Path Computation Element Communications Protocol (PCEP) Management Information Base (MIB) Module
draft-ietf-pce-pcep-mib-11

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 describes managed objects for modeling of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 27, 2015.
1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.
PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE).

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a MIB module that can be used to monitor PCEP interactions between a PCC and a PCE, or between two PCEs.

The scope of this document is to provide a MIB module for the PCEP base protocol defined in [RFC5440]. Extensions to the PCEP base protocol are beyond the scope for this document.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Terminology

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- Path Computation Request message (PCReq).
- Path Computation Reply message (PCRep).
- Notification message (PCNtf).
- Error message (PCErr).
- Request Parameters object (RP).
- Synchronization Vector object (SVEC).
- Explicit Route object (ERO).

This document uses the term "PCEP entity" to refer to a local PCEP speaker, "peer" to refer to a remote PCEP speaker and "PCEP speaker" where it is not necessary to distinguish between local and remote.

2. 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].

3. PCEP MIB Module Architecture

The PCEP MIB module contains the following information:

a. PCE and PCC local entity status (see pcePcepEntityTable).

b. PCEP peer information (see pcePcepPeerTable).

c. PCEP session information (see pcePcepSessTable).

d. Notifications to indicate PCEP session changes.

The PCEP MIB module is limited to "read-only" access except for pcePcepNotificationsMaxRate, which is used to throttle the rate at which the implementation generates notifications.

3.1. pcePcepEntityTable

The PCEP MIB module may contain status information for multiple logical local PCEP entities. There are several scenarios in which there may be more than one local PCEP entity, including the following:

- A physical router, which is partitioned into multiple virtual routers, each with its own PCC.

- A PCE device which front-ends a cluster of compute resources, each with a different set of capabilities that are accessed via different IP addresses.

The pcePcepEntityTable contains one row for each local PCEP entity. Each row is read-only and contains current status information plus the PCEP entity’s running configuration.
The pcePcepEntityTable is indexed by pcePcepEntityIndex, which also acts as the primary index for the other tables in this MIB module.

3.2. pcePcepPeerTable

The pcePcepPeerTable contains one row for each peer that the local PCEP entity knows about. Each row is read-only and contains information to identify the peer, the running configuration relating to that peer and statistics that track the messages exchanged with that peer and its response times.

A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

The pcePcepPeerTable is indexed first by pcePcepEntityIndex, then by pcePcepPeerAddrType and pcePcepPeerAddr. This indexing structure allows each local PCEP entity to report its own set of peers.

Since PCEP sessions can be ephemeral, the pcePcepPeerTable tracks a peer even when no PCEP session currently exists to that peer. The statistics contained in pcePcepPeerTable are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard a row from the pcePcepPeerTable if and only if no PCEP session exists to the corresponding peer.

3.3. pcePcepSessTable

The pcePcepSessTable contains one row for each PCEP session that the PCEP entity (PCE or PCC) is currently participating in. Each row is read-only and contains the running configuration that is applied to the session, plus identifiers and statistics for the session.

The statistics in pcePcepSessTable are semantically different from those in pcePcepPeerTable since the former apply to the current session only, whereas the latter are the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids there from being more than one active PCEP session between a given pair of PCEP entities at any one time, there is a window during session establishment where the pcePcepSessTable may contain two rows for a given peer, one representing a session initiated by the local PCEP entity and one representing a session initiated by the peer. If either of these sessions reaches active state, then the other is discarded.
The pcePcepSessTable is indexed first by pcePcepEntityIndex, then by pcePcepPeerAddrType and pcePcepPeerAddr, and finally by pcePcepSessInitiator. This indexing structure allows each local PCEP entity to report its own set of active sessions. The pcePcepSessInitiator index allows two rows to exist transiently for a given peer, as discussed above.

3.4. PCEP Notifications

The PCEP MIB module contains notifications for the following conditions.

a. pcePcepSessUp: PCEP Session has gone up.
b. pcePcepSessDown: PCEP Session has gone down.
c. pcePcepSessLocalOverload: Local PCEP entity has sent an overload PCNtf on this session.
d. pcePcepSessLocalOverloadClear: Local PCEP entity has sent an overload-cleared PCNtf on this session.
e. pcePcepSessPeerOverload: Peer has sent an overload PCNtf on this session.
f. pcePcepSessPeerOverloadClear: Peer has sent an overload-cleared PCNtf on this session.

3.5. Relationship to other MIB modules

The PCEP MIB module imports the following textual conventions from the INET-ADDRESS-MIB defined in RFC 4001 [RFC4001]:

- InetAddressType
- InetAddress

PCEP relies on existing protocols which have specialized MIB objects to monitor their own activities. Consequently this document considers that the monitoring of underlying protocols is out of scope of the PCEP MIB module.

3.6. Illustrative example

The following diagram illustrates the relationships between the pcePcepEntityTable, pcePcepPeerTable and pcePcepSessTable.
4. Object Definitions

4.1. PCE-PCEP-MIB

PCE-PCEP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    mib-2,
    NOTIFICATION-TYPE,
    Unsigned32,
Counter32
FROM SNMPv2-SMI -- RFC 2578
TruthValue,
TimeStamp
FROM SNMPv2-TC -- RFC 2579
MODULE-COMPLIANCE,
OBJECT-GROUP,
NOTIFICATION-GROUP
FROM SNMPv2-CONF -- RFC 2580
InetAddressType,
InetAddress
FROM INET-ADDRESS-MIB; -- RFC 4001

pcePcepMIB MODULE-IDENTITY
LAST-UPDATED "201410241200Z" -- 24 October 2014
ORGANIZATION "IETF Path Computation Element (PCE) Working Group"
CONTACT-INFO "Email: pce@ietf.org
WG charter:
http://www.ietf.org/html.charters/pce-charter.html"

DESCRIPTION "This MIB module defines a collection of objects for managing
Path Computation Element communications Protocol (PCEP).

Copyright (C) The IETF Trust (2014). 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 & remove this note
REVISION "201410241200Z" -- 24 October 2014
DESCRIPTION "Initial version, published as RFC YYYY."
-- RFC Ed.: replace YYYY with actual RFC number & remove this note
::= { mib-2 XXX }
-- RFC Ed.: replace XXX with IANA-assigned number & remove this note

pcePcepNotifications OBJECT IDENTIFIER ::= { pcePcepMIB 0 }
pcePcepObjects OBJECT IDENTIFIER ::= { pcePcepMIB 1 }
pcePcepConformance OBJECT IDENTIFIER ::= { pcePcepMIB 2 }

--
-- PCEP Entity Objects
--

pcePcepEntityTable OBJECT-TYPE
SYNTAX      SEQUENCE OF PcePcepEntityEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This table contains information about local PCEP entities. The entries in this table are read-only."
 ::= { pcePcepObjects 1 }

pcePcepEntityEntry OBJECT-TYPE
SYNTAX      PcePcepEntityEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This entry represents a local PCEP entity."
INDEX       {  pcePcepEntityIndex  }
 ::= { pcePcepEntityTable 1 }

PcePcepEntityEntry ::= SEQUENCE {
    pcePcepEntityIndex                Unsigned32,
pcePcepEntityAdminStatus          INTEGER,
pcePcepEntityOperStatus           INTEGER,
pcePcepEntityAddrType             InetAddressType,
pcePcepEntityAddr                 InetAddress,
pcePcepEntityConnectTimer         Unsigned32,
pcePcepEntityConnectMaxRetry      Unsigned32,
pcePcepEntityInitBackoffTimer     Unsigned32,
pcePcepEntityMaxBackoffTimer      Unsigned32,
pcePcepEntityOpenWaitTimer        Unsigned32,
pcePcepEntityKeepWaitTimer        Unsigned32,
pcePcepEntityKeepAliveTimer       Unsigned32,
pcePcepEntityDeadTimer            Unsigned32,
pcePcepEntityAllowNegotiation     TruthValue,
pcePcepEntityMaxKeepAliveTimer    Unsigned32,
pcePcepEntityMaxDeadTimer         Unsigned32,
pcePcepEntityMinKeepAliveTimer    Unsigned32,
pcePcepEntityMinDeadTimer         Unsigned32,
pcePcepEntitySyncTimer            Unsigned32,
pcePcepEntityRequestTimer         Unsigned32,
pcePcepEntityMaxSessions          Unsigned32,
pcePcepEntityMaxUnknownReqs       Unsigned32,
pcePcepEntityMaxUnknownMsgs       Unsigned32
}

pcePcepEntityIndex OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

This index is used to uniquely identify the PCEP entity.

pcePcepEntityAdminStatus OBJECT-TYPE
SYNTAX INTEGER {
   adminStatusUp(1),
   adminStatusDown(2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The administrative status of this PCEP Entity. This is the desired operational status as currently set by an operator or by default in the implementation. The value of pcePcepEntityOperStatus represents the current status of an attempt to reach this desired status."

pcePcepEntityOperStatus OBJECT-TYPE
SYNTAX INTEGER {
   operStatusUp(1),
   operStatusDown(2),
   operStatusGoingUp(3),
   operStatusGoingDown(4),
   operStatusFailed(5),
   operStatusFailedPerm(6)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The operational status of the PCEP entity. Takes one of the following values.
- operStatusUp(1): the PCEP entity is active.
- operStatusDown(2): the PCEP entity is inactive.
- operStatusGoingUp(3): the PCEP entity is activating.
- operStatusGoingDown(4): the PCEP entity is deactivating.
- operStatusFailed(5): the PCEP entity has failed and will recover when possible.
- operStatusFailedPerm(6): the PCEP entity has failed and will not recover without operator intervention."

pcePcepEntityAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The type of the PCEP entity’s Internet address. This object specifies how the value of the pcePcepEntityAddr object should be interpreted. Only values unknown(0), ipv4(1), or ipv6(2) are supported."

::= { pcePcepEntityEntry 4 }

pcePcepEntityAddr OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The local Internet address of this PCEP entity. The type is given by pcePcepEntityAddrType.

If operating as a PCE server, the PCEP entity listens on this address. If operating as a PCC, the PCEP entity binds outgoing TCP connections to this address.

It is possible for the PCEP entity to operate both as a PCC and a PCE Server, in which case it uses this address both to listen for incoming TCP connections and to bind outgoing TCP connections."

::= { pcePcepEntityEntry 5 }

pcePcepEntityConnectTimer OBJECT-TYPE
SYNTAX      Unsigned32 (1..65535)
UNITS       "seconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The time that the PCEP entity will wait to establish a TCP connection with a peer. If a TCP connection is not established within this time then PCEP aborts the session setup attempt."

::= { pcePcepEntityEntry 6 }

pcePcepEntityConnectMaxRetry OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The maximum number of times the system tries to establish a TCP connection to a peer before the session with the peer transitions to the idle state.

When the session transitions to the idle state:
- pcePcepPeerSessionExists transitions to false(2)
- the associated PcePcepSessEntry is deleted"
pcePcepEntityInitBackoffTimer OBJECT-TYPE
SYNTAX Unsigned32 (1..65535)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The initial back-off time for retrying a failed session
setup attempt to a peer.

The back-off time increases for each failed session setup
attempt, until a maximum back-off time is reached. The
maximum back-off time is pcePcepEntityMaxBackoffTimer."

::= { pcePcepEntityEntry 7 }

pcePcepEntityMaxBackoffTimer OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The maximum back-off time for retrying a failed session
setup attempt to a peer.

The back-off time increases for each failed session setup
attempt, until this maximum value is reached. Session
setup attempts then repeat periodically without any
further increase in back-off time."

::= { pcePcepEntityEntry 8 }

pcePcepEntityOpenWaitTimer OBJECT-TYPE
SYNTAX Unsigned32 (1..65535)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The time that the PCEP entity will wait to receive an Open
message from a peer after the TCP connection has come up.
If no Open message is received within this time then PCEP
terminates the TCP connection and deletes the associated
PcePcepSessEntry."

::= { pcePcepEntityEntry 9 }

pcePcepEntityKeepWaitTimer OBJECT-TYPE
SYNTAX Unsigned32 (1..65535)
UNITS "seconds"
max-access  read-only
status      current
description
"The time that the PCEP entity will wait to receive a
Keepalive or PCErr message from a peer during session
initialization after receiving an Open message. If no
Keepalive or PCErr message is received within this time then
PCEP terminates the TCP connection and deletes the
associated PcePcepSessEntry."
 ::= { pcePcepEntityEntry 11 }

pcePcepEntityKeepAliveTimer OBJECT-TYPE
syntax      Unsigned32 (0..255)
units       "seconds"
max-access  read-only
status      current
description
"The keep alive transmission timer that this PCEP entity will
propose in the initial OPEN message of each session it is
involved in. This is the maximum time between two
consecutive messages sent to a peer. Zero means that
the PCEP entity prefers not to send Keepalives at all.

Note that the actual Keepalive transmission intervals, in
either direction of an active PCEP session, are determined
by negotiation between the peers as specified by RFC
5440, and so may differ from this configured value. For
the actually negotiated values (per-session), see
pcePcepSessKeepaliveTimer and
pcePcepSessPeerKeepaliveTimer."
 ::= { pcePcepEntityEntry 12 }

pcePcepEntityDeadTimer OBJECT-TYPE
syntax      Unsigned32 (0..255)
units       "seconds"
max-access  read-only
status      current
description
"The dead timer that this PCEP entity will propose in the
initial OPEN message of each session it is involved in.
This is the time after which a peer should declare a
session down if it does not receive any PCEP messages.
Zero suggests that the peer does not run a dead timer at
all."
 ::= { pcePcepEntityEntry 13 }

pcePcepEntityAllowNegotiation OBJECT-TYPE
syntax      TruthValue

MAX-ACCESS: read-only
STATUS: current
DESCRIPTION:
"Whether the PCEP entity will permit negotiation of session
parameters."
 ::= { pcePcepEntityEntry 14 }

pcePcepEntityMaxKeepAliveTimer OBJECT-TYPE
SYNTAX: Unsigned32 (0..255)
UNITS: "seconds"
MAX-ACCESS: read-only
STATUS: current
DESCRIPTION:
"In PCEP session parameter negotiation, the maximum value
that this PCEP entity will accept from a peer for the
interval between Keepalive transmissions. Zero means that
the PCEP entity will allow no Keepalive transmission at
all."
 ::= { pcePcepEntityEntry 15 }

pcePcepEntityMaxDeadTimer OBJECT-TYPE
SYNTAX: Unsigned32 (0..255)
UNITS: "seconds"
MAX-ACCESS: read-only
STATUS: current
DESCRIPTION:
"In PCEP session parameter negotiation, the maximum value
that this PCEP entity will accept from a peer for the Dead
timer. Zero means that the PCEP entity will allow not
running a Dead timer."
 ::= { pcePcepEntityEntry 16 }

pcePcepEntityMinKeepAliveTimer OBJECT-TYPE
SYNTAX: Unsigned32 (0..255)
UNITS: "seconds"
MAX-ACCESS: read-only
STATUS: current
DESCRIPTION:
"In PCEP session parameter negotiation, the minimum value
that this PCEP entity will accept for the interval between
Keepalive transmissions. Zero means that the PCEP entity
insists on no Keepalive transmission at all."
 ::= { pcePcepEntityEntry 17 }

pcePcepEntityMinDeadTimer OBJECT-TYPE
SYNTAX: Unsigned32 (0..255)
UNITS: "seconds"
MAX-ACCESS: read-only
"In PCEP session parameter negotiation, the minimum value that this PCEP entity will accept for the Dead timer. Zero means that the PCEP entity insists on not running a Dead timer."

::= { pcePcepEntityEntry 18 }

pcePcepEntitySyncTimer OBJECT-TYPE
SYNTAX Unsigned32 (0..65535)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value of SyncTimer is used in the case of synchronized path computation request using the SVEC object.

Consider the case where a PCReq message is received by a PCE that contains the SVEC object referring to M synchronized path computation requests. If after the expiration of the SyncTimer all the M path computation requests have not been received, a protocol error is triggered and the PCE MUST cancel the whole set of path computation requests.

The aim of the SyncTimer is to avoid the storage of unused synchronized requests should one of them get lost for some reasons (for example, a misbehaving PCC).

A value of zero is returned if and only if the entity does not use the SyncTimer."

::= { pcePcepEntityEntry 19 }

pcePcepEntityRequestTimer OBJECT-TYPE
SYNTAX Unsigned32 (1..65535)
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The maximum time that the PCEP entity will wait for a response to a PCReq message."

::= { pcePcepEntityEntry 20 }

pcePcepEntityMaxSessions OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Maximum number of sessions involving this PCEP entity"
that can exist at any time."
::= { pcePcepEntityEntry 21 }

pcePcepEntityMaxUnknownReqs OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The maximum number of unrecognized requests and replies that
any session on this PCEP entity is willing to accept per
minute before terminating the session.

A PCRep message contains an unrecognized reply if it
contains an RP object whose request ID does not correspond
to any in-progress request sent by this PCEP entity.

A PCReq message contains an unrecognized request if it
contains an RP object whose request ID is zero."
::= { pcePcepEntityEntry 22 }

pcePcepEntityMaxUnknownMsgs OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The maximum number of unknown messages that any session
on this PCEP entity is willing to accept per minute before
terminating the session."
::= { pcePcepEntityEntry 23 }

--
-- The PCEP Peer Table
--

pcePcepPeerTable OBJECT-TYPE
SYNTAX      SEQUENCE OF PcePcepPeerEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This table contains information about peers known by
the local PCEP entity. The entries in this table are
read-only.

This table gives peer information that spans PCEP
sessions. Information about current PCEP sessions can be
found in the pcePcepSessTable table."
::= { pcePcepObjects 2 }

### pcePcepPeerEntry

**OBJECT-TYPE**

**SYNTAX** `PcePcepPeerEntry`

**MAX-ACCESS** not-accessible

**STATUS** current

**DESCRIPTION**

"Information about a single peer which spans all PCEP sessions to that peer."

**INDEX** `{ pcePcepEntityIndex, pcePcepPeerAddrType, pcePcepPeerAddr }`

```plaintext
::= { pcePcepPeerTable 1 }
```

```plaintext
PcePcepPeerEntry ::= SEQUENCE {
  pcePcepPeerAddrType                 InetAddressType,
  pcePcepPeerAddr                     InetAddress,
  pcePcepPeerRole                     INTEGER,
  pcePcepPeerDiscontinuityTime        TimeStamp,
  pcePcepPeerInitiateSession          TruthValue,
  pcePcepPeerSessionExists            TruthValue,
  pcePcepPeerNumSessSetupOK           Counter32,
  pcePcepPeerNumSessSetupFail         Counter32,
  pcePcepPeerSessionUpTime            TimeStamp,
  pcePcepPeerSessionFailTime          TimeStamp,
  pcePcepPeerSessionFailUpTime        TimeStamp,
  pcePcepPeerAvgRspTime               Unsigned32,
  pcePcepPeerLWMRspTime               Unsigned32,
  pcePcepPeerHWMRspTime               Unsigned32,
  pcePcepPeerNumPCReqSent             Counter32,
  pcePcepPeerNumPCReqRcvd             Counter32,
  pcePcepPeerNumPCRepSent             Counter32,
  pcePcepPeerNumPCRepRcvd             Counter32,
  pcePcepPeerNumPCErrSent             Counter32,
  pcePcepPeerNumPCErrRcvd             Counter32,
  pcePcepPeerNumPCNtfSent             Counter32,
  pcePcepPeerNumPCNtfRcvd             Counter32,
  pcePcepPeerNumKeepaliveSent         Counter32,
  pcePcepPeerNumKeepaliveRcvd         Counter32,
  pcePcepPeerNumUnknownRcvd           Counter32,
  pcePcepPeerNumCorruptRcvd           Counter32,
  pcePcepPeerNumReqSent               Counter32,
  pcePcepPeerNumSvecSent              Counter32,
  pcePcepPeerNumSvecReqSent           Counter32,
  pcePcepPeerNumReqSentPendRep        Counter32,
  pcePcepPeerNumReqSentEroRcvd        Counter32,
  pcePcepPeerNumReqSentNoPathRcvd     Counter32,
  pcePcepPeerNumReqSentCancelRcvd     Counter32,
  pcePcepPeerNumReqSentErrorRcvd      Counter32,
  pcePcepPeerNumReqSentTimeout        Counter32,
}
```
pcePcepPeerNumReqSentCancelSent Counter32,
pcePcepPeerNumReqSentClosed Counter32,
pcePcepPeerNumReqRcvd Counter32,
pcePcepPeerNumSvecRcvd Counter32,
pcePcepPeerNumSvecReqRcvd Counter32,
pcePcepPeerNumReqRcvdPendRep Counter32,
pcePcepPeerNumReqRcvdEroSent Counter32,
pcePcepPeerNumReqRcvdNoPathSent Counter32,
pcePcepPeerNumReqRcvdCancelSent Counter32,
pcePcepPeerNumReqRcvdErrorSent Counter32,
pcePcepPeerNumReqRcvdCancelRcvd Counter32,
pcePcepPeerNumReqRcvdClosed Counter32,
pcePcepPeerNumRepRcvdUnknown Counter32,
pcePcepPeerNumReqRcvdUnknown Counter32
}

pcePcepPeerAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The type of the peer’s Internet address. This object specifies how the value of the pcePcepPeerAddr object should be interpreted. Only values unknown(0), ipv4(1), or ipv6(2) are supported."
 ::= { pcePcepPeerEntry 1 }

pcePcepPeerAddr OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The Internet address of the peer. The type is given by pcePcepPeerAddrType."
 ::= { pcePcepPeerEntry 2 }

pcePcepPeerRole OBJECT-TYPE
SYNTAX INTEGER {
    unknown(0),
    pcc(1),
    pce(2),
    pccAndPce(3)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The role that this peer took the last time a session was established. Takes one of the following values."
- unknown(0): this peer’s role is not known.
- pcc(1): this peer is a Path Computation Client (PCC).
- pce(2): this peer is a Path Computation Server (PCE).
- pccAndPce(3): this peer is both a PCC and a PCE.

\[
::= \{ \text{pcePcepPeerEntry 3} \}
\]

\textbf{pcePcepPeerDiscontinuityTime \text{OBJECT-TYPE}}

\begin{verbatim}
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value of sysUpTime at the time that the information and
statistics in this row were last reset."
\end{verbatim}

\[
::= \{ \text{pcePcepPeerEntry 4} \}
\]

\textbf{pcePcepPeerInitiateSession \text{OBJECT-TYPE}}

\begin{verbatim}
SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"Indicates whether the local PCEP entity initiates sessions
to this peer, or waits for the peer to initiate a session."
\end{verbatim}

\[
::= \{ \text{pcePcepPeerEntry 5} \}
\]

\textbf{pcePcepPeerSessionExists \text{OBJECT-TYPE}}

\begin{verbatim}
SYNTAX      TruthValue
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"Indicates whether a session with this peer currently
exists."
\end{verbatim}

\[
::= \{ \text{pcePcepPeerEntry 6} \}
\]

\textbf{pcePcepPeerNumSessSetupOK \text{OBJECT-TYPE}}

\begin{verbatim}
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of PCEP sessions successfully established with
the peer, including any current session. This counter is
incremented each time a session with this peer is
successfully established."
\end{verbatim}

\[
::= \{ \text{pcePcepPeerEntry 7} \}
\]

\textbf{pcePcepPeerNumSessSetupFail \text{OBJECT-TYPE}}

\begin{verbatim}
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
\end{verbatim}
DESCRIPTION
"The number of PCEP sessions with the peer that have been
attempted but failed before being fully established.
This counter is incremented each time a session retry to
this peer fails."
::= { pcePcepPeerEntry 8 }

pcePcepPeerSessionUpTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime the last time a session with this
peer was successfully established.

If pcePcepPeerNumSessSetupOK is zero, then this object
contains zero."
::= { pcePcepPeerEntry 9 }

pcePcepPeerSessionFailTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime the last time a session with this
peer failed to be established.

If pcePcepPeerNumSessSetupFail is zero, then this object
contains zero."
::= { pcePcepPeerEntry 10 }

pcePcepPeerSessionFailUpTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime the last time a session with this
peer failed from active.

If pcePcepPeerNumSessSetupOK is zero, then this object
contains zero."
::= { pcePcepPeerEntry 11 }

pcePcepPeerAvgRspTime OBJECT-TYPE
SYNTAX Unsigned32
UNITS "milliseconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "The average response time for this peer.
  If an average response time has not been calculated for this peer then this object has the value zero.
  If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."
::= { pcePcepPeerEntry 12 }

pcePcepPeerLWMRspTime OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The smallest (low-water mark) response time seen from this peer.
  If no responses have been received from this peer then this object has the value zero.
  If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."
::= { pcePcepPeerEntry 13 }

pcePcepPeerHWMRspTime OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The greatest (high-water mark) response time seen from this peer.
  If no responses have been received from this peer then this object has the value zero.
  If pcePcepPeerRole is pcc then this field is meaningless and is set to zero."
::= { pcePcepPeerEntry 14 }

pcePcepPeerNumPCReqSent OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The number of PCReq messages sent to this peer."
pcePcepPeerNumPCReqRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCReq messages received from this peer."
 ::= { pcePcepPeerEntry 15 }

pcePcepPeerNumPCRepSent OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCRep messages sent to this peer."
 ::= { pcePcepPeerEntry 16 }

pcePcepPeerNumPCRepRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCRep messages received from this peer."
 ::= { pcePcepPeerEntry 17 }

pcePcepPeerNumPCErrSent OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCErr messages sent to this peer."
 ::= { pcePcepPeerEntry 18 }

pcePcepPeerNumPCErrRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCErr messages received from this peer."
 ::= { pcePcepPeerEntry 19 }

pcePcepPeerNumPCNtfSent OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCNtf messages sent to this peer."
 ::= { pcePcepPeerEntry 20 }

pcePcepPeerNumPCNtfRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION  "The number of PCNtf messages received from this peer."
pcePcepPeerNumPCNtfRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCNtf messages received from this peer."

pcePcepPeerNumKeepaliveSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of Keepalive messages sent to this peer."

pcePcepPeerNumKeepaliveRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of Keepalive messages received from this peer."

pcePcepPeerNumUnknownRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of unknown messages received from this peer."

pcePcepPeerNumCorruptRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of corrupted PCEP message received from this peer."

pcePcepPeerNumReqSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent to this peer. A request corresponds 1:1 with an RP object in a PCReq message.

This might be greater than pcepPcepPeerNumPCReqSent because multiple requests can be batched into a single PCReq message."

 ::= { pcepPcepPeerEntry 27 }

pcepPcepPeerNumSvecSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of SVEC objects sent to this peer in PCReq messages. An SVEC object represents a set of synchronized requests."

 ::= { pcepPcepPeerEntry 28 }

pcepPcepPeerNumSvecReqSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent to this peer that appeared in one or more SVEC objects."

 ::= { pcepPcepPeerEntry 29 }

pcepPcepPeerNumReqSentPendRep OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that have been sent to this peer for which a response is still pending."

 ::= { pcepPcepPeerEntry 30 }

pcepPcepPeerNumReqSentEroRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that have been sent to this peer for which a response with an ERO object was received. Such responses indicate that a path was successfully computed by the peer."

 ::= { pcepPcepPeerEntry 31 }

pcepPcepPeerNumReqSentNoPathRcvd OBJECT-TYPE
pcePcepPeerNumReqSentCancelRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests that were cancelled by the peer with a PCNtf message.

This might be different than pcePcepPeerNumPCNtfRcvd because not all PCNtf messages are used to cancel requests, and a single PCNtf message can cancel multiple requests."
::= { pcePcepPeerEntry 33 }

pcePcepPeerNumReqSentErrorRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests that were rejected by the peer with a PCErr message.

This might be different than pcePcepPeerNumPCErrRcvd because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."
::= { pcePcepPeerEntry 34 }

pcePcepPeerNumReqSentTimeout OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests that have been sent to a peer and have been abandoned because the peer has taken too long to respond to them."
::= { pcePcepPeerEntry 35 }

pcePcepPeerNumReqSentCancelSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests that have been sent to this peer for which a response with a NO-PATH object was received. Such responses indicate that the peer could not find a path to satisfy the request."
::= { pcePcepPeerEntry 32 }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of requests that were sent to the peer and
explicitly canceled by the local PCEP entity sending a
PCNtf."
::= { pcePcepPeerEntry 36 }

pcePcepPeerNumReqSentClosed OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of requests that were sent to the peer and
implicitly canceled when the session they were sent over was
closed."
::= { pcePcepPeerEntry 37 }

pcePcepPeerNumReqRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of requests received from this peer. A request
corresponds 1:1 with an RP object in a PCReq message.
This might be greater than pcePcepPeerNumPCReqRcvd because
multiple requests can be batched into a single PCReq
message."
::= { pcePcepPeerEntry 38 }

pcePcepPeerNumSvecRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of SVEC objects received from this peer in PCReq
messages. An SVEC object represents a set of synchronized
requests."
::= { pcePcepPeerEntry 39 }

pcePcepPeerNumSvecReqRcvd OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The number of requests received from this peer that appeared
in one or more SVEC objects."
::= { pcePcepPeerEntry 40 }

pcePcepPeerNumReqRcvdPendRep OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that have been received from this peer for which a response is still pending."
::= { pcePcepPeerEntry 41 }

pcePcepPeerNumReqRcvdEroSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that have been received from this peer for which a response with an ERO object was sent. Such responses indicate that a path was successfully computed by the local PCEP entity."
::= { pcePcepPeerEntry 42 }

pcePcepPeerNumReqRcvdNoPathSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that have been received from this peer for which a response with a NO-PATH object was sent. Such responses indicate that the local PCEP entity could not find a path to satisfy the request."
::= { pcePcepPeerEntry 43 }

pcePcepPeerNumReqRcvdCancelSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests received from this peer that were cancelled by the local PCEP entity sending a PCNtf message.

This might be different than pcePcepPeerNumPCNtfSent because not all PCNtf messages are used to cancel requests, and a single PCNtf message can cancel multiple requests."
::= { pcePcepPeerEntry 44 }

pcePcepPeerNumReqRcvdErrorSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent to the peer at which the peer received an error."
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests received from this peer that were
rejected by the local PCEP entity sending a PCErr message.
This might be different than pcePcepPeerNumPCErrSent because
not all PCErr messages are used to reject requests, and a
single PCErr message can reject multiple requests."
 ::= { pcePcepPeerEntry 45 }

pcePcepPeerNumReqRcvdCancelRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that were received from the peer and
explicitly canceled by the peer sending a PCNtf."
 ::= { pcePcepPeerEntry 46 }

pcePcepPeerNumReqRcvdClosed OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that were received from the peer and
implicitly canceled when the session they were received over
was closed."
 ::= { pcePcepPeerEntry 47 }

pcePcepPeerNumRepRcvdUnknown OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of responses to unknown requests received from
this peer. A response to an unknown request is a response
whose RP object does not contain the request ID of any
request that is currently outstanding on the session."
 ::= { pcePcepPeerEntry 48 }

pcePcepPeerNumReqRcvdUnknown OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of unknown requests that have been received from
a peer. An unknown request is a request whose RP object
contains a request ID of zero."
::= { pcepPcepPeerEntry 49 }

--
-- The PCEP Sessions Table
--

pcePcepSessTable OBJECT-TYPE
SYNTAX      SEQUENCE OF PcePcepSessEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"A table of PCEP sessions that involve the local PCEP entity. Each entry in this table represents a single
session. The entries in this table are read-only.

An entry appears in this table when the corresponding PCEP session transitions out of idle state. If the PCEP session
transitions back into idle state then the corresponding entry in this table is removed."
::= { pcepPcepObjects 3 }

pcePcepSessEntry OBJECT-TYPE
SYNTAX      PcePcepSessEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This entry represents a single PCEP session in which the local PCEP entity participates.

This entry exists only if the corresponding PCEP session has been initialized by some event, such as manual user
configuration, autodiscovery of a peer, or an incoming TCP connection."
INDEX { pcePcepEntityIndex,
pcePcepPeerAddrType,
pcePcepPeerAddr,
pcePcepSessInitiator }
::= { pcePcepSessTable 1 }

PcePcepSessEntry ::= SEQUENCE {
pcePcepSessInitiator                INTEGER,
pcePcepSessStateLastChange          TimeStamp,
pcePcepSessState                    INTEGER,
pcePcepSessConnectRetry             Counter32,
pcePcepSessLocalID                  Unsigned32,
pcePcepSessRemoteID                 Unsigned32,
pcePcepSessKeepaliveTimer           Unsigned32,
pcePcepSessPeerKeepaliveTimer       Unsigned32,
pcePcepSessDeadTimer                Unsigned32,
pcePcepSessPeerDeadTimer            Unsigned32,
pcePcepSessKAHoldTimeRem            Unsigned32,
pcePcepSessOverloaded               TruthValue,
pcePcepSessOverloadTime             Unsigned32,
pcePcepSessPeerOverloaded           TruthValue,
pcePcepSessPeerOverloadTime         Unsigned32,
pcePcepSessDiscontinuityTime        TimeStamp,
pcePcepSessAvgRspTime               Unsigned32,
pcePcepSessLWMRspTime               Unsigned32,
pcePcepSessHWMRspTime               Unsigned32,
pcePcepSessNumPCReqSent             Counter32,
pcePcepSessNumPCReqRcvd             Counter32,
pcePcepSessNumPCRepSent             Counter32,
pcePcepSessNumPCRepRcvd             Counter32,
pcePcepSessNumPCErrSent             Counter32,
pcePcepSessNumPCErrRcvd             Counter32,
pcePcepSessNumPCNtfSent             Counter32,
pcePcepSessNumPCNtfRcvd             Counter32,
pcePcepSessNumKeepaliveSent         Counter32,
pcePcepSessNumKeepaliveRcvd         Counter32,
pcePcepSessNumUnknownRcvd           Counter32,
pcePcepSessNumCorruptRcvd           Counter32,
pcePcepSessNumReqSent               Counter32,
pcePcepSessNumSvecSent              Counter32,
pcePcepSessNumSvecReqSent           Counter32,
pcePcepSessNumReqSentPendRep        Counter32,
pcePcepSessNumReqSentEroRcvd        Counter32,
pcePcepSessNumReqSentNoPathRcvd     Counter32,
pcePcepSessNumReqSentCancelRcvd     Counter32,
pcePcepSessNumReqSentErrorRcvd      Counter32,
pcePcepSessNumReqSentTimeout        Counter32,
pcePcepSessNumReqSentCancelSent     Counter32,
pcePcepSessNumReqRcvd               Counter32,
pcePcepSessNumSvecRcvd              Counter32,
pcePcepSessNumSvecReqRcvd           Counter32,
SYNTAX INTEGER {
    local(1),
    remote(2)
}
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "The initiator of the session, that is, whether the TCP
connection was initiated by the local PCEP entity or the peer.

There is a window during session initialization where two
sessions can exist between a pair of PCEP speakers, each
initiated by one of the speakers. One of these sessions is
always discarded before it leaves OpenWait state. However,
before it is discarded, two sessions to the given peer
appear transiently in this MIB module. The sessions are
distinguished by who initiated them, and so this field is an
index for the pcePcepSessTable."
::= { pcePcepSessEntry 1 }

pcePcepSessStateLastChange OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value of sysUpTime at the time this session entered its
current state as denoted by the pcePcepSessState object."
::= { pcePcepSessEntry 2 }

pcePcepSessState OBJECT-TYPE
SYNTAX INTEGER {
    tcpPending(1),
    openWait(2),
    keepWait(3),
    sessionUp(4)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The current state of the session.

The set of possible states excludes the idle state since
entries do not exist in this table in the idle state."
::= { pcePcepSessEntry 3 }

pcePcepSessConnectRetry OBJECT-TYPE
SYNTAX Counter32
The number of times that the local PCEP entity has attempted to establish a TCP connection for this session without success. The PCEP entity gives up when this reaches pcePcepEntityConnectMaxRetry.

::= { pcePcepSessEntry 4 }

**pcePcepSessLocalID** OBJECT-TYPE
SYNTAX      Unsigned32 (0..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value of the PCEP session ID used by the local PCEP entity in the Open message for this session.

If pcePcepSessState is tcpPending then this is the session ID that will be used in the Open message. Otherwise, this is the session ID that was sent in the Open message."

::= { pcePcepSessEntry 5 }

**pcePcepSessRemoteID** OBJECT-TYPE
SYNTAX      Unsigned32 (0..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The value of the PCEP session ID used by the peer in its Open message for this session.

If pcePcepSessState is tcpPending or openWait then this field is not used and MUST be set to zero."

::= { pcePcepSessEntry 6 }

**pcePcepSessKeepaliveTimer** OBJECT-TYPE
SYNTAX      Unsigned32 (0..255)
UNITS       "seconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The agreed maximum interval at which the local PCEP entity transmits PCEP messages on this PCEP session. Zero means that the local PCEP entity never sends Keepalives on this session.

This field is used if and only if pcePcepSessState is sessionUp. Otherwise, it is not used and MUST be set to zero."
pcePcepSessPeerKeepaliveTimer OBJECT-TYPE
SYNTAX     Unsigned32 (0..255)
UNITS      "seconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The agreed maximum interval at which the peer transmits PCEP
messages on this PCEP session. Zero means that the peer
never sends Keepalives on this session.

This field is used if and only if pcePcepSessState is
sessionUp. Otherwise, it is not used and MUST be set to
zero."

pcePcepSessDeadTimer OBJECT-TYPE
SYNTAX     Unsigned32 (0..255)
UNITS      "seconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The DeadTimer interval for this PCEP session."

pcePcepSessPeerDeadTimer OBJECT-TYPE
SYNTAX     Unsigned32 (0..255)
UNITS      "seconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The peer’s DeadTimer interval for this PCEP session.

If pcePcepSessState is tcpPending or openWait then this
field is not used and MUST be set to zero."

pcePcepSessKAHoldTimeRem OBJECT-TYPE
SYNTAX     Unsigned32 (0..255)
UNITS      "seconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The keep alive hold time remaining for this session.

If pcePcepSessState is tcpPending or openWait then this
field is not used and MUST be set to zero."
pcePcepSessOverloaded OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"If the local PCEP entity has informed the peer that it is currently overloaded, then this is set to true. Otherwise, it is set to false."
 ::= { pcePcepSessEntry 11 }  

pcePcepSessOverloadTime OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The interval of time that is remaining until the local PCEP entity will cease to be overloaded on this session.
This field is only used if pcePcepSessOverloaded is set to true. Otherwise, it is not used and MUST be set to zero."
 ::= { pcePcepSessEntry 12 }  

pcePcepSessPeerOverloaded OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"If the peer has informed the local PCEP entity that it is currently overloaded, then this is set to true. Otherwise, it is set to false."
 ::= { pcePcepSessEntry 13 }  

pcePcepSessPeerOverloadTime OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The interval of time that is remaining until the peer will cease to be overloaded. If it is not known how long the peer will stay in overloaded state, this field is set to zero.
This field is only used if pcePcepSessPeerOverloaded is set to true. Otherwise, it is not used and MUST be set to zero."
 ::= { pcePcepSessEntry 14 }
zero.
::= { pcePcepSessEntry 15 }

pcePcepSessDiscontinuityTime OBJECT-TYPE
SYNTAX     TimeStamp
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The value of sysUpTime at the time that the statistics in
this row were last reset."
::= { pcePcepSessEntry 16 }

pcePcepSessAvgRspTime OBJECT-TYPE
SYNTAX     Unsigned32
UNITS "milliseconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The average response time for this peer on this session.
If an average response time has not been calculated for this
peer then this object has the value zero."
::= { pcePcepSessEntry 17 }

pcePcepSessLWMRspTime OBJECT-TYPE
SYNTAX     Unsigned32
UNITS "milliseconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The smallest (low-water mark) response time seen from this
peer on this session.
If no responses have been received from this peer then this
object has the value zero."
::= { pcePcepSessEntry 18 }

pcePcepSessHWMRspTime OBJECT-TYPE
SYNTAX     Unsigned32
UNITS "milliseconds"
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The greatest (high-water mark) response time seen from this
peer on this session.
If no responses have been received from this peer then this
object has the value zero."
::= { pcePcepSessEntry 19 }

pcePcepSessNumPCReqSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCReq messages sent on this session."
::= { pcePcepSessEntry 20 }

pcePcepSessNumPCReqRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCReq messages received on this session."
::= { pcePcepSessEntry 21 }

pcePcepSessNumPCRepSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCRep messages sent on this session."
::= { pcePcepSessEntry 22 }

pcePcepSessNumPCRepRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCRep messages received on this session."
::= { pcePcepSessEntry 23 }

pcePcepSessNumPCErrSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCErr messages sent on this session."
::= { pcePcepSessEntry 24 }

pcePcepSessNumPCErrRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of PCErr messages received on this session."
<table>
<thead>
<tr>
<th>Object Description</th>
<th>Syntax</th>
<th>Access</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of PCNtf messages sent on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of PCNtf messages sent on this session.</td>
</tr>
<tr>
<td>Number of PCNtf messages received on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of PCNtf messages received on this session.</td>
</tr>
<tr>
<td>Number of Keepalive messages sent on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of Keepalive messages sent on this session.</td>
</tr>
<tr>
<td>Number of Keepalive messages received on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of Keepalive messages received on this session.</td>
</tr>
<tr>
<td>Number of unknown messages received on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of unknown messages received on this session.</td>
</tr>
<tr>
<td>Number of corrupted PCEP message received on this session.</td>
<td>Counter32</td>
<td>read-only</td>
<td>current</td>
<td>The number of corrupted PCEP message received on this session.</td>
</tr>
</tbody>
</table>
pcePcepSessEntry 31

pcePcepSessNumReqSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests sent on this session. A request corresponds 1:1 with an RP object in a PCReq message. This might be greater than pcePcepSessNumPCReqSent because multiple requests can be batched into a single PCReq message."

::= { pcePcepSessEntry 32 }

pcePcepSessNumSvecSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of SVEC objects sent on this session in PCReq messages. An SVEC object represents a set of synchronized requests."

::= { pcePcepSessEntry 33 }

pcePcepSessNumSvecReqSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests sent on this session that appeared in one or more SVEC objects."

::= { pcePcepSessEntry 34 }

pcePcepSessNumReqSentPendRep OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of requests that have been sent on this session for which a response is still pending."

::= { pcePcepSessEntry 35 }

pcePcepSessNumReqSentEroRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of successful responses received on this session. A response corresponds 1:1 with an RP object in a PCRep message. A successful response is a response for which an ERO was successfully computed."
::= { pcePcepSessEntry 36 }

pcePcepSessNumReqSentNoPathRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of unsuccessful responses received on this session. A response corresponds 1:1 with an RP object in a PCRep message. An unsuccessful response is a response with a NO-PATH object."
::= { pcePcepSessEntry 37 }

pcePcepSessNumReqSentCancelRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent on this session that were cancelled by the peer with a PCNtf message. This might be different than pcePcepSessNumPCNtfRcvd because not all PCNtf messages are used to cancel requests, and a single PCNtf message can cancel multiple requests."
::= { pcePcepSessEntry 38 }

pcePcepSessNumReqSentErrorRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent on this session that were rejected by the peer with a PCErr message. This might be different than pcePcepSessNumPCErrRcvd because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."
::= { pcePcepSessEntry 39 }

pcePcepSessNumReqSentTimeout OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent on this session that have been
sent to a peer and have been abandoned because the peer has
taken too long to respond to them."
 ::= { pcePcepSessEntry 40 }

pcePcepSessNumReqSentCancelSent OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests sent on this session that were sent
to the peer and explicitly canceled by the local PCEP
entity sending a PCNtf."
 ::= { pcePcepSessEntry 41 }

pcePcepSessNumReqRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests received on this session. A request
 corresponds 1:1 with an RP object in a PCReq message.
This might be greater than pcePcepSessNumPCReqRcvd because
multiple requests can be batched into a single PCReq
message."
 ::= { pcePcepSessEntry 42 }

pcePcepSessNumSvecRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of SVEC objects received on this session in PCReq
messages. An SVEC object represents a set of synchronized
requests."
 ::= { pcePcepSessEntry 43 }

pcePcepSessNumSvecReqRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests received on this session that
appeared in one or more SVEC objects."
 ::= { pcePcepSessEntry 44 }
pcePcepSessNumReqRcvdPendRep OBJECT-TYPE
SYNTAX     Counter32
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The number of requests that have been received on this
session for which a response is still pending."
 ::= { pcePcepSessEntry 45 }

pcePcepSessNumReqRcvdEroSent OBJECT-TYPE
SYNTAX     Counter32
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The number of successful responses sent on this session.  A
response corresponds 1:1 with an RP object in a PCRep
message.  A successful response is a response for which an
ERO was successfully computed."
 ::= { pcePcepSessEntry 46 }

pcePcepSessNumReqRcvdNoPathSent OBJECT-TYPE
SYNTAX     Counter32
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The number of unsuccessful responses sent on this session.  A
response corresponds 1:1 with an RP object in a PCRep
message.  An unsuccessful response is a response with a
NO-PATH object."
 ::= { pcePcepSessEntry 47 }

pcePcepSessNumReqRcvdCancelSent OBJECT-TYPE
SYNTAX     Counter32
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The number of requests received on this session that were
cancelled by the local PCEP entity sending a PCNtf message.
This might be different than pcePcepSessNumPCNtfSent because
not all PCNtf messages are used to cancel requests, and a
single PCNtf message can cancel multiple requests."
 ::= { pcePcepSessEntry 48 }

pcePcepSessNumReqRcvdErrorSent OBJECT-TYPE
SYNTAX     Counter32
MAX-ACCESS read-only
STATUS     current
DESCRIPTION
"The number of requests received on this session that were rejected by the local PCEP entity sending a PCErr message. This might be different than pcePcepSessNumPCErrSent because not all PCErr messages are used to reject requests, and a single PCErr message can reject multiple requests."
::= { pcePcepSessEntry 49 }

pcePcepSessNumReqRcvdCancelRcvd OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of requests that were received on this session and explicitly canceled by the peer sending a PCNtf."
::= { pcePcepSessEntry 50 }

pcePcepSessNumRepRcvdUnknown OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of responses to unknown requests received on this session. A response to an unknown request is a response whose RP object does not contain the request ID of any request that is currently outstanding on the session."
::= { pcePcepSessEntry 51 }

pcePcepSessNumReqRcvdUnknown OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of unknown requests that have been received on this session. An unknown request is a request whose RP object contains a request ID of zero."
::= { pcePcepSessEntry 52 }

---
--- Notifications Configuration
---

pcePcepNotificationsMaxRate OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This variable indicates the maximum number of notifications issued per second. If events occur more rapidly, the implementation may simply fail to emit these notifications during that period, or may queue them until an appropriate time. A value of 0 means no notifications are emitted and all should be discarded (that is, not queued)."

::= { pcePcepObjects 4 }

--- Notifications ---

pcePcepSessUp NOTIFICATION-TYPE
OBJECTS     {
              pcePcepSessState,
              pcePcepSessStateLastChange
            }
STATUS      current
DESCRIPTION  
"This notification is sent when the value of 'pcePcepSessState' enters the 'sessionUp' state."
::= { pcePcepNotifications 1 }

pcePcepSessDown NOTIFICATION-TYPE
OBJECTS     {
              pcePcepSessState,
              pcePcepSessStateLastChange
            }
STATUS      current
DESCRIPTION  
"This notification is sent when the value of 'pcePcepSessState' leaves the 'sessionUp' state."
::= { pcePcepNotifications 2 }

pcePcepSessLocalOverload NOTIFICATION-TYPE
OBJECTS     {
              pcePcepSessOverloaded,
              pcePcepSessOverloadTime
            }
STATUS      current
DESCRIPTION  
"This notification is sent when the local PCEP entity enters overload state for a peer."
::= { pcePcepNotifications 3 }

pcePcepSessLocalOverloadClear NOTIFICATION-TYPE
OBJECTS     {

This notification is sent when the local PCEP entity leaves overload state for a peer.
::= { pcepPcepNotifications 4 }  

pcepPcepSessPeerOverload NOTIFICATION-TYPE
OBJECTS
   { pcepPcepSessPeerOverloaded, pcepPcepSessPeerOverloadTime }  

This notification is sent when a peer enters overload state.
::= { pcepPcepNotifications 5 }  

pcepPcepSessPeerOverloadClear NOTIFICATION-TYPE
OBJECTS
   { pcepPcepSessPeerOverloaded }  

This notification is sent when a peer leaves overload state.
::= { pcepPcepNotifications 6 }  

--  
-- Module Conformance Statement  
--  

pcepPcepCompliances
   OBJECT IDENTIFIER ::= { pcepPcepConformance 1 }  

pcepPcepGroups
   OBJECT IDENTIFIER ::= { pcepPcepConformance 2 }  

--  
-- Read-Only Compliance  
--  

pcepPcepModuleReadOnlyCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
   "The Module is implemented with support for read-only. In other words, only monitoring is available by implementing
this MODULE-COMPLIANCE."

MODULE -- this module
MANDATORY-GROUPS {
    pcePcepGeneralGroup,
    pcePcepNotificationsGroup
}

OBJECT       pcePcepEntityAddrType
SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
DESCRIPTION  "Only unknown(0), ipv4(1) and ipv6(2) support
is required."

OBJECT       pcePcepPeerAddrType
SYNTAX       InetAddressType { unknown(0), ipv4(1), ipv6(2) }
DESCRIPTION  "Only unknown(0), ipv4(1) and ipv6(2) support
is required."

::= { pcePcepCompliances 1 }

-- units of conformance

pcePcepGeneralGroup OBJECT-GROUP
OBJECTS { pcePcepEntityAdminStatus,
    pcePcepEntityOperStatus,
    pcePcepEntityAddrType,
    pcePcepEntityAddr,
    pcePcepEntityConnectTimer,
    pcePcepEntityConnectMaxRetry,
    pcePcepEntityInitBackoffTimer,
    pcePcepEntityMaxBackoffTimer,
    pcePcepEntityOpenWaitTimer,
    pcePcepEntityKeepWaitTimer,
    pcePcepEntityKeepAliveTimer,
    pcePcepEntityDeadTimer,
    pcePcepEntityAllowNegotiation,
    pcePcepEntityMaxKeepAliveTimer,
    pcePcepEntityMaxDeadTimer,
    pcePcepEntityMinKeepAliveTimer,
    pcePcepEntityMinDeadTimer,
    pcePcepEntitySyncTimer,
    pcePcepEntityRequestTimer,
    pcePcepEntityMaxSessions,
    pcePcepEntityMaxUnknownReqs,
    pcePcepEntityMaxUnknownMsgs,
    pcePcepPeerRole,
    pcePcepPeerDiscontinuityTime,
    pcePcepPeerInitiateSession,
pcePcepPeerSessionExists,
pcePcepPeerNumSessSetupOK,
pcePcepPeerNumSessSetupFail,
pcePcepPeerSessionUpTime,
pcePcepPeerSessionFailTime,
pcePcepPeerSessionFailUpTime,
pcePcepPeerAvgRspTime,
pcePcepPeerLowRspTime,
pcePcepPeerHighRspTime,
pcePcepPeerNumPCReqSent,
pcePcepPeerNumPCReqRcvd,
pcePcepPeerNumPCRepSent,
pcePcepPeerNumPCRepRcvd,
pcePcepPeerNumPCErrSent,
pcePcepPeerNumPCErrRcvd,
pcePcepPeerNumPCNtfSent,
pcePcepPeerNumPCNtfRcvd,
pcePcepPeerNumKeepaliveSent,
pcePcepPeerNumKeepaliveRcvd,
pcePcepPeerNumUnknownRcvd,
pcePcepPeerNumUnknownRcvd,
pcePcepPeerNumCorruptRcvd,
pcePcepPeerNumReqSent,
pcePcepPeerNumSvecSent,
pcePcepPeerNumSvecRcvd,
pcePcepPeerNumSvecReqSent,
pcePcepPeerNumSvecReqRcvd,
pcePcepPeerNumReqSentPendRep,
pcePcepPeerNumReqSentEroRcvd,
pcePcepPeerNumReqSentNoPathRcvd,
pcePcepPeerNumReqSentCancelRcvd,
pcePcepPeerNumReqSentErrRcvd,
pcePcepPeerNumReqSentTimeout,
pcePcepPeerNumReqSentCancelSent,
pcePcepPeerNumReqSentClosed,
pcePcepPeerNumReqRcvd,
pcePcepPeerNumSvecRcvd,
pcePcepPeerNumSvecReqRcvd,
pcePcepPeerNumSvecReqRcvdPendRep,
pcePcepPeerNumReqRcvdEroSent,
pcePcepPeerNumReqRcvdNoPathSent,
pcePcepPeerNumReqRcvdCancelSent,
pcePcepPeerNumReqRcvdErrorSent,
pcePcepPeerNumReqRcvdCancelRcvd,
pcePcepPeerNumReqRcvdClosed,
pcePcepPeerNumRepRcvdUnknown,
pcePcepPeerNumRepRcvdUnknown,
pcePcepSessStateLastChange,
pcePcepSessState,
pcePcepSessConnectRetry,
pcePcepSessLocalID,
pcePcepSessRemoteID,
pcePcepSessKeepaliveTimer,
pcePcepSessPeerKeepaliveTimer,
pcePcepSessDeadTimer,
pcePcepSessPeerDeadTimer,
pcePcepSessKAHoldTimeRem,
pcePcepSessOverloaded,
pcePcepSessOverloadTime,
pcePcepSessPeerOverloaded,
pcePcepSessPeerOverloadTime,
pcePcepSessDiscontinuityTime,
pcePcepSessAvgRspTime,
pcePcepSessLWMRspTime,
pcePcepSessHMRspTime,
pcePcepSessNumPCReqSent,
pcePcepSessNumPCReqRcvd,
pcePcepSessNumPCRepSent,
pcePcepSessNumPCRepRcvd,
pcePcepSessNumPCErrSent,
pcePcepSessNumPCErrRcvd,
pcePcepSessNumPCNtfSent,
pcePcepSessNumPCNtfRcvd,
pcePcepSessNumKeepaliveSent,
pcePcepSessNumKeepaliveRcvd,
pcePcepSessNumUnknownRcvd,
pcePcepSessNumCorruptRcvd,
pcePcepSessNumReqSent,
pcePcepSessNumSvecSent,
pcePcepSessNumSvecReqSent,
pcePcepSessNumReqSentPendRep,
pcePcepSessNumReqSentEroRcvd,
pcePcepSessNumReqSentNoPathRcvd,
pcePcepSessNumReqSentCancelRcvd,
pcePcepSessNumReqSentErrorRcvd,
pcePcepSessNumReqSentTimeout,
pcePcepSessNumReqSentCancelSent,
pcePcepSessNumReqRcvd,
pcePcepSessNumSvecRcvd,
pcePcepSessNumSvecReqRcvd,
pcePcepSessNumReqRcvdPendRep,
pcePcepSessNumReqRcvdEroSent,
pcePcepSessNumReqRcvdNoPathSent,
pcePcepSessNumReqRcvdCancelSent,
pcePcepSessNumReqRcvdErrorSent,
pcePcepSessNumReqRcvdCancelRcvd,
pcePcepSessNumRepRcvdUnknown,
5. Security Considerations

The pcePcepNotificationsMaxRate object defined in this MIB module has a MAX-ACCESS clause of read-write. 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. In particular, pcePcepNotificationsMaxRate may be used improperly to stop notifications being issued, or to permit a flood of notifications to be sent to the management agent at a high rate.

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 since, collectively, they provide information about the amount and frequency of path computation requests and responses within the network and can reveal some aspects of its configuration. 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.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), 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.
Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

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.

6. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>OBJECT IDENTIFIER value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcePcepMIB</td>
<td>{ mib-2 XXX }</td>
</tr>
</tbody>
</table>

Editor’s Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the ‘mib-2’ subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

7. Acknowledgement

The authors would like to thank Santanu Mazumder, Meral Shirazipour and Adrian Farrel for their valuable input.

Funding for the RFC Editor function is currently provided by the Internet Society.

8. References

8.1. Normative References

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Leela Palace
Bangalore, Karnataka  560008
India
EMail: dhruv.ietf@gmail.com

Appendix B. PCEP MIB Module Example

This example considers the set of PCC / PCE relationships shown in the following figure. The example shows the contents of the PCEP MIB module as read at PCE2 and PCCb.

```
PCE1---PCE2   PCE3
   |   /  |    /  |
   |  /   |   /   |
PCCa/ PCCb   PCCc
```

The IP addresses of the PCE speakers in this diagram are given in the following table.

```
+----+-------------+
| PCE1 | 1.1.1.1     |
+----+-------------+
| PCE2 | 2.2.2.2     |
+----+-------------+
| PCE3 | 3.3.3.3     |
+----+-------------+
| PCCa | 11.11.11.11 |
+----+-------------+
| PCCB | 22.22.22.22 |
+----+-------------+
| PCCC | 33.33.33.33 |
+----+-------------+
```

In this example, the PCEP session between PCCb and PCE3 is currently down.

B.1. Contents of PCEP MIB module at PCE2

At PCE2, there is a single local PCEP entity which has three peers (PCCa, PCCb and PCE1). There is a session active to all of these peers.
The contents of the PCEP MIB module as read at PCE2 are as follows.

In pcePcepEntityTable {
    pcePcepEntityIndex 1,
    pcePcepEntityAdminStatus adminStatusUp(1),
    pcePcepEntityOperStatus operStatusUp(1),
    pcePcepEntityAddrType ipv4(1),
    pcePcepEntityAddr 2.2.2.2, -- PCE2
    pcePcepEntityConnectTimer 60,
    pcePcepEntityConnectMaxRetry 5,
    pcePcepEntityInitBackoffTimer 30,
    pcePcepEntityMaxBackoffTimer 3600,
    pcePcepEntityOpenWaitTimer 60,
    pcePcepEntityKeepWaitTimer 60,
    pcePcepEntityKeepAliveTimer 1,
    pcePcepEntityDeadTimer 4,
    pcePcepEntityAllowNegotiation true(1),
    pcePcepEntityMaxKeepAliveTimer 60,
    pcePcepEntityMaxDeadTimer 240,
    pcePcepEntityMinKeepAliveTimer 1,
    pcePcepEntityMinDeadTimer 4,
    pcePcepEntitySyncTimer 60,
    pcePcepEntityRequestTimer 120,
    pcePcepEntityMaxSessions 999,
    pcePcepEntityMaxUnknownReqs 5,
    pcePcepEntityMaxUnknownMsgs 5
}

In pcePcepPeerTable {
    pcePcepPeerAddrType ipv4(1), -- PCE1
    pcePcepPeerAddr 1.1.1.1,
    pcePcepPeerRole pccAndPce(3),
    pcePcepPeerDiscontinuityTime TimeStamp,
    pcePcepPeerInitiateSession true(1),
    pcePcepPeerSessionExists true(1),
    pcePcepPeerNumSessSetupOK 1,
    pcePcepPeerNumSessSetupFail 0,
    pcePcepPeerSessionUpTime TimeStamp,
    pcePcepPeerSessionFailTime 0,
    pcePcepPeerSessionFailUpTime TimeStamp,
    pcePcepPeerAvgRspTime 0,
    pcePcepPeerLWRspTime 0,
    pcePcepPeerHWRspTime 0,
    pcePcepPeerNumPCReqSent 0,
    pcePcepPeerNumPCReqRcvd 0,
    pcePcepPeerNumPCRepSent 0,
    pcePcepPeerNumPCRepRcvd 0,
    pcePcepPeerNumPCErrSent 0,
pcePcepPeerNumPCErrRcvd 0,
pcePcepPeerNumPCNtfSent 0,
pcePcepPeerNumPCNtfRcvd 0,
pcePcepPeerNumKeepaliveSent 123,
pcePcepPeerNumKeepaliveRcvd 123,
pcePcepPeerNumUnknownRcvd 0,
pcePcepPeerNumCorruptRcvd 0,
pcePcepPeerNumReqSent 0,
pcePcepPeerNumSvecSent 0,
pcePcepPeerNumSvecReqSent 0,
pcePcepPeerNumReqSentPendRep 0,
pcePcepPeerNumReqSentEroRcvd 0,
pcePcepPeerNumReqSentNoPathRcvd 0,
pcePcepPeerNumReqSentCancelRcvd 0,
pcePcepPeerNumReqSentErrorRcvd 0,
pcePcepPeerNumReqSentTimeout 0,
pcePcepPeerNumReqSentCancelSent 0,
pcePcepPeerNumReqSentClosed 0,
pcePcepPeerNumReqRcvd 0,
pcePcepPeerNumSvecRcvd 0,
pcePcepPeerNumSvecReqRcvd 0,
pcePcepPeerNumReqRcvdPendRep 0,
pcePcepPeerNumReqRcvdEroSent 0,
pcePcepPeerNumReqRcvdNoPathSent 0,
pcePcepPeerNumReqRcvdCancelSent 0,
pcePcepPeerNumReqRcvdErrorSent 0,
pcePcepPeerNumReqRcvdCancelRcvd 0,
pcePcepPeerNumReqRcvdClosed 0,
pcePcepPeerNumRepRcvdUnknown 0,
pcePcepPeerNumReqRcvdUnknown 0
}

pcePcepPeerAddrType ipv4(1), --PCCa
pcePcepPeerAddr 11.11.11.11,
pcePcepPeerRole pcc(1),
pcePcepPeerDiscontinuityTime TimeStamp,
pcePcepPeerInitiateSession false(0),
pcePcepPeerSessionExists true(1),
pcePcepPeerNumSessSetupOK 1,
pcePcepPeerNumSessSetupFail 0,
pcePcepPeerSessionUpTime TimeStamp,
pcePcepPeerSessionFailTime 0,
pcePcepPeerSessionFailUpTime TimeStamp,
pcePcepPeerAvgRspTime 200,
pcePcepPeerLWMRspTime 100,
pcePcepPeerHWMRspTime 300,
pcePcepPeerNumPCReqSent 0,
pcePcepPeerNumPCReqRcvd 3,
pcePcepPeerNumPCRepSent            3,
pcePcepPeerNumPCRepRcvd             0,
pcePcepPeerNumPCErrSent             0,
pcePcepPeerNumPCErrRcvd             0,
pcePcepPeerNumPCNtfSent             0,
pcePcepPeerNumPCNtfRcvd             0,
pcePcepPeerNumKeepaliveSent         123,
pcePcepPeerNumKeepaliveRcvd         123,
pcePcepPeerNumUnknownRcvd           0,
pcePcepPeerNumCorruptRcvd           0,
pcePcepPeerNumReqSent               0,
pcePcepPeerNumSvecSent              0,
pcePcepPeerNumSvecRcvd              0,
pcePcepPeerNumSvecReqSent           0,
pcePcepPeerNumSvecRcvd              0,
pcePcepPeerNumReqRcvd               3,
pcePcepPeerNumReqRcvdUnknown        0,
pcePcepPeerAddrType                 ipv4(1), -- PCCb
pcePcepPeerAddr                     22.22.22.22,
pcePcepPeerRole                     pcc(1),
pcePcepPeerDiscontinuityTime        TimeStamp,
pcePcepPeerInitiateSession          true(1),
pcePcepPeerSessionExists            true(1),
pcePcepPeerNumSessSetupOK           1,
pcePcepPeerNumSessSetupFail         0,
pcePcepPeerSessionUpTime            TimeStamp,
pcePcepPeerSessionFailTime          0,
pcePcepPeerSessionFailUpTime        TimeStamp,
pcePcepPeerAvgRspTime               200,
pcePcepPeerLWMRspTime               100,
Internet-Draft                  PCEP MIB                    October 2014

pcePcepPeerHWMRspTime               300,
pcePcepPeerNumPCReqSent             0,
pcePcepPeerNumPCReqRcvd             4,
pcePcepPeerNumPCRepSent             4,
pcePcepPeerNumPCRepRcvd             0,
pcePcepPeerNumPCErrSent             0,
pcePcepPeerNumPCErrRcvd             0,
pcePcepPeerNumPCNtfSent             0,
pcePcepPeerNumPCNtfRcvd             0,
pcePcepPeerNumKeepaliveSent         123,
pcePcepPeerNumKeepaliveRcvd         123,
pcePcepPeerNumUnknownRcvd           0,
pcePcepPeerNumCorruptRcvd           0,
pcePcepPeerNumReqSent               0,
pcePcepPeerNumSvecSent              0,
pcePcepPeerNumSvecReqSent           0,
pcePcepPeerNumReqSentPendRep        0,
pcePcepPeerNumReqSentEroRcvd        0,
pcePcepPeerNumReqSentNoPathRcvd     0,
pcePcepPeerNumReqSentCancelRcvd     0,
pcePcepPeerNumReqSentErrorRcvd      0,
pcePcepPeerNumReqSentTimeout        0,
pcePcepPeerNumReqSentCancelSent     0,
pcePcepPeerNumReqSentClosed         0,
pcePcepPeerNumReqRcvd               4,
pcePcepPeerNumSvecRcvd              0,
pcePcepPeerNumSvecReqRcvd           0,
pcePcepPeerNumReqRcvdPendRep        0,
pcePcepPeerNumReqRcvdEroSent        3,
pcePcepPeerNumReqRcvdNoPathSent     1,
pcePcepPeerNumReqRcvdCancelSent     0,
pcePcepPeerNumReqRcvdErrorSent      0,
pcePcepPeerNumReqRcvdCancelRcvd     0,
pcePcepPeerNumReqRcvdClosed         0,
pcePcepPeerNumReqRcvdUnknown        0,
pcePcepPeerNumReqRcvdUnknown        0,

In pcePcepSessTable {
  pcePcepSessInitiator                local(1), --PCE1
  pcePcepSessStateLastChange          TimeStamp,
  pcePcepSessState                    sessionUp(4),
  pcePcepSessConnectRetry             0,
  pcePcepSessLocalID                  1,
  pcePcepSessRemoteID                 2,
  pcePcepSessKeepaliveTimer           1,
  pcePcepSessPeerKeepaliveTimer       1,
  pcePcepSessDeadTimer                4,
pcePcepSessPeerDeadTimer 4,
pcePcepSessKAHoldTimeRem 1,
pcePcepSessOverloaded false(0),
pcePcepSessOverloadTime 0,
pcePcepSessPeerOverloaded false(0),
pcePcepSessPeerOverloadTime 0,
pcePcepSessDiscontinuityTime TimeStamp,
pcePcepSessAvgRspTime 0,
pcePcepSessLWMRspTime 0,
pcePcepSessHWMRspTime 0,
pcePcepSessNumPCReqSent 0,
pcePcepSessNumPCRepSent 0,
pcePcepSessNumPCReqRcvd 0,
pcePcepSessNumPCRepRcvd 0,
pcePcepSessNumPCErrSent 0,
pcePcepSessNumPCErrRcvd 0,
pcePcepSessNumPCNtfSent 0,
pcePcepSessNumPCNtfRcvd 0,
pcePcepSessNumKeepaliveSent 123,
pcePcepSessNumKeepaliveRcvd 123,
pcePcepSessNumUnknownRcvd 0,
pcePcepSessNumCorruptRcvd 0,
pcePcepSessNumReqSent 0,
pcePcepSessNumSvecSent 0,
pcePcepSessNumSvecReqSent 0,
pcePcepSessNumReqSentPendRep 0,
pcePcepSessNumReqSentEroRcvd 0,
pcePcepSessNumReqSentNoPathRcvd 0,
pcePcepSessNumReqSentCancelRcvd 0,
pcePcepSessNumReqSentErrorRcvd 0,
pcePcepSessNumReqSentTimeout 0,
pcePcepSessNumReqSentCancelSent 0,
pcePcepSessNumReqRcvd 0,
pcePcepSessNumSvecRcvd 0,
pcePcepSessNumSvecReqRcvd 0,
pcePcepSessNumReqRcvdPendRep 0,
pcePcepSessNumReqRcvdEroSent 0,
pcePcepSessNumReqRcvdNoPathSent 0,
pcePcepSessNumReqRcvdCancelSent 0,
pcePcepSessNumReqRcvdErrorSent 0,
pcePcepSessNumReqRcvdCancelRcvd 0,
pcePcepSessNumRepRcvdUnknown 0,
pcePcepSessNumReqRcvdUnknown 0
}
{
  pcePcepSessInitiator remote(2), --PCCA
  pcePcepSessStateLastChange TimeStamp,
  pcePcepSessState sessionUp(4),
}

pcePcepSessConnectRetry 0,
pcePcepSessLocalID 2,
pcePcepSessRemoteID 1,
pcePcepSessKeepaliveTimer 1,
pcePcepSessPeerKeepaliveTimer 1,
pcePcepSessDeadTimer 4,
pcePcepSessPeerDeadTimer 4,
pcePcepSessKAHoldTimeRem 1,
pcePcepSessOverloaded false(0),
pcePcepSessOverloadTime 0,
pcePcepSessPeerOverloaded false(0),
pcePcepSessPeerOverloadTime 0,
pcePcepSessDiscontinuityTime TimeStamp,
pcePcepSessAvgRspTime 200,
pcePcepSessLWMRspTime 100,
pcePcepSessHWMRspTime 300,
pcePcepSessNumPCReqSent 0,
pcePcepSessNumPCReqRcvd 1,
pcePcepSessNumPCRepSent 1,
pcePcepSessNumPCRepRcvd 0,
pcePcepSessNumPCErrSent 0,
pcePcepSessNumPCErrRcvd 0,
pcePcepSessNumPCNtfSent 0,
pcePcepSessNumPCNtfRcvd 0,
pcePcepSessNumKeepaliveSent 123,
pcePcepSessNumKeepaliveRcvd 123,
pcePcepSessNumUnknownRcvd 0,
pcePcepSessNumUnknownRcvd 0,
pcePcepSessNumReqSent 0,
pcePcepSessNumSvecSent 0,
pcePcepSessNumSvecRcvd 0,
pcePcepSessNumReqSentPendRep 0,
pcePcepSessNumReqSentEroRcvd 0,
pcePcepSessNumReqSentNoPathRcvd 0,
pcePcepSessNumReqSentCancelRcvd 0,
pcePcepSessNumReqSentErrorRcvd 0,
pcePcepSessNumReqSentTimeout 0,
pcePcepSessNumReqSentCancelSent 0,
pcePcepSessNumReqSent 3,
pcePcepSessNumSvecRcvd 0,
pcePcepSessNumSvecRcvd 0,
pcePcepSessNumReqRcvdPendRep 0,
pcePcepSessNumReqRcvdEroSent 3,
pcePcepSessNumReqRcvdNoPathSent 0,
pcePcepSessNumReqRcvdCancelSent 0,
pcePcepSessNumReqRcvdErrorSent 0,
pcePcepSessNumReqRcvdCancelRcvd 0,
pcePcepSessNumRepRcvdUnknown 0,
pcePcepSessNumReqRcvdUnknown 0,
}
{
pcePcepSessInitiator remote(2), --PCCh
pcePcepSessStateLastChange TimeStamp,
pcePcepSessState sessionUp(4),
pcePcepSessConnectRetry 0,
pcePcepSessLocalID 2,
pcePcepSessRemoteID 1,
pcePcepSessKeepaliveTimer 1,
pcePcepSessPeerKeepaliveTimer 1,
pcePcepSessDeadTimer 4,
pcePcepSessPeerDeadTimer 4,
pcePcepSessKAHoldTimeRem 1,
pcePcepSessOverloaded false(0),
pcePcepSessOverloadTime 0,
pcePcepSessPeerOverloaded false(0),
pcePcepSessPeerOverloadTime 0,
pcePcepSessDiscontinuityTime TimeStamp,
pcePcepSessAvgRspTime 200,
pcePcepSessLWMRspTime 100,
pcePcepSessHWMRspTime 300,
pcePcepSessNumPCReqSent 0,
pcePcepSessNumPCReqRcvd 4,
pcePcepSessNumPCRepSent 4,
pcePcepSessNumPCRepRcvd 0,
pcePcepSessNumPCErrSent 0,
pcePcepSessNumPCErrRcvd 0,
pcePcepSessNumPCNtfSent 0,
pcePcepSessNumPCNtfRcvd 0,
pcePcepSessNumKeepaliveSent 123,
pcePcepSessNumKeepaliveRcvd 123,
pcePcepSessNumUnknownRcvd 0,
pcePcepSessNumCorruptRcvd 0,
pcePcepSessNumReqSent 0,
pcePcepSessNumSvecSent 0,
pcePcepSessNumSvecReqSent 0,
pcePcepSessNumReqSentPendRep 0,
pcePcepSessNumReqSentEroRcvd 0,
pcePcepSessNumReqSentNoPathRcvd 0,
pcePcepSessNumReqSentCancelRcvd 0,
pcePcepSessNumReqSentErrorRcvd 0,
pcePcepSessNumReqSentTimeout 0,
pcePcepSessNumReqSentCancelSent 0,
pcePcepSessNumReqRcvd 4,
pcePcepSessNumSvecRcvd 0,
pcePcepSessNumSvecReqRcvd 0,
pcePcepSessNumReqRcvdPendRep 0,
B.2. Contents of PCEP MIB module at PCCb

At PCCb, there is a single local PCEP entity which has two peers (PCE2 and PCE3). There is a session active to PCE2, but the session to PCE3 is currently down.

The contents of the PCEP MIB module as read at PCCb are as follows.

In pcePcepEntityTable {
  pcePcepEntityIndex 1,
  pcePcepEntityAdminStatus adminStatusUp(1),
  pcePcepEntityOperStatus operStatusUp(1),
  pcePcepEntityAddrType ipv4(1),
  pcePcepEntityAddr 22.22.22.22, -- PCCb
  pcePcepEntityConnectTimer 60,
  pcePcepEntityConnectMaxRetry 5,
  pcePcepEntityInitBackoffTimer 30,
  pcePcepEntityMaxBackoffTimer 3600,
  pcePcepEntityOpenWaitTimer 60,
  pcePcepEntityKeepWaitTimer 60,
  pcePcepEntityKeepAliveTimer 1,
  pcePcepEntityDeadTimer 4,
  pcePcepEntityAllowNegotiation true(1),
  pcePcepEntityMaxKeepAliveTimer 60,
  pcePcepEntityMaxDeadTimer 240,
  pcePcepEntityMinKeepAliveTimer 1,
  pcePcepEntityMinDeadTimer 4,
  pcePcepEntitySyncTimer 60,
  pcePcepEntityRequestTimer 120,
  pcePcepEntityMaxSessions 999,
  pcePcepEntityMaxUnknownReqs 5,
  pcePcepEntityMaxUnknownMsgs 5
}

In pcePcepPeerTable {
  pcePcepPeerAddrType ipv4(1), --PCE2
  pcePcepPeerAddr 2.2.2.2,
  pcePcepPeerRole pce(2),
  pcePcepPeerDiscontinuityTime TimeStamp,

pcePcepPeerInitiateSession  true(1),
pcePcepPeerSessionExists    true(1)),
pcePcepPeerNumSessSetupOK   0,
pcePcepPeerNumSessSetupFail 1,
pcePcepPeerSessionUpTime     TimeStamp,
pcePcepPeerSessionFailTime  TimeStamp,
pcePcepPeerSessionFailUpTime TimeStamp,
pcePcepPeerAvgRspTime       0,
pcePcepPeerLWMRspTime        0,
pcePcepPeerHWMRspTime        0,
pcePcepPeerNumPCReqSent      4,
pcePcepPeerNumPCReqRcvd      0,
pcePcepPeerNumPCRepSent      0,
pcePcepPeerNumPCRepRcvd      4,
pcePcepPeerNumPCErrSent      0,
pcePcepPeerNumPCErrRcvd      0,
pcePcepPeerNumPCNtfSent      0,
pcePcepPeerNumPCNtfRcvd      0,
pcePcepPeerNumKeepaliveSent  0,
pcePcepPeerNumKeepaliveRcvd  0,
pcePcepPeerNumUnknownRcvd    0,
pcePcepPeerNumUnknownRcvd    0,
pcePcepPeerNumReqSent       4,
pcePcepPeerNumSvecSent      0,
pcePcepPeerNumSvecReqSent   0,
pcePcepPeerNumSvecReqSent   0,
pcePcepPeerNumReqSentPendRep 0,
pcePcepPeerNumReqSentEroRcvd 3,
pcePcepPeerNumReqSentNoPathRcvd 1,
pcePcepPeerNumReqSentCancelRcvd 0,
pcePcepPeerNumReqSentErrorRcvd 0,
pcePcepPeerNumReqSentTimeout 0,
pcePcepPeerNumReqSentCancelSent 0,
pcePcepPeerNumReqSentClosed 0,
pcePcepPeerNumReqRcvd       0,
pcePcepPeerNumSvecRcvd      0,
pcePcepPeerNumSvecReqRcvd   0,
pcePcepPeerNumReqRcvdPendRep 0,
pcePcepPeerNumReqRcvdEroSent 0,
pcePcepPeerNumReqRcvdNoPathSent 0,
pcePcepPeerNumReqRcvdCancelSent 0,
pcePcepPeerNumReqRcvdErrorSent 0,
pcePcepPeerNumReqRcvdCancelRcvd 0,
pcePcepPeerNumReqRcvdClosed 0,
pcePcepPeerNumReqRcvdUnknown 0,
pcePcepPeerNumReqRcvdUnknown 0,

},
{
  pcePcepPeerAddrType     ipv4(1),  --PCE3
pcePcepPeerAddr   3.3.3.3,
pcePcepPeerRole   pce(2),
pcePcepPeerDiscontinuityTime   TimeStamp,
pcePcepPeerInitiateSession   true(1),
pcePcepPeerSessionExists   false(0),
pcePcepPeerNumSessSetupOK   1,
pcePcepPeerNumSessSetupFail   0,
pcePcepPeerSessionUpTime   TimeStamp,
pcePcepPeerSessionFailTime   TimeStamp,
pcePcepPeerSessionFailUpTime   TimeStamp,
pcePcepPeerInitiateSession   true(1),
pcePcepPeerSessionExists   false(0),
pcePcepPeerNumSessSetupOK   1,
pcePcepPeerNumSessSetupFail   0,
pcePcepPeerSessionUpTime   TimeStamp,
pcePcepPeerSessionFailTime   TimeStamp,
pcePcepPeerSessionFailUpTime   TimeStamp,
pcePcepPeerAvgRspTime   200,
pcePcepPeerLWMRspTime   100,
pcePcepPeerHWMRspTime   300,
pcePcepPeerNumPCReqSent   4,
pcePcepPeerNumPCReqRcvd   0,
pcePcepPeerNumPCRepSent   0,
pcePcepPeerNumPCRepRcvd   3,
pcePcepPeerNumCERrrSent   0,
pcePcepPeerNumCERrrRcvd   0,
pcePcepPeerNumPCNtfSent   0,
pcePcepPeerNumPCNtfRcvd   0,
pcePcepPeerNumKeepaliveSent   123,
pcePcepPeerNumKeepaliveRcvd   123,
pcePcepPeerNumUnknownRcvd   0,
pcePcepPeerNumCorruptRcvd   0,
pcePcepPeerNumReqSent   4,
pcePcepPeerNumSvecSent   0,
pcePcepPeerNumSvecReqSent   0,
pcePcepPeerNumReqSentPendRep   0,
pcePcepPeerNumReqSentEroRcvd   3,
pcePcepPeerNumReqSentNoPathRcvd   0,
pcePcepPeerNumReqSentCancelRcvd   0,
pcePcepPeerNumReqSentErrorRcvd   0,
pcePcepPeerNumReqSentTimeout   0,
pcePcepPeerNumReqSentCancelSent   1,
pcePcepPeerNumReqRcvd   0,
pcePcepPeerNumSvecRcvd   0,
pcePcepPeerNumSvecReqRcvd   0,
pcePcepPeerNumReqRcvdPendRep   0,
pcePcepPeerNumReqRcvdEroSent   0,
pcePcepPeerNumReqRcvdNoPathSent   0,
pcePcepPeerNumReqRcvdCancelSent   0,
pcePcepPeerNumReqRcvdErrorSent   0,
pcePcepPeerNumReqRcvdCancelRcvd   0,
pcePcepPeerNumReqRcvdClosed   0,
pcePcepPeerNumRepRcvdUnknown   0,
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcepPcepSessInitiator</td>
<td>local(1), --PCE2</td>
</tr>
<tr>
<td>pcepPcepSessStateLastChange</td>
<td>TimeStamp,</td>
</tr>
<tr>
<td>pcepPcepSessState</td>
<td>sessionUp(4),</td>
</tr>
<tr>
<td>pcepPcepSessConnectRetry</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessLocalID</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessRemoteID</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessKeepaliveTimer</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessPeerKeepaliveTimer</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessDeadTimer</td>
<td>4,</td>
</tr>
<tr>
<td>pcepPcepSessPeerDeadTimer</td>
<td>4,</td>
</tr>
<tr>
<td>pcepPcepSessKAHoldTimeRem</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessOverloaded</td>
<td>false(0),</td>
</tr>
<tr>
<td>pcepPcepSessOverloadTime</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessPeerOverloaded</td>
<td>false(0),</td>
</tr>
<tr>
<td>pcepPcepSessPeerOverloadTime</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessDiscontinuityTime</td>
<td>TimeStamp,</td>
</tr>
<tr>
<td>pcepPcepSessAvgRspTime</td>
<td>200,</td>
</tr>
<tr>
<td>pcepPcepSessLWMRspTime</td>
<td>100,</td>
</tr>
<tr>
<td>pcepPcepSessHWMRspTime</td>
<td>300,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCReqSent</td>
<td>4,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCReqRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCRepSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCRepRcvd</td>
<td>4,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCErrSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCErrRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCNtfSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumPCNtfRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumKeepaliveSent</td>
<td>123,</td>
</tr>
<tr>
<td>pcepPcepSessNumKeepaliveRcvd</td>
<td>123,</td>
</tr>
<tr>
<td>pcepPcepSessNumUnknownRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumCorruptRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSent</td>
<td>4,</td>
</tr>
<tr>
<td>pcepPcepSessNumSvecSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumSvecReqSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentPendRep</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentEroRcvd</td>
<td>3,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentNoPathRcvd</td>
<td>1,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentCancelRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentErrorRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentTimeout</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqSentCancelSent</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumSvecRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumSvecRcvdRcvd</td>
<td>0,</td>
</tr>
<tr>
<td>pcepPcepSessNumReqRcvdPendRep</td>
<td>0,</td>
</tr>
</tbody>
</table>
pcePcepSessNumReqRcvdEroSent 0,
pcePcepSessNumReqRcvdNoPathSent 0,
pcePcepSessNumReqRcvdCancelSent 0,
pcePcepSessNumReqRcvdErrorSent 0,
pcePcepSessNumReqRcvdCancelRcvd 0,
pcePcepSessNumRepRcvdUnknown 0,
pcePcepSessNumReqRcvdUnknown 0
}

-- no session to PCE3

Authors’ Addresses

Agrahara Kiran Koushik
Brocade Communications Inc.
EMail: kkoushik@brocade.com

Emile Stephan
Orange
2 avenue Pierre Marzin
Lannion F-22307
France
EMail: emile.stephan@orange.com

Quintin Zhao
Huawei Technology
125 Nagog Technology Park
Acton, MA  01719
US
EMail: qzhao@huawei.com

Daniel King
Old Dog Consulting
EMail: daniel@olddog.co.uk
PCEP Requirements for WSON Routing and Wavelength Assignment

draft-ietf-pce-wson-routing-wavelength-15.txt

Abstract

This memo provides application-specific requirements for the Path Computation Element communication Protocol (PCEP) for the support of Wavelength Switched Optical Networks (WSON). Lightpath provisioning in WSONs requires a routing and wavelength assignment (RWA) process. From a path computation perspective, wavelength assignment is the process of determining which wavelength can be used on each hop of a path and forms an additional routing constraint to optical light path computation. Requirements for PCEP extensions in support of optical impairments will be addressed in a separate document.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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."
1. Introduction ...................................................3
2. WSON RWA Processes & Architecture ..............................4
3. Requirements ...................................................6
  3.1. Path Computation Type Option ................................6
  3.2. RWA Processing .............................................6
  3.3. Bulk RWA Path Request/Reply ...............................7
  3.4. RWA Path Re-optimization Request/Reply .....................7
  3.5. Wavelength Range Constraint ..............................8
  3.6. Wavelength Assignment Preference ..........................8
  3.7. Signal Processing Capability Restriction ..................8
4. Manageability Considerations ................................9
  4.1. Control of Function and Policy ...........................9
  4.2. Information and Data Models, e.g. MIB module ..........9
  4.3. Liveness Detection and Monitoring .......................10
  4.4. Verifying Correct Operation .............................10
1. Introduction

[RFC4655] defines the PCE-based architecture and explains how a Path Computation Element (PCE) may compute Label Switched Paths (LSP) in Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS)-controlled networks at the request of Path Computation Clients (PCCs). A PCC is shown to be any network component that makes such a request and may be for instance an optical switching element within a Wavelength Division Multiplexing (WDM) network. The PCE, itself, can be located anywhere within the network, and may be within an optical switching element, a Network Management System (NMS) or Operational Support System (OSS), or may be an independent network server.

The PCE communication Protocol (PCEP) is the communication protocol used between PCC and PCE, and may also be used between cooperating PCEs. [RFC4657] sets out the common protocol requirements for PCEP. Additional application-specific requirements for PCEP are deferred to separate documents.

This document provides a set of application-specific PCEP requirements for support of path computation in Wavelength Switched Optical Networks (WSON). WSON refers to WDM-based optical networks in which switching is performed selectively based on the wavelength of an optical signal.

The path in WSON is referred to as a lightpath. A lightpath may span multiple fiber links and the path should be assigned a wavelength for each link.

A transparent optical network is made up of optical devices that can switch but not convert from one wavelength to another. In a transparent optical network, a lightpath operates on the same wavelength across all fiber links that it traverses. In such case, the lightpath is said to satisfy the wavelength-continuity
constraint. Two lightpaths that share a common fiber link cannot be assigned the same wavelength. To do otherwise would result in both signals interfering with each other. Note that advanced additional multiplexing techniques such as polarization based multiplexing are not addressed in this document since the physical layer aspects are not currently standardized. Therefore, assigning the proper wavelength on a lightpath is an essential requirement in the optical path computation process.

When a switching node has the ability to perform wavelength conversion the wavelength-continuity constraint can be relaxed, and a lightpath may use different wavelengths on different links along its path from origin to destination. It is, however, to be noted that wavelength converters may be limited for cost reasons, while the number of WDM channels that can be supported in a fiber is also limited. As a WSON can be composed of network nodes that cannot perform wavelength conversion, nodes with limited wavelength conversion, and nodes with full wavelength conversion abilities, wavelength assignment is an additional routing constraint to be considered in all lightpath computations.

In this document we first review the processes for routing and wavelength assignment (RWA) used when wavelength continuity constraints are present and then specify requirements for PCEP to support RWA. Requirements for optical impairments will be addressed in a separate document.

The remainder of this document uses terminology from [RFC4655].

2. WSON RWA Processes & Architecture

In [RFC6163] three alternative process architectures were given for performing routing and wavelength assignment. These are shown schematically in Figure 1. R stands for Routing, WA for Wavelength Assignment, and DWA for Distributed Wavelength Assignment.
These alternatives have the following properties and impact on PCEP requirements in this document.

(a) Combined Processes (R&WA)

Here path selection and wavelength assignment are performed as a single process. The requirements for PCC-PCE interaction with such a combined RWA process PCE is addressed in this document.

(b) Routing separate from Wavelength Assignment (R+WA)

Here the routing process furnishes one or more potential paths to the wavelength assignment process that then performs final path selection and wavelength assignment. The requirements for PCE-PCE interaction with one PCE implementing the routing process and another implementing the wavelength assignment process are not addressed in this document.

(b’) Routing and distributed Wavelength Assignment (R+DWA)

Here a standard path computation (unaware of detailed wavelength availability) takes place, then wavelength assignment is performed along this path in a distributed manner via signaling (RSVP-TE). This alternative is a particular case of R+WA and it should be covered by GMPLS PCEP extensions and does not present new WSON-specific requirements.

In the previous section various process architectures for implementing RWA have been reviewed. Figure 2 shows one typical PCE-based implementation, which is referred to as Combined Process (R&WA). With this architecture, the two processes of routing and wavelength assignment are accessed via a single PCE. This architecture is the base architecture from which the requirements are specified in this document.
3. Requirements

The requirements for the PCC to PCE interface of Figure 2 are specified in this section.

3.1. Path Computation Type Option

A PCEP request MAY include the path computation type. This can be:

(i) Both Routing and Wavelength Assignment (RWA),

(ii) Routing only.

This requirement is needed to differentiate between the currently supported routing with distributed wavelength assignment option and combined RWA. In case of distributed wavelength assignment option, wavelength assignment will be performed at each node of the route.

3.2. RWA Processing

(a) When the request is a RWA path computation type, the request MUST further include the wavelength assignment options. At the minimum, the following option should be supported:

(i) Explicit Label Control (ELC) [RFC3473]

(ii) A set of recommended labels for each hop. The PCC can select the label based on local policy.

Note that option (ii) may also be used in R+WA or R+DWA.

(b) In case of a RWA computation type, the response MUST include the wavelength(s) assigned to the path and an indication of which label assignment option has been applied (ELC or label set).
In the case where a valid path is not found, the response MUST include why the path is not found (e.g., network disconnected, wavelength not found, or both, etc.). Note that ‘wavelength not found’ may include several sub-cases such as wavelength continuity not met, unsupported FEC/Modulation type, etc.

3.3. Bulk RWA Path Request/Reply

Sending simultaneous path requests for "routing only" computation is supported by PCEP specification [RFC5440]. To remain consistent the following requirements are added.

(a) A PCEP request MUST be able to specify an option for bulk RWA path request. Bulk path request is an ability to request a number of simultaneous RWA path requests.

(b) The PCEP response MUST include the path and the assigned wavelength assigned for each RWA path request specified in the original bulk request.

3.4. RWA Path Re-optimization Request/Reply

1. For a re-optimization request, the request MUST provide both the path and current wavelength to be re-optimized and MAY include the following options:
   a. Re-optimize the path keeping the same wavelength(s)
   b. Re-optimize wavelength(s) keeping the same path
   c. Re-optimize allowing both the wavelength and the path to change

2. The corresponding response to the re-optimized request MUST provide the re-optimized path and wavelengths even when the request asked for the path or the wavelength to remain unchanged.

3. In case that the new path is not found, the response MUST include why the path is not found (e.g., network disconnected, wavelength not found, or both, etc.). Note that ‘wavelength not found’ may include several sub-cases such as wavelength continuity not met, unsupported FEC/Modulation type, etc.
3.5. Wavelength Range Constraint

For any RWA computation type request, the requester (PCC) MUST be allowed to specify a restriction on the wavelengths to be used. The requester MAY use this option to restrict the assigned wavelength for explicit label or label set. This restriction may for example come from the tuning ability of a laser transmitter, any optical element, or a policy-based restriction.

Note that the requester (e.g., PCC) is not required to furnish any range restrictions.

3.6. Wavelength Assignment Preference

1. A RWA computation type request MAY include the requester preference for, e.g., random assignment, descending order, ascending order, etc. A response SHOULD follow the requestor preference unless it conflicts with operator’s policy.

2. A request for two or more paths MUST allow the requester to include an option constraining the paths to have the same wavelength(s) assigned. This is useful in the case of protection with single transponder (e.g., 1+1 link disjoint paths).

In a network with wavelength conversion capabilities (e.g. sparse 3R regenerators), a request SHOULD be able to indicate whether a single, continuous wavelength should be allocated or not. In other words, the requesting PCC SHOULD be able to specify the precedence of wavelength continuity even if wavelength conversion is available.

3.7. Signal Processing Capability Restriction

Signal processing compatibility is an important constraint for optical path computation. The signal type for an end-to-end optical path must match at source and at destination.

The PCC MUST be allowed to specify the signal type at the endpoints (i.e., at source and at destination). The following signal processing capabilities should be supported at a minimum:
The PCC MUST also be allowed to state whether transit modification is acceptable for the above signal processing capabilities.

4. Manageability Considerations

Manageability of WSON Routing and Wavelength Assignment (RWA) with PCE must address the following considerations:

4.1. Control of Function and Policy

In addition to the parameters already listed in Section 8.1 of [RFC5440], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCC:

- The ability to send a WSON RWA request.

In addition to the parameters already listed in Section 8.1 of [RFC5440], a PCEP implementation SHOULD allow configuring the following PCEP session parameters on a PCE:

- The support for WSON RWA.

- The maximum number of bulk path requests associated with WSON RWA per request message.

These parameters may be configured as default parameters for any PCEP session the PCEP speaker participates in, or may apply to a specific session with a given PCEP peer or a specific group of sessions with a specific group of PCEP peers.

4.2. Information and Data Models, e.g. MIB module

As this document only concerns the requirements to support WSON RWA, no additional MIB module is defined in this document. However, the corresponding solution draft will list the information that should be added to the PCE MIB module defined in [PCEP-MIB].
4.3. Liveness Detection and Monitoring

No new mechanism is defined in this document that implies any new liveness detection and monitoring requirements in addition to those already listed in section 8.3 of [RFC5440].

4.4. Verifying Correct Operation

No new mechanism is defined in this document that implies any new verification requirements in addition to those already listed in section 8.4 of [RFC5440].

4.5. Requirements on Other Protocols and Functional Components

If PCE discovery mechanisms ([RFC5089] and [RFC5088]) were to be extended for technology-specific capabilities, advertising WSON RWA path computation capability should be considered.

4.6. Impact on Network Operation

No new mechanism is defined in this document that implies any new network operation requirements in addition to those already listed in section 8.6 of [RFC5440].

5. Security Considerations

This document has no requirement for a change to the security models within PCEP [RFC5440]. However, the additional information distributed in order to address the RWA problem represents a disclosure of network capabilities that an operator may wish to keep private. Consideration should be given to securing this information.

Solutions that address the requirements in this document need to verify that existing PCEP security mechanisms adequately protect the additional network capabilities and must include new mechanisms as necessary.
6. IANA Considerations

This informational document does not make any requests for IANA action.

7. Acknowledgments

The authors would like to thank Adrian Farrel, Cycil Margaria and Ramon Casellas for many helpful comments that greatly improved the contents of this draft.

This document was prepared using 2-Word-v2.0.template.dot.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Young Lee (Ed.)
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75245, USA
Phone: (469)277-5838
Email: leeyoung@huawei.com

Greg Bernstein (Ed.)
Grotto Networking
Fremont, CA, USA
Phone: (510) 573-2237
Email: gregb@grotto-networking.com

Jonas Martensson
Acreo
Email:Jonas.Martensson@acreo.se

Tomonori Takeda
NTT Corporation
3-9-11, Midori-Cho
Musashino-Shi, Tokyo 180-8585, Japan
Email: takeda.tomonori@lab.ntt.co.jp