Stream:	Internet Engineering Task Force (IETF)			
RFC:	9826			
Category:	Standards Track			
Published:	July 2025			
ISSN:	2070-1721			
Authors:	D. Dhody, Ed. *Huawei*	V. Beeram *Juniper Networks*	J. Hardwick	J. Tantsura *Nvidia*

RFC 9826
A YANG Data Model for the Path Computation Element Communication Protocol (PCEP)

Abstract

This document defines a YANG data model for the management of the Path Computation Element Communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9826.

Table of Contents

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; it 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 Label Switched Paths (LSPs). [RFC8664] and [RFC9603] extend PCEP to support Segment Routing in MPLS and IPv6, respectively.

This document defines a YANG 1.1 [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. Further, this document also includes the PCEP statistics YANG module "ietf-pcep-stats", which provides statistics, counters, and telemetry data.

The YANG modules in this document conform to the 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:

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

This document uses the following terms defined in [RFC7420]:

PCEP entity: a local PCEP speaker

PCEP peer: a remote PCEP speaker

PCEP speaker: term used when it is not necessary to distinguish between local and remote.

Further, this document uses the following terms defined in [RFC8231]:

- Stateful PCE, Passive Stateful PCE, Active Stateful PCE
- Delegation, Revocation, Redelegation
- Path Computation LSP State Report (PCRpt) message
- Path Computation LSP Update Request (PCUpd) message
- PLSP-ID (a PCEP-specific identifier for the LSP)
- Stateful PCE Request Parameter (SRP)

This document also uses the following terms defined in [RFC8281]:

- PCE-initiated LSP
- Path Computation LSP Initiate (PCInitiate) message

Last, this document uses the following terms, which are defined in the RFCs indicated below:

- Path Setup Type (PST) [RFC8408]
- Segment Routing (SR) [RFC8664]
- Objective Function (OF) [RFC5541]
- Association [RFC8697]
- Configuration data [RFC6241]
- State data [RFC6241]

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, the 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	[RFC9645]
tlsc	ietf-tls-client	[RFC9645]
ospf	ietf-ospf	[RFC9129]
isis	ietf-isis	[RFC9130]

Table 1: Prefixes and Corresponding YANG Modules

3.3. References in the Model

The 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]

Documents	Reference
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]
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]
Common YANG Data Types for Traffic Engineering	[RFC8776]
YANG Groupings for TLS Clients and TLS Servers	[RFC9645]
Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing	[RFC8664]
Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)	[RFC8697]
YANG Data Model for the OSPF Protocol	[RFC9129]
YANG Data Model for the IS-IS Protocol	[RFC9130]

Documents	Reference
Path Computation Element Communication Protocol (PCEP) Extensions for GMPLS	[RFC8779]
"ASSOCIATION Type Field" registry (in the "Path Computation Element Protocol (PCEP) Numbers" registry group)	[IANA-PCEP]
"Path Computation Element (PCE) Capability Flags" registry (in the "Interior Gateway Protocol (IGP) Parameters" registry group)	[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 Label Switched Paths and Virtual Networks	[RFC9358]
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 PCEP, which are listed below and further detailed in the subsequent subsections.

- The local PCEP entity
- The PCEP peer
- The PCEP session
- Notifications
- RPC

```
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 domain]
        |     +--...
        +--rw capabilities
        |  +--...
        +--rw auth
        |  +--...
        +--rw pce-info
        |  +--rw scope
        |  |  +--...
        |  +--rw neighbor-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 Database (LSP-DB) information.

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

```
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 domain]
        |     +--rw type      identityref
```

```
|     +--rw domain    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-mpls {sr-mpls}?
|  |  +--rw enabled?        boolean
|  |  +--rw no-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
+--ro msd?                           uint8 {sr-mpls}?
+--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 neighbor-domains
|  |  +--rw domain* [type domain]
|  |     +--rw type      identityref
|  |     +--rw domain    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     leafref
        |  |     +--ro lsp-id     leafref
        |  +--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               leafref
        |        +--ro source           leafref
        |        +--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 pcc-requester?   -> /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 for that peer. The statistics contained are an aggregate of the statistics for all successive sessions with 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 for the corresponding peer.

The data model for PCEP peers 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 a PCE address in the network via the 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 domain]
              |     +--rw type      identityref
              |     +--rw domain    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-mpls {sr-mpls}?
|  |  +--rw enabled?        boolean
|  |  +--rw no-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
+--ro msd?                       uint8 {sr-mpls}?
+--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 neighbor-domains
|     +--rw domain* [type domain]
|        +--rw type      identityref
|        +--rw domain    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 the 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 the 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?                        -> ../../../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 data model defines a list of notifications to inform clients 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?   leafref
  |  +--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?      leafref
  |  +--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?           leafref
  |  +--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 data model defines an RPC to trigger state resynchronization at the PCE for a 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 "ietf-pcep-stats" module augments the "ietf-pcep" module to include statistics at the PCEP peer and session level. It also includes an RPC to reset all PCEP statistics across all peers and sessions through mechanisms such as walking a list of pointers to those peer and session statistics.

If this mechanism is not supported, implementations must reset PCEP statistics by invoking the action 'reset-statistics' for each peer and session.

```
module: ietf-pcep-stats

  augment /pcep:pcep/pcep:entity/pcep:peers/pcep:peer:
    +--ro stats
       +--ro discontinuity-time?   yang:timestamp
       +--ro pce
       |  +--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 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 rep-rcvd-unknown?       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 pcc
       |  +--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 req-rcvd-unknown?       yang:counter32
       +--ro svec {pcep:svec}?
       |  +--ro pce
       |  |  +--ro svec-sent?       yang:counter32
       |  |  +--ro svec-req-sent?   yang:counter32
       |  +--ro pcc
       |     +--ro svec-rcvd?       yang:counter32
       |     +--ro svec-req-rcvd?   yang:counter32
       +--ro stateful {pcep:stateful}?
       |  +--ro pce
       |  |  +--ro pcrpt-sent?             yang:counter32
       |  |  +--ro pcupd-rcvd?             yang:counter32
       |  |  +--ro rpt-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 pcc
        |  |  +--ro pcrpt-rcvd?            yang:counter32
        |  |  +--ro pcupd-sent?            yang:counter32
        |  |  +--ro rpt-rcvd?              yang:counter32
        |  |  +--ro rpt-rcvd-error-sent?   yang:counter32
        |  |  +--ro upd-sent?              yang:counter32
        |  +--ro initiation {pcep:pce-initiated}?
        |     +--ro pcc
        |     |  +--ro pcinitiate-sent?   yang:counter32
        |     |  +--ro initiate-sent?     yang:counter32
        |     +--ro pce
        |        +--ro pcinitiate-rcvd?            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
        +---x reset-statistics
        |  +---w input
        |  |  +---w reset-at?   yang:date-and-time
        |  +--ro output
        |     +--ro reset-finished-at?   yang:date-and-time
        +--ro sess-setup-ok?        yang:counter32
        +--ro sess-setup-fail?      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 pce
        |  +--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 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 rep-rcvd-unknown?       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 pcc
      |  +--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 req-rcvd-unknown?        yang:counter32
      +--ro svec {pcep:svec}?
      |  +--ro pce
      |  |  +--ro svec-sent?        yang:counter32
      |  |  +--ro svec-req-sent?    yang:counter32
      |  +--ro pcc
      |     +--ro svec-rcvd?        yang:counter32
      |     +--ro svec-req-rcvd?    yang:counter32
      +--ro stateful {pcep:stateful}?
      |  +--ro pce
      |  |  +--ro pcrpt-sent?              yang:counter32
      |  |  +--ro pcupd-rcvd?              yang:counter32
      |  |  +--ro rpt-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 pcc
      |  |  +--ro pcrpt-rcvd?             yang:counter32
      |  |  +--ro pcupd-sent?             yang:counter32
      |  |  +--ro rpt-rcvd?               yang:counter32
      |  |  +--ro rpt-rcvd-error-sent?    yang:counter32
      |  |  +--ro upd-sent?               yang:counter32
      |  +--ro initiation {pcep:pce-initiated}?
      |     +--ro pcc
      |     |  +--ro pcinitiate-sent?    yang:counter32
      |     |  +--ro initiate-sent?      yang:counter32
      |     +--ro pce
      |        +--ro pcinitiate-rcvd?             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
      +---x reset-statistics
         +---w input
         |  +---w reset-at?   yang:date-and-time
         +--ro output
            +--ro reset-finished-at?   yang:date-and-time

rpcs:
  +---x reset-pcep-statistics-all {reset-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 such as:

- Capability and scope
- Domain information (local/neighbor)
- Path-key
- Objective Function (OF)
- Global Concurrent Optimization (GCO)
- Point-to-Multipoint (P2MP)
- GMPLS
- Inter-layer
- Stateful PCE
- Segment Routing (SR) for the MPLS data plane
- Authentication including PCEPS (TLS)
- Hierarchical PCE (H-PCE)

Segment Routing in the IPv6 data plane is out of the scope of this document. Refer to [YANG-PCEP-SR] for the PCEP-SRv6 YANG module.

6.1. Stateful PCE's LSP-DB

In the operational datastore of stateful PCE, the list of LSP state is 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 [YANG-TE]. A reference to the LSP state in the TE model is maintained.

7. Other Considerations

7.1. PCEP over TLS (PCEPS)

[RFC8253] describes the use of TLS 1.2 [RFC5246] or later in PCEP. Further, [PCEPS-UPDATE] specifies 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 PCC acting as the TLS client opens the TLS connection, and the PCE acting as the TLS server listens for incoming connections as per the TLS specifications ([RFC8446] and [RFC5246]). [RFC8253] specifies the StartTLS procedure in PCEP that initiates the TLS connection before exchanging PCEP messages; thus, the identity verification is completed before the PCEP session is established.

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 is true for TLS as well).

The YANG module uses the TLS grouping in [RFC9645]. Note that any TLS version can be configured, but [RFC9645] 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 the feature "tls12" even though it is marked with status as "deprecated".

8. PCEP YANG Modules

8.1. ietf-pcep Module

```
<CODE BEGINS> file "ietf-pcep@2025-07-24.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 Access Control Model";
  }
  import ietf-tls-server {
```

```
    prefix tlss;
    reference
      "RFC 9645: YANG Groupings for TLS Clients and TLS Servers";
  }
  import ietf-tls-client {
    prefix tlsc;
    reference
      "RFC 9645: YANG Groupings for TLS Clients and TLS Servers";
  }
  import ietf-ospf {
    prefix ospf;
    reference
      "RFC 9129: YANG Data Model for the OSPF Protocol";
  }
  import ietf-isis {
    prefix isis;
    reference
      "RFC 9130: YANG Data Model for the 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 the Path Computation Element
     Communication Protocol (PCEP).

     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.

     Copyright (c) 2025 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 9826; see the
     RFC itself for full legal notices.";

  revision 2025-07-24 {
    description
      "Initial revision.";
    reference
      "RFC 9826:  A YANG Data Model for the Path Computation
       Element Communication 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 (PCC).";
    }
    enum pce {
      value 2;
      description
        "The role of a Path Computation Element (PCE).";
    }
    enum pcc-and-pce {
      value 3;
      description
        "The role of both Path Computation Client (PCC) and
         Path Computation Element (PCE).";
    }
  }
  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
            "Signaled.";
        }
```

```
      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 signaled.";
      }
    }
    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: Path Computation Element Communication Protocol
       (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-mpls {
    description
      "Support Segment Routing (SR) for MPLS in PCEP.";
    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 brought 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 virtual-network {
    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 domain {
      type domain;
      description
        "The domain information.";
    }
  }

  grouping domain-info {
    description
```

```
       "This grouping specifies all information that
        may be 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 domain";
         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-AO).";
              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;
          default "false";
          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;
          default "true";
          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;
          default "true";
          description
            "Support PCE-triggered resynchronization.";
          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;
          default "true";
          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;
        default "true";
        description
          "Support incremental (delta) synchronization.";
        reference
          "RFC 8232: Optimizations of Label Switched Path State
           Synchronization Procedures for a Stateful PCE";
      }
      container sr-mpls {
        if-feature "sr-mpls";
        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;
          default "false";
          description
            "Set to true if SR-MPLS is enabled.";
        }
        leaf no-msd-limit {
          type boolean;
          default "false";
          description
            "True indicates no limit on Maximum SID Depth (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 GMPLS";
        leaf enabled {
          type boolean;
          default "false";
          description
```

```
            "Set to true if stateful GMPLS is enabled.";
        }
      }
      leaf inter-layer {
        if-feature "inter-layer";
        type boolean;
        default "false";
        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;
          default "false";
          description
            "Set to true if H-PCE is enabled.";
        }
        leaf stateful {
          if-feature "stateful";
          type boolean;
          default "false";
          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-mpls";
      type uint8;
      config false;
      description
```

```
        "Maximum SID Depth (MSD) for SR-MPLS (i.e., the label stack
         depth that a PCC 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
       that 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 that 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 neighbor-domains {
        description
          "The list of neighbor PCE domains
           toward which a PCE can compute
           paths.";
        list domain {
          key "type domain";
          description
            "The neighbor domain.";
          uses domain {
            description
              "The PCE neighbor 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.";
    uses notification-instance-hdr;
    leaf session-initiator {
      type leafref {
        path "/pcep/entity/peers/peer[addr=current()/../peer-addr]/"
           + "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 keystring {
              leaf keystring {
                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-list addr {
        type inet:ip-address-no-zone;
        min-elements 1;
        ordered-by user;
        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 based on the address family.  It is
           possible for the PCEP entity to operate as both 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;
        must '(. != "unknown")' {
          error-message "The PCEP entity role cannot be unknown";
        }
        mandatory true;
        description
          "The role that this entity can play.
           Takes one of the following values:
           - 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 PECP 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 domain-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;
            default "false";
            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), same as local PCE entity IP address.";
          }
        }
      }
      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";
        mandatory true;
        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";
        mandatory true;
        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 thus 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 the negotiation
           of session parameters.";
      }
      leaf max-keepalive-timer {
        type uint8;
        units "seconds";
        mandatory true;
        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";
        mandatory true;
        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";
        mandatory true;
        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";
        mandatory true;
        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.  If after the expiration of
             the SyncTimer all the path computation requests
             have not been received, a protocol error is
             triggered, and the PCE must cancel the whole set
             of path computation requests.
             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";
          mandatory true;
          description
            "The maximum time that the PCEP entity will wait
             for a response to a PCReq message.";
        }
        leaf max-sessions {
          type uint32;
          mandatory true;
          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";
          mandatory true;
          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 behaviors.  The max value
             represents infinity.";
          reference
            "RFC 8231: Path Computation Element Communication
             Protocol (PCEP) Extensions for Stateful PCE";
        }
        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";
          must '(. < ../state-timeout)' {
            error-message "The Redelegation Timeout must be "
                        + "less than the State Timeout";
          }
          mandatory true;
          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.";
          reference
            "RFC 8231: Path Computation Element Communication
             Protocol (PCEP) Extensions for Stateful 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 fields 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[plsp-id=current()/"
                   + "../plsp-id]/pcc-id";
              }
              description
                "Reference to PCC-ID in LSP-DB.";
            }
            leaf lsp-id {
              type leafref {
                path "/pcep/entity/lsp-db/"
                   + "lsp[plsp-id=current()/../plsp-id]"
                   + "[pcc-id=current()/../pcc-id]/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
               reserved.";
          }
          leaf pcc-id {
            type inet:ip-address-no-zone;
            description
              "The local IP 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 {
              when '../enabled' {
                description
                  "The LSP must be delegated.";
              }
              type leafref {
                path "/pcep/entity/peers/peer/addr";
              }
              description
                "At the PCC, the reference to the PCEP peer to
                 which LSP is delegated; at the PCE, the
                 reference to the PCEP peer that delegated this
                 LSP.";
            }
            leaf srp-id {
              type uint32 {
                range "1..4294967294";
              }
              description
                "The last SRP-ID-number associated with this
                 LSP.  The values 0x00000000 and 0xFFFFFFFF
                 are reserved.";
            }
          }
          container initiation {
            if-feature "pce-initiated";
            description
              "The parameters related to PCE initiation.";
            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 {
              when '../enabled' {
                description
                  "The LSP must be PCE-initiated.";
              }
              type leafref {
                path "/pcep/entity/peers/peer/addr";
              }
              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[type=current()/"
                   + "../type]/id";
              }
              description
                "PCEP Association ID.";
            }
            leaf source {
              type leafref {
                path "/pcep/entity/lsp-db/"
                   + "association-list[type=current()/../type]"
                   + "[id=current()/../id]/source";
              }
              description
                "PCEP Association Source.";
            }
            leaf global-source {
              type leafref {
                path "/pcep/entity/lsp-db/"
```

```
                  + "association-list[type=current()/../type]"
                  + "[id=current()/../id]"
                  + "[source=current()/../source]"
                  + "/global-source";
            }
            description
              "PCEP Global Association Source.";
          }
          leaf extended-id {
            type leafref {
              path "/pcep/entity/lsp-db/"
                 + "association-list[type=current()/../type]"
                 + "[id=current()/../id]"
                 + "[source=current()/../source]"
                 + "[global-source=current()/../global-source]"
                 + "/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 (EROs).";
            leaf index {
              type uint32;
              description
                "ERO subobject index.";
            }
            uses te-types:explicit-route-hop;
          }
        }
        leaf pcc-requester {
          type leafref {
            path "/pcep/entity/peers/peer/addr";
          }
          description
            "Reference to PCC peer address that
             issued the original request that led
             to the creation of the path-key.";
        }
        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 {
          when '../retrieved' {
            description
              "The path-key should be retrieved.";
          }
          type leafref {
            path "/pcep/entity/peers/peer/addr";
          }
          description
            "Reference to PCC peer address that
             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 same time.";
          }
          leaf description {
            type string;
            description
              "Description of the PCEP peer
               configured by the user.";
          }
          uses domain-info {
            description
              "PCE peer information.";
          }
          container pce-info {
            uses pce-info {
              description
                "Using the PCE peer information grouping.";
            }
            description
              "The PCE peer information.";
          }
```

```
            leaf delegation-pref {
              if-feature "stateful";
              type uint8 {
                range "0..7";
              }
              mandatory true;
              description
                "The PCE peer delegation preference, where
                 7 reflects the highest 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 waits 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 "../../../role";
                }
                description
                  "The 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 the 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;
                description
                  "The value of the PCEP session ID used by the
                   peer in its Open message for this session.

                   If the state is TCPPending or OpenWait, then
                   this leaf is not used and MUST be set to
                   zero.";
                reference
                  "RFC 5440: Path Computation Element (PCE)
                   Communication Protocol (PCEP)";
              }
              leaf keepalive-timer {
                type uint8;
                units "seconds";
                description
                  "The agreed maximum interval at which the local
                   PCEP entity transmits PCEP messages on this PCEP
                   session.  Zero means that the local PCEP entity
                   never sends Keepalives on this session.

                   This field is used if and only if the state
                   is session-up.  Otherwise, it is not used and
                   MUST be set to zero.";
                reference
                  "RFC 5440: Path Computation Element (PCE)
                   Communication Protocol (PCEP)";
              }
              leaf peer-keepalive-timer {
                type uint8;
```

```
                units "seconds";
                description
                  "The agreed maximum interval at which the peer
                   transmits PCEP messages on this PCEP session.
                   Zero means that the peer never sends Keepalives
                   on this session.

                   This field is used if and only if state is
                   session-up.  Otherwise, it is not used and MUST
                   be set to zero.";
                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";
                description
                  "The peer's DeadTimer interval for this PCEP
                   session.

                   If the state is TCPPending or OpenWait, then
                   this leaf is not used and MUST be set to zero.";
                reference
                  "RFC 5440: Path Computation Element (PCE)
                   Communication Protocol (PCEP)";
              }
              leaf ka-hold-time-rem {
                type uint8;
                units "seconds";
                description
                  "The Keepalive hold time remaining for this
                   session.

                   If the state is TCPPending or OpenWait, then
                   this field is not used and MUST be set to
                   zero.";
              }
              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' {
                  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";
                description
                  "The interval of time that is remaining until the
                   local PCEP entity will cease to be overloaded on
                   this session.

                   This field is only used if overloaded is set to
                   true.  Otherwise, it is not used and MUST be set
                   to zero.";
                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' {
                  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";
                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.

                   This field is only used if peer-overloaded
                   is set to true.  Otherwise, it is not used
                   and MUST be 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-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-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-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";
        }
        mandatory true;
        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@2025-07-24.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 9826: A YANG Data Model for the Path Computation
       Element Communication 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) 2025 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 9826; see the
      RFC itself for full legal notices.";

  revision 2025-07-24 {
    description
      "Initial revision.";
    reference
      "RFC 9826:  A YANG Data Model for the Path Computation
       Element Communication Protocol (PCEP)";
  }

  /*
   * Features
   */

  feature reset-all {
    description
      "Support resetting of all PCEP statistics.";
  }

  /*
   * Groupings
   */

  grouping stats {
    description
      "This grouping defines statistics for PCEP.  It is used
       for both peer and current sessions.  Since this grouping
       includes a relative path, care needs to be taken while
       using it.";
    leaf discontinuity-time {
      type yang:timestamp;
      description
        "The timestamp value of the time when the
         statistics were last reset.";
    }
    container pce {
      when "../../pcep:role = 'pce'"
         + "or "
         + "../../pcep:role = 'pcc-and-pce'" {
        description
          "Valid for PCEP peer as PCE.";
      }
      leaf rsp-time-avg {
        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 {
        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 {
        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 {
        type yang:counter32;
        description
          "The number of PCReq messages sent.";
      }
      leaf pcreq-rcvd {
        type yang:counter32;
        description
          "The number of PCReq messages received.";
      }
      leaf pcrep-sent {
        type yang:counter32;
        description
          "The number of PCRep messages sent.";
      }
      leaf pcrep-rcvd {
        type yang:counter32;
        description
          "The number of PCRep messages received.";
      }
      leaf req-sent {
        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 {
        type yang:counter32;
        description
          "The number of requests that have been sent for
           which a response is still pending.";
      }
      leaf req-sent-ero-rcvd {
        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 {
          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 {
          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 {
          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 {
          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 {
          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 rep-rcvd-unknown {
          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.";
        }
        description
          "The stats related to PCE as peer.";
      }
      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.";
      }
      container pcc {
        when "../../pcep:role = 'pcc'"
           + "or "
           + "../../pcep:role = 'pcc-and-pce'" {
          description
            "Valid for PCEP peer as PCC.";
        }
        leaf req-rcvd {
          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 {
          type yang:counter32;
          description
```

```
            "The number of requests that have been received for
             which a response is still pending.";
        }
        leaf req-rcvd-ero-sent {
          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 {
          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 {
          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 {
          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 {
          type yang:counter32;
          description
            "The number of requests that were received from the
             peer and explicitly cancelled by the peer sending
             a PCNtf.";
        }
        leaf req-rcvd-unknown {
          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.";
        }
        description
          "The stats related to PCC as peer.";
      }
      container svec {
        if-feature "pcep:svec";
```

```
        description
          "If synchronized path computation is supported.";
        container pce {
          when "../../../pcep:role = 'pce'"
             + "or "
             + "../../../pcep:role = 'pcc-and-pce'" {
            description
              "Valid for PCEP peer as PCE.";
          }
          leaf svec-sent {
            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 {
            type yang:counter32;
            description
              "The number of requests sent that appeared in one
               or more SVEC objects.";
          }
          description
            "The SVEC stats related to PCE.";
        }
        container pcc {
          when "../../../pcep:role = 'pcc'"
             + "or "
             + "../../../pcep:role = 'pcc-and-pce'" {
            description
              "Valid for PCEP peer as PCC.";
          }
          leaf svec-rcvd {
            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 {
            type yang:counter32;
            description
              "The number of requests received that appeared
               in one or more SVEC objects.";
          }
          description
            "The SVEC stats related to PCC as peer.";
        }
      }
      container stateful {
        if-feature "pcep:stateful";
        description
          "Stateful PCE-related statistics.";
        container pce {
          when "../../../pcep:role = 'pce'"
             + "or "
             + "../../../pcep:role = 'pcc-and-pce'" {
            description
```

```
            "Valid for PCEP peer as PCE.";
        }
        leaf pcrpt-sent {
          type yang:counter32;
          description
            "The number of PCRpt messages sent.";
        }
        leaf pcupd-rcvd {
          type yang:counter32;
          description
            "The number of PCUpd messages received.";
        }
        leaf rpt-sent {
          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 upd-rcvd {
          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 {
          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 {
          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 {
          type yang:counter32;
          description
            "The number of updates to LSPs received that were
             responded by the local PCEP entity by sending a
             PCErr message.";
        }
        description
          "The stateful stats related to PCE as peer";
```

```
      }
      container pcc {
        when "../../../pcep:role = 'pcc'"
           + "or "
           + "../../../pcep:role = 'pcc-and-pce'" {
          description
            "Valid for PCEP peer as PCC.";
        }
        leaf pcrpt-rcvd {
          type yang:counter32;
          description
            "The number of PCRpt messages received.";
        }
        leaf pcupd-sent {
          type yang:counter32;
          description
            "The number of PCUpd messages sent.";
        }
        leaf rpt-rcvd {
          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 {
          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 {
          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.";
        }
        description
          "The stateful stats related to PCC as peer.";
      }
      container initiation {
        if-feature "pcep:pce-initiated";
        description
          "PCE-initiated related statistics.";
        container pcc {
          when "../../../../pcep:role = 'pcc'"
             + "or "
             + "../../../../pcep:role = 'pcc-and-pce'" {
            description
              "Valid for PCEP peer as PCC.";
          }
```

```
          leaf pcinitiate-sent {
            type yang:counter32;
            description
              "The number of PCInitiate messages sent.";
          }
          leaf initiate-sent {
            type yang:counter32;
            description
              "The number of LSP initiations 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.";
          }
          description
            "The initiation stats related to PCC as peer.";
        }
        container pce {
          when "../../../../pcep:role = 'pce'"
             + "or "
             + "../../../../pcep:role = 'pcc-and-pce'" {
            description
              "Valid for PCEP peer as PCE.";
          }
          leaf pcinitiate-rcvd {
            type yang:counter32;
            description
              "The number of PCInitiate messages received.";
          }
          leaf initiate-rcvd {
            type yang:counter32;
            description
              "The number of LSP initiations 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 {
            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.";
          }
          description
            "The initiation stats related to PCE as peer.";
        }
      }
    }
    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 the same
           path-key.";
      }
      leaf path-key-no-attempt {
        type yang:counter32;
        description
          "The number of expired path-keys with no attempt to
           expand it.";
      }
    }
    action reset-statistics {
      description
        "The reset action will clear the statistics at the
         associated container.";
      input {
        leaf reset-at {
          type yang:date-and-time;
          description
            "The time when the reset was issued.";
        }
      }
      output {
        leaf reset-finished-at {
          type yang:date-and-time;
          description
            "The time when the reset finished.";
        }
      }
    }
  }

  /*
   * Augment modules to add statistics
   */

  augment "/pcep:pcep/pcep:entity/pcep:peers/pcep:peer" {
    description
      "Augmenting the statistics.";
    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 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.";
      }
      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 reset-pcep-statistics-all {
    if-feature "reset-all";
    description
      "Reset all the PCEP statistics collected across all peers
       and sessions.  This RPC is used if the implementation
       supports a mechanism to reset all PCEP statistics across
       all peers and sessions through mechanisms such as by
       walking a list of pointers to those peers and sessions.

       If this mechanism is not supported, implementations must
       reset PCEP statistics individually by invoking the action
       for each peer and session.";
  }
}

<CODE ENDS>
```

9. Security Considerations

This section is modeled after the template described in Section 3.7 of [YANG-GUIDELINES].

The "ietf-pcep" and "ietf-pcep-stats" YANG modules define data models that are designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The "ietf-pcep" YANG module:

There are a number of data nodes defined in the "ietf-pcep" YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be reasonably sensitive or vulnerable in some network environments. Write operations (e.g.,

edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

- /pcep/entity/: Configures local parameters, capabilities, etc.
- /pcep/entity/peers: Configures remote peers to set up a PCEP session.

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

Some of the readable data nodes in this 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. Specifically, the following subtrees and data nodes have particular sensitivities/ vulnerabilities:

- /pcep/lsp-db: All the LSPs in the network. Unauthorized access to this could provide 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 or action 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. Specifically, the following operation has particular sensitivities/ vulnerabilities:

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

This YANG module uses groupings from other YANG modules that define nodes that may be considered sensitive or vulnerable in network environments. Refer to the Security Considerations of respective RFCs for information as to which nodes may be considered sensitive or vulnerable in network environments.

The YANG module defines a set of identities, types, and groupings. These nodes are intended to be reused by other YANG modules. The module by itself does not expose any data nodes that are writable, data nodes that contain read-only state, or RPCs. As such, there are no additional security issues related to the YANG module that need to be considered.

Modules that use the groupings that are defined in this document should identify the corresponding security considerations.

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 a significant notification load. Otherwise, this notification load may have some side effects on the system stability and may be exploited as an attack vector.

The "auth" container includes various authentication and security options for PCEP. Further, Section 7.1 describes how to configure TLS 1.2 and TLS 1.3 for a PCEP session via this YANG module.

The "ietf-pcep-stats" YANG module:

This document also includes another YANG module (called "ietf-pcep-stats") for maintaining the statistics by augmenting the "ietf-pcep" YANG module.

There are no particularly sensitive writable data nodes.

The readable data nodes in this 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. The statistics could provide information related to the current usage patterns of the network.

Some of the RPC or action 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. Specifically, the following operation has particular sensitivities/vulnerabilities:

- reset-pcep-statistics-all: The RPC is used to reset all PCEP statistics across all peers and sessions. An unauthorized reset could impact monitoring.

10. IANA Considerations

IANA has registered the following URIs in the "IETF XML Registry" [RFC3688]:

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.

IANA has registered the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name: ietf-pcep
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix: pcep
Reference: RFC 9826

Name: ietf-pcep-stats
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep-stats
Prefix: pcep-stats
Reference: RFC 9826

11. References

11.1. Normative References

[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>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[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, JL., Vasseur, JP., 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, JL., 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>.

[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>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[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>.

[RFC9645] Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", RFC 9645, DOI 10.17487/RFC9645, October 2024, <https://www.rfc-editor.org/info/rfc9645>.

11.2. Informative References

[IANA-IGP] IANA, "Path Computation Element (PCE) Capability Flags", <https://www.iana.org/assignments/igp-parameters>.

[IANA-PCEP] IANA, "ASSOCIATION Type Field", <https://www.iana.org/assignments/pcep>.

[PCEPS-UPDATE] Dhody, D., Turner, S., and R. Housley, "Updates for PCEPS: TLS Connection Establishment Restrictions", Work in Progress, Internet-Draft, draft-ietf-pce-pceps-tls13-04, 9 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-pce-pceps-tls13-04>.

[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>.

[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>.

[RFC9603] Li, C., Ed., Kaladharan, P., Sivabalan, S., Koldychev, M., and Y. Zhu, "Path Computation Element Communication Protocol (PCEP) Extensions for IPv6 Segment Routing", RFC 9603, DOI 10.17487/RFC9603, July 2024, <https://www.rfc-editor.org/info/rfc9603>.

[YANG-GUIDELINES] Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

[YANG-PCEP-SR] Li, C., Sivabalan, S., Peng, S., Koldychev, M., and L. Ndifor, "A YANG Data Model for Segment Routing (SR) Policy and SR in IPv6 (SRv6) support in Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-srv6-yang-07, 21 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-srv6-yang-07>.

[YANG-TE] Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-38, 29 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-38>.

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 domain]
        |     +--rw type      identityref
        |     +--rw domain    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-mpls {sr-mpls}?
        |  |  +--rw enabled?        boolean
        |  |  +--rw no-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
        +--ro msd?                           uint8 {sr-mpls}?
        +--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 neighbor-domains
         |  |  +--rw domain* [type domain]
         |  |     +--rw type      identityref
         |  |     +--rw domain    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     leafref
         |  |     +--ro lsp-id     leafref
         |  +--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               leafref
|        +--ro source           leafref
|        +--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 pcc-requester?   -> /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 domain]
      |     +--rw type      identityref
      |     +--rw domain    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-mpls {sr-mpls}?
      |  |  +--rw enabled?        boolean
      |  |  +--rw no-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
+--ro msd?                    uint8 {sr-mpls}?
+--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 neighbor-domains
|     +--rw domain* [type domain]
|        +--rw type      identityref
|        +--rw domain    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?                         -> ../../../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?   leafref
    |  +--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?      leafref
    |  +--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?           leafref
    |  +--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 provides an overview of PCEP peer session information and LSP-DB in the YANG module.

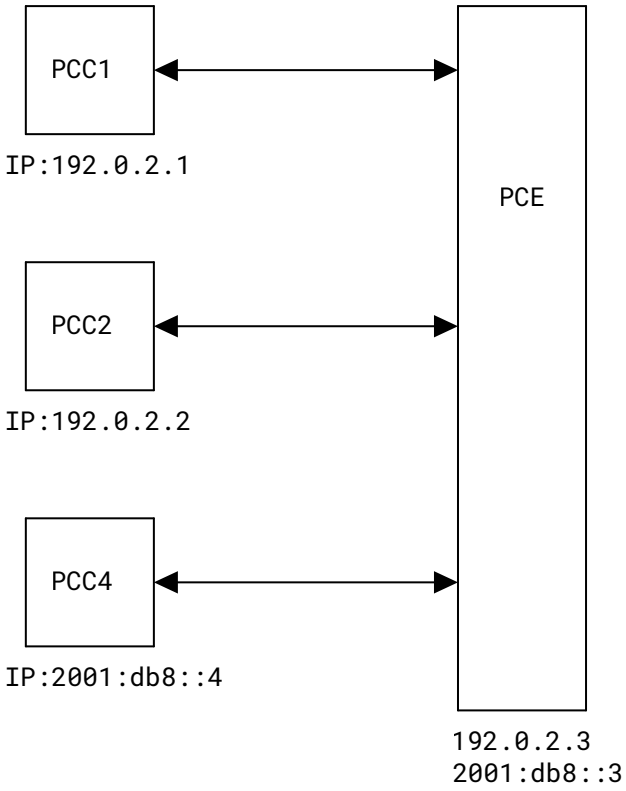

Figure 1: Example

```
at PCE:
{
  "ietf-pcep:pcep": {
    "entity": {
      "addr": [
        "192.0.2.3",
        "2001:db8::3"
      ],
      "role": "pce",
```

```
        "oper-status": "oper-status-up",
        "capabilities": {
          "capability": "active passive"
        },
        "init-back-off-timer": 5,
        "max-back-off-timer": 3600,
        "max-keepalive-timer": 255,
        "max-dead-timer": 255,
        "min-keepalive-timer": 1,
        "min-dead-timer": 30,
        "request-timer": 300,
        "max-sessions": 2400,
        "pcep-notification-max-rate": 5,
        "stateful-parameter": {
          "state-timeout": 300
        },
        "lsp-db": {
          "lsp": [
            {
              "plsp-id": 3,
              "pcc-id": "192.0.2.1",
              "source": "192.0.2.1",
              "destination": "192.0.2.4",
              "tunnel-id": 16,
              "lsp-id": 3,
              "extended-tunnel-id": "0.0.0.0",
              "operational-state": "up",
              "delegated": {
                "enabled": true
              },
              "symbolic-path-name": "iewauh"
            },
            {
              "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": "0.0.0.0",
              "operational-state": "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"
                }
              ]
            }
          },
          {
            "addr": "192.0.2.2",
            "role": "pcc",
            "capabilities": {
              "capability": "active passive"
            },
            "sessions": {
              "session": [
                {
                  "initiator": "remote",
                  "role": "pcc"
                }
              ]
            }
          },
          {
            "addr": "2001:db8::4",
            "role": "pcc",
            "capabilities": {
              "capability": "active passive"
            },
            "sessions": {
              "session": [
                {
                  "initiator": "remote",
                  "role": "pcc"
                }
              ]
            }
          }
        ]
      }
    }
  }
}
```

Similarly, a PCEP session with an IPv6 address between a PCE (2001:db8::3) and a PCC (2001:db8::4) could also be set up.

Appendix C. Design Objectives

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

- Existing implementations need to map the data model defined in this document to their proprietary 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 that 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
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

YANG Data Nodes	PCEP MIB Objects
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

YANG Data Nodes in /pcep/entity/peers/peer	PCEP MIB Objects
/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
/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

YANG Data Nodes in /pcep/entity/peers/peer	PCEP MIB Objects
/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

YANG Data Nodes in /pcep/entity/peers/peer/sessions/session	PCEP MIB Objects
ka-hold-time-rem	pcePcepSessKAHoldTimeRem
overloaded	pcePcepSessOverloaded
overloaded-timestamp	pcePcepSessOverloadTime
peer-overloaded	pcePcepSessPeerOverloaded
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

YANG Data Nodes in /pcep/entity/peers/peer/sessions/session	PCEP MIB Objects
/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
pcep-session-local-overload-clear	pcePcepSessLocalOverloadClear

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

Table 7: Relationship with PCEP MIB Notification

Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. The authors of this document would like to thank the authors of the above document.

Thanks to Martin Bjorklund and Tom Petch for the detailed review. Thanks to Mahesh Jethanandani and Jan Lindblad for the YANGDOCTOR review. Thanks to Scott Kelly for the SECDIR review. Thanks to Gyan Mishra and Matthew Bocci for the RTGDIR review.

Contributors

Rohit Pobbathi
Nokia Networks
India
Email: rohit.pobbathi_ashok@nokia.com

Vinod KumarS
India
Email: vinods.kumar@gmail.com

Zafar Ali
Cisco Systems
Canada
Email: zali@cisco.com

Xufeng Liu
Email: xufeng.liu.ietf@gmail.com

Young Lee
Samsung
Email: younglee.tx@gmail.com

Udayasree Palle
Email: udayasreereddy@gmail.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen
518129
China
Email: zhang.xian@huawei.com

Avantika
ECI Telecom
India
Email: avantika.srm@gmail.com

Shashikanth
India
Email: 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
United Kingdom
Email: jonathan.e.hardwick@gmail.com

Jeff Tantsura
Nvidia
United States of America
Email: jefftant.ietf@gmail.com