,  
pktcSigDevToneFirstFrequency,  
pktcSigDevToneSecondFrequency,  
pktcSigDevToneThirdFrequency,  
pktcSigDevToneFourthFrequency,  
pktcSigDevToneNumOnOffTimes,  
pktcSigDevToneFirstToneOn,  
pktcSigDevToneFirstToneOff,  
pktcSigDevToneSecondToneOn,  
pktcSigDevToneSecondToneOff,  
pktcSigDevToneThirdToneOn,  
pktcSigDevToneThirdToneOff,  
pktcSigDevToneFourthToneOn,  
pktcSigDevToneFourthToneOff,  
pktcSigDevToneWholeToneRepeatCount,  
pktcSigDevToneSteady  
}
```

STATUS current

DESCRIPTION

" Group of objects that extend the behavior of existing objects to support operations in the widest possible set of international marketplaces. Note that many of these objects represent a superset of behaviors described in other objects within this MIB Module."

::= { pktcSigGroups 3 }

END

5. Acknowledgments

This document is a production of the PacketCable Working Group.

The current editors wish to express gratitude to:

Angela Lyda	Arris Interactive
Sasha Medvinsky	Motorola
Roy Spitzer	Telogy Networks, Inc.
Rick Vetter	Motorola

Itay Sherman	Texas Instruments
Klaus Hermanns	Cisco Systems
Eugene Nechamkin	Broadcom Corp
Matt A. Osman	CableLabs
Jean-Francois Mule	CableLabs
Rich Woundy	Comcast
Bert Wijnen	Lucent

Beacham/Kumar/Channabasappa Expires - July 2004

[Page 54]

PacketCable/IPCablecom NCS Signaling MIB

February 2004

Mike Heard Consultant

6. Normative References

- [RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC3289] Baker, F., Chan, K., Smith, A., "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.
- [RFC3291] Daniel, M., Haberman, B., Routhier, S., Schoenwaelder, J., "Textual Conventions for Internet Network Addresses", RFC 3291, May 2002.

--

-- NOTES TO RFC EDITOR (to be removed prior to publication)

--

-- The I-D <draft-ietf-ops-rfc3291bis-01.txt> (or a successor) is
-- expected to eventually replace RFC 3291. If that draft (or a
-- successor) is published as an RFC prior to, or concurrently with
-- this document, then the normative reference [RFC3291] should be
-- updated to point to the replacement RFC, and the reference tag
-- [RFC3291] should be updated to match.

--

[PKT-SP-MGCP-IO9-040113] Packetcable Network-Based Call Signaling

Protocol Specification PKT-SP-EC-MGCP-
IO9-040113.

[PKT-SP-PROV-IO8-040113] Packetcable MTA Device Provisioning
Specification PKT-SP-PROV-IO8-040113.

[PKT-SP-CODEC-IO5-040113] Packetcable Audio/Video Codecs
Specification PKT-SP-CODEC-IO5-040113.

7. Informative References

[RFC3261] Rosenberg, J., Schulzrinne H., Camarillo, G., Johnston,

Beacham/Kumar/Channabasappa Expires - July 2004

[Page 55]

PacketCable/IPCablecom NCS Signaling MIB

February 2004

A., Peterson, J., Sparks, R., Handley, M., and Schooler,
E. "SIP: Session Initiation Protocol", RFC 3261, June
2002.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart,
"Introduction and Applicability Statements for Internet-
Standard Management Framework", RFC 3410, December 2002.

[RFC3435] Andreasen, F., Foster, B., "Media Gateway Control Protocol
(MGCP)", RFC 3435, January 2003.

[PKT-SP-MIB-MTA-IO8-040113] Packetcable MTA MIB Specification PKT-
SP-MIB-MTA-IO8-040113.

[ETSI TS 101 909-4] ETSI TS 101 909-4:"Access and Terminals (AT);
Digital Broadband Cable Access to the Public
Telecommunications Network; IP Multimedia Time
Critical Services; Part 4: Network Call
Signaling Protocol".

[ETSI TS 101 909-9] ETSI TS 101 909-9:"Access and Terminals (AT);
Digital Broadband Cable Access to the Public
Telecommunications Network; IP Multimedia Time
Critical Services; Part 9: IPCablecom Network
Call Signalling (NCS) MIB Requirements".

[EN 300 001] EN 300 001 V1.5.1 (1998-10):"European Standard
(Telecommunications series) Attachments to Public
Switched Telephone Network (PSTN); General technical
requirements for equipment connected to an analogue
subscriber interface in the PSTN; Chapter 3: Ringing
signal characteristics (national deviations are in

Table 3.1.1)".

- [EN 300 324-1] EN 300 324-1 V2.1.1 (2000-04): "V Interfaces at the digital Loop Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification".
- [EN 300 659-1] EN 300 659-1: "Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 1: On hook data transmission".
- [ITU-T E.180] ITU-T: "Various Tones Used in National Networks, Supplement 2 to Recommendation E.180".
- [TR 101 183] TR 101 183: "Public Switched Telephone Network (PSTN) Analogue Ringing Signals".

8. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The following Differentiated Services Code Point (DSCP) and mask objects are used to differentiate between various types of traffic in the service provider network:

```
pktcSigDefCallSigDscp
pktcSigDefMediaStreamDscp
pktcSigServiceClassNameMask
```

These objects may contain information that may be sensitive from a business perspective. For example, they may represent a customer's service contract that a service provider chooses to apply to a customer's ingress or egress traffic. If these objects are SET maliciously, it may permit unmarked or inappropriately marked signaling and media traffic to enter the service provider network, resulting in unauthorized levels of service for customers.

The following objects determine ring cadence, repeatable

characteristics, signal duration, and caller id subscriber line protocol for telephony operation:

```
pktcSigDevR0Cadence
pktcSigDevR1Cadence
pktcSigDevR2Cadence
pktcSigDevR3Cadence
pktcSigDevR4Cadence
pktcSigDevR5Cadence
pktcSigDevR6Cadence
pktcSigDevR7Cadence
pktcSigDevRgCadence
pktcSigDevRsCadence
pktcSigDevCallerIdSigProtocol
pktcSigPulseSignalDuration
pktcSigPulseSignalPauseDuration
```

If these objects are SET maliciously, it may result in unwanted operation, or a failure to obtain telephony service from client (MTA) devices.

The objects in the `pktcNcsEndPntConfigTable` are used for end point signaling. The `pktcNcsEndPntConfigCallAgentId` object contains the name of the call agent, which includes the call agent FQDN. If this object is SET maliciously, the MTA will not be able to communicate with the call agent, resulting in a disruption of telephony service. The `pktcNcsEndPntConfigCallAgentUdpPort` object identifies the UDP port for NCS signaling traffic. If this object is SET maliciously, the call agent will not receive NCS signaling traffic from the MTA, also resulting in a disruption of telephony service.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. The most sensitive is `pktcNcsEndPntStatusCallIpAddress` within `pktcNcsEndPntConfigTable`. This information itself may be valuable to would-be attackers.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is

allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances

of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Authors' Addresses

Gordon Beacham
Motorola BCS/DCG
6450 Sequence Drive, Bldg. 1
San Diego, CA 92121, USA

+1 858-404-2335
gordon.beacham@motorola.com

Satish Kumar Mudugere Eswaraiah
Texas Instruments India (P) Ltd.,
Golf view, Wind Tunnel Road
Murugesh Palya
Bangalore - 560 017, INDIA
+91 80 5269451
satish.kumar@ti.com

Sumanth Channabasappa
Alopa Networks Inc
248, McCaslin Blvd #101,
Louisville, Colorado - 80027
U.S.A.
+1 303 604 6595
Sumanth@alopa.com

11. Full Copyright Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such

proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to

others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.