

Workgroup: PCE Working Group
Internet-Draft: draft-ietf-pce-pcep-yang-22
Published: 11 September 2023
Intended Status: Standards Track
Expires: 14 March 2024
Authors: D. Dhody, Ed. V. Beeram J. Hardwick
 Huawei Juniper Networks Microsoft
 J. Tantsura
 Nvidia

A YANG Data Model for Path Computation Element Communications Protocol (PCEP)

Abstract

This document defines a YANG data model for the management 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. The data model includes configuration and state data.

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 <https://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 14 March 2024.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Requirements Language](#)
- [3. Terminology and Notation](#)
 - [3.1. Tree Diagrams](#)
 - [3.2. Prefixes in Data Node Names](#)
 - [3.3. References in the Model](#)
- [4. The Design of PCEP Data Model](#)
 - [4.1. The Entity](#)
 - [4.1.1. The Peer List](#)
 - [4.1.1.1. The Session List](#)
 - [4.2. Notifications](#)
 - [4.3. RPC](#)
- [5. The Design of PCEP Statistics Data Model](#)
- [6. Advanced PCE Features](#)
 - [6.1. Stateful PCE's LSP-DB](#)
- [7. Other Considerations](#)
 - [7.1. PCEP over TLS \(PCEPS\)](#)
- [8. PCEP YANG Modules](#)
 - [8.1. ietf-pcep module](#)
 - [8.2. ietf-pcep-stats module](#)
- [9. Security Considerations](#)
- [10. IANA Considerations](#)
- [11. Implementation Status](#)
- [12. Acknowledgements](#)
- [13. References](#)
 - [13.1. Normative References](#)
 - [13.2. Informative References](#)
- [Appendix A. The Full PCEP Data Model](#)
- [Appendix B. Example](#)
- [Appendix C. Design Objectives](#)
- [Appendix D. Relationship with PCEP MIB](#)
- [Appendix E. Contributor Addresses](#)
- [Authors' Addresses](#)

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). [RFC8231] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC7950] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data.

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

The PCEP operational state is included in the same tree as the PCEP configuration consistent with Network Management Datastore Architecture (NMDA) [RFC8342]. The origin of the data is indicated as per the origin metadata annotation.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology and Notation

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 also uses the following terms defined in [RFC7420]:

*PCEP entity: a local PCEP speaker.

*PCEP peer: to refer to a remote PCEP speaker.

*PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [\[RFC8231\]](#) :

*Stateful PCE, Passive Stateful PCE, Active Stateful PCE

*Delegation, Revocation, Redelegation

*LSP State Report, Path Computation Report message (PCRpt).

*LSP State Update, Path Computation Update message (PCUpd).

*PLSP-ID: a PCEP-specific identifier for the LSP.

*SRP: Stateful PCE Request Parameters

[\[RFC8281\]](#) :

*PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[\[RFC8408\]](#) :

*Path Setup Type (PST).

[\[RFC8664\]](#) :

*Segment Routing (SR).

[\[RFC5541\]](#) :

*Objective Function (OF).

[\[RFC8697\]](#) :

*Association.

[\[RFC6241\]](#) :

*Configuration data.

*State data.

3.1. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is defined in [[RFC8340](#)].

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in [Table 1](#).

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	[RFC8776]
key-chain	ietf-key-chain	[RFC8177]
nacm	ietf-netconf-acm	[RFC8341]
tlss	ietf-tls-server	[I-D.ietf-netconf-tls-client-server]
tlsc	ietf-tls-client	[I-D.ietf-netconf-tls-client-server]
ospf	ietf-ospf	[RFC9129]
isis	ietf-isis	[RFC9130]

Table 1: Prefixes and corresponding YANG modules

3.3. References in the Model

Following documents are referenced in the model defined in this document -

Documents	Reference
OSPF Protocol Extensions for Path Computation Element (PCE) Discovery	[RFC5088]
IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery	[RFC5089]
Path Computation Element (PCE) Communication Protocol (PCEP)	[RFC5440]
Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism	[RFC5520]
Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)	[RFC5541]

Documents	Reference
Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization	[RFC5557]
Common YANG Data Types	[RFC6991]
YANG Data Model for Key Chains	[RFC8177]
Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE	[RFC8231]
Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE	[RFC8232]
PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)	[RFC8253]
Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model	[RFC8281]
Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths	[RFC8306]
Network Configuration Access Control Model	[RFC8341]
Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages	[RFC8408]
Traffic Engineering Common YANG Types	[RFC8776]
A YANG Data Model for Traffic Engineering Tunnels and Interfaces	[I-D.ietf-teas-yang-te]
YANG Groupings for TLS Clients and TLS Servers	[I-D.ietf-netconf-tls-client-server]
PCEP Extensions for Segment Routing	[RFC8664]
PCEP Extensions for Establishing Relationships Between Sets of LSPs	[RFC8697]
YANG Data Model for OSPF Protocol	[RFC9129]

Documents	Reference
YANG Data Model for IS-IS Protocol	[RFC9130]
PCEP extensions for GMPLS	[RFC8779]
ASSOCIATION Type Field in Path Computation Element Protocol (PCEP) Numbers	[IANA-PCEP]
Path Computation Element (PCE) Capability Flags in Interior Gateway Protocol (IGP) Parameters	[IANA-IGP]
Path Computation Element Communication Protocol (PCEP) Extension for Flow Specification	[RFC9168]
Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture	[RFC8685]
Hierarchical Stateful Path Computation Element (PCE)	[RFC8751]
IGP Extension for Path Computation Element Communication Protocol (PCEP) Security Capability Support in PCE Discovery (PCED)	[RFC9353]
Stateful Path Computation Element (PCE) Protocol Extensions for Usage with Point-to-Multipoint TE Label Switched Paths (LSPs)	[RFC8623]
The TCP Authentication Option	[RFC5925]
Path Computation Element Communication Protocol (PCEP) Extensions for Associating Working and Protection Label Switched Paths (LSPs) with Stateful PCE	[RFC8745]
Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling	[RFC8800]
Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of	[RFC9358]

Documents	Reference
Label Switched Paths and Virtual Networks	
Extensions to the Path Computation Element Communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering	[RFC8282]
Path Computation Element Communication Protocol (PCEP) Extension for Associating Policies and Label Switched Paths (LSPs)	[RFC9005]

Table 2: References in the YANG modules

4. The Design of PCEP Data Model

The PCEP YANG module defined in this document has all the common building blocks for the PCEP protocol.


```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--rw addr                inet:ip-address-no-zone
    +--rw enabled?            boolean
    +--rw role                 role
    +--rw description?        string
    +--rw speaker-entity-id?  string {sync-opt}?
    +--rw admin-status?       boolean
    +--ro index?              uint32
    +--ro oper-status?        oper-status
    +--rw domains
      | +--rw domain* [type info]
      |   +--...
    +--rw capabilities
      | +--...
    +--rw auth
      | +--...
    +--rw pce-info
      | +--rw scope
      | | +--...
      | +--rw neighbour-domains
      | | +--...
      | +--rw path-key {path-key}?
      |   +--...
    +--...
    +--ro lsp-db {stateful}?
      | +--ro db-ver?          uint64 {sync-opt}?
      | +--ro association-list*
      | | [type id source global-source extended-id]
      | | {association}?
      | | +--...
      | +--ro lsp* [plsp-id pcc-id lsp-id]
      |   +--...
    +--ro path-keys {path-key}?
      | +--ro path-key* [key]
      |   +--...
    +--rw peers
      +--rw peer* [addr]
      +--...
      +--ro sessions
        +--ro session* [initiator]
        +--...

rpcs:
+---x trigger-resync {stateful, sync-opt}?
+---w input
+---w pcc?   -> /pcep/entity/peers/peer/addr

```

notifications:

```
+---n pcep-session-up
| +---...
+---n pcep-session-down
| +---...
+---n pcep-session-local-overload
| +---...
+---n pcep-session-local-overload-clear
| +---...
+---n pcep-session-peer-overload
| +---...
+---n pcep-session-peer-overload-clear
+---...
```

4.1. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using ietf-inet-types [[RFC6991](#)]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [[RFC7420](#)] uses an entity list and a system generated entity index as a primary index to the read only entity table.

The local PCEP entity contains various information related to this entity such as its domain, capabilities, security parameters etc. When the local entity is PCE it could also have path-key and the LSP-DB information.

Note that the timer names in the YANG uses hyphen compared to [[RFC5440](#)] (for example, dead-timer in YANG is called DeadTimer in the protocol specification).

```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--rw addr                inet:ip-address-no-zone
    +--rw enabled?           boolean
    +--rw role                role
    +--rw description?       string
    +--rw speaker-entity-id? string {sync-opt}?
    +--rw admin-status?      boolean
    +--ro index?             uint32
    +--ro oper-status?       oper-status
    +--rw domains
      | +--rw domain* [type info]
      |   +--rw type      identityref
      |   +--rw info      domain
    +--rw capabilities
      | +--rw capability?          bits
      | +--rw pce-initiated?       boolean {pce-initiated}?
      | +--rw include-db-ver?      boolean {stateful,sync-opt}?
      | +--rw trigger-resync?      boolean {stateful,sync-opt}?
      | +--rw trigger-initial-sync? boolean {stateful,sync-opt}?
      | +--rw incremental-sync?    boolean {stateful,sync-opt}?
      | +--rw sr {sr}?
      | | +--rw enabled?           boolean
      | | +--rw msd-limit?        boolean
      | | +--rw nai?               boolean
      | +--rw stateful-gmpls {stateful,gmpls}?
      | | +--rw enabled?          boolean
      | +--rw inter-layer?        boolean {inter-layer}?
      | +--rw h-pce {h-pce}?
      |   +--rw enabled?          boolean
      |   +--rw stateful?         boolean {stateful}?
      |   +--rw role?             hpce-role
    +--rw msd?                    uint8 {sr}?
    +--rw auth
      | +--rw (auth-type-selection)?
      |   +--:(auth-key-chain)
      |     | +--rw key-chain?
      |     |   key-chain:key-chain-ref
      |     +--:(auth-key)
      |       | +--rw crypto-algorithm          identityref
      |       | +--rw (key-string-style)?
      |       |   +--:(keystring)
      |       |     | +--rw keystring?          string
      |       |     +--:(hexadecimal) {key-chain:hex-key-string}?
      |       |       +--rw hexadecimal-string? yang:hex-string
      |     +--:(auth-tls) {tls}?
      |       +--rw (role)?
      |         +--:(server)

```

```

|         | +--rw tls-server
|         |   +---...
|         +---:(client)
|         +--rw tls-client
|         +---...
+--rw pce-info
| +--rw scope
| | +--rw path-scope?      bits
| | +--rw intra-area-pref? uint8
| | +--rw inter-area-pref? uint8
| | +--rw inter-as-pref?  uint8
| | +--rw inter-layer-pref? uint8
| +--rw neighbour-domains
| | +--rw domain* [type info]
| |   +--rw type      identityref
| |   +--rw info      domain
| +--rw path-key {path-key}?
|   +--rw enabled?    boolean
|   +--rw discard-timer? uint32
|   +--rw reuse-time?  uint32
|   +--rw pce-id?     inet:ip-address-no-zone
+--rw connect-timer?      uint16
+--rw connect-max-retry?  uint32
+--rw init-back-off-timer? uint16
+--rw max-back-off-timer? uint32
+--ro open-wait-timer?    uint16
+--ro keep-wait-timer?    uint16
+--rw keepalive-timer?    uint8
+--rw dead-timer?        uint8
+--rw allow-negotiation?  boolean
+--rw max-keepalive-timer? uint8
+--rw max-dead-timer?     uint8
+--rw min-keepalive-timer? uint8
+--rw min-dead-timer?     uint8
+--rw sync-timer?        uint16 {svec}?
+--rw request-timer?     uint16
+--rw max-sessions?      uint32
+--rw max-unknown-reqs?  uint32
+--rw max-unknown-msgs?  uint32
+--rw pcep-notification-max-rate uint32
+--rw stateful-parameter {stateful}?
| +--rw state-timeout?    uint32
| +--rw redelegation-timeout? uint32
| +--rw rpt-non-pcep-lsp? boolean
+--rw of-list {objective-function}?
| +--rw objective-function* [of]
|   +--rw of      identityref
+--ro lsp-db {stateful}?
| +--ro db-ver?          uint64 {sync-opt}?

```

```

| +--ro association-list*
| |     [type id source global-source extended-id]
| |     {association}?
| |     +--ro type             identityref
| |     +--ro id               uint16
| |     +--ro source           inet:ip-address-no-zone
| |     +--ro global-source    uint32
| |     +--ro extended-id     string
| |     +--ro lsp* [plsp-id pcc-id lsp-id]
| |         +--ro plsp-id     -> /pcep/entity/lsp-db/lsp/plsp-id
| |         +--ro pcc-id     -> /pcep/entity/lsp-db/lsp/pcc-id
| |         +--ro lsp-id     -> /pcep/entity/lsp-db/lsp/lsp-id
| +--ro lsp* [plsp-id pcc-id lsp-id]
|     +--ro plsp-id           uint32
|     +--ro pcc-id           inet:ip-address-no-zone
|     +--ro source?         inet:ip-address-no-zone
|     +--ro destination?   inet:ip-address-no-zone
|     +--ro tunnel-id?     uint16
|     +--ro lsp-id         uint16
|     +--ro extended-tunnel-id? inet:ip-address-no-zone
|     +--ro admin-state?   boolean
|     +--ro operational-state? operational-state
|     +--ro delegated
|     | +--ro enabled?    boolean
|     | +--ro peer?      -> /pcep/entity/peers/peer/addr
|     | +--ro srp-id?   uint32
|     +--ro initiation {pce-initiated}?
|     | +--ro enabled?    boolean
|     | +--ro peer?      -> /pcep/entity/peers/peer/addr
|     +--ro symbolic-path-name? string
|     +--ro last-error?     identityref
|     +--ro pst?           identityref
|     +--ro association-list*
|         [type id source global-source extended-id]
|         {association}?
|         +--ro type
|         |     -> /pcep/entity/lsp-db/association-list/type
|         +--ro id
|         |     -> /pcep/entity/lsp-db/association-list/id
|         +--ro source
|         |     -> /pcep/entity/lsp-db/association-list/source
|         +--ro global-source    leafref
|         +--ro extended-id     leafref
+--ro path-keys {path-key}?
| +--ro path-key* [key]
|     +--ro key             uint16
|     +--ro cps
|     | +--ro explicit-route-objects* [index]
|     | +--ro index             uint32

```

```

|     |     +--ro (type)?
|     |     +--:(numbered-node-hop)
|     |     | +--ro numbered-node-hop
|     |     |   +--ro node-id      te-node-id
|     |     |   +--ro hop-type?    te-hop-type
|     |     +--:(numbered-link-hop)
|     |     | +--ro numbered-link-hop
|     |     |   +--ro link-tp-id    te-tp-id
|     |     |   +--ro hop-type?    te-hop-type
|     |     |   +--ro direction?   te-link-direction
|     |     +--:(unnumbered-link-hop)
|     |     | +--ro unnumbered-link-hop
|     |     |   +--ro link-tp-id    te-tp-id
|     |     |   +--ro node-id      te-node-id
|     |     |   +--ro hop-type?    te-hop-type
|     |     |   +--ro direction?   te-link-direction
|     |     +--:(as-number)
|     |     | +--ro as-number-hop
|     |     |   +--ro as-number    inet:as-number
|     |     |   +--ro hop-type?    te-hop-type
|     |     +--:(label)
|     |     | +--ro label-hop
|     |     |   +--ro te-label
|     |     |
|     |     |     ...
|     +--ro pcc-original?    -> /pcep/entity/peers/peer/addr
|     +--ro req-id?          uint32
|     +--ro retrieved?       boolean
|     +--ro pcc-retrieved?   -> /pcep/entity/peers/peer/addr
|     +--ro creation-time?   yang:timestamp
|     +--ro discard-time?    uint32
|     +--ro reuse-time?      uint32
+--rw peers
   +--rw peer* [addr]
     +--...

```

4.1.1. The Peer List

The peer list contains peer(s) that the local PCEP entity knows about. 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.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained 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 this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address.

This peer list includes peers that are explicitly configured at the local PCEP entity as well as peers that are learned dynamically. For example, at a PCC, the remote PCE peer to use could be explicitly configured. A PCC could also learn PCE in the network via IGP discovery and it will show up in this list. When a session is initiated at a PCE, the remote PCC peer information is also added by the system to the peer list.

```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--...
    +--rw peers
      +--rw peer* [addr]
        +--rw addr                inet:ip-address-no-zone
        +--rw role                role
        +--rw description?       string
        +--rw domains
          | +--rw domain* [type info]
          |   +--rw type  identityref
          |   +--rw info  domain
        +--rw capabilities
          | +--rw capability?      bits
          | +--rw pce-initiated?   boolean {pce-initiated}?
          | +--rw include-db-ver?  boolean
          | | {stateful, sync-opt}?
          | +--rw trigger-resync?  boolean
          | | {stateful, sync-opt}?
          | +--rw trigger-initial-sync?  boolean
          | | {stateful, sync-opt}?
          | +--rw incremental-sync?  boolean
          | | {stateful, sync-opt}?
          | +--rw sr {sr}?
          | | +--rw enabled?    boolean
          | | +--rw msd-limit?  boolean
          | | +--rw nai?        boolean
          | +--rw stateful-gmpls {stateful, gmpls}?
          | | +--rw enabled?    boolean
          | +--rw inter-layer?    boolean {inter-layer}?
          | +--rw h-pce {h-pce}?
          |   +--rw enabled?    boolean
          |   +--rw stateful?   boolean {stateful}?
          |   +--rw role?       hpce-role
        +--rw msd?                uint8 {sr}?
      +--rw pce-info
        | +--rw scope
        | | +--rw path-scope?    bits
        | | +--rw intra-area-pref?  uint8
        | | +--rw inter-area-pref?  uint8
        | | +--rw inter-as-pref?   uint8
        | | +--rw inter-layer-pref? uint8
        | +--rw neighbour-domains
        |   +--rw domain* [type info]
        |     +--rw type  identityref
        |     +--rw info  domain
        +--rw delegation-pref?    uint8 {stateful}?
      +--rw auth

```



```

| +--rw (auth-type-selection)?
|   +--:(auth-key-chain)
|     | +--rw key-chain?
|       |         key-chain:key-chain-ref
|     +--:(auth-key)
|       | +--rw crypto-algorithm          identityref
|       | +--rw (key-string-style)?
|       |   +--:(keystring)
|       |     | +--rw keystring?          string
|       |     +--:(hexadecimal) {key-chain:hex-key-string}?
|       |       +--rw hexadecimal-string? yang:hex-string
|     +--:(auth-tls) {tls}?
|       +--rw (role)?
|         +--:(server)
|           | +--rw tls-server
|             |         ...
|         +--:(client)
|           +--rw tls-client
|             |         ...
+--ro discontinuity-time?      yang:timestamp
+--ro initiate-session?       boolean
+--ro session-exists?         boolean
+--ro session-up-time?        yang:timestamp
+--ro session-fail-time?      yang:timestamp
+--ro session-fail-up-time?   yang:timestamp
+--ro sessions
  +--ro session* [initiator]
    +--...

```

4.1.1.1. The Session List

The session list contains PCEP sessions that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [[RFC5440](#)] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. When one of these sessions reaches the active state, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

```

module: ietf-pcep
  +--rw pcep!
    +--rw entity
      +--...
      +--rw peers
        +--rw peer* [addr]
          +--...
          +--ro sessions
            +--ro session* [initiator]
              +--ro initiator                initiator
              +--ro role?
              |      -> /pcep/entity/role
              +--ro state-last-change?      yang:timestamp
              +--ro state?                  sess-state
              +--ro session-creation?       yang:timestamp
              +--ro connect-retry?          yang:counter32
              +--ro local-id?               uint8
              +--ro remote-id?              uint8
              +--ro keepalive-timer?         uint8
              +--ro peer-keepalive-timer?   uint8
              +--ro dead-timer?              uint8
              +--ro peer-dead-timer?         uint8
              +--ro ka-hold-time-rem?        uint8
              +--ro overloaded?              boolean
              +--ro overloaded-timestamp?    yang:timestamp
              +--ro overload-time?           uint32
              +--ro peer-overloaded?         boolean
              +--ro peer-overloaded-timestamp? yang:timestamp
              +--ro peer-overload-time?      uint32
              +--ro lspdb-sync?              sync-state
              |      {stateful}?
              +--ro recv-db-ver?             uint64
              |      {stateful, sync-opt}?
              +--ro of-list {objective-function}?
              | +--ro objective-function* [of]
              | +--ro of      identityref
              +--ro pst-list
              | +--ro path-setup-type* [pst]
              | +--ro pst      identityref
              +--ro assoc-type-list {association}?
              | +--ro assoc-type* [at]
              | +--ro at      identityref
              +--ro speaker-entity-id?      string
              {sync-opt}?

```

4.2. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB [[RFC7420](#)] notifications.

notifications:

```
+---n pcep-session-up
| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |           -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro state-last-change? yang:timestamp
| +--ro state?             sess-state
+---n pcep-session-down
| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator? initiator
| +--ro state-last-change? yang:timestamp
| +--ro state?             sess-state
+---n pcep-session-local-overload
| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |           -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro overloaded?         boolean
| +--ro overloaded-timestamp? yang:timestamp
| +--ro overload-time?      uint32
+---n pcep-session-local-overload-clear
| +--ro peer-addr?
| |           -> /pcep/entity/peers/peer/addr
| +--ro overloaded?         boolean
| +--ro overloaded-clear-timestamp? yang:timestamp
+---n pcep-session-peer-overload
| +--ro peer-addr?
| |           -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |           -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro peer-overloaded?    boolean
| +--ro peer-overloaded-timestamp? yang:timestamp
| +--ro peer-overload-time?  uint32
+---n pcep-session-peer-overload-clear
  +--ro peer-addr?
  |           -> /pcep/entity/peers/peer/addr
  +--ro peer-overloaded?    boolean
  +--ro peer-overloaded-clear-timestamp? yang:timestamp
```

4.3. RPC

This YANG model defines a RPC to trigger state resynchronize at the PCE for sanity check with a particular PCC.

rpcs:

```
+---x trigger-resync {stateful, sync-opt}?  
+---w input  
+---w pcc? -> /pcep/entity/peers/peer/addr
```

5. The Design of PCEP Statistics Data Model

The module, "ietf-pcep-stats", augments the ietf-pcep module to include statistics at the PCEP peer and session level. It includes a RPC to reset statistics.

module: ietf-pcep-stats

augment /pcep:pcep/pcep:entity/pcep:peers/pcep:peer:

```
  +--ro sess-setup-ok?      yang:counter32
  +--ro sess-setup-fail?   yang:counter32
  +--ro stats
    +--ro discontinuity-time? yang:timestamp
    +--ro rsp-time-avg?      uint32
    +--ro rsp-time-lwm?     uint32
    +--ro rsp-time-hwm?     uint32
    +--ro pcreq-sent?       yang:counter32
    +--ro pcreq-rcvd?      yang:counter32
    +--ro pcrep-sent?      yang:counter32
    +--ro pcrep-rcvd?     yang:counter32
    +--ro pcerr-sent?      yang:counter32
    +--ro pcerr-rcvd?     yang:counter32
    +--ro pcntf-sent?      yang:counter32
    +--ro pcntf-rcvd?     yang:counter32
    +--ro keepalive-sent?  yang:counter32
    +--ro keepalive-rcvd? yang:counter32
    +--ro unknown-rcvd?   yang:counter32
    +--ro corrupt-rcvd?   yang:counter32
    +--ro req-sent?       yang:counter32
    +--ro req-sent-pend-rep? yang:counter32
    +--ro req-sent-ero-rcvd? yang:counter32
    +--ro req-sent-nopath-rcvd? yang:counter32
    +--ro req-sent-cancel-rcvd? yang:counter32
    +--ro req-sent-error-rcvd? yang:counter32
    +--ro req-sent-timeout? yang:counter32
    +--ro req-sent-cancel-sent? yang:counter32
    +--ro req-rcvd?       yang:counter32
    +--ro req-rcvd-pend-rep? yang:counter32
    +--ro req-rcvd-ero-sent? yang:counter32
    +--ro req-rcvd-nopath-sent? yang:counter32
    +--ro req-rcvd-cancel-sent? yang:counter32
    +--ro req-rcvd-error-sent? yang:counter32
    +--ro req-rcvd-cancel-rcvd? yang:counter32
    +--ro rep-rcvd-unknown? yang:counter32
    +--ro req-rcvd-unknown? yang:counter32
    +--ro svec {pcep:svec}?
      | +--ro svec-sent?      yang:counter32
      | +--ro svec-req-sent?  yang:counter32
      | +--ro svec-rcvd?     yang:counter32
      | +--ro svec-req-rcvd? yang:counter32
    +--ro stateful {pcep:stateful}?
      | +--ro pcrpt-sent?     yang:counter32
      | +--ro pcrpt-rcvd?    yang:counter32
      | +--ro pcupd-sent?    yang:counter32
      | +--ro pcupd-rcvd?    yang:counter32
```

```

| +--ro rpt-sent?          yang:counter32
| +--ro rpt-rcvd?         yang:counter32
| +--ro rpt-rcvd-error-sent? yang:counter32
| +--ro upd-sent?         yang:counter32
| +--ro upd-rcvd?         yang:counter32
| +--ro upd-rcvd-unknown? yang:counter32
| +--ro upd-rcvd-undelegated? yang:counter32
| +--ro upd-rcvd-error-sent? yang:counter32
| +--ro initiation {pcep:pce-initiated}?
|   +--ro pcinitiate-sent?      yang:counter32
|   +--ro pcinitiate-rcvd?      yang:counter32
|   +--ro initiate-sent?        yang:counter32
|   +--ro initiate-rcvd?        yang:counter32
|   +--ro initiate-rcvd-error-sent? yang:counter32
+--ro path-key {pcep:path-key}?
| +--ro unknown-path-key?      yang:counter32
| +--ro exp-path-key?          yang:counter32
| +--ro dup-path-key?          yang:counter32
| +--ro path-key-no-attempt?    yang:counter32
+--ro req-sent-closed?          yang:counter32
+--ro req-rcvd-closed?         yang:counter32
augment /pcep:pcep/pcep:entity/pcep:peers/pcep:peer/pcep:sessions
  /pcep:session:
+--ro stats
  +--ro discontinuity-time?     yang:timestamp
  +--ro rsp-time-avg?           uint32
  +--ro rsp-time-lwm?           uint32
  +--ro rsp-time-hwm?           uint32
  +--ro pcreq-sent?             yang:counter32
  +--ro pcreq-rcvd?             yang:counter32
  +--ro pcrep-sent?             yang:counter32
  +--ro pcrep-rcvd?             yang:counter32
  +--ro pcerr-sent?             yang:counter32
  +--ro pcerr-rcvd?             yang:counter32
  +--ro pcntf-sent?             yang:counter32
  +--ro pcntf-rcvd?             yang:counter32
  +--ro keepalive-sent?         yang:counter32
  +--ro keepalive-rcvd?         yang:counter32
  +--ro unknown-rcvd?           yang:counter32
  +--ro corrupt-rcvd?           yang:counter32
  +--ro req-sent?               yang:counter32
  +--ro req-sent-pend-rep?      yang:counter32
  +--ro req-sent-ero-rcvd?      yang:counter32
  +--ro req-sent-nopath-rcvd?   yang:counter32
  +--ro req-sent-cancel-rcvd?   yang:counter32
  +--ro req-sent-error-rcvd?    yang:counter32
  +--ro req-sent-timeout?       yang:counter32
  +--ro req-sent-cancel-sent?   yang:counter32
  +--ro req-rcvd?               yang:counter32

```

```

+--ro req-rcvd-pend-rep?      yang:counter32
+--ro req-rcvd-ero-sent?     yang:counter32
+--ro req-rcvd-nopath-sent?  yang:counter32
+--ro req-rcvd-cancel-sent?  yang:counter32
+--ro req-rcvd-error-sent?   yang:counter32
+--ro req-rcvd-cancel-rcvd?  yang:counter32
+--ro rep-rcvd-unknown?      yang:counter32
+--ro req-rcvd-unknown?      yang:counter32
+--ro svec {pcep:svec}?
| +--ro svec-sent?           yang:counter32
| +--ro svec-req-sent?       yang:counter32
| +--ro svec-rcvd?           yang:counter32
| +--ro svec-req-rcvd?       yang:counter32
+--ro stateful {pcep:stateful}?
| +--ro pcrpt-sent?          yang:counter32
| +--ro pcrpt-rcvd?          yang:counter32
| +--ro pcupd-sent?          yang:counter32
| +--ro pcupd-rcvd?          yang:counter32
| +--ro rpt-sent?            yang:counter32
| +--ro rpt-rcvd?            yang:counter32
| +--ro rpt-rcvd-error-sent? yang:counter32
| +--ro upd-sent?            yang:counter32
| +--ro upd-rcvd?            yang:counter32
| +--ro upd-rcvd-unknown?    yang:counter32
| +--ro upd-rcvd-undelegated? yang:counter32
| +--ro upd-rcvd-error-sent? yang:counter32
| +--ro initiation {pcep:pce-initiated}?
|   +--ro pcinitiate-sent?    yang:counter32
|   +--ro pcinitiate-rcvd?    yang:counter32
|   +--ro initiate-sent?      yang:counter32
|   +--ro initiate-rcvd?      yang:counter32
|   +--ro initiate-rcvd-error-sent? yang:counter32
+--ro path-key {pcep:path-key}?
  +--ro unknown-path-key?     yang:counter32
  +--ro exp-path-key?         yang:counter32
  +--ro dup-path-key?         yang:counter32
  +--ro path-key-no-attempt?  yang:counter32

```

rpcs:

```

+---x statistics-reset
+---w input
  +---w (peer-or-all)?
    +--:(peer)
      | +---w peer-addr?  -> /pcep:pcep/entity/peers/peer/addr
    +--:(all)

```

6. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [[RFC5440](#)] data model.

This document further handles advanced PCE features like -

- *Capability and Scope
- *Domain information (local/neighbour)
- *Path-Key
- *Objective Function (OF)
- *Global Concurrent Optimization (GCO)
- *P2MP
- *GMPLS
- *Inter-Layer
- *Stateful PCE
- *Segment Routing (SR) for MPLS data plane
- *Authentication including PCEPS (TLS)
- *Hierarchical PCE (H-PCE)

6.1. Stateful PCE's LSP-DB

In the operational datastore of stateful PCE, the list of LSP state are maintained in the LSP-DB. The key is the PLSP-ID, the PCC's IP address, and the LSP-ID.

The PCEP data model contains the operational state of LSPs (/pcep/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [[I-D.ietf-teas-yang-te](#)]. A reference to LSP state in TE model is maintained.

7. Other Considerations

7.1. PCEP over TLS (PCEPS)

[[RFC8253](#)] describes the use of TLSv1.2 [[RFC5246](#)] or later in PCEP. Further, [[I-D.dhody-pce-pceps-tls13](#)] specify how to protect PCEP messages with TLS 1.3 [[RFC8446](#)] by disallowing the use of early data (0-RTT) and listing the cipher suites that need to be supported with TLS 1.3.

The peer acting as the PCEP client must act as the TLS client. The TLS client actively opens the TLS connection and the TLS server passively listens for the incoming TLS connections. The well-known TCP port number 4189 is used by PCEP servers to listen for TCP connections established by PCEP over TLS clients. The TLS client sends the TLS ClientHello message to begin the TLS handshake. The TLS server sends a CertificateRequest in order to request a certificate from the TLS client. Once the TLS handshake has finished, the client and the server begin to exchange PCEP messages. Client and server identity verification is done before the PCEP open message is sent. This means that the identity verification is completed before the PCEP session is started.

Note that, a PCEP speaker could act as both a client (PCC) and a server (PCE). The role within the context of a PCEP session is determined by the relationship it has with its peer (the same holds good for TLS as well).

The YANG module uses the TLS grouping in [\[I-D.ietf-netconf-tls-client-server\]](#). Note that any TLS version can be configured but [\[I-D.ietf-netconf-tls-client-server\]](#) recommends the use of TLS 1.3 only. At the time of publication of this document, TLS 1.2 is still in common use for PCEP and can still be enabled with feature "tls12" even though it is marked with status as "deprecated".

8. PCEP YANG Modules

8.1. ietf-pcep module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2023-09-11.yang"
```

```
module ietf-pcep {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
  prefix pcep;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC 8177: YANG Data Model for Key Chains";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Protocol (NETCONF) Access
      Control Model";
  }
  import ietf-tls-server {
    prefix tlss;
    reference
      "I-D.ietf-netconf-tls-client-server: YANG Groupings for TLS
      Clients and TLS Servers";
  }
  import ietf-tls-client {
    prefix tlsc;
    reference
      "I-D.ietf-netconf-tls-client-server: YANG Groupings for TLS
      Clients and TLS Servers";
  }
  import ietf-ospf {
    prefix ospf;
    reference
      "RFC 9129: YANG Data Model for OSPF Protocol";
  }
}
```

```

}
import ietf-isis {
  prefix isis;
  reference
    "RFC 9130: YANG Data Model for IS-IS Protocol";
}

organization
  "IETF PCE (Path Computation Element) Working Group";
contact
  "WG Web: <https://datatracker.ietf.org/wg/pce/>
  WG List: <mailto:pce@ietf.org>
  Editor: Dhruv Dhody
    <mailto:dhruv.ietf@gmail.com>";
description
  "The YANG module defines a generic configuration and
  operational model for Path Computation Element
  Communication Protocol (PCEP).

  Copyright (c) 2023 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Revised BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see the
  RFC itself for full legal notices.

  The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
  NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
  'MAY', and 'OPTIONAL' in this document are to be interpreted as
  described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.";

revision 2023-09-11 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for Path Computation
    Element Communications Protocol (PCEP)";
}

/*
 * Typedefs
 */

```

```

typedef role {
    type enumeration {
        enum unknown {
            value 0;
            description
                "An unknown role";
        }
        enum pcc {
            value 1;
            description
                "The role of a Path Computation Client";
        }
        enum pce {
            value 2;
            description
                "The role of Path Computation Element";
        }
        enum pcc-and-pce {
            value 3;
            description
                "The role of both Path Computation Client and
                Path Computation Element";
        }
    }
    description
        "The role of a PCEP speaker.
        Takes one of the following values
        - unknown(0): the role is not known.
        - pcc(1): the role is of a Path Computation
          Client (PCC).
        - pce(2): the role is of a Path Computation
          Server (PCE).
        - pcc-and-pce(3): the role is of both a PCC and
          a PCE.";
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}

```

```

typedef oper-status {
    type enumeration {
        enum oper-status-up {
            value 1;
            description
                "The PCEP entity is active";
        }
        enum oper-status-down {
            value 2;
            description

```

```

        "The PCEP entity is inactive";
    }
    enum oper-status-going-up {
        value 3;
        description
            "The PCEP entity is activating";
    }
    enum oper-status-going-down {
        value 4;
        description
            "The PCEP entity is deactivating";
    }
    enum oper-status-failed {
        value 5;
        description
            "The PCEP entity has failed and will recover
            when possible.";
    }
    enum oper-status-failed-perm {
        value 6;
        description
            "The PCEP entity has failed and will not recover
            without operator intervention";
    }
}
description
    "The operational status of the PCEP entity.
    Takes one of the following values
    - oper-status-up(1): Active
    - oper-status-down(2): Inactive
    - oper-status-going-up(3): Activating
    - oper-status-going-down(4): Deactivating
    - oper-status-failed(5): Failed
    - oper-status-failed-perm(6): Failed Permanently";
reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

typedef initiator {
    type enumeration {
        enum local {
            value 1;
            description
                "The local PCEP entity initiated the session";
        }
        enum remote {
            value 2;
            description

```

```

        "The remote PCEP peer initiated the session";
    }
}
description
    "The initiator of the session, that is, whether the TCP
    connection was initiated by the local PCEP entity or
    the remote peer.
    Takes one of the following values
    - local(1): Initiated locally
    - remote(2): Initiated remotely";
}

typedef sess-state {
    type enumeration {
        enum tcp-pending {
            value 1;
            description
                "The TCPPending state of PCEP session.";
        }
        enum open-wait {
            value 2;
            description
                "The OpenWait state of PCEP session.";
        }
        enum keep-wait {
            value 3;
            description
                "The KeepWait state of PCEP session.";
        }
        enum session-up {
            value 4;
            description
                "The SessionUP state of PCEP session.";
        }
    }
}
description
    "The current state of the session.
    The set of possible states excludes the idle state
    since entries do not exist in the idle state.
    Takes one of the following values
    - tcp-pending(1): PCEP TCPPending state
    - open-wait(2): PCEP OpenWait state
    - keep-wait(3): PCEP KeepWait state
    - session-up(4): PCEP SessionUP state";
reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

```

```

typedef domain {
    type union {
        type ospf:area-id-type;
        type isis:area-address;
        type inet:as-number;
    }
    description
        "The Domain Information";
}

typedef operational-state {
    type enumeration {
        enum down {
            value 0;
            description
                "not active.";
        }
        enum up {
            value 1;
            description
                "signalled.";
        }
        enum active {
            value 2;
            description
                "up and carrying traffic.";
        }
        enum going-down {
            value 3;
            description
                "LSP is being torn down, resources are
                being released.";
        }
        enum going-up {
            value 4;
            description
                "LSP is being signalled.";
        }
    }
    description
        "The operational status of the LSP";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}

typedef sync-state {
    type enumeration {
        enum pending {

```

```

    value 0;
    description
        "The state synchronization
        has not started.";
}
enum ongoing {
    value 1;
    description
        "The state synchronization
        is ongoing.";
}
enum finished {
    value 2;
    description
        "The state synchronization
        is finished.";
}
}
description
    "The LSP-DB state synchronization operational
    status.";
reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

typedef hpce-role {
    type enumeration {
        enum unknown {
            value 0;
            description
                "An unknown role";
        }
        enum child {
            value 1;
            description
                "The PCE is acting as child PCE.";
        }
        enum parent {
            value 2;
            description
                "The PCE is acting as parent PCE.";
        }
    }
}
description
    "The H-PCE role of the PCE.";
reference
    "RFC 8685: Path Computation Element Communication Protocol
    (PCEP) Extensions for the Hierarchical Path Computation

```



```

        Element (H-PCE) Architecture";
    }

/*
 * Features
 */

feature svec {
    description
        "Support synchronized path computation.";
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}

feature gmpls {
    description
        "Support GMPLS.";
    reference
        "RFC 8779: PCEP extensions for GMPLS";
}

feature objective-function {
    description
        "Support OF as per RFC 5541.";
    reference
        "RFC 5541: Encoding of Objective Functions in the Path
        Computation Element Communication Protocol (PCEP)";
}

feature global-concurrent {
    description
        "Support Global Concurrent Optimization (GCO) as per RFC
        5557.";
    reference
        "RFC 5557: Path Computation Element Communication Protocol
        (PCEP) Requirements and Protocol Extensions in Support of
        Global Concurrent Optimization";
}

feature path-key {
    description
        "Support path-key as per RFC 5520.";
    reference
        "RFC 5520: Preserving Topology Confidentiality in Inter-
        Domain Path Computation Using a Path-Key-Based Mechanism";
}

feature p2mp {

```

```

description
  "Support Point-to-Multipoint (P2MP) as per RFC 8306.";
reference
  "RFC 8306: Extensions to the Path Computation Element
  Communication Protocol (PCEP) for Point-to-Multipoint
  Traffic Engineering Label Switched Paths";
}

feature stateful {
  description
    "Support Stateful PCE as per RFC 8231.";
  reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

feature sync-opt {
  description
    "Support Stateful state synchronization optimization
    as per RFC 8232";
  reference
    "RFC 8232: Optimizations of Label Switched Path State
    Synchronization Procedures for a Stateful PCE";
}

feature pce-initiated {
  description
    "Support PCE-Initiated LSP as per
    RFC 8281.";
  reference
    "RFC 8281: Path Computation Element Communication Protocol
    (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful
    PCE Model";
}

feature tls {
  description
    "Support PCEP over TLS as per RFC 8253.";
  reference
    "RFC 8253: PCEPS: Usage of TLS to Provide a Secure Transport
    for the Path Computation Element Communication Protocol
    (PCEP)";
}

feature sr {
  description
    "Support Segment Routing (SR) for PCE.";
  reference
    "RFC 8664: Path Computation Element Communication Protocol

```

```

        (PCEP) Extensions for Segment Routing";
    }

feature association {
    description
        "Support Association in PCEP.";
    reference
        "RFC 8697: Path Computation Element Communication Protocol
        (PCEP) Extensions for Establishing Relationships between
        Sets of Label Switched Paths (LSPs)";
}

feature flowspec {
    description
        "Support Flow Specification in PCEP.";
    reference
        "RFC 9168: Path Computation Element Communication Protocol
        (PCEP) Extension for Flow Specification";
}

feature h-pce {
    description
        "Support Hierarchical-PCE (H-PCE).";
    reference
        "RFC 8685: Path Computation Element Communication
        Protocol (PCEP) Extensions for the Hierarchical Path
        Computation Element (H-PCE) Architecture";
}

feature inter-layer {
    description
        "Support Inter-layer path computation.";
    reference
        "RFC 8282: Extensions to the Path Computation
        Element Communication Protocol (PCEP) for Inter-
        Layer MPLS and GMPLS Traffic Engineering";
}

/*
 * Identities
 */

identity domain-type {
    description
        "Base Domain Type for PCE";
}

identity ospf-area {
    base domain-type;
}

```

```
    description
      "The OSPF area.";
  }

  identity isis-area {
    base domain-type;
    description
      "The IS-IS area.";
  }

  identity autonomous-system {
    base domain-type;
    description
      "The Autonomous System (AS).";
  }

  identity lsp-error {
    if-feature "stateful";
    description
      "Base LSP error";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity no-error-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "No error, LSP is fine.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity unknown-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "Unknown reason. LSP Error Code value = 1.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity limit-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
```

```
        "Limit reached for PCE-controlled LSPs. LSP Error Code
        value = 2.";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}

identity pending-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
        "Too many pending LSP update requests. LSP Error Code
        value = 3.";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}

identity unacceptable-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
        "Unacceptable parameters. LSP Error Code value = 4.";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}

identity internal-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
        "Internal error. LSP Error Code value = 5.";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}

identity admin-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
        "LSP administratively bought down. LSP Error Code value
        = 6.";
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}
```

```
identity preempted-lsp-error {
  if-feature "stateful";
  base lsp-error;
  description
    "LSP preempted. LSP Error Code value = 7.";
  reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

identity rsvp-lsp-error {
  if-feature "stateful";
  base lsp-error;
  description
    "RSVP signaling error. LSP Error Code value = 8.";
  reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

identity path-protection {
  base te-types:association-type;
  description
    "Path Protection Association";
  reference
    "RFC 8745: Path Computation Element Communication Protocol
    (PCEP) Extensions for Associating Working and Protection
    Label Switched Paths (LSPs) with Stateful PCE";
}

identity disjoint {
  base te-types:association-type;
  description
    "Disjoint Association";
  reference
    "RFC 8800: Path Computation Element Communication Protocol
    (PCEP) Extension for Label Switched Path (LSP) Diversity
    Constraint Signaling";
}

identity policy {
  base te-types:association-type;
  description
    "Policy Association";
  reference
    "RFC 9005: Path Computation Element Communication Protocol
    (PCEP) Extension for Associating Policies and Label Switched
    Paths (LSPs)";
}
```

```

identity vn {
  base te-types:association-type;
  description
    "Virtual Network (VN) Association";
  reference
    "RFC 9358: Path Computation Element Communication Protocol
    (PCEP) Extensions for Establishing Relationships between
    Sets of Label Switched Paths and Virtual Networks";
}

/*
 * Groupings
 */

grouping domain {
  description
    "This grouping specifies a Domain where the
    PCEP speaker has topology visibility.";
  leaf type {
    type identityref {
      base domain-type;
    }
    description
      "The domain type.";
  }
  leaf info {
    type domain;
    description
      "The domain Information.";
  }
}

grouping info {
  description
    "This grouping specifies all information which
    maybe relevant to both PCC and PCE.
    This information corresponds to PCE auto-discovery
    information.
    The scope relates to either a local entity or a peer.";
  container domains {
    description
      "The domain for the local PCEP entity or a peer.";
    list domain {
      key "type info";
      description
        "The domain information.";
      uses domain {
        description

```

```

        "The domain for the local PCEP entity or a peer.";
    }
}
}
container capabilities {
    description
        "The PCEP entity or peer capability information.
        This may be relevant to PCE selection as well.
        This information corresponds to PCE auto-
        discovery information.";
    reference
        "IANA IGP: Path Computation Element (PCE) Capability
        Flags in Interior Gateway Protocol (IGP) Parameters
        RFC 5088: OSPF Protocol Extensions for Path
        Computation Element (PCE) Discovery
        RFC 5089: IS-IS Protocol Extensions for Path
        Computation Element (PCE) Discovery
        RFC 9353: IGP Extension for Path Computation Element
        Communication Protocol (PCEP) Security Capability
        Support in PCE Discovery (PCED)";
    leaf capability {
        type bits {
            bit gmpls {
                if-feature "gmpls";
                description
                    "Path computation with GMPLS link
                    constraints.";
            }
            bit bi-dir {
                description
                    "Bidirectional path computation.";
            }
            bit diverse {
                description
                    "Diverse path computation.";
            }
            bit load-balance {
                description
                    "Load-balanced path computation.";
            }
            bit synchronize {
                if-feature "svec";
                description
                    "Synchronized paths computation.";
            }
            bit objective-function {
                if-feature "objective-function";
                description
                    "Support for multiple objective functions.";
            }
        }
    }
}

```



```

}
bit add-path-constraint {
    description
        "Support for additive path constraints (max
        hop count, etc.).";
}
bit prioritization {
    description
        "Support for request prioritization.";
}
bit multi-request {
    description
        "Support for multiple requests per message.";
}
bit global-concurrent {
    if-feature "global-concurrent";
    description
        "Support for Global Concurrent Optimization
        (GCO).";
    reference
        "RFC 5557: Path Computation Element Communication
        Protocol (PCEP) Requirements and Protocol
        Extensions in Support of Global Concurrent
        Optimization";
}
bit p2mp {
    if-feature "p2mp";
    description
        "Support for P2MP path computation.";
    reference
        "RFC 8306: Extensions to the Path Computation
        Element Communication Protocol (PCEP) for
        Point-to-Multipoint Traffic Engineering Label
        Switched Paths";
}
bit active {
    if-feature "stateful";
    description
        "Support for active stateful PCE.";
    reference
        "RFC 8231: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
}
bit passive {
    if-feature "stateful";
    description
        "Support for passive stateful PCE.";
    reference

```

```
        "RFC 8231: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
    }
    bit p2mp-active {
        if-feature "stateful";
        if-feature "p2mp";
        description
            "Support for active stateful PCE for P2MP.";
        reference
            "RFC 8623: Stateful Path Computation Element
            (PCE) Protocol Extensions for Usage with
            Point-to-Multipoint TE Label Switched Paths
            (LSPs)";
    }
    bit p2mp-passive {
        if-feature "stateful";
        if-feature "p2mp";
        description
            "Support for passive stateful PCE for P2MP.";
        reference
            "RFC 8623: Stateful Path Computation Element
            (PCE) Protocol Extensions for Usage with
            Point-to-Multipoint TE Label Switched Paths
            (LSPs)";
    }
    bit p2mp-pce-initiated {
        if-feature "stateful";
        if-feature "pce-initiated";
        if-feature "p2mp";
        description
            "Support for PCE-initiated LSP for P2MP.";
        reference
            "RFC 8623: Stateful Path Computation Element
            (PCE) Protocol Extensions for Usage with
            Point-to-Multipoint TE Label Switched Paths
            (LSPs)";
    }
    bit flowspec {
        if-feature "flowspec";
        description
            "Support for Flow specification.";
        reference
            "RFC 9168: Path Computation Element Communication
            Protocol (PCEP) Extension for Flow Specification";
    }
    bit tcp-ao {
        description
            "Support for The TCP Authentication Option
```

```

        (TCP-A0)";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)
        RFC 5925: The TCP Authentication Option";
    }
    bit tls {
        if-feature "tls";
        description
            "Support for TLS";
        reference
            "RFC 8253: PCEPS: Usage of TLS to Provide a
            Secure Transport for the Path Computation
            Element Communication Protocol (PCEP)";
    }
}
description
    "The bits string indicating the capabilities";
reference
    "IANA IGP: Path Computation Element (PCE) Capability
    Flags in Interior Gateway Protocol (IGP) Parameters
    RFC 5088: OSPF Protocol Extensions for Path
    Computation Element (PCE) Discovery
    RFC 5089: IS-IS Protocol Extensions for Path
    Computation Element (PCE) Discovery
    RFC 9353: IGP Extension for Path Computation Element
    Communication Protocol (PCEP) Security Capability
    Support in PCE Discovery (PCED)";
}
leaf pce-initiated {
    if-feature "pce-initiated";
    type boolean;
    description
        "Set to true if PCE-initiated LSP capability is
        enabled.";
    reference
        "RFC 8281: Path Computation Element Communication
        Protocol (PCEP) Extensions for PCE-Initiated LSP
        Setup in a Stateful PCE Model";
}
leaf include-db-ver {
    if-feature "stateful";
    if-feature "sync-opt";
    type boolean;
    description
        "Support inclusion of LSP-DB-VERSION
        in LSP object";
    reference
        "RFC 8232: Optimizations of Label Switched Path State

```

```

        Synchronization Procedures for a Stateful PCE";
    }
    leaf trigger-resync {
        if-feature "stateful";
        if-feature "sync-opt";
        type boolean;
        description
            "Support PCE triggered re-synchronization";
        reference
            "RFC 8232: Optimizations of Label Switched Path State
            Synchronization Procedures for a Stateful PCE";
    }
    leaf trigger-initial-sync {
        if-feature "stateful";
        if-feature "sync-opt";
        type boolean;
        description
            "PCE triggered initial synchronization";
        reference
            "RFC 8232: Optimizations of Label Switched Path State
            Synchronization Procedures for a Stateful PCE";
    }
    leaf incremental-sync {
        if-feature "stateful";
        if-feature "sync-opt";
        type boolean;
        description
            "Support incremental (delta) sync";
        reference
            "RFC 8232: Optimizations of Label Switched Path State
            Synchronization Procedures for a Stateful PCE";
    }
    container sr {
        if-feature "sr";
        description
            "If segment routing for MPLS is supported at the local
            entity or a peer.";
        reference
            "RFC 8664: Path Computation Element Communication Protocol
            (PCEP) Extensions for Segment Routing";
        leaf enabled {
            type boolean;
            description
                "Set to true if SR-MPLS is enabled";
        }
        leaf msd-limit {
            type boolean;
            default "false";
            description

```

```

        "True indicates no limit on MSD, the
        leaf msd is ignored";
    }
    leaf nai {
        type boolean;
        default "false";
        description
            "True indicates the capability to resolve Node or
            Adjacency Identifier (NAI) to Segment
            Identifier (SID)";
    }
}
container stateful-gmpls {
    if-feature "stateful";
    if-feature "gmpls";
    description
        "If Stateful GMPLS is supported for a local entity
        or a peer";
    reference
        "RFC 8779: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE Usage in GMPLS-controlled
        Networks";
    leaf enabled {
        type boolean;
        description
            "Set to true if Stateful GMPLS is enabled";
    }
}
leaf inter-layer {
    if-feature "inter-layer";
    type boolean;
    description
        "If inter-layer path computation is supported for
        local entity or a peer";
    reference
        "RFC 8282: Extensions to the Path Computation
        Element Communication Protocol (PCEP) for Inter-
        Layer MPLS and GMPLS Traffic Engineering";
}
container h-pce {
    if-feature "h-pce";
    description
        "If Hierarchical PCE (H-PCE) is supported for local
        entity or a peer";
    reference
        "RFC 8685: Path Computation Element Communication
        Protocol (PCEP) Extensions for the Hierarchical Path
        Computation Element (H-PCE) Architecture";
}

```

```

leaf enabled {
  type boolean;
  description
    "Set to true if Stateful GMPLS is enabled";
}
leaf stateful {
  if-feature "stateful";
  type boolean;
  description
    "Set to true if Stateful H-PCE is enabled";
  reference
    "RFC 8751: Hierarchical Stateful Path Computation
    Element (PCE)";
}
leaf role {
  when "../.../role = 'pce'"
    + "or "
    + "../.../role = 'pcc-and-pce'" {
  description
    "These fields are applicable when the role is PCE.";
  }
  type hpce-role;
  description
    "The H-PCE role of the PCE.";
}
}
}
leaf msd {
  if-feature "sr";
  type uint8;
  description
    "Maximum SID Depth for SR-MPLS i.e. the label stack depth
    that a router is capable of imposing on a packet.";
  reference
    "RFC 8664: Path Computation Element Communication Protocol
    (PCEP) Extensions for Segment Routing";
}
}

grouping pce-info {
  description
    "This grouping specifies all PCE information
    which may be relevant to the PCE selection.
    This information corresponds to PCE auto-discovery
    information.";
  container scope {
    description
      "This container defines PCE path computation scope
      information which may be relevant to PCE selection.

```

```

    This information corresponds to PCE auto-discovery
    information.";
leaf path-scope {
  type bits {
    bit intra-area-scope {
      description
        "PCE can compute intra-area paths (L bit).";
    }
    bit inter-area-scope {
      description
        "PCE can compute inter-area paths (R bit).";
    }
    bit inter-area-scope-default {
      description
        "PCE can act as a default PCE for inter-area
        path computation. (Rd bit)";
    }
    bit inter-as-scope {
      description
        "PCE can compute inter-AS paths (S bit).";
    }
    bit inter-as-scope-default {
      description
        "PCE can act as a default PCE for inter-AS
        path computation (Sd bit).";
    }
    bit inter-layer-scope {
      description
        "PCE can compute inter-layer paths (Y bit).";
    }
  }
  description
    "The field corresponding to the path scope bits";
}
leaf intra-area-pref {
  type uint8 {
    range "0..7";
  }
  description
    "The PCE's preference for intra-area TE LSP
    computation (PrefL field). Where 7 reflects
    the highest preference.";
}
leaf inter-area-pref {
  type uint8 {
    range "0..7";
  }
  description
    "The PCE's preference for inter-area TE LSP

```

```

        computation (PrefR field). Where 7 reflects
        the highest preference.";
    }
    leaf inter-as-pref {
        type uint8 {
            range "0..7";
        }
        description
            "The PCE's preference for inter-AS TE LSP
            computation (PrefS field). Where 7 reflects
            the highest preference.";
    }
    leaf inter-layer-pref {
        type uint8 {
            range "0..7";
        }
        description
            "The PCE's preference for inter-layer TE LSP
            computation (PrefY field). Where 7 reflects
            the highest preference.";
    }
    reference
        "RFC 5088: OSPF Protocol Extensions for Path
        Computation Element (PCE) Discovery
        RFC 5089: IS-IS Protocol Extensions for Path
        Computation Element (PCE) Discovery";
}
container neighbour-domains {
    description
        "The list of neighbour PCE-Domain
        toward which a PCE can compute
        paths";
    list domain {
        key "type info";
        description
            "The neighbour domain.";
        uses domain {
            description
                "The PCE neighbour domain.";
        }
    }
}
}

grouping notification-instance-hdr {
    description
        "This group describes common instance-specific data
        for notifications.";
    leaf peer-addr {

```



```

    type leafref {
      path "/pcep/entity/peers/peer/addr";
    }
    description
      "Reference to peer address";
  }
}

grouping notification-session-hdr {
  description
    "This group describes common session instance specific
    data for notifications.";
  leaf session-initiator {
    type leafref {
      path "/pcep/entity/peers/peer/sessions/"
        + "session/initiator";
    }
    description
      "Reference to pcep session initiator leaf";
  }
}

grouping of-list {
  description
    "List of Objective Functions (OF)";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP)";
  list objective-function {
    key "of";
    description
      "The list of authorized OF";
    leaf of {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "The OF authorized";
    }
  }
}

grouping auth {
  description
    "The Authentication options";
  container auth {
    description
      "The Authentication options";
    choice auth-type-selection {

```

```

description
  "Options for expressing authentication
  setting.";
case auth-key-chain {
  leaf key-chain {
    type key-chain:key-chain-ref;
    description
      "key-chain name.";
  }
}
case auth-key {
  leaf crypto-algorithm {
    type identityref {
      base key-chain:crypto-algorithm;
    }
    mandatory true;
    description
      "Cryptographic algorithm associated
      with key.";
  }
  choice key-string-style {
    description
      "Key string styles";
    case keystack {
      leaf keystack {
        nacm:default-deny-all;
        type string;
        description
          "Key string in ASCII format.";
      }
    }
    case hexadecimal {
      if-feature "key-chain:hex-key-string";
      leaf hexadecimal-string {
        nacm:default-deny-all;
        type yang:hex-string;
        description
          "Key in hexadecimal string format. When
          compared to ASCII, specification in
          hexadecimal affords greater key entropy
          with the same number of internal
          key-string octets. Additionally, it
          discourages usage of well-known words or
          numbers.";
      }
    }
  }
}
}
case auth-tls {

```

```

if-feature "tls";
choice role {
  description
    "The role of the local entity";
  case server {
    container tls-server {
      uses tlss:tls-server-grouping {
        description
          "Server TLS information.";
      }
      description
        "TLS related information";
    }
  }
  case client {
    container tls-client {
      uses tlsc:tls-client-grouping {
        description
          "Client TLS information.";
      }
      description
        "TLS related information";
    }
  }
}
}
}
}
}

/*
 * Configuration data nodes
 */

container pcep {
  presence "The PCEP is enabled";
  description
    "Parameters for list of configured PCEP entities
    on the device.";
  container entity {
    description
      "The configured PCEP entity on the device.";
    leaf addr {
      type inet:ip-address-no-zone;
      mandatory true;
      description
        "The local Internet address of this PCEP entity.
        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.";
    }
    leaf enabled {
        type boolean;
        default "true";
        description
            "The administrative status of this PCEP
            Entity; set to true when UP.";
    }
    leaf role {
        type role;
        mandatory true;
        description
            "The role that this entity can play.
            Takes one of the following values.
            - unknown(0): this PCEP Entity role is not
            known.
            - pcc(1): this PCEP Entity is a PCC.
            - pce(2): this PCEP Entity is a PCE.
            - pcc-and-pce(3): this PCEP Entity is both
            a PCC and a PCE.";
    }
    leaf description {
        type string;
        description
            "Description of the PCEP entity configured
            by the user";
    }
    leaf speaker-entity-id {
        if-feature "sync-opt";
        type string;
        description
            "The Speaker Entity Identifier";
        reference
            "RFC 8232: Optimizations of Label Switched
            Path State Synchronization Procedures for
            a Stateful PCE";
    }
    leaf admin-status {
        type boolean;
        default "true";
        description
            "The administrative status of this PCEP Entity.
            The value true represents admin status as up.
            This is the desired operational status as

```

```

        currently set by an operator or by default in
        the implementation. The value of oper-status
        represents the current status of an attempt to
        reach this desired status.";
    }
    leaf index {
        type uint32;
        config false;
        description
            "The index of the operational PCEP entity";
    }
    leaf oper-status {
        type oper-status;
        config false;
        description
            "The operational status of the PCEP entity.
            Takes one of the following values.
            - oper-status-up(1): the PCEP entity is active.
            - oper-status-down(2): the PCEP entity is inactive.
            - oper-status-going-up(3): the PCEP entity is
            activating.
            - oper-status-going-down(4): the PCEP entity is
            deactivating.
            - oper-status-failed(5): the PCEP entity has
            failed and will recover when possible.
            - oper-status-failed-perm(6): the PCEP entity
            has failed and will not recover without
            operator intervention.";
    }
    uses info {
        description
            "Local PCEP entity information";
    }
    uses auth {
        description
            "Local authorization and security parameters";
    }
    container pce-info {
        when "../role = 'pce'"
            + "or "
            + "../role = 'pcc-and-pce'" {
            description
                "These fields are applicable when the role is PCE.";
        }
        description
            "The Local PCE Entity PCE information";
        uses pce-info {
            description
                "Local PCE information";
        }
    }

```

```

}
container path-key {
  if-feature "path-key";
  description
    "Path-Key Configuration";
  reference
    "RFC 5520: Preserving Topology Confidentiality in
    Inter-Domain Path Computation Using a Path-Key
    -Based Mechanism";
  leaf enabled {
    type boolean;
    description
      "Enabled or Disabled; set to true when Enabled";
  }
  leaf discard-timer {
    type uint32;
    units "minutes";
    default "10";
    description
      "A timer to discard unwanted path-keys";
  }
  leaf reuse-time {
    type uint32 {
      range "30..max";
    }
    units "minutes";
    default "30";
    description
      "A time after which the path-keys could be reused";
  }
  leaf pce-id {
    type inet:ip-address-no-zone;
    description
      "PCE Address to be used in each Path-Key Subobject
      (PKS)";
  }
}
}
leaf connect-timer {
  type uint16 {
    range "1..max";
  }
  units "seconds";
  default "60";
  description
    "The time in seconds 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.";
}

```

```

reference
  "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf connect-max-retry {
  type uint32;
  default "5";
  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.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
      Protocol (PCEP)";
}
leaf init-back-off-timer {
  type uint16 {
    range "1..max";
  }
  units "seconds";
  description
    "The initial back-off time in seconds 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 the
      max-back-off-timer leaf.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
      Protocol (PCEP)";
}
leaf max-back-off-timer {
  type uint32;
  units "seconds";
  description
    "The maximum back-off time in seconds 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.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
      Protocol (PCEP)";
}
leaf open-wait-timer {
  type uint16;
  units "seconds";

```

```

config false;
description
  "The time in seconds 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 sessions.";
reference
  "RFC 5440: Path Computation Element (PCE) Communication
  Protocol (PCEP)";
}
leaf keep-wait-timer {
  type uint16;
  units "seconds";
  config false;
  description
    "The time in seconds 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
    sessions.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf keepalive-timer {
  type uint8;
  units "seconds";
  default "30";
  description
    "The Keepalive 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.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf dead-timer {

```



```

type uint8;
units "seconds";
must '(. > ../keepalive-timer)' {
    error-message "The DeadTimer must be "
        + "larger than the Keepalive timer";
}
default "120";
description
    "The DeadTimer 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 DeadTimer at all.";
reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf allow-negotiation {
    type boolean;
    default "true";
    description
        "Whether the PCEP entity will permit negotiation of
        session parameters.";
}
leaf max-keepalive-timer {
    type uint8;
    units "seconds";
    description
        "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.";
}
leaf max-dead-timer {
    type uint8;
    units "seconds";
    description
        "The maximum value in seconds, that this PCEP
        entity will accept from a peer for the DeadTimer.
        Zero means that the PCEP entity will allow not
        running a DeadTimer.";
}
leaf min-keepalive-timer {
    type uint8;
    units "seconds";
    description
        "The minimum value in seconds, 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.";
    }
    leaf min-dead-timer {
        type uint8;
        units "seconds";
        description
            "The minimum value in seconds, that this PCEP
            entity will accept for the DeadTimer. Zero
            means that the PCEP entity insists on not
            running a DeadTimer.";
    }
    leaf sync-timer {
        if-feature "svec";
        type uint16;
        units "seconds";
        default "60";
        description
            "The value of SyncTimer in seconds 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 reason (for example, a
            misbehaving PCC).
            Zero means that the PCEP entity does not use the
            SyncTimer.";
        reference
            "RFC 5440: Path Computation Element (PCE)
            Communication Protocol (PCEP)";
    }
    leaf request-timer {
        type uint16 {
            range "1..max";
        }
        units "seconds";
        description
            "The maximum time that the PCEP entity will wait
            for a response to a PCReq message.";
    }
}

```

```

leaf max-sessions {
  type uint32;
  description
    "Maximum number of sessions involving this PCEP
    entity that can exist at any time.";
}
leaf max-unknown-reqs {
  type uint32;
  default "5";
  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.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf max-unknown-msgs {
  type uint32;
  default "5";
  description
    "The maximum number of unknown messages that any
    session on this PCEP entity is willing to accept
    per minute before terminating the session.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf pcep-notification-max-rate {
  type uint32;
  mandatory true;
  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).";
}

```

```

container stateful-parameter {
  if-feature "stateful";
  description
    "The configured stateful PCE parameters";
  leaf state-timeout {
    type uint32;
    units "seconds";
    description
      "When a PCEP session is terminated, a PCC
      waits for this time period before flushing
      LSP state associated with that PCEP session
      and reverting to operator-defined default
      parameters or behaviours.";
  }
  leaf redelegation-timeout {
    when "../..role = 'pcc'"
      + "or "
      + "../..role = 'pcc-and-pce'" {
    description
      "This field is applicable when the role is
      PCC";
    }
    type uint32;
    units "seconds";
    description
      "When a PCEP session is terminated, a PCC
      waits for this time period before revoking
      LSP delegation to a PCE and attempting to
      redelegate LSPs associated with the
      terminated PCEP session to an alternate
      PCE.";
  }
  leaf rpt-non-pcep-lsp {
    when "../..role = 'pcc'"
      + "or "
      + "../..role = 'pcc-and-pce'" {
    description
      "This field is applicable when the role is
      PCC";
    }
    type boolean;
    default "true";
    description
      "If set, a PCC reports LSPs that are not
      controlled by any PCE (for example, LSPs
      that are statically configured at the
      PCC). ";
  }
}
reference

```

```

    "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
}
container of-list {
  when "../role = 'pce'"
    + "or "
    + "../role = 'pcc-and-pce'" {
    description
      "These field are applicable when the role is
        PCE";
  }
  if-feature "objective-function";
  uses of-list;
  description
    "The authorized OF-List at PCE for all peers";
}
container lsp-db {
  if-feature "stateful";
  config false;
  description
    "The LSP-DB";
  leaf db-ver {
    when "../..../role = 'pcc'"
      + "or "
      + "../..../role = 'pcc-and-pce'" {
      description
        "This field is applicable when the role is
          PCC";
    }
    if-feature "sync-opt";
    type uint64;
    description
      "The LSP State Database Version Number";
  }
}
list association-list {
  if-feature "association";
  key "type id source global-source extended-id";
  description
    "List of all PCEP associations";
  reference
    "RFC 8697: Path Computation Element Communication
      Protocol (PCEP) Extensions for Establishing
      Relationships between Sets of Label Switched
      Paths (LSPs)";
  leaf type {
    type identityref {
      base te-types:association-type;
    }
    description

```

```

    "The PCEP Association Type";
reference
    "IANA PCEP: ASSOCIATION Type Field in Path
    Computation Element Protocol (PCEP) Numbers
    RFC 8697: Path Computation Element Communication
    Protocol (PCEP) Extensions for Establishing
    Relationships between Sets of Label Switched
    Paths (LSPs)";
}
leaf id {
    type uint16;
    description
        "PCEP Association ID";
}
leaf source {
    type inet:ip-address-no-zone;
    description
        "PCEP Association Source.";
}
leaf global-source {
    type uint32;
    description
        "PCEP Global Association Source.";
}
leaf extended-id {
    type string;
    description
        "Additional information to support unique
        identification (Extended Association ID).";
}
list lsp {
    key "plsp-id pcc-id lsp-id";
    description
        "List of all LSP in this association";
    leaf plsp-id {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "lsp/plsp-id";
        }
        description
            "Reference to PLSP-ID in LSP-DB";
    }
    leaf pcc-id {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "lsp/pcc-id";
        }
        description
            "Reference to PCC-ID in LSP-DB";
    }
}

```

```

    }
    leaf lsp-id {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "lsp/lsp-id";
        }
        description
            "Reference to LSP ID in LSP-DB";
    }
}
}
list lsp {
    key "plsp-id pcc-id lsp-id";
    description
        "List of all LSPs in LSP-DB";
    leaf plsp-id {
        type uint32 {
            range "1..1048575";
        }
        description
            "A PCEP-specific identifier for the LSP. A PCC
            creates a unique PLSP-ID for each LSP that is
            constant for the lifetime of a PCEP session.
            PLSP-ID is 20 bits with 0 and 0xFFFFF are
            reserved";
    }
    leaf pcc-id {
        type inet:ip-address-no-zone;
        description
            "The local internet address of the PCC, that
            generated the PLSP-ID.";
    }
    leaf source {
        type inet:ip-address-no-zone;
        description
            "Tunnel sender address extracted from
            LSP-IDENTIFIERS TLV";
        reference
            "RFC 8231: Path Computation Element
            Communication Protocol (PCEP) Extensions
            for Stateful PCE";
    }
    leaf destination {
        type inet:ip-address-no-zone;
        description
            "Tunnel endpoint address extracted from
            LSP-IDENTIFIERS TLV";
        reference
            "RFC 8231: Path Computation Element

```

```

        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
    }
leaf tunnel-id {
    type uint16;
    description
        "Tunnel identifier used in the LSP-IDENTIFIERS
        TLV that remains constant over the life
        of the tunnel.";
    reference
        "RFC 8231: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
}
leaf lsp-id {
    type uint16;
    description
        "Identifier used in the LSP-IDENTIFIERS TLV
        that can be changed to allow a sender to share
        resources with itself.";
    reference
        "RFC 8231: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
}
leaf extended-tunnel-id {
    type inet:ip-address-no-zone;
    description
        "Extended Tunnel ID of the LSP in LSP-IDENTIFIERS
        TLV. The all-zeros format is represented as
        0.0.0.0 and ::.";
    reference
        "RFC 8231: Path Computation Element
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
}
leaf admin-state {
    type boolean;
    default "true";
    description
        "The desired operational state";
}
leaf operational-state {
    type operational-state;
    description
        "The operational status of the LSP";
}
container delegated {
    description

```



```

    "The delegation related parameters";
leaf enabled {
    type boolean;
    default "false";
    description
        "LSP is delegated or not; set to true when
        delegated";
}
leaf peer {
    type leafref {
        path "/pcep/entity/peers/peer/addr";
    }
    must '(!./enabled = true())' {
        error-message "The LSP must be delegated";
    }
    description
        "At the PCC, the reference to the PCEP peer to
        which LSP is delegated; At the PCE, the
        reference to the PCEP peer which delegated this
        LSP";
}
leaf srp-id {
    type uint32 {
        range "1..4294967294";
    }
    description
        "The last SRP-ID-number associated with this
        LSP. The value 0x00000000 and 0xFFFFFFFF
        are reserved.";
}
}
container initiation {
    if-feature "pce-initiated";
    description
        "The PCE initiation related parameters";
    reference
        "RFC 8281: Path Computation Element Communication
        Protocol (PCEP) Extensions for PCE-Initiated LSP
        Setup in a Stateful PCE Model";
    leaf enabled {
        type boolean;
        default "false";
        description
            "Set to true if this LSP is initiated by a PCE";
    }
    leaf peer {
        type leafref {
            path "/pcep/entity/peers/peer/addr";
        }
    }
}

```

```

    must '(!./enabled = true())' {
        error-message "The LSP must be PCE-Initiated";
    }
    description
        "If the role is PCC, this leaf refers to the PCEP
        peer (PCE) that initiated this LSP. If the role
        is PCE, this leaf refers to the PCEP peer (PCC)
        where the LSP is initiated";
    }
}
leaf symbolic-path-name {
    type string;
    description
        "The symbolic path name associated with the LSP.";
    reference
        "RFC 8231: Path Computation Element Communication
        Protocol (PCEP) Extensions for Stateful PCE";
}
leaf last-error {
    type identityref {
        base lsp-error;
    }
    description
        "The last error for the LSP.";
}
leaf pst {
    type identityref {
        base te-types:path-signaling-type;
    }
    default "te-types:path-setup-rsvp";
    description
        "The Path Setup Type (PST). Note that the
        te-types model uses the term Path Signaling
        Type";
    reference
        "RFC 8408: Conveying Path Setup Type in PCE
        Communication Protocol (PCEP) Messages";
}
list association-list {
    if-feature "association";
    key "type id source global-source extended-id";
    description
        "List of all PCEP associations";
    leaf type {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "association-list/type";
        }
        description

```

```

        "PCEP Association Type";
    }
    leaf id {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "association-list/id";
        }
        description
            "PCEP Association ID";
    }
    leaf source {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "association-list/source";
        }
        description
            "PCEP Association Source.";
    }
    leaf global-source {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "association-list/global-source";
        }
        description
            "PCEP Global Association Source.";
    }
    leaf extended-id {
        type leafref {
            path "/pcep/entity/lsp-db/"
                + "association-list/extended-id";
        }
        description
            "Additional information to
            support unique identification.";
    }
    reference
        "RFC 8697: Path Computation Element Communication
        Protocol (PCEP) Extensions for Establishing
        Relationships between Sets of Label Switched
        Paths (LSPs)";
    }
}
}
}
container path-keys {
    when "../role = 'pce' or ../role = 'pcc-and-pce'" {
        description
            "These fields are applicable when the role is
            PCE";
    }
}

```

```

if-feature "path-key";
config false;
description
    "The path-keys generated by the PCE";
reference
    "RFC 5520: Preserving Topology Confidentiality
    in Inter-Domain Path Computation Using a Path-
    Key-Based Mechanism";
list path-key {
    key "key";
    description
        "The list of path-keys generated by the PCE";
    leaf key {
        type uint16;
        description
            "The identifier, or token used to represent
            the Confidential Path Segment (CPS) within
            the context of the PCE";
    }
    container cps {
        description
            "The Confidential Path Segment (CPS)";
        list explicit-route-objects {
            key "index";
            description
                "List of explicit route objects";
            leaf index {
                type uint32;
                description
                    "ERO subobject index";
            }
            uses te-types:explicit-route-hop;
        }
    }
}
leaf pcc-original {
    type leafref {
        path "/pcep/entity/peers/peer/addr";
    }
    description
        "Reference to PCC peer address of
        the original request";
}
leaf req-id {
    type uint32;
    description
        "The request ID of the original PCReq.";
}
leaf retrieved {
    type boolean;
}

```

```

        description
            "If path-key has been retrieved yet";
    }
    leaf pcc-retrieved {
        type leafref {
            path "/pcep/entity/peers/peer/addr";
        }
        must '(!./retrieved = true())' {
            error-message "The Path-key should be retrieved";
        }
        description
            "Reference to PCC peer address which
            retrieved the path-key";
    }
    leaf creation-time {
        type yang:timestamp;
        description
            "The timestamp value at the time this Path-Key
            was created.";
    }
    leaf discard-time {
        type uint32;
        units "minutes";
        description
            "A time after which this path-keys will be
            discarded";
    }
    leaf reuse-time {
        type uint32;
        units "minutes";
        description
            "A time after which this path-keys could be
            reused";
    }
}
}
}
container peers {
    description
        "The list of configured peers for the
        entity (remote PCE)";
    list peer {
        key "addr";
        description
            "The peer configured for the entity.
            (remote PCE)";
        leaf addr {
            type inet:ip-address-no-zone;
            description
                "The local Internet address of this

```

```

        PCEP peer.";
    }
leaf role {
    type role;
    must '(. != "pcc-and-pce")' {
        error-message
            "The PCEP peer cannot be both
            PCE and PCC at the same time";
    }
    mandatory true;
    description
        "The role of the PCEP Peer.
        Takes one of the following values.
        - unknown(0): this PCEP peer role is not
        known.
        - pcc(1): this PCEP peer is a PCC.
        - pce(2): this PCEP peer is a PCE.
        - pcc-and-pce(3): is not allowed as PCEP
        peer cannot be acting as both a PCC and a
        PCE at the sametime.";
}
leaf description {
    type string;
    description
        "Description of the PCEP peer
        configured by the user";
}
uses info {
    description
        "PCE Peer information";
}
container pce-info {
    uses pce-info {
        description
            "PCE Peer information";
    }
    description
        "The PCE Peer information";
}
leaf delegation-pref {
    if-feature "stateful";
    type uint8 {
        range "0..7";
    }
    description
        "The PCE peer delegation preference.";
}
uses auth {
    description

```

```
        "The PCE peer authorization and security
          parameters";
    }
    leaf discontinuity-time {
        type yang:timestamp;
        config false;
        description
            "The timestamp of the time when the information and
              statistics were last reset.";
    }
    leaf initiate-session {
        type boolean;
        config false;
        description
            "Indicates whether the local PCEP entity initiates
              sessions to this peer, or wait for the peer to
              initiate a session.";
    }
    leaf session-exists {
        type boolean;
        config false;
        description
            "Indicates whether a session with
              this peer currently exists.";
    }
    leaf session-up-time {
        type yang:timestamp;
        config false;
        description
            "The timestamp value of the last time a
              session with this peer was successfully
              established.";
    }
    leaf session-fail-time {
        type yang:timestamp;
        config false;
        description
            "The timestamp value of the last time a
              session with this peer failed to be
              established.";
    }
    leaf session-fail-up-time {
        type yang:timestamp;
        config false;
        description
            "The timestamp value of the last time a
              session with this peer failed from
              active.";
    }
}
```

```

container sessions {
  config false;
  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, auto-discovery of a peer,
    or an incoming TCP connection.";
  list session {
    key "initiator";
    description
      "The list of sessions, note that for a time being
      two sessions may exist for a peer";
    leaf initiator {
      type initiator;
      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 YANG module. The
        sessions are distinguished by who initiated
        them, and so this field is the key.";
    }
    leaf role {
      type leafref {
        path "/pcep/entity/role";
      }
      description
        "The reference to peer role .";
    }
    leaf state-last-change {
      type yang:timestamp;
      description
        "The timestamp value at the time this
        session entered its current state as
        denoted by the state leaf.";
    }
    leaf state {
      type sess-state;
      description
        "The current state of the session.

```



```

        The set of possible states excludes the
        idle state since entries do not exist
        in the idle state.";
    }
    leaf session-creation {
        type yang:timestamp;
        description
            "The timestamp value at the time this
            session was created.";
    }
    leaf connect-retry {
        type yang:counter32;
        description
            "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 connect-max-retry.";
    }
    leaf local-id {
        type uint8;
        description
            "The value of the PCEP session ID used by
            the local PCEP entity in the Open message
            for this session. If state is tcp-pending
            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.";
        reference
            "RFC 5440: Path Computation Element (PCE)
            Communication Protocol (PCEP)";
    }
    leaf remote-id {
        type uint8;
        must "(../state != 'tcp-pending'"
            + "and "
            + "../state != 'open-wait' )"
            + "or "
            + "(../state = 'tcp-pending'"
            + " or "
            + "../state = 'open-wait' )"
            + "and (. = 0))" {
            error-message "Invalid remote-id";
            description
                "If state is TCPPending or OpenWait then
                this leaf is not used and MUST be set to
                zero.";
        }
    }
}

```

```

description
    "The value of the PCEP session ID used by the
    peer in its Open message for this session.";
reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCE)";
}
leaf keepalive-timer {
    type uint8;
    units "seconds";
    must "(./state = 'session-up'"
        + "or "
        + "(./state != 'session-up'"
        + "and (. = 0)))" {
        error-message "Invalid Keepalive timer";
        description
            "This field is used if and only if state is
            session-up. Otherwise, it is not used and
            MUST be set to zero.";
    }
}
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.";
reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCE)";
}
leaf peer-keepalive-timer {
    type uint8;
    units "seconds";
    must "(./state = 'session-up'"
        + "or "
        + "(./state != 'session-up'"
        + "and "
        + "(. = 0)))" {
        error-message "Invalid Peer Keepalive timer";
        description
            "This field is used if and only if state is
            session-up. Otherwise, it is not used and MUST
            be set to zero.";
    }
}
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.";
reference

```

```

        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
    }
leaf dead-timer {
    type uint8;
    units "seconds";
    description
        "The DeadTimer interval for this PCEP session.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}
leaf peer-dead-timer {
    type uint8;
    units "seconds";
    must "((../state != 'tcp-pending'"
        + "and "
        + "../state != 'open-wait' )"
        + "or "
        + "((../state = 'tcp-pending'"
        + " or "
        + "../state = 'open-wait' )"
        + "and "
        + "(. = 0)))" {
    error-message "Invalid Peer DeadTimer";
    description
        "If state is TCPPending or OpenWait then this
        leaf is not used and MUST be set to zero.";
}
description
    "The peer's DeadTimer interval
    for this PCEP session.";
reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf ka-hold-time-rem {
    type uint8;
    units "seconds";
    must "((../state != 'tcp-pending'"
        + "and "
        + "../state != 'open-wait' )"
        + "or "
        + "((../state = 'tcp-pending'"
        + "or "
        + "../state = 'open-wait' )"
        + "and "
        + "(. = 0)))" {
    error-message

```

```

        "Invalid Keepalive hold time remaining";
    description
        "If state is TCPending or OpenWait then this
        field is not used and MUST be set to zero.";
    }
    description
        "The Keepalive hold time remaining for this
        session.";
    }
    leaf overloaded {
        type boolean;
        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.";
        reference
            "RFC 5440: Path Computation Element (PCE)
            Communication Protocol (PCEP)";
    }
    leaf overloaded-timestamp {
        when '../overloaded = true()' {
            description
                "Valid when overloaded";
        }
        type yang:timestamp;
        description
            "The timestamp value of the time when the
            overloaded field was set to true.";
    }
    leaf overload-time {
        type uint32;
        units "seconds";
        must '((../overloaded = true()) '
            + 'or ((../overloaded != true()) '
            + 'and (. = 0))' {
            error-message "Invalid overload-time";
            description
                "This field is only used if overloaded is set to
                true. Otherwise, it is not used and MUST be set
                to zero.";
        }
        description
            "The interval of time that is remaining until the
            local PCEP entity will cease to be overloaded on
            this session.";
        reference
            "RFC 5440: Path Computation Element (PCE)
            Communication Protocol (PCEP)";
    }
}

```

```

leaf peer-overloaded {
    type boolean;
    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.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}
leaf peer-overloaded-timestamp {
    when '../peer-overloaded = true()' {
        description
            "Valid when Peer is overloaded";
    }
    type yang:timestamp;
    description
        "The timestamp value of the time when the
        peer-overloaded field was set to true.";
}
leaf peer-overload-time {
    type uint32;
    units "seconds";
    must '((../peer-overloaded = '
        + 'true()) or '
        + '((../peer-overloaded !='
        + 'true()))'
        + ' and '
        + '(. = 0)))' {
        error-message "Invalid peer overload time";
        description
            "This field is only used if peer-overloaded
            is set to true. Otherwise, it is not used
            and MUST be set to zero.";
    }
    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 leaf is set to zero.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}
leaf lspdb-sync {
    if-feature "stateful";
    type sync-state;
    description
        "The LSP-DB state synchronization status.";
}

```

```

reference
    "RFC 8231: Path Computation Element Communication
    Protocol (PCEP) Extensions for Stateful PCE";
}
leaf recv-db-ver {
    when "../role = 'pcc'"
        + "or "
        + "../role = 'pcc-and-pce'" {
    description
        "This field is applicable when the role is
        PCC";
    }
    if-feature "stateful";
    if-feature "sync-opt";
    type uint64;
    description
        "The last received LSP State Database Version
        Number";
    reference
        "RFC 8231: Path Computation Element Communication
        Protocol (PCEP) Extensions for Stateful PCE";
}
container of-list {
    when "../role = 'pce'"
        + "or "
        + "../role = 'pcc-and-pce'" {
    description
        "These fields are applicable when the role is
        PCE";
    }
    if-feature "objective-function";
    uses of-list;
    description
        "Indicate the list of supported OF on this
        session";
    reference
        "RFC 5541: Encoding of Objective Functions in
        the Path Computation Element Communication
        Protocol (PCEP)";
}
container pst-list {
    when "../role = 'pce'"
        + "or "
        + "../role = 'pcc-and-pce'" {
    description
        "These fields are applicable when the role is
        PCE";
    }
    description

```

```

    "Indicate the list of supported
      PST on this session";
reference
  "RFC 8408: Conveying Path Setup Type in PCE
    Communication Protocol (PCEP) Messages";
list path-setup-type {
  key "pst";
  description
    "The list of PST";
  leaf pst {
    type identityref {
      base te-types:path-signaling-type;
    }
    description
      "The PST supported";
  }
}
}
container assoc-type-list {
  if-feature "association";
  description
    "Indicate the list of supported association types
      on this session";
reference
  "RFC 8697: Path Computation Element Communication
    Protocol (PCEP) Extensions for Establishing
    Relationships between Sets of Label Switched
    Paths (LSPs)";
list assoc-type {
  key "at";
  description
    "The list of authorized association types";
  leaf at {
    type identityref {
      base te-types:association-type;
    }
    description
      "The association type authorized";
  }
}
}
leaf speaker-entity-id {
  if-feature "sync-opt";
  type string;
  description
    "The Speaker Entity Identifier";
reference
  "RFC 8232: Optimizations of Label Switched
    Path State Synchronization Procedures for

```

```

        a Stateful PCE";
    }
}
}
}
}
}
}
}

/*
 * Notifications
 */

notification pcep-session-up {
    description
        "This notification is sent when the value of
        '/pcep/peers/peer/sessions/session/state'
        enters the 'session-up' state.";
    uses notification-instance-hdr;
    uses notification-session-hdr;
    leaf state-last-change {
        type yang:timestamp;
        description
            "The timestamp value at the time this session
            entered its current state as denoted by the state
            leaf.";
    }
    leaf state {
        type sess-state;
        description
            "The current state of the session.
            The set of possible states excludes the idle state
            since entries do not exist in the idle state.";
    }
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}

notification pcep-session-down {
    description
        "This notification is sent when the value of
        '/pcep/peers/peer/sessions/session/state'
        leaves the 'session-up' state.";
    uses notification-instance-hdr;
    leaf session-initiator {
        type initiator;
        description
            "The initiator of the session.";
    }
}

```



```

}
leaf state-last-change {
  type yang:timestamp;
  description
    "The timestamp value at the time this session
    entered its current state as denoted by the state
    leaf.";
}
leaf state {
  type sess-state;
  description
    "The current state of the session.
    The set of possible states excludes the idle state
    since entries do not exist in the idle state.";
}
reference
  "RFC 5440: Path Computation Element (PCE) Communication
  Protocol (PCEP)";
}

notification pcep-session-local-overload {
  description
    "This notification is sent when the local PCEP entity
    enters overload state for a peer.";
  uses notification-instance-hdr;
  uses notification-session-hdr;
  leaf overloaded {
    type boolean;
    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.";
  }
  leaf overloaded-timestamp {
    type yang:timestamp;
    description
      "The timestamp value of the time when the
      overloaded field was set to true.";
  }
  leaf overload-time {
    type uint32;
    units "seconds";
    description
      "The interval of time that is remaining until the
      local PCEP entity will cease to be overloaded on
      this session.";
  }
}
reference
  "RFC 5440: Path Computation Element (PCE) Communication

```

```

    Protocol (PCEP)";
}

notification pcep-session-local-overload-clear {
  description
    "This notification is sent when the local PCEP entity
    leaves overload state for a peer.";
  uses notification-instance-hdr;
  leaf overloaded {
    type boolean;
    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.";
  }
  leaf overloaded-clear-timestamp {
    type yang:timestamp;
    description
      "The timestamp value of the time when the
      overloaded field was set to false.";
  }
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

notification pcep-session-peer-overload {
  description
    "This notification is sent when a peer enters overload
    state.";
  uses notification-instance-hdr;
  uses notification-session-hdr;
  leaf peer-overloaded {
    type boolean;
    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.";
  }
  leaf peer-overloaded-timestamp {
    type yang:timestamp;
    description
      "The timestamp value of the time when the
      peer-overloaded field was set to true.";
  }
  leaf peer-overload-time {
    type uint32;
    units "seconds";
    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 leaf is set to zero.";
    }
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}

notification pcep-session-peer-overload-clear {
    description
        "This notification is sent when a peer leaves overload
        state.";
    uses notification-instance-hdr;
    leaf peer-overloaded {
        type boolean;
        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.";
    }
    leaf peer-overloaded-clear-timestamp {
        type yang:timestamp;
        description
            "The timestamp value of the time when the
            peer-overloaded field was set to false.";
    }
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}

/*
 * RPC
 */

rpc trigger-resync {
    if-feature "stateful";
    if-feature "sync-opt";
    nacm:default-deny-all;
    description
        "Trigger the resynchronization at the PCE";
    reference
        "RFC 8232: Optimizations of Label Switched Path State
        Synchronization Procedures for a Stateful PCE";
    input {
        leaf pcc {
            type leafref {

```

```
    path "/pcep/entity/peers/peer/addr";
  }
  description
    "The IP address to identify the PCC. The state
    synchronization is re-triggered for all LSPs from
    the PCC. The rpc on the PCC will be ignored.";
  }
}
}
```

<CODE ENDS>

8.2. ietf-pcep-stats module

```
<CODE BEGINS> file "ietf-pcep-stats@2023-09-11.yang"
```

```
module ietf-pcep-stats {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-pcep-stats";
  prefix pcep-stats;

  import ietf-pcep {
    prefix pcep;
    reference
      "RFC XXXX: A YANG Data Model for Path Computation
      Element Communications Protocol (PCEP)";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  organization
    "IETF PCE (Path Computation Element) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/pce/>
    WG List: <mailto:pce@ietf.org>
    Editor: Dhruv Dhody
           <mailto:dhruv.ietf@gmail.com>";
  description
    "The YANG module augments the Path Computation Element
    Communication Protocol (PCEP) YANG operational
    model with statistics, counters and telemetry data.

    Copyright (c) 2023 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Revised BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see the
    RFC itself for full legal notices.";

  revision 2023-09-11 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for Path Computation
```

```

        Element Communications Protocol (PCEP)";
    }

/*
 * Groupings
 */

grouping stats {
    description
        "This grouping defines statistics for PCEP. It is used
        for both peer and current sessions.";
    leaf discontinuity-time {
        type yang:timestamp;
        description
            "The timestamp value of the time when the
            statistics were last reset.";
    }
    leaf rsp-time-avg {
        when "../..//pcep:role = 'pce'"
            + "or "
            + "../..//pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type uint32;
        units "milliseconds";
        description
            "The average response time. If an average response time
            has not been calculated then this leaf has the value
            zero.";
    }
    leaf rsp-time-lwm {
        when "../..//pcep:role = 'pce'"
            + "or "
            + "../..//pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type uint32;
        units "milliseconds";
        description
            "The smallest (low-water mark) response time seen.
            If no responses have been received then this leaf has
            the value zero.";
    }
    leaf rsp-time-hwm {
        when "../..//pcep:role = 'pce'"
            + "or "
            + "../..//pcep:role = 'pcc-and-pce'" {

```

```

    description
      "Valid for PCEP Peer as PCE";
  }
  type uint32;
  units "milliseconds";
  description
    "The greatest (high-water mark) response time seen.
    If no responses have been received then this object
    has the value zero.";
}
leaf pcreq-sent {
  when "../..//pcep:role = 'pce'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCReq messages sent.";
}
leaf pcreq-rcvd {
  when "../..//pcep:role = 'pcc'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCReq messages received.";
}
leaf pcrep-sent {
  when "../..//pcep:role = 'pcc'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCRep messages sent.";
}
leaf pcrep-rcvd {
  when "../..//pcep:role = 'pce'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }

```



```

    }
    type yang:counter32;
    description
        "The number of PCRep messages received.";
}
leaf pcerr-sent {
    type yang:counter32;
    description
        "The number of PCErr messages sent.";
}
leaf pcerr-rcvd {
    type yang:counter32;
    description
        "The number of PCErr messages received.";
}
leaf pcntf-sent {
    type yang:counter32;
    description
        "The number of PCNtf messages sent.";
}
leaf pcntf-rcvd {
    type yang:counter32;
    description
        "The number of PCNtf messages received.";
}
leaf keepalive-sent {
    type yang:counter32;
    description
        "The number of Keepalive messages sent.";
}
leaf keepalive-rcvd {
    type yang:counter32;
    description
        "The number of Keepalive messages received.";
}
leaf unknown-rcvd {
    type yang:counter32;
    description
        "The number of unknown messages received.";
}
leaf corrupt-rcvd {
    type yang:counter32;
    description
        "The number of corrupted PCEP messages received.";
}
leaf req-sent {
    when "../..//pcep:role = 'pce'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {

```

```

    description
        "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of requests sent. A request corresponds
    1:1 with an RP object in a PCReq message. This might
    be greater than pcreq-sent because multiple
    requests can be batched into a single PCReq
    message.";
}
leaf req-sent-pend-rep {
    when "../..pcep:role = 'pce'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of requests that have been sent for
    which a response is still pending.";
}
leaf req-sent-ero-rcvd {
    when "../..pcep:role = 'pce'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of requests that have been sent for
    which a response with an ERO object was received.
    Such responses indicate that a path was
    successfully computed by the peer.";
}
leaf req-sent-nopath-rcvd {
    when "../..pcep:role = 'pce'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of requests that have been sent 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.";
    }
leaf req-sent-cancel-rcvd {
    when "../..//pcep:role = 'pce'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of requests that were cancelled with
        a PCNtf message. This might be different than
        pcntf-rcvd because not all PCNtf messages are
        used to cancel requests, and a single PCNtf message
        can cancel multiple requests.";
}
leaf req-sent-error-rcvd {
    when "../..//pcep:role = 'pce'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of requests that were rejected with a
        PCErr message. This might be different than
        pcerr-rcvd because not all PCErr messages are
        used to reject requests, and a single PCErr message
        can reject multiple requests.";
}
leaf req-sent-timeout {
    when "../..//pcep:role = 'pce'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    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.";
}
leaf req-sent-cancel-sent {
    when "../..//pcep:role = 'pce'"
        + "or "

```

```

    + "../..pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of requests that were sent to the peer and
    explicitly cancelled by the local PCEP entity sending
    a PCNtf.";
}
leaf req-rcvd {
    when "../..pcep:role = 'pcc'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;
description
    "The number of requests received. A request
    corresponds 1:1 with an RP object in a PCReq
    message.
    This might be greater than pcreq-rcvd because
    multiple requests can be batched into a single
    PCReq message.";
}
leaf req-rcvd-pend-rep {
    when "../..pcep:role = 'pcc'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;
description
    "The number of requests that have been received for
    which a response is still pending.";
}
leaf req-rcvd-ero-sent {
    when "../..pcep:role = 'pcc'"
        + "or "
        + "../..pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;
description
    "The number of requests that have been received for
    which a response with an ERO object was sent. Such

```

```

        responses indicate that a path was successfully
        computed by the local PCEP entity.";
    }
leaf req-rcvd-nopath-sent {
    when "../..//pcep:role = 'pcc'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests that have been received 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.";
}
leaf req-rcvd-cancel-sent {
    when "../..//pcep:role = 'pcc'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests received that were cancelled
        by the local PCEP entity sending a PCNtf message.
        This might be different than pcntf-sent because
        not all PCNtf messages are used to cancel requests,
        and a single PCNtf message can cancel multiple
        requests.";
}
leaf req-rcvd-error-sent {
    when "../..//pcep:role = 'pcc'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests received that were cancelled
        by the local PCEP entity sending a PCErr message.
        This might be different than pcerr-sent because
        not all PCErr messages are used to cancel requests,
        and a single PCErr message can cancel multiple
        requests.";
}
}

```

```

leaf req-rcvd-cancel-rcvd {
  when "../..//pcep:role = 'pcc'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests that were received from the
    peer and explicitly cancelled by the peer sending
    a PCNtf.";
}
leaf rep-rcvd-unknown {
  when "../..//pcep:role = 'pce'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of responses to unknown requests
    received. 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.";
}
leaf req-rcvd-unknown {
  when "../..//pcep:role = 'pcc'"
    + "or "
    + "../..//pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of unknown requests that have been
    received. An unknown request is a request
    whose RP object contains a request ID of zero.";
}
container svec {
  if-feature "pcep:svec";
  description
    "If synchronized path computation is supported";
  leaf svec-sent {
    when "../..//..//pcep:role = 'pce'"
      + "or "
      + "../..//..//pcep:role = 'pcc-and-pce'" {

```

```

        description
            "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of SVEC objects sent in PCReq messages.
        An SVEC object represents a set of synchronized
        requests.";
    }
    leaf svec-req-sent {
        when "../.../pcep:role = 'pce'"
            + "or "
            + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests sent that appeared in one
            or more SVEC objects.";
    }
    leaf svec-rcvd {
        when "../.../pcep:role = 'pcc'"
            + "or "
            + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of SVEC objects received in PCReq
            messages. An SVEC object represents a set of
            synchronized requests.";
    }
    leaf svec-req-rcvd {
        when "../.../pcep:role = 'pcc'"
            + "or "
            + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of requests received that appeared
            in one or more SVEC objects.";
    }
}
container stateful {
    if-feature "pcep:stateful";

```

```

description
  "Stateful PCE related statistics";
leaf pcrpt-sent {
  when "../.../..../pcep:role = 'pce'"
    + "or "
    + "../.../..../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCRpt messages sent.";
}
leaf pcrpt-rcvd {
  when "../.../..../pcep:role = 'pcc'"
    + "or "
    + "../.../..../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCRpt messages received.";
}
leaf pcupd-sent {
  when "../.../..../pcep:role = 'pcc'"
    + "or "
    + "../.../..../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCUpd messages sent.";
}
leaf pcupd-rcvd {
  when "../.../..../pcep:role = 'pce'"
    + "or "
    + "../.../..../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCUpd messages received.";
}
leaf rpt-sent {
  when "../.../..../pcep:role = 'pce'"
    + "or "

```



```

    + "../../../pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
    "The number of LSP Reports sent. An LSP report
    corresponds 1:1 with an LSP object in a PCRpt
    message. This might be greater than
    pcrpt-sent because multiple reports can
    be batched into a single PCRpt message.";
}
leaf rpt-rcvd {
when "../../../pcep:role = 'pcc'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;
description
    "The number of LSP Reports received. An LSP report
    corresponds 1:1 with an LSP object in a PCRpt
    message.
    This might be greater than pcrpt-rcvd because
    multiple reports can be batched into a single
    PCRpt message.";
}
leaf rpt-rcvd-error-sent {
when "../../../pcep:role = 'pcc'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;
description
    "The number of reports of LSPs received that were
    responded by the local PCEP entity by sending a
    PCErr message.";
}
leaf upd-sent {
when "../../../pcep:role = 'pcc'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
description
    "Valid for PCEP Peer as PCC";
}
type yang:counter32;

```

```

description
  "The number of LSP updates sent. An LSP update
  corresponds 1:1 with an LSP object in a PCUpd
  message. This might be greater than
  pcupd-sent because multiple updates can
  be batched into a single PCUpd message.";
}
leaf upd-rcvd {
  when "../../../pcep:role = 'pce'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of LSP Updates received. An LSP update
    corresponds 1:1 with an LSP object in a PCUpd
    message.
    This might be greater than pcupd-rcvd because
    multiple updates can be batched into a single
    PCUpd message.";
}
leaf upd-rcvd-unknown {
  when "../../../pcep:role = 'pce'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of updates to unknown LSPs
    received. An update to an unknown LSP is a
    update whose LSP object does not contain the
    PLSP-ID of any LSP that is currently
    present.";
}
leaf upd-rcvd-undelegated {
  when "../../../pcep:role = 'pce'"
    + "or "
    + "../../../pcep:role = 'pcc-and-pce'" {
  description
    "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of updates to not delegated LSPs
    received. An update to an undelegated LSP is a

```

```

        update whose LSP object does not contain the
        PLSP-ID of any LSP that is currently
        delegated to the current PCEP session.";
    }
leaf upd-rcvd-error-sent {
    when "../../../pcep:role = 'pce'"
        + "or "
        + "../../../pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of updates to LSPs received that were
        responded by the local PCEP entity by sending a
        PCErr message.";
}
container initiation {
    if-feature "pcep:pce-initiated";
    description
        "PCE-Initiated related statistics";
    leaf pcinitiate-sent {
        when "../../../pcep:role = 'pcc'"
            + "or "
            + "../../../pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of PCInitiate messages sent.";
    }
    leaf pcinitiate-rcvd {
        when "../../../pcep:role = 'pce'"
            + "or "
            + "../../../pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of PCInitiate messages received.";
    }
}
leaf initiate-sent {
    when "../../../pcep:role = 'pcc'"
        + "or "
        + "../../../pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCC";
}

```

```

    }
    type yang:counter32;
    description
        "The number of LSP Initiation sent via PCE.
        An LSP initiation corresponds 1:1 with an LSP
        object in a PCInitiate message. This might be
        greater than pcinitiate-sent because
        multiple initiations can be batched into a
        single PCInitiate message.";
    }
    leaf initiate-rcvd {
        when "../.../.../pcep:role = 'pce'"
            + "or "
            + "../.../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of LSP Initiation received from
            PCE. An LSP initiation corresponds 1:1 with
            an LSP object in a PCInitiate message. This
            might be greater than pcinitiate-rcvd
            because multiple initiations can be batched
            into a single PCInitiate message.";
    }
    leaf initiate-rcvd-error-sent {
        when "../.../.../pcep:role = 'pce'"
            + "or "
            + "../.../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of initiations of LSPs received
            that were responded to by the local PCEP entity
            by sending a PCErr message.";
    }
    }
}
}
container path-key {
    when "../.../pcep:role = 'pcc'"
        + "or "
        + "../.../pcep:role = 'pcc-and-pce'" {
        description
            "Valid for PCEP Peer as PCC";
    }
}
if-feature "pcep:path-key";

```

```

description
  "If Path-Key is supported";
leaf unknown-path-key {
  type yang:counter32;
  description
    "The number of attempts to expand an unknown
    path-key.";
}
leaf exp-path-key {
  type yang:counter32;
  description
    "The number of attempts to expand an expired
    path-key.";
}
leaf dup-path-key {
  type yang:counter32;
  description
    "The number of duplicate attempts to expand same
    path-key.";
}
leaf path-key-no-attempt {
  type yang:counter32;
  description
    "The number of expired path-keys with no attempt to
    expand it.";
}
}
}

/*
 * Augment modules to add statistics
 */

augment "/pcep:pcep/pcep:entity/pcep:peers/pcep:peer" {
  description
    "Augmenting the statistics";
  leaf sess-setup-ok {
    type yang:counter32;
    config false;
    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.";
  }
  leaf sess-setup-fail {
    type yang:counter32;
    config false;
    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.";
    }
    container stats {
        config false;
        description
            "The container for all statistics at peer level.";
        uses stats {
            description
                "Since PCEP sessions can be ephemeral, the peer statistics
                tracks a peer even when no PCEP session currently exists
                to that peer. The statistics contained are an aggregate of
                the statistics for all successive sessions to that peer.";
        }
        leaf req-sent-closed {
            when "../pcep:role = 'pce'"
                + "or "
                + "../pcep:role = 'pcc-and-pce'" {
                description
                    "Valid for PCEP Peer as PCE";
            }
            type yang:counter32;
            description
                "The number of requests that were sent to the peer and
                implicitly cancelled when the session they were sent
                over was closed.";
        }
        leaf req-rcvd-closed {
            when "../pcep:role = 'pcc'"
                + "or "
                + "../pcep:role = 'pcc-and-pce'" {
                description
                    "Valid for PCEP Peer as PCC";
            }
            type yang:counter32;
            description
                "The number of requests that were received from the peer
                and implicitly cancelled when the session they were
                received over was closed.";
        }
    }
}

augment "/pcep:pcep/pcep:entity/pcep:peers/pcep:peer/"
    + "pcep:sessions/pcep:session" {
    description

```

```

    "Augmenting the statistics";
container stats {
  description
    "The container for all statistics at session level.";
  uses stats {
    description
      "The statistics contained are for the current sessions to
      that peer. These are lost when the session goes down.";
  }
}
}

rpc statistics-reset {
  description
    "Reset statistics collected.";
  input {
    choice peer-or-all {
      description
        "Resets statistics for a particular peer or
        all";
      case peer {
        leaf peer-addr {
          type leafref {
            path "/pcep:pcep/pcep:entity/pcep:peers"
              + "/pcep:peer/pcep:addr";
          }
          description
            "Reference to peer address";
        }
        description
          "This resets all the statistics collected for
          the peer.";
      }
      case all {
        description
          "This resets all the statistics collected.";
      }
    }
  }
}
}
}

```

<CODE ENDS>

9. Security Considerations

The YANG modules defined in this document are designed to be accessed via network management protocol such as NETCONF [[RFC6241](#)]

or RESTCONF [[RFC8040](#)]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [[RFC6242](#)]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [[RFC8446](#)]

The NETCONF access control model [[RFC8341](#)] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the ietf-pcep YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

`/pcep/entity/` - configure local parameters, capabilities etc.

`/pcep/entity/peers` - configure remote peers to setup PCEP session.

Unauthorized access to above list can adversely affect the PCEP session between the local entity and the peers. This may lead to inability to compute new paths, stateful operations on the delegated as well as PCE-initiated LSPs.

Some of the readable data nodes in the ietf-pcep YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`/pcep/lsp-db` - All the LSPs in the network. Unauthorized access to this could provide the all path and network usage information.

`/pcep/path-keys/` - The Confidential Path Segments (CPS) are hidden using path-keys. Unauthorized access to this could leak confidential path information.

Some of the RPC operations in the ietf-pcep YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`trigger-resync` - trigger resynchronization with the PCE. Unauthorized access to this could force a PCEP session into continuous state synchronization.

The actual authentication key data (whether locally specified or part of a key-chain) is sensitive and needs to be kept secret from unauthorized parties; compromise of the key data would allow an attacker to forge PCEP traffic that would be accepted as authentic, potentially compromising the TE domain.

The model describes several notifications, implementations must rate-limit the generation of these notifications to avoid creating significant notification load. Otherwise, this notification load may have some side effects on the system stability and may be exploited as an attack vector.

Further, this document also include another YANG module (called `ietf-pcep-stats`) for maintaining the statistics by augmenting the `ietf-pcep` YANG module. There are no data nodes defined in this module which are writable/creatable/deletable (i.e., `config true`). The readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. The statistics could provide information related to the current usage patterns of the network. It is thus important to control read access (e.g., via `get`, `get-config`, or notification).

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`statistics-reset` - The RPC is used to reset statistics.
Unauthorized reset could impact monitoring.

The "auth" container includes various authentication and security options for PCEP. Further, [Section 7.1](#) describes how to configure TLS1.2 and TLS1.3 for a PCEP session via this YANG module.

10. IANA Considerations

This document request the IANA to register two URIs in the "IETF XML Registry" [[RFC3688](#)]. Following the format in RFC 3688, the following registrations are requested -

URI: `urn:ietf:params:xml:ns:yang:ietf-pcep`
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

URI: `urn:ietf:params:xml:ns:yang:ietf-pcep-stats`
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

This document request the IANA to register two YANG modules in the "YANG Module Names" registry [[RFC6020](#)], as follows -

Name: ietf-pcep
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix: pcep
Reference: [This.I-D]

Name: ietf-pcep-stats
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep-stats
Prefix: pcep-stats
Reference: [This.I-D]

11. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to RFC 7942.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [[RFC7942](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [[RFC7942](#)], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Currently, there are no known implementations of the YANG Module as specified.

12. Acknowledgements

The initial document is based on the PCEP MIB [[RFC7420](#)]. We would like to thank the authors of the aforementioned documents.

Thanks to Martin Bjorklund and Tom Petch for detailed review and providing guidance regarding the YANG guidelines. Thanks to Mahesh

Jethanandani for the YANGDOCTOR review. Thanks to Scott Kelly for the SECDIR review. Thanks to Gyan Mishra for the RTGDIR review.

13. References

13.1. Normative References

[I-D.ietf-netconf-tls-client-server]

Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tls-client-server-33, 17 April 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-tls-client-server-33>>.

[I-D.ietf-teas-yang-te] Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. de Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-33, 4 July 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-33>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<https://www.rfc-editor.org/info/rfc5088>>.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<https://www.rfc-editor.org/info/rfc5089>>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520,

DOI 10.17487/RFC5520, April 2009, <<https://www.rfc-editor.org/info/rfc5520>>.

- [RFC5541]** Le Roux, J.L., Vasseur, J.P., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<https://www.rfc-editor.org/info/rfc5541>>.
- [RFC5557]** Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <<https://www.rfc-editor.org/info/rfc5557>>.
- [RFC5925]** Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6020]** Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241]** Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol

(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.

[RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE

Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

[RFC8282] Oki, E., Takeda, T., Farrel, A., and F. Zhang, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 8282, DOI 10.17487/RFC8282, December 2017, <<https://www.rfc-editor.org/info/rfc8282>>.

[RFC8306] Zhao, Q., Dhody, D., Ed., Palleti, R., and D. King, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 8306, DOI 10.17487/RFC8306, November 2017, <<https://www.rfc-editor.org/info/rfc8306>>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

[RFC8408] Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages", RFC 8408, DOI 10.17487/RFC8408, July 2018, <<https://www.rfc-editor.org/info/rfc8408>>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

[RFC8623] Palle, U., Dhody, D., Tanaka, Y., and V. Beeram, "Stateful Path Computation Element (PCE) Protocol Extensions for Usage with Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 8623, DOI 10.17487/RFC8623, June 2019, <<https://www.rfc-editor.org/info/rfc8623>>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.

[RFC8685] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas, R., and D. King, "Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path

Computation Element (H-PCE) Architecture", RFC 8685, DOI 10.17487/RFC8685, December 2019, <<https://www.rfc-editor.org/info/rfc8685>>.

[RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <<https://www.rfc-editor.org/info/rfc8697>>.

[RFC8745] Ananthakrishnan, H., Sivabalan, S., Barth, C., Minei, I., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extensions for Associating Working and Protection Label Switched Paths (LSPs) with Stateful PCE", RFC 8745, DOI 10.17487/RFC8745, March 2020, <<https://www.rfc-editor.org/info/rfc8745>>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

[RFC8779] Margaria, C., Ed., Gonzalez de Dios, O., Ed., and F. Zhang, Ed., "Path Computation Element Communication Protocol (PCEP) Extensions for GMPLS", RFC 8779, DOI 10.17487/RFC8779, July 2020, <<https://www.rfc-editor.org/info/rfc8779>>.

[RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.

[RFC9005] Litkowski, S., Sivabalan, S., Tantsura, J., Hardwick, J., and C. Li, "Path Computation Element Communication Protocol (PCEP) Extension for Associating Policies and Label Switched Paths (LSPs)", RFC 9005, DOI 10.17487/RFC9005, March 2021, <<https://www.rfc-editor.org/info/rfc9005>>.

[RFC9129] Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "YANG Data Model for the OSPF Protocol", RFC 9129, DOI 10.17487/RFC9129, October 2022, <<https://www.rfc-editor.org/info/rfc9129>>.

[RFC9130] Litkowski, S., Ed., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for the IS-IS Protocol",

RFC 9130, DOI 10.17487/RFC9130, October 2022, <<https://www.rfc-editor.org/info/rfc9130>>.

[RFC9168] Dhody, D., Farrel, A., and Z. Li, "Path Computation Element Communication Protocol (PCEP) Extension for Flow Specification", RFC 9168, DOI 10.17487/RFC9168, January 2022, <<https://www.rfc-editor.org/info/rfc9168>>.

[RFC9353] Lopez, D., Wu, Q., Dhody, D., Ma, Q., and D. King, "IGP Extension for Path Computation Element Communication Protocol (PCEP) Security Capability Support in PCE Discovery (PCED)", RFC 9353, DOI 10.17487/RFC9353, January 2023, <<https://www.rfc-editor.org/info/rfc9353>>.

[RFC9358] Lee, Y., Zheng, H., and D. Ceccarelli, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths and Virtual Networks", RFC 9358, DOI 10.17487/RFC9358, February 2023, <<https://www.rfc-editor.org/info/rfc9358>>.

13.2. Informative References

[I-D.dhody-pce-pceps-tls13] Dhody, D., Turner, S., and R. Housley, "Updates for PCEPS", Work in Progress, Internet-Draft, draft-dhody-pce-pceps-tls13-02, 13 March 2023, <<https://datatracker.ietf.org/doc/html/draft-dhody-pce-pceps-tls13-02>>.

[IANA-IGP] "Path Computation Element (PCE) Capability Flags in Interior Gateway Protocol (IGP) Parameters", <<https://www.iana.org/assignments/igp-parameters/igp-parameters.xhtml#pce-capability-flags>>.

[IANA-PCEP] "ASSOCIATION Type Field in Path Computation Element Protocol (PCEP) Numbers", <<https://www.iana.org/assignments/pcep/pcep.xhtml#association-type-field>>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication

Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <<https://www.rfc-editor.org/info/rfc7420>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

[RFC8751] Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE)", RFC 8751, DOI 10.17487/RFC8751, March 2020, <<https://www.rfc-editor.org/info/rfc8751>>.

Appendix A. The Full PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker. The tree depth in the tree is set to 10.

```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--rw addr                inet:ip-address-no-zone
    +--rw enabled?            boolean
    +--rw role                 role
    +--rw description?        string
    +--rw speaker-entity-id?  string {sync-opt}?
    +--rw admin-status?       boolean
    +--ro index?              uint32
    +--ro oper-status?        oper-status
    +--rw domains
      | +--rw domain* [type info]
      |   +--rw type      identityref
      |   +--rw info      domain
    +--rw capabilities
      | +--rw capability?          bits
      | +--rw pce-initiated?       boolean {pce-initiated}?
      | +--rw include-db-ver?       boolean {stateful,sync-opt}?
      | +--rw trigger-resync?       boolean {stateful,sync-opt}?
      | +--rw trigger-initial-sync? boolean {stateful,sync-opt}?
      | +--rw incremental-sync?     boolean {stateful,sync-opt}?
      | +--rw sr {sr}?
      | | +--rw enabled?            boolean
      | | +--rw msd-limit?         boolean
      | | +--rw nai?               boolean
      | +--rw stateful-gmpls {stateful,gmpls}?
      | | +--rw enabled?           boolean
      | +--rw inter-layer?         boolean {inter-layer}?
      | +--rw h-pce {h-pce}?
      |   +--rw enabled?           boolean
      |   +--rw stateful?         boolean {stateful}?
      |   +--rw role?             hpce-role
    +--rw msd?                    uint8 {sr}?
    +--rw auth
      | +--rw (auth-type-selection)?
      |   +--:(auth-key-chain)
      |     | +--rw key-chain?
      |     |   key-chain:key-chain-ref
      |     +--:(auth-key)
      |       | +--rw crypto-algorithm      identityref
      |       | +--rw (key-string-style)?
      |       |   +--:(keystring)
      |       |     | +--rw keystring?      string
      |       |     +--:(hexadecimal) {key-chain:hex-key-string}?
      |       |       +--rw hexadecimal-string? yang:hex-string
      |     +--:(auth-tls) {tls}?
      |       +--rw (role)?
      |         +--:(server)

```

```

|         | |--rw tls-server
|         | |--rw server-identity
|         | | |--rw (auth-type)
|         | |     ...
|         | |--rw client-authentication!
|         | |     {client-auth-supported}?
|         | | |--rw ca-certs! {client-auth-x509-cert}?
|         | | |     ...
|         | | |--rw ee-certs! {client-auth-x509-cert}?
|         | | |     ...
|         | | |--rw raw-public-keys!
|         | | |     {client-auth-raw-public-key}?
|         | | |     ...
|         | |--rw tls12-psks?         empty
|         | |     {client-auth-tls12-psk}?
|         | |--rw tls13-epsks?       empty
|         | |     {client-auth-tls13-epsk}?
|         |--rw hello-params {tlscmn:hello-params}?
|         | |--rw tls-versions
|         | |     ...
|         | |--rw cipher-suites
|         | |     ...
|         |--rw keepalives {tls-server-keepalives}?
|         | |--rw peer-allowed-to-send? empty
|         | |--rw test-peer-aliveness!
|         | |     ...
| |--:(client)
| |--rw tls-client
| |--rw client-identity!
| | |--rw (auth-type)
| |     ...
| |--rw server-authentication
| | |--rw ca-certs! {server-auth-x509-cert}?
| | |     ...
| | |--rw ee-certs! {server-auth-x509-cert}?
| | |     ...
| | |--rw raw-public-keys!
| | |     {server-auth-raw-public-key}?
| | |     ...
| |--rw tls12-psks?         empty
| |     {server-auth-tls12-psk}?
| |--rw tls13-epsks?       empty
| |     {server-auth-tls13-epsk}?
|--rw hello-params {tlscmn:hello-params}?
| |--rw tls-versions
| |     ...
| |--rw cipher-suites
| |     ...
|--rw keepalives {tls-client-keepalives}?

```

```

|                                     +--rw peer-allowed-to-send?  empty
|                                     +--rw test-peer-aliveness!
|                                     ...
+--rw pce-info
| +--rw scope
| | +--rw path-scope?           bits
| | +--rw intra-area-pref?     uint8
| | +--rw inter-area-pref?     uint8
| | +--rw inter-as-pref?       uint8
| | +--rw inter-layer-pref?    uint8
| +--rw neighbour-domains
| | +--rw domain* [type info]
| |   +--rw type      identityref
| |   +--rw info      domain
| +--rw path-key {path-key}?
|   +--rw enabled?      boolean
|   +--rw discard-timer? uint32
|   +--rw reuse-time?   uint32
|   +--rw pce-id?       inet:ip-address-no-zone
+--rw connect-timer?          uint16
+--rw connect-max-retry?      uint32
+--rw init-back-off-timer?    uint16
+--rw max-back-off-timer?     uint32
+--ro open-wait-timer?        uint16
+--ro keep-wait-timer?        uint16
+--rw keepalive-timer?        uint8
+--rw dead-timer?            uint8
+--rw allow-negotiation?      boolean
+--rw max-keepalive-timer?    uint8
+--rw max-dead-timer?         uint8
+--rw min-keepalive-timer?    uint8
+--rw min-dead-timer?         uint8
+--rw sync-timer?            uint16 {svec}?
+--rw request-timer?         uint16
+--rw max-sessions?          uint32
+--rw max-unknown-reqs?      uint32
+--rw max-unknown-msgs?      uint32
+--rw pcep-notification-max-rate  uint32
+--rw stateful-parameter {stateful}?
| +--rw state-timeout?        uint32
| +--rw redelegation-timeout? uint32
| +--rw rpt-non-pcep-lsp?     boolean
+--rw of-list {objective-function}?
| +--rw objective-function* [of]
|   +--rw of      identityref
+--ro lsp-db {stateful}?
| +--ro db-ver?                uint64 {sync-opt}?
| +--ro association-list*
| |   [type id source global-source extended-id]

```

```

| |         {association}?
| | +--ro type             identityref
| | +--ro id               uint16
| | +--ro source           inet:ip-address-no-zone
| | +--ro global-source    uint32
| | +--ro extended-id      string
| | +--ro lsp* [plsp-id pcc-id lsp-id]
| |   +--ro plsp-id        -> /pcep/entity/lsp-db/lsp/plsp-id
| |   +--ro pcc-id         -> /pcep/entity/lsp-db/lsp/pcc-id
| |   +--ro lsp-id         -> /pcep/entity/lsp-db/lsp/lsp-id
| +--ro lsp* [plsp-id pcc-id lsp-id]
|   +--ro plsp-id          uint32
|   +--ro pcc-id           inet:ip-address-no-zone
|   +--ro source?          inet:ip-address-no-zone
|   +--ro destination?    inet:ip-address-no-zone
|   +--ro tunnel-id?       uint16
|   +--ro lsp-id           uint16
|   +--ro extended-tunnel-id? inet:ip-address-no-zone
|   +--ro admin-state?     boolean
|   +--ro operational-state? operational-state
|   +--ro delegated
|     | +--ro enabled?     boolean
|     | +--ro peer?        -> /pcep/entity/peers/peer/addr
|     | +--ro srp-id?      uint32
|   +--ro initiation {pce-initiated}?
|     | +--ro enabled?     boolean
|     | +--ro peer?        -> /pcep/entity/peers/peer/addr
|   +--ro symbolic-path-name? string
|   +--ro last-error?      identityref
|   +--ro pst?              identityref
|   +--ro association-list*
|     [type id source global-source extended-id]
|     {association}?
|     +--ro type
|       | -> /pcep/entity/lsp-db/association-list/type
|     +--ro id
|       | -> /pcep/entity/lsp-db/association-list/id
|     +--ro source
|       | -> /pcep/entity/lsp-db/association-list/source
|     +--ro global-source    leafref
|     +--ro extended-id      leafref
+--ro path-keys {path-key}?
| +--ro path-key* [key]
|   +--ro key                uint16
|   +--ro cps
|     | +--ro explicit-route-objects* [index]
|     |   +--ro index                uint32
|     |   +--ro (type)?
|     |   +--:(numbered-node-hop)

```

```

|   |   |   |--ro numbered-node-hop
|   |   |   |--ro node-id-uri?   nw:node-id
|   |   |   |--ro node-id?       te-node-id
|   |   |   |--ro hop-type?      te-hop-type
|   |   |--:(numbered-link-hop)
|   |   |   |--ro numbered-link-hop
|   |   |   |--ro link-tp-id     te-tp-id
|   |   |   |--ro hop-type?     te-hop-type
|   |   |   |--ro direction?    te-link-direction
|   |   |--:(unnumbered-link-hop)
|   |   |   |--ro unnumbered-link-hop
|   |   |   |--ro link-tp-id-uri? nt:tp-id
|   |   |   |--ro link-tp-id?    te-tp-id
|   |   |   |--ro node-id-uri?   nw:node-id
|   |   |   |--ro node-id?       te-node-id
|   |   |   |--ro hop-type?     te-hop-type
|   |   |   |--ro direction?    te-link-direction
|   |   |--:(as-number)
|   |   |   |--ro as-number-hop
|   |   |   |--ro as-number     inet:as-number
|   |   |   |--ro hop-type?     te-hop-type
|   |   |--:(label)
|   |   |   |--ro label-hop
|   |   |   |--ro te-label
|   |   |
|   |   |   ...
|   |--ro pcc-original?   -> /pcep/entity/peers/peer/addr
|   |--ro req-id?         uint32
|   |--ro retrieved?      boolean
|   |--ro pcc-retrieved?  -> /pcep/entity/peers/peer/addr
|   |--ro creation-time?  yang:timestamp
|   |--ro discard-time?   uint32
|   |--ro reuse-time?     uint32
+--rw peers
  +--rw peer* [addr]
    +--rw addr             inet:ip-address-no-zone
    +--rw role             role
    +--rw description?     string
    +--rw domains
      +--rw domain* [type info]
        +--rw type         identityref
        +--rw info         domain
    +--rw capabilities
      +--rw capability?    bits
      +--rw pce-initiated? boolean {pce-initiated}?
      +--rw include-db-ver? boolean
      |   {stateful, sync-opt}?
      +--rw trigger-resync? boolean
      |   {stateful, sync-opt}?
      +--rw trigger-initial-sync? boolean

```

```

| |         {stateful, sync-opt}?
| +--rw incremental-sync?         boolean
| |         {stateful, sync-opt}?
| +--rw sr {sr}?
| |   +--rw enabled?             boolean
| |   +--rw msd-limit?          boolean
| |   +--rw nai?                 boolean
| +--rw stateful-gmpls {stateful, gmpls}?
| |   +--rw enabled?            boolean
| +--rw inter-layer?             boolean {inter-layer}?
| +--rw h-pce {h-pce}?
|   +--rw enabled?              boolean
|   +--rw stateful?             boolean {stateful}?
|   +--rw role?                 hpce-role
+--rw msd?                       uint8 {sr}?
+--rw pce-info
| +--rw scope
| |   +--rw path-scope?          bits
| |   +--rw intra-area-pref?    uint8
| |   +--rw inter-area-pref?    uint8
| |   +--rw inter-as-pref?      uint8
| |   +--rw inter-layer-pref?   uint8
| +--rw neighbour-domains
|   +--rw domain* [type info]
|     +--rw type                identityref
|     +--rw info                domain
+--rw delegation-pref?           uint8 {stateful}?
+--rw auth
| +--rw (auth-type-selection)?
|   +--:(auth-key-chain)
|   |   +--rw key-chain?
|   |   |   key-chain:key-chain-ref
|   +--:(auth-key)
|   |   +--rw crypto-algorithm          identityref
|   |   +--rw (key-string-style)?
|   |   |   +--:(keystring)
|   |   |   |   +--rw keystring?          string
|   |   |   |   +--:(hexadecimal) {key-chain:hex-key-string}?
|   |   |   |   +--rw hexadecimal-string? yang:hex-string
|   +--:(auth-tls) {tls}?
|   |   +--rw (role)?
|   |   |   +--:(server)
|   |   |   |   +--rw tls-server
|   |   |   |   ...
|   |   +--:(client)
|   |   |   +--rw tls-client
|   |   |   ...
+--ro discontinuity-time?        yang:timestamp
+--ro initiate-session?         boolean

```

```

+--ro session-exists?          boolean
+--ro session-up-time?         yang:timestamp
+--ro session-fail-time?       yang:timestamp
+--ro session-fail-up-time?    yang:timestamp
+--ro sessions
  +--ro session* [initiator]
    +--ro initiator              initiator
    +--ro role?
      |      -> /pcep/entity/role
    +--ro state-last-change?     yang:timestamp
    +--ro state?                  sess-state
    +--ro session-creation?       yang:timestamp
    +--ro connect-retry?          yang:counter32
    +--ro local-id?               uint8
    +--ro remote-id?              uint8
    +--ro keepalive-timer?        uint8
    +--ro peer-keepalive-timer?   uint8
    +--ro dead-timer?             uint8
    +--ro peer-dead-timer?        uint8
    +--ro ka-hold-time-rem?       uint8
    +--ro overloaded?             boolean
    +--ro overloaded-timestamp?   yang:timestamp
    +--ro overload-time?          uint32
    +--ro peer-overloaded?        boolean
    +--ro peer-overloaded-timestamp? yang:timestamp
    +--ro peer-overload-time?     uint32
    +--ro lspdb-sync?             sync-state
      |      {stateful}?
    +--ro recv-db-ver?            uint64
      |      {stateful, sync-opt}?
    +--ro of-list {objective-function}?
      | +--ro objective-function* [of]
      | +--ro of      identityref
    +--ro pst-list
      | +--ro path-setup-type* [pst]
      | +--ro pst      identityref
    +--ro assoc-type-list {association}?
      | +--ro assoc-type* [at]
      | +--ro at      identityref
    +--ro speaker-entity-id?      string
      {sync-opt}?

```

rpcs:

```

+---x trigger-resync {stateful, sync-opt}?
  +---w input
    +---w pcc?   -> /pcep/entity/peers/peer/addr

```

notifications:

```

+---n pcep-session-up

```



```

| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |       -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro state-last-change? yang:timestamp
| +--ro state?              sess-state
+---n pcep-session-down
| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator? initiator
| +--ro state-last-change? yang:timestamp
| +--ro state?              sess-state
+---n pcep-session-local-overload
| +--ro peer-addr?          -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |       -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro overloaded?         boolean
| +--ro overloaded-timestamp? yang:timestamp
| +--ro overload-time?      uint32
+---n pcep-session-local-overload-clear
| +--ro peer-addr?
| |       -> /pcep/entity/peers/peer/addr
| +--ro overloaded?         boolean
| +--ro overloaded-clear-timestamp? yang:timestamp
+---n pcep-session-peer-overload
| +--ro peer-addr?
| |       -> /pcep/entity/peers/peer/addr
| +--ro session-initiator?
| |       -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro peer-overloaded?    boolean
| +--ro peer-overloaded-timestamp? yang:timestamp
| +--ro peer-overload-time? uint32
+---n pcep-session-peer-overload-clear
  +--ro peer-addr?
  |       -> /pcep/entity/peers/peer/addr
  +--ro peer-overloaded?    boolean
  +--ro peer-overloaded-clear-timestamp? yang:timestamp

```

Appendix B. Example

The example below provide an overview of PCEP peer session informations and LSP-DB in the Yang Module.


```
    "plsp-id": 4,
    "pcc-id": "192.0.2.2",
    "source": "192.0.2.2",
    "destination": "192.0.2.5",
    "tunnel-id": 17,
    "lsp-id": 4,
    "extended-tunnel-id": 0,
    "operational-status": "oper-status-up",
    "delegated": {
      "enabled": true
    },
    "symbolic-path-name": "iewauhiewauh"
  }
},
],
"peers": [
  {
    "peer": {
      "addr": "192.0.2.1",
      "role": "pcc",
      "capabilities": {
        "capability": "active passive"
      },
      "sessions": [
        {
          "session": {
            "initiator": "remote",
            "role": "pcc"
          }
        }
      ]
    }
  },
  {
    "peer": {
      "addr": "192.0.2.2",
      "role": "pcc",
      "capabilities": {
        "capability": "active passive"
      },
      "sessions": [
        {
          "session": {
            "initiator": "remote",
            "role": "pcc"
          }
        }
      ]
    }
  }
]
```

```

    }
  ]
}
}
}

```

Similarly a PCEP session with IPv6 address between PCE (2001:DB8::3) and a PCC (2001:DB8::4) could also be setup.

Appendix C. Design Objectives

This section describes some of the design objectives for the model:

- *In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.
- *The data model should be suitable for new implementations to use as is.
- *Mapping to the PCEP MIB Module [[RFC7420](#)] should be clear.
- *The data model should allow for static configurations of peers.
- *The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors. This could be in a separate model which augments the base data model.
- *It should be fairly straightforward to augment the base data model for advanced PCE features.

Appendix D. Relationship with PCEP MIB

If a node implements the PCEP-MIB [[RFC7420](#)], data nodes from the YANG module can be mapped to table entries in the PCEP-MIB.

YANG Data Nodes	PCEP MIB Objects
/pcep/entity	PcePcepEntityEntry
/pcep/entity/peers/peer	pcePcepPeerEntry
/pcep/entity/peers/peer/sessions/session	pcePcepSessEntry

Table 3: High Level Relationship with PCEP MIB

YANG Data Nodes	PCEP MIB Objects
-	pcePcepEntityIndex
admin-status	pcePcepEntityAdminStatus

YANG Data Nodes	PCEP MIB Objects
oper-status	pcePcepEntityOperStatus
addr	pcePcepEntityAddrType, pcePcepEntityAddr
connect-timer	pcePcepEntityConnectTimer
connect-max-retry	pcePcepEntityConnectMaxRetry
init-back-off-timer	pcePcepEntityInitBackoffTimer
max-back-off-timer	pcePcepEntityMaxBackoffTimer
open-wait-timer	pcePcepEntityOpenWaitTimer
keep-wait-timer	pcePcepEntityKeepWaitTimer
keepalive-timer	pcePcepEntityKeepAliveTimer
dead-timer	pcePcepEntityDeadTimer
allow-negotiation	pcePcepEntityAllowNegotiation
max-keepalive-timer	pcePcepEntityMaxKeepAliveTimer
max-dead-timer	pcePcepEntityMaxDeadTimer
min-keepalive-timer	pcePcepEntityMinKeepAliveTimer
min-dead-timer	pcePcepEntityMinDeadTimer
sync-timer	pcePcepEntitySyncTimer
request-timer	pcePcepEntityRequestTimer
max-sessions	pcePcepEntityMaxSessions
max-unknown-reqs	pcePcepEntityMaxUnknownReqs
max-unknown-msgs	pcePcepEntityMaxUnknownMsgs

Table 4: Relationship with PCEP MIB for Entity

YANG Data Nodes in /pcep/entity/peers/peer	PCEP MIB Objects
addr	pcePcepPeerAddrType, pcePcepPeerAddr
role	pcePcepPeerRole
discontinuity-time	pcePcepPeerDiscontinuityTime
initiate-session	pcePcepPeerInitiateSession
session-exists	pcePcepPeerSessionExists
sess-setup-ok	pcePcepPeerNumSessSetupOK
sess-setup-fail	pcePcepPeerNumSessSetupFail
session-up-time	pcePcepPeerSessionUpTime
session-fail-time	pcePcepPeerSessionFailTime
session-fail-up-time	pcePcepPeerSessionFailUpTime
/stats/rsp-time-avg	pcePcepPeerAvgRspTime
/stats/rsp-time-lwm	pcePcepPeerLWMRspTime
/stats/rsp-time-hwm	pcePcepPeerHWMRspTime
/stats/pcreq-sent	pcePcepPeerNumPCReqSent
/stats/pcreq-rcvd	pcePcepPeerNumPCReqRcvd
/stats/pcrep-sent	pcePcepPeerNumPCRepSent
/stats/pcrep-rcvd	pcePcepPeerNumPCRepRcvd
/stats/pcerr-sent	pcePcepPeerNumPCErrSent
/stats/pcerr-rcvd	pcePcepPeerNumPCErrRcvd
/stats/pcntf-sent	pcePcepPeerNumPCntfSent
/stats/pcntf-rcvd	pcePcepPeerNumPCntfRcvd

YANG Data Nodes in /pcep/entity/peers/peer	PCEP MIB Objects
/stats/keepalive-sent	pcePcepPeerNumKeepaliveSent
/stats/keepalive-rcvd	pcePcepPeerNumKeepaliveRcvd
/stats/unknown-rcvd	pcePcepPeerNumUnknownRcvd
/stats/corrupt-rcvd	pcePcepPeerNumCorruptRcvd
/stats/req-sent	pcePcepPeerNumReqSent
/stats/svec/svec-sent	pcePcepPeerNumSvecSent
/stats/svec/svec-req-sent	pcePcepPeerNumSvecReqSent
/stats/req-sent-pend-rep	pcePcepPeerNumReqSentPendRep
/stats/req-sent-ero-rcvd	pcePcepPeerNumReqSentEroRcvd
/stats/req-sent-nopath-rcvd	pcePcepPeerNumReqSentNoPathRcvd
/stats/req-sent-cancel-rcvd	pcePcepPeerNumReqSentCancelRcvd
/stats/req-sent-error-rcvd	pcePcepPeerNumReqSentErrorRcvd
/stats/req-sent-timeout	pcePcepPeerNumReqSentTimeout
/stats/req-sent-cancel-sent	pcePcepPeerNumReqSentCancelSent
/stats/req-sent-closed	pcePcepPeerNumReqSentClosed
/stats/req-rcvd	pcePcepPeerNumReqRcvd
/stats/svec/svec-rcvd	pcePcepPeerNumSvecRcvd
/stats/svec/svec-req-rcvd	pcePcepPeerNumSvecReqRcvd
/stats/req-rcvd-pend-rep	pcePcepPeerNumReqRcvdPendRep
/stats/req-rcvd-ero-sent	pcePcepPeerNumReqRcvdEroSent
/stats/req-rcvd-nopath-sent	pcePcepPeerNumReqRcvdNoPathSent
/stats/req-rcvd-cancel-sent	pcePcepPeerNumReqRcvdCancelSent
/stats/req-rcvd-error-sent	pcePcepPeerNumReqRcvdErrorSent
/stats/req-rcvd-cancel-rcvd	pcePcepPeerNumReqRcvdCancelRcvd
/stats/req-rcvd-closed	pcePcepPeerNumReqRcvdClosed
/stats/rep-rcvd-unknown	pcePcepPeerNumRepRcvdUnknown
/stats/req-rcvd-unknown	pcePcepPeerNumReqRcvdUnknown

Table 5: Relationship with PCEP MIB for Peer

YANG Data Nodes in /pcep/entity/peers/peer/sessions/session	PCEP MIB Objects
initiator	pcePcepSessInitiator
state-last-change	pcePcepSessStateLastChange
state	pcePcepSessState
connect-retry	pcePcepSessConnectRetry
local-id	pcePcepSessLocalID
remote-id	pcePcepSessRemoteID
keepalive-timer	pcePcepSessKeepaliveTimer
peer-keepalive-timer	pcePcepSessPeerKeepaliveTimer
dead-timer	pcePcepSessDeadTimer
peer-dead-timer	pcePcepSessPeerDeadTimer
ka-hold-time-rem	pcePcepSessKAHoldTimeRem
overloaded	pcePcepSessOverloaded
overloaded-timestamp	pcePcepSessOverloadTime
peer-overloaded	pcePcepSessPeerOverloaded

YANG Data Nodes in /pcep/entity/peers/peer/sessions/session	PCEP MIB Objects
peer-overloaded-timestamp	pcePcepSessPeerOverloadTime
/stats/discontinuity-time	pcePcepSessDiscontinuityTime
/stats/rsp-time-avg	pcePcepSessAvgRspTime
/stats/rsp-time-lwm	pcePcepSessLWMRspTime
/stats/rsp-time-hwm	pcePcepSessHWMRspTime
/stats/pcreq-sent	pcePcepSessNumPCReqSent
/stats/pcreq-rcvd	pcePcepSessNumPCReqRcvd
/stats/pcrep-sent	pcePcepSessNumPCRepSent
/stats/pcrep-rcvd	pcePcepSessNumPCRepRcvd
/stats/pcerr-sent	pcePcepSessNumPCErrSent
/stats/pcerr-rcvd	pcePcepSessNumPCErrRcvd
/stats/pcntf-sent	pcePcepSessNumPCntfSent
/stats/pcntf-rcvd	pcePcepSessNumPCntfRcvd
/stats/keepalive-sent	pcePcepSessNumKeepaliveSent
/stats/keepalive-rcvd	pcePcepSessNumKeepaliveRcvd
/stats/unknown-rcvd	pcePcepSessNumUnknownRcvd
/stats/corrupt-rcvd	pcePcepSessNumCorruptRcvd
/stats/req-sent	pcePcepSessNumReqSent
/stats/svec/svec-sent	pcePcepSessNumSvecSent
/stats/svec/svec-req-sent	pcePcepSessNumSvecReqSent
/stats/req-sent-pend-rep	pcePcepSessNumReqSentPendRep
/stats/req-sent-ero-rcvd	pcePcepSessNumReqSentEroRcvd
/stats/req-sent-nopath-rcvd	pcePcepSessNumReqSentNoPathRcvd
/stats/req-sent-cancel-rcvd	pcePcepSessNumReqSentCancelRcvd
/stats/req-sent-error-rcvd	pcePcepSessNumReqSentErrorRcvd
/stats/req-sent-timeout	pcePcepSessNumReqSentTimeout
/stats/req-sent-cancel-sent	pcePcepSessNumReqSentCancelSent
/stats/req-rcvd	pcePcepSessNumReqRcvd
/stats/svec/svec-rcvd	pcePcepSessNumSvecRcvd
/stats/svec/svec-req-rcvd	pcePcepSessNumSvecReqRcvd
/stats/req-rcvd-pend-rep	pcePcepSessNumReqRcvdPendRep
/stats/req-rcvd-ero-sent	pcePcepSessNumReqRcvdEroSent
/stats/req-rcvd-nopath-sent	pcePcepSessNumReqRcvdNoPathSent
/stats/req-rcvd-cancel-sent	pcePcepSessNumReqRcvdCancelSent
/stats/req-rcvd-error-sent	pcePcepSessNumReqRcvdErrorSent
/stats/req-rcvd-cancel-rcvd	pcePcepSessNumReqRcvdCancelRcvd
/stats/rep-rcvd-unknown	pcePcepSessNumRepRcvdUnknown
/stats/req-rcvd-unknown	pcePcepSessNumReqRcvdUnknown

Table 6: Relationship with PCEP MIB for Session

YANG notifications	PCEP MIB NOTIFICATIONS
pcep-session-up	pcePcepSessUp
pcep-session-down	pcePcepSessDown
pcep-session-local-overload	pcePcepSessLocalOverload

YANG notifications	PCEP MIB NOTIFICATIONS
pcep-session-local-overload-clear	pcePcepSessLocalOverloadClear
pcep-session-peer-overload	pcePcepSessPeerOverload
pcep-session-peer-overload-clear	pcePcepSessPeerOverloadClear

Table 7: Relationship with PCEP MIB Notification

Appendix E. Contributor Addresses

Rohit Pobbathi
Nokia Networks
India

E-Mail: rohit.pobbathi_ashok@nokia.com

Vinod KumarS
India

E-Mail: vinods.kumar@gmail.com

Zafar Ali
Cisco Systems
Canada

E-Mail: zali@cisco.com

Xufeng Liu
Alef Edge

E-Mail: xufeng.liu.ietf@gmail.com

Young Lee
Samsung

E-Mail: younglee.tx@gmail.com

Udayasree Palle

E-Mail: udayasreereddy@gmail.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

E-Mail: zhang.xian@huawei.com

Avantika
ECI Telecom
India

E-Mail: avantika.srm@gmail.com

Shashikanth
India

E-Mail: shashivh@gmail.com

Authors' Addresses

Dhruv Dhody (editor)
Huawei
India

Email: dhruv.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks
India

Email: vbeeram@juniper.net

Jonathan Hardwick
Microsoft
United Kingdom

Email: jonhardwick@microsoft.com

Jeff Tantsura
Nvidia
United States of America

Email: jefftant.ietf@gmail.com