

Segment Routing and Path Computation Element

Using traffic engineering to optimize path placement and efficiency in IP/MPLS networks

Operators of IP/MPLS networks need to optimize their existing network infrastructure, increase the options available to existing services and potentially create new service offerings. In this environment, traffic engineering is vital to success. Segment Routing (SR) provides traffic engineering capabilities without the scaling constraints of Resource Reservation Protocol- Traffic Engineering (RSVP-TE). A path Computation Element (PCE) resolves some of the issues of distributed path computation. This paper explains how SR combined with PCE can optimize path placement and improve efficiency in networks.

Table of contents

Introduction	/ 1
Segment Routing technology overview	/ 2
How SR works	/ 2
Segment identifiers	/ 3
Path placement	/ 5
Overview	/ 5
Shortest path routing	/ 5
ECMP and load-balancing	/ 6
Disjointness	/ 9
Traffic engineering and PCE	/ 10
Challenges	/ 10
About PCE servers	/ 11
PCC-initiated LSPs	/ 13
PCE-initiated LSPs	/ 16
End-to-end path placement	/ 18
Disjointness and path profiles	/ 19
Topology discovery	/ 21
Overview	/ 21
Link state information	/ 21
Egress Peer Engineering	/ 23
Conclusion	/ 24
Acronyms	/ 25
References	/ 25

Introduction

Operators of IP/MPLS networks need to optimize their existing network infrastructure, increase the options available to existing services and potentially create new service offerings. Segment Routing (SR) and a Path Computation Element (PCE) deliver the functionality required to meet these needs.

Segment Routing is an emerging technology for IP/MPLS networks that enables source routing. Segment Routing allows the network operator to specify a path from ingress to egress using a forwarding path that is completely abstract from the Interior Gateway Protocol (IGP) shortest path.

In the SR domain, nodes and links are assigned Segment Identifiers (SIDs), which are advertised into the domain by each SR router using extensions to Intermediate System-Intermediate System/Open Shortest Path First (IS-IS/OSPF). These SIDs allow an ingress node to select a path through the network using either a single SID to represent the destination node or using a series of SIDs, called a segment list, which specifies a particular path through the network that an SR tunnel should traverse.

Segment Identifiers or segment lists can be encoded as one or more Multi-protocol Label Switching (MPLS) labels or as one or more IPv6 addresses. Where MPLS is used, SR does not have a requirement for the Label Distribution Protocol (LDP) and/or Resource Reservation Protocol – Traffic Engineering (RSVP-TE) transport signaling control plane.

No state is held in the network with the exception of the ingress SR router. This allows SR to scale significantly better than RSVP-TE while providing most of the same functions:

- SR can offer IGP-based MPLS tunnels to services such as VPRN or VPWS without the addition of any other transport signaling protocol.
- SR can offer Fast-Reroute capability using a pre-computed backup path that can provide full coverage and does not have any topology dependencies.
- SR provides the ability to source route using a combination of loose and/or strict hops. It allows for centralized or distributed traffic engineering models with most of the capabilities of RSVP-TE (including Admin-Groups and Shared Risk Link Groups) without the associated midpoint¹ state.

While SR enables building the forwarding paths across the network, some abstract intelligence is required to instruct ingress routers which paths to use through the network, and for which services. This intelligence is delivered by an external traffic engineering controller, which, among other things, functions as a stateful active Path Computation Element (PCE), providing end-to-end control of network resources based on real-time network state. This ensures that expensive Wide Area Network (WAN) capacity is effectively utilized, and because of the PCE's network-wide visibility, ensures that the network can deliver specific service requirements such as *disjointness* where required. (Disjointness is a term used to describe two or more services that must be completely disjoint of each other.)

The use of a centralized controller also drives the adoption of Software Defined Networking (SDN) in the WAN, offering a more agile way of networking by automating the creation and/or removal of bandwidth made available to particular services. Segment Routing Traffic Engineering (SR-TE) simplifies this process further by limiting instantiation and deletion of tunnel state to only the ingress

¹ A midpoint refers to a router through which a path transits and which is neither a head-end nor a tail-end router.

Label Switched Router (LSR). This in turn allows for the introduction of services such as bandwidth calendaring and bandwidth on demand to accommodate the dynamic nature of SDN applications.

This paper focuses on the interaction between the SR-enabled router and the external controller/PCE, with particular emphasis on path placement and path diversity.

Segment Routing technology overview

How SR works

Segment Routing provides a tunnelling mechanism that enables source routing in IP/MPLS networks. An SR path (SR tunnel) is encoded as a sequential list of sub-paths called *segments*, which are advertised to the SR domain using extensions to link-state routing protocols such as IS-IS or OSPF.

An SR tunnel can contain a single segment that represents the destination node or it can contain a segment list that represents the set of segments that a given tunnel must traverse.

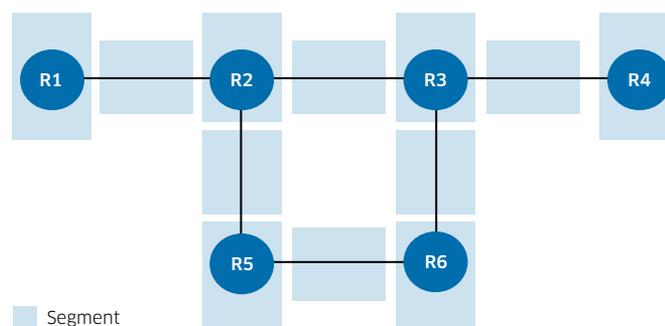
The SR tunnel can be established over an IPv4/IPv6 MPLS infrastructure or over an IPv6 infrastructure and is encoded as:

- A single MPLS label or an ordered list of hops represented as a stack of labels (with no change to the MPLS data-plane)
- A single IPv6 address or an ordered list of hops represented by a number of IPv6 addresses contained in an IPv6 extension header

The segments that an ingress router imposes for a particular tunnel act a set of instructions, such as “go to Node M using the shortest path” or “go to Node N using link/node/explicit-route L”

Figure 1 shows an SR domain represented as segments.

Figure 1. SR domain represented as segments



When MPLS is used to instantiate SR tunnels, the MPLS forwarding plane does not change. Segment Routing uses extensions to the link-state IGP to flood SIDs in the form of MPLS labels. No LDP and/or RSVP control plane is required although it is acceptable to run these in conjunction with SR: because the LDP and RSVP label spaces do not overlap, they do not affect each other.

Each SID is a 32-bit entity with the MPLS label encoded as the 20 right-most bits of the segment.

When SR is instantiated over the MPLS data plane, the following actions apply:

- A list of segments is represented as a stack of labels.
- The active segment is the top label.
- The CONTINUE operation is implemented as an MPLS swap operation.
- The NEXT operation is implemented as an MPLS pop operation.
- The PUSH operation is implemented as an MPLS push operation.

Unlike RSVP-TE based LSPs, in which the mid-points hold state, SR requires that only the ingress provider-edge (PE) holds state. For any transit or egress SR routers, any required state information is contained in the segment list.

The remainder of this chapter focuses on the instantiation of SR using MPLS, the likely choice for most early adopters. The term LSP is used interchangeably between RSVP-TE and SR.

Segment identifiers

In an SR domain, each segment is known as a Prefix-SID or an Adjacency-SID (Adj-SID). A Prefix-SID is globally unique within the IGP/SR domain. The SID value is allocated from a unique pool called the SR Global Block (SRGB).

In an MPLS network, the SRGB is a set of labels reserved specifically for SR use. A Prefix-SID represents the Equal Cost Multi-Path (ECMP)-aware shortest-path route to the related prefix and is typically a multi-hop path.

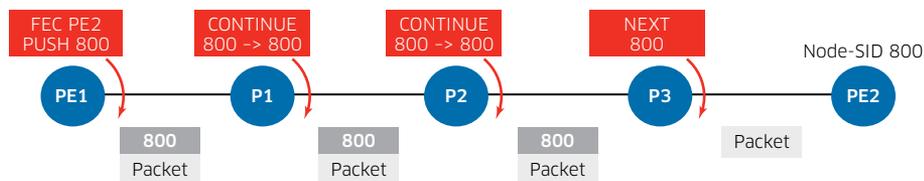
Prefix-SIDs: Node-SID and Anycast-SID

A Node-SID is a special type of Prefix-SID used to identify a particular router (loopback/system address) in the domain. The Node-SID is identified by an N flag set to 1 in the Prefix-SID sub-Type Length Value (TLV) that IS-IS or OSPF uses to advertise the SID. Because the Node-SID is a Prefix-SID, it also represents the ECMP-aware shortest-path route to the related prefix, typically a multi-hop path.

When an SR router advertises its Node-SID to the SR domain, all routers in the domain install the node segment in the data-plane². In the example shown in Figure 2, PE2 advertises a Node-SID of 800 to the SR domain. When PE1 wants to forward SR tunnel-encapsulated traffic towards PE2, it pushes on the node segment {800} and forwards the packet using its shortest-path towards PE2. Routers P1 and P2 each implement a CONTINUE (swap) action in the data plane.

In this example, router P3 also implements a CONTINUE (swap) action and may implement a NEXT (pop) action if the egress router has the P flag set to 0 in its advertised Prefix-SID Sub-TLV. This behavior is analogous to Penultimate Hop Popping (PHP) in MPLS.

Figure 2. SR tunnel with Prefix-SID (Node-SID)



² This paper assumes the use of absolute SID values, in which all SR routers in the domain use a single consistent SRGB. Indexing, an alternative option, is possible: SR routers have a different start point (start label) in the SRGB and advertise an offset label called an SID index. The advertised index is summed with the start point to create behavior analogous to LDP in independent label-distribution mode.

A second type of Prefix-SID is an Anycast-SID. An Anycast-SID is a prefix segment specifying a set of routers and represents the ECMP-aware shortest-path IGP route to the closest node of the *anycast set*. An Anycast-SID can be useful for coarse traffic engineering where an operator may choose to route through Plane A of a dual-plane network or route through Region B of a multi-region network

Adj-SIDs

An Adj-SID is a segment that identifies an adjacency or a set of adjacencies to another router that must be known in the IGP . The value of an Adj-SID is local to the router that advertises it, and every SR router in the domain can potentially use the same segment (label) space. Therefore, only the advertising router can install an Adj-SID in the forwarding information base.

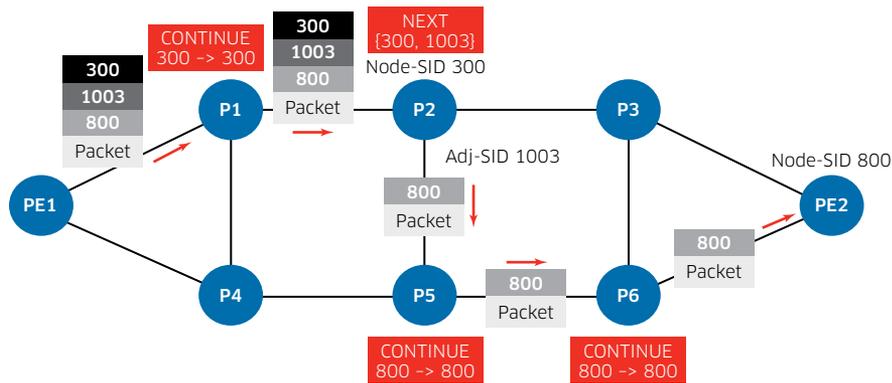
If an Adj-SID is used, the behavior is as follows (assuming AB is the Node-SID of Node N and ABC is an Adj-SID at Node N to an adjacency over link L):

1. A packet with segment list {AB, ABC} is forwarded along the shortest path to Node N.
2. The packet is switched by Node N toward link L without considering shortest-path routing as if a label-swap is set to implicit null.

Adj-SIDs can be used to define a source-routed explicit hop-by-hop path from ingress to egress. However, constructing lists using only Adj-SIDs can potentially create a deep segment list depth (or in the case of an MPLS data plane, deep label stack). An alternative method is to combine Node-SIDs and Adj-SIDs to exercise ECMP paths to the next specified Node-SID in the segment list and to enforce the use of a particular link (or links) from that node.

Figure 3 shows an example in which PE1 has an SR tunnel to PE2 but has a requirement to traverse the P2-to-P5 link and avoid the P2-to-P3 link PE1 therefore imposes the segment list {300, 1003, 800}: the Node-SID for P2, P2's Adj-SID for link P2-to-P5, and the Node-SID for PE2, respectively.

Figure 3. SR tunnel with combined Node-SID and Adj-SID



3 Where a number of adjacencies exist between two routers, a router can advertise an Adjacency-Set SID (Adj-Set SID) in addition to the Adj-SID. Traffic that subsequently arrives at the advertising router with this Adj-Set SID is load-balanced (potentially with weighting) across the available links of the Adj-Set.

Path placement

Overview

There are a number of methods that determine how a path (SR tunnel) is placed between ingress and egress routers in an SR domain. The path can be computed:

- At the ingress PE by executing a Constrained Shortest Path First (CSPF) using information obtained from traffic engineering extensions to the IGP. This is generically referred to as distributed TE: path decisions are made autonomously by each computing router. However, unlike RSVP-TE, SR has no control plane setup for an LSP. When RSVP-TE is in use there is a PATH message in the forward direction and a RESV message in the reverse direction that allow LSP bandwidth to be signaled and known at every hop.

After the LSP is established, the bandwidth it consumes is subtracted from the available bandwidth at each hop and the new available bandwidth value is flooded into the IGP. As a result, the bandwidth available on every interface is known to the head-end router executing the CSPF.

Because SR has no control plane for LSP setup, bandwidth requirements are not made known to the network. Therefore, available bandwidth is not flooded into the IGP. As a result, a CSPF cannot include bandwidth as a constraint.

- Using a CSPF at a centralized device that has knowledge of the network topology and current utilization. The computed path is then passed to the ingress PE using any of a number of protocols. This is generically referred to as centralized TE and is discussed in the next chapter.
- At the ingress PE by following the IGP shortest path to the destination.

The simplest form of SR tunnel placement is the last approach using shortest-path routing. Therefore, we'll look at that approach first before discussing alternative approaches that encompass TE capability.

Shortest path routing

All SR routers in the domain that function as transit or termination points for SR tunnels advertise a unique node segment into the IGP using the Prefix-SID Sub-TLV extension to IS-IS/OSPF. Each advertised Node-SID is installed in the Label Forwarding Information Base (LFIB) of all other SR routers in the domain.

When shortest-path routing to a destination Node-SID is employed, the head-end and each transit router forward SR-encapsulated packets using their shortest path towards the prefix associated with the Node-SID; this also includes any ECMP paths.

Using the example shown in Figure 4:

- PE2 advertises a Node-SID with label value 800 into the IGP and installs a NEXT (pop) entry in the LFIB for that label. Whether traffic arrives labeled or unlabeled depends on the setting of the P-bit in the advertised Prefix-SID Sub-TLV. The example assumes that the P-bit is set, meaning PHP is not in use.
- All of the SR routers in the domain install a PUSH entry and a CONTINUE (swap) entry for the advertised prefix/label in the LFIB.

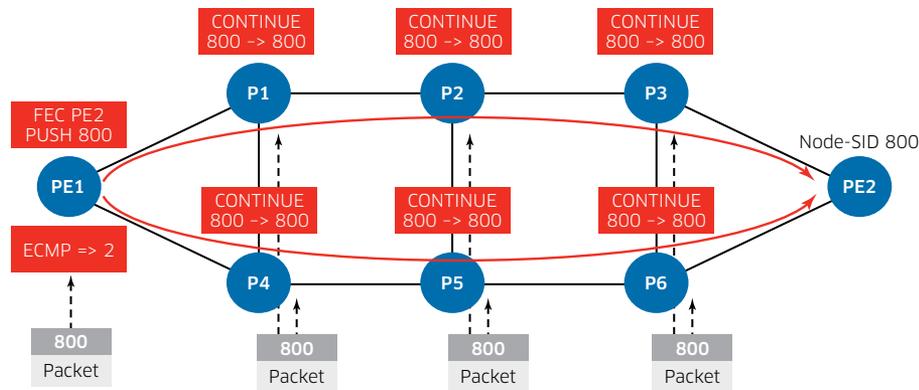
For SR-encapsulated traffic from PE1 to PE2 (noting that absolute SID values are used and all SR routers in the domain use a single consistent SRGB):

- PE1 pushes on segment list {800} and uses its shortest IGP path to reach PE2. If ECMP is set to 2 or more, all ECMP paths are exercised.

In Figure 3, there are two ECMP paths: P1-P2-P3-PE2 and P4-P5-P6-PE2. As a result of these ECMP paths, based on hash output some flows are routed P1-P2-P3-PE2 while other flows are routed P4-P5-P6-PE2.

- Routers P1-P2 and P4-P5 install an Incoming Label Map (ILM) entry of {label = 800, Next-Hop-Label-Forwarding Entry (NHLFE) = 800, Next-Hop = shortest path to PE2}.
- Routers P3 and P6 also install an ILM entry of {label = 800, NHLFE = 800, Next-Hop = shortest-path to PE2} but could instead implement PHP if the egress router signals its Prefix-SID Sub-TLV with the P flag set to 0.

Figure 4. SR tunnel with shortest-path routing



Based on network requirements and the subsequent configuration, the SR tunnel can then be made available to Layer-2/Layer-3 VPN services as well as infrastructure forwarding such as IGP/BGP shortcuts.

SR tunnels using shortest-path routing work well for many applications, but like LDP, SR tunnels slavishly follow the IGP. What if, for example, the link P2-P3 becomes congested to the point where there is a requirement to move some traffic or services away from the shortest-path route? Similarly, what if the link P5-P6 is subject to a failure at the transmission layer and is restored over a very sub-optimal path? This situation may breach the latency requirements of services carried at the IP layer such that there is a requirement to move those services away from that link. In both examples given, traffic engineering becomes a requirement.

ECMP and load-balancing

As previously discussed, a Node-SID represents the ECMP-aware shortest-path route to the related prefix. In the network diagram shown in Figure 4, PE1 has two ECMP routes to PE2. If PE1 has ECMP configured to a value of 2 or more, it will exercise both of the paths.

As the ingress LSR, PE1 has access to IP and UDP/TCP header information and will probably use that information as hash key input to decide which flows are mapped to which ECMP route (noting that a given flow must always follow the same path to avoid packet re-ordering). This typically provides for fairly granular hashing.

In many networks LSRs also have ECMP paths to a given destination, and the network operator wants to use these ECMP paths. Or the network may consist of Link Aggregation Groups (LAGs), which use hash input that is similar (but not identical) to ECMP hashing.

Ideally, the LSR would have access to the same hash key input as the ingress LSR. However, in an MPLS network this may require a fairly deep packet inspection. In many cases this is either not desirable or simply not possible due to hardware limitations. As a result, LSRs frequently use the MPLS label stack as the hash key input, which can lead to very coarse load-balancing, particularly where fat pseudowires exist in a network.

To eliminate the need for deep packet inspection while still allowing varying hash key input for LSR load-balancing, the Entropy Label (sometimes called the flow label or hash label) was introduced for the MPLS data plane. The Entropy Label (EL) is an additional MPLS label imposed by the ingress LSR.

Because the ingress LSR has access to un-encapsulated flow information, it can impose a different EL value for different flows. This provides entropy to hash input keys at LSRs to enable more granular load-balancing on a per-flow basis. If the ingress LSR and the egress LSR can support ELs for a given service, the ingress LSR can impose ELs on a per-flow basis while the egress LSR can simply pop those same labels.

This process implies two things:

- There needs to be mutual consent between ingress and egress on the use of an EL to ensure that both can process the additional label.
- Because it is the ingress LSR that imposes the EL, it follows that the egress LSR must be able to distinguish between ELs and conventional application labels.

To agree on the use of an EL between ingress and egress LSRs, explicit provisioning is possible at ingress and egress. This explicit provisioning negates any requirement for control plane extensions.

Dynamic negotiation of EL capability is possible but it needs to be coupled with the protocol used to distribute labels.

- RFC 6391: *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*⁴ makes provision for negotiation of ELs for LDP-signaled pseudowire services using a new pseudowire Interface Parameter Sub-TLV known as the Flow Label Sub-TLV. This is service-level negotiation of ELs.
- RFC 6790: *The Use of Entropy Labels in MPLS Forwarding*⁵ is more generic and defines the Entropy Label Capability (ELC) as a means to negotiate the use of ELs. For quantified reasons, this RFC opted to associate ELs (and ELC) with MPLS tunnels rather than MPLS applications (such as pseudowires) and made the ELC applicable to a number of MPLS tunneling protocols.

4 IETF, RFC 6391: *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*, November 2011. <https://tools.ietf.org/html/rfc6391>

5 IETF, RFC 6790: *The Use of Entropy Labels in MPLS Forwarding*, November 2012. <https://tools.ietf.org/html/rfc6790>

For LDP, an ELC TLV is defined that is an optional parameter of a LABEL MAPPING message. For BGP, a new non-transitive ELC attribute is introduced. For RSVP-TE, an ELC flag is added to the Attribute Flags TLV of the LSP_ATTRIBUTES object.

- With the introduction of SR, in which labels are signaled using extensions to the IGP and not conventional MPLS signaling protocols (LDP, BGP, RSVP-TE), the preceding methods are largely ineffective. The IETF draft *Signaling Entropy Label Capability Using Interior Gateway Protocols*⁶ therefore proposes advertising EL capability in OSPF/IS-IS using a new ELC TLV carried in the Router Information LSA (OSPF) and/or a new ELC Sub-TLV carried in the Router Capability TLV (IS-IS).

With the EL negotiation options in place, the egress LSR still needs the ability to distinguish the EL from application labels. In some early implementations, the EL is assumed to be bottom of the label stack (the Bottom-of-Stack bit is set to 1)⁷. While this approach suits most environments, in certain environments such as LSP tunneling (also called LSP hierarchy or stacked tunnels) an EL may appear more than once in a label stack: on the outer LSP and also on the encapsulated LSP.

So that an EL can always be unambiguously identified, a new Entropy Label Identifier (ELI) is defined that must always appear as the label preceding an EL. The ELI has a reserved label value (7) to uniquely identify it, and that value essentially says “an Entropy Label follows”.

Using an ELI to indicate the presence of an EL theoretically means that hashing could be carried out at every level of the LSP hierarchy and at the egress LSR. The reality, however, is somewhat different.

The imposition of the additional EL/ELI labels, coupled with the ability in SR to stack a combination of Node-SID and Adj-SID labels, combine to create the potential for a deep label stack. This in turn can create hardware dependencies at some LSRs, which may not be able to process labels below a certain label stack depth. All routers in the network therefore need to understand the depth at which label stacks can be processed as well as any Maximum Transmission Unit (MTU) implications.

The IETF draft *Entropy labels for source routed stacked tunnels*⁸ suggests that if an ingress LSR is made aware of the maximum label depth (called the Readable Label Depth [RLD]) of all routers in the domain, it can make informed decisions about where to place the EL/ELI labels and how many it can or should insert.

As described, for SR the RLD would be advertised in an ELC TLV of a Router Information LSA or ELC Sub-TLV of a Router Information TLV (IS-IS).

To date, the use of the ELI is not widely adopted, but it is reasonable to expect that it will become more prevalent in the near future.

6 IETF, Internet Draft *Signaling Entropy Label Capability Using Interior Gateway Protocols*, September 6, 2013. <https://tools.ietf.org/html/draft-xu-mpls-el-capability-signaling-igp-00>

7 In the absence of Control Word the EL is placed between the label stack and payload. In the presence of Control Word, the EL is placed between the label stack and Control Word.

8 IETF, Internet Draft *Entropy labels for source routed stacked tunnels*, February 14, 2014. <https://tools.ietf.org/html/draft-kini-mpls-spring-entropy-label-00>

Disjointness

A common requirement for Ethernet Line (E-Line) services is disjointness. The term disjointness is used to describe two or more services that must be completely disjoint of each other. These services should not share any common network infrastructure, including ingress and egress LSR, so that that if one fails, the other must always be active. This requirement comes largely from enterprises that replace legacy SDH circuits with E-Line services.

Many networks use a dual-plane strategy in which each plane has an element of diversity and redundancy from the opposing plane, be it different router platforms from different vendors, or different fibre paths, or both. Edge routers then connect redundantly to each plane or have a direct link to one plane and an indirect link to the opposing plane through another edge router.

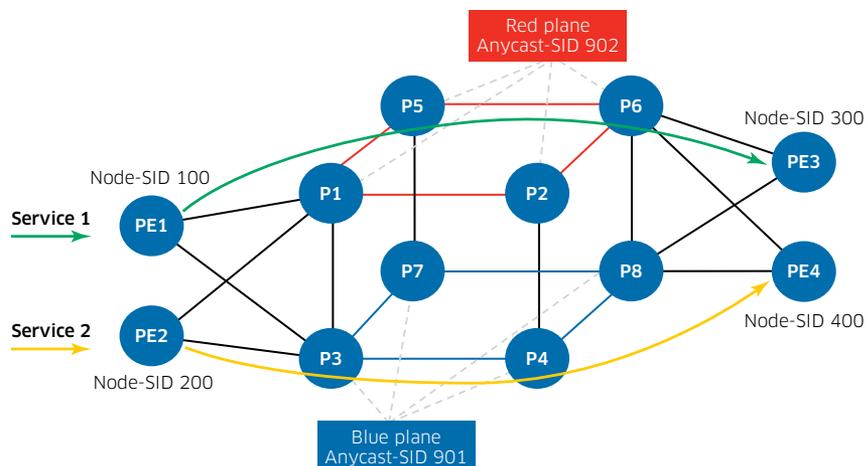
In this environment, it is possible to provide a coarse level of disjointness using an Anycast-SID to specify a set of routers. Each SR router in the Anycast set advertises the same Anycast-SID, which then represents the ECMP-aware, shortest-path IGP route to the closest node of that Anycast set.

Consider the dual-plane topology in Figure 5. Routers P3-P4 and P7-P8 belong to the blue plane Anycast set and advertise the Anycast-SID {901}. Similarly, routers P1-P2 and P5-P6 belong to the red plane Anycast set and advertise the Anycast-SID {902}. Service 1 is an E-Line service from PE1 to PE3 while Service 2 is an E-Line service from PE2 to PE4.

Wherever possible these services should be disjoint from each other. To effect this, Service 1 at PE1 imposes the segment list {902, 300}, which means that it will follow the ECMP-aware shortest path to the red plane Anycast set before reaching PE3 with Node-SID 300.

Service 2 at PE2 imposes the segment list {901, 400}, which means that it will follow the ECMP-aware shortest path to the blue plane Anycast set before reaching PE4 with Node-SID 400.

Figure 5. Disjointness using Anycast-SIDs



In the example of Figure 5, the Anycast-SID approach achieves disjointness between Service 1 and Service 2 under normal operating conditions. However, this disjointness cannot be guaranteed under failure conditions or any type of topology change. Both services can become routed through some common piece of infrastructure that represents a Shared Risk Group (SRG), and no indication would be given until that piece of infrastructure failed.

An alternative method of achieving disjointness is to use separate IGP instances in each plane and include the PE router loopbacks in each instance. Services requiring disjointness can then be bound to SR tunnels that are routed through each IGP instance. However, this may not be a realistic option for many network operators who already have production networks.

What is needed is a methodology to achieve guaranteed service disjointness: a methodology whereby when the path for Service 1 is being computed, a constraint is applied that stipulates “avoid the path that Service 2 takes”. Equally, when the path for Service 2 is being computed, a constraint is applied that stipulates “avoid the path that Service 1 takes”. Although this is possible to some extent with RSVP-TE and Administrative Groups, a full-mesh of RSVP-TE LSPs in a large network already creates scaling challenges. Because disjointness is a per-service requirement, it cannot be guaranteed with a full-mesh topology. In the worst case, the requirement could be to create RSVP-TE LSPs on a per-service basis, which is clearly not scalable.

To achieve guaranteed service disjointness requires traffic engineering with scalability.

Traffic engineering and PCE

Challenges

By imposing a segment list consisting of a series of Node-SIDs and/or Adj-SIDs at the ingress router, SR provides a traffic engineering capability (SR-TE) that is highly scalable. No path state is held at midpoint and/or tail-end routers, and path state is held by only the ingress router. Traffic engineering information that is flooded throughout the IGP domain using OSPF/IS-IS TE extensions can be used at the head-end in constraint-based SPF for SR tunnels in the same way it is currently used for RSVP-TE based LSPs. This approach is called distributed traffic engineering.

To compute a traffic engineered path, the computing element must run a constraint-based SPF with the relevant constraints applied (bandwidth, administrative-groups, Shared Risk Link Groups [SRLGs] and so on). In large networks, this path computation can become complex and CPU-intensive when there are a large number of paths to compute.

Because SR does not suffer from the same scaling constraints as RSVP-TE based LSPs, it can be used not only to deliver a transport-layer topological mesh but also to deliver performance-engineered tunnels at a service level. However, this capability creates the potential to increase the number of constraint-based LSPs that a head-end may need to compute.

Aside from the potential number of SR-TE-based LSPs, multi-area networks also impose limitations when distributed TE is used. In multi-area networks it is not possible to compute an optimal end-to-end path with a constrained SPF due to the lack of visibility of the entire network topology.

Options to compute an optimal end-to-end path do exist with RSVP-TE using Explicit Route Object (ERO) expansion at area border routers (ABRs)/Level 1-Level 2 routers. However, SPFs are computed for only the portion of the path that the computing router resides in, not the overall end-to-end path, which could potentially be more optimal.

About PCE servers

A Path Computation Element (PCE)⁹ is an entity that is capable of computing a constraint-based path based on information known about a given network topology. The PCE can be located in a router, or more likely, it can be an off-network server. The objective of a PCE is to resolve some of the issues of distributed path computation:

- Distributed TE lacks network-wide path scheduling. TE routers signal (and re-signal) asynchronously, which leads to collisions, re-tries and path packing problems.
- Distributed TE cannot enforce constraints on paths that originate/terminate on different routers. This includes the requirement for disjointness/diversity and bi-directionality.
- Distributed TE lacks end-to-end path visibility in multi-area networks.
- With distributed TE, each hop computes local Fast-Reroute protection in an autonomous way, considering only its direct adjacencies (nodes/links). A TE controller is capable of computing end-to-end protection for the TE tunnel, simplifying the protection strategy.

Using a PCE server to offload path computation is not a new idea. PCE has, to date, been used to calculate RSVP-TE-based LSPs. However, because of the scaling concerns, RSVP-TE has not been deployed in a significant number of large multi-area/multi-domain networks. As a result, there have also not been a significant number of production deployments of PCE servers.

Segment Routing provides the same ability as RSVP-TE to source route, but due to its lack of midpoint state, SR has significantly higher scaling properties than RSVP-TE. However, this same lack of midpoint state makes it impossible for the ingress LSR to compute a path through the network based on available bandwidth at intermediate hops. The increased scalability of SR together with the ability to signal a path based on bandwidth constraints means that the use of PCE has re-emerged as a viable proposition for large networks.

A PCE can be stateful or stateless.

A stateful PCE server has strict synchronization between the PCE and the network state using the Traffic Engineering Database (TED). A stateful PCE server also maintains state on the set of active paths and their reserved resources in use in the network.

A stateless PCE server computes paths based on the TED but processes each path independently and does not need to remember any previously computed paths. As a result, the server can be slightly out of synchronization with actual network state. Because it has no view of current active LSP state, it is impossible for a stateless PCE server to attempt any kind of re-optimization of active paths.

Use of a stateful PCE server increases the likelihood of an optimal path computation but requires a reliable synchronization mechanism between itself and the network.

In summary, a stateful PCE server performs the following functions:

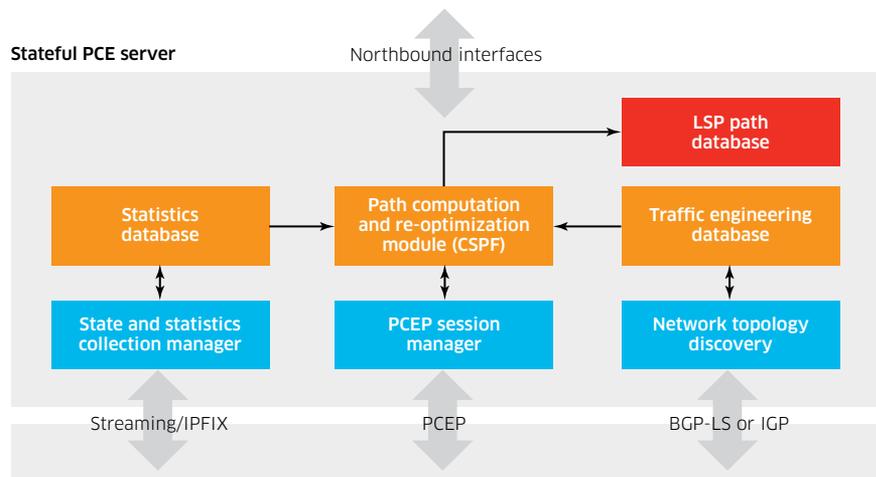
- Discovers the network topology and resources by listening to IGP/TE-resource updates. This can be accomplished by directly tapping into the IGP using a route-listener or through some other means such as BGP Link-State (BGP-LS).
- Collects information on link statistics using, for example, IPFIX, SNMP or XML-based accounting.

⁹ IETF, RFC 4655: *A Path Computation Element (PCE)-Based Architecture*, August 2006. <https://tools.ietf.org/html/rfc4655>

- Accepts requests from network elements or management systems for computing paths using the Path Computation Element Protocol (PCEP)
- May communicate with PCEs in other domains/areas and/or layers for path computation. There can be multiple PCE servers managing a single multi-domain/multi-area network or there can be PCE servers at both the optical and IP layers, and they may need to communicate with each other to ensure that an optimal path is selected that meets the requirements of both layers.
- Monitors network state, measuring such things as real-time (or near real-time) traffic demand and LSP statistics, and re-optimizes path placement based on that knowledge.
- Supports both RSVP and SR LSP types.

Figure 6 shows the functional blocks of a stateful PCE server.

Figure 6. Stateful PCE server



A router or network element that has a requirement to communicate with a PCE server to request a path computation is called a Path Computation Client (PCC). The PCE Protocol (PCEP) is used to establish this interface.

The PCE Protocol operates over TCP (port 4189) and consists of a common header followed by a number of mandatory and optional objects that specify the characteristics of the LSP.

The original PCEP specification, defined in RFC 5440: *Path Computation Element (PCE) Communication Protocol*¹⁰, was intended to be stateless and focused purely on RSVP-TE for LSP placement. The specification was subsequently extended in the IETF draft *PCEP Extensions for Stateful PCE*¹¹ to include extensions for stateful PCE operation. More recently, further extensions were defined in the IETF draft *PCEP Extensions for Segment Routing*¹² to include support for SR.

After a PCEP session has been established, the process of establishing and/or updating an LSP can be PCC-initiated or PCE-initiated dependent on mutual consent and exchange of the relevant parameters and capabilities between the peers.

¹⁰ IETF, RFC 5440: *Path Computation Element (PCE) Communication Protocol*, March 2009. <https://tools.ietf.org/html/rfc5440>

¹¹ IETF, Internet Draft *PCEP Extensions for Stateful PCE*, February 28, 2012. <https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-00>

¹² IETF, Internet Draft *PCEP Extensions for Segment Routing*, October 26, 2014. <https://tools.ietf.org/html/draft-ietf-pce-segment-routing-00>

PCC-initiated LSPs

Using only the stateful extensions to PCE, the process of path setup and teardown is always controlled by the PCC. However, the process of path maintenance and control of a given LSP differs depending on whether the PCE server is acting as an active stateful PCE or a passive stateful PCE for this LSP.

The active/passive PCE server role can be configured on a per-LSP basis by enabling or disabling the delegation of the LSP to the PCE server. In addition, when a PCC is trying to select a PCE server among candidate servers, the PCC can operate the LSP in passive stateful mode until it selects which PCE is the best choice. After the selection is made, the PCC can then delegate the LSP to that specific PCE.

Passive stateful PCE

When a passive stateful PCE is in use, the PCE uses LSP state information learned from PCCs to make path computations. The PCEP session begins with an exchange of OPEN messages containing information such as PCEP version, Keepalive timers and a PCEP Session ID used to uniquely identify the given session. The Open object contained in the OPEN message can also carry a Stateful-PCE-Capability TLV to indicate support for stateful PCE operation and an SR-PCE-Capability-TLV to indicate support for SR within the PCEP session.

If the parameters are acceptable to the PCC and PCE, KEEPALIVE messages are exchanged, after which the session is considered established.

The PCC then synchronizes its LSP state with the PCE server using Path Computation Report (PCRpt) LSP state reports. Each PCRpt message contains details about one or more LSPs and sets a Sync flag in the mandatory LSP object to indicate that the synchronization process is still in progress. When the last PCRpt message is sent by the PCC, the PCC sets the LSP object Sync flag to 0 to indicate to the PCE server that the synchronization process is complete.

When the synchronization process is completed, any event at the PCC that triggers the computation of a path causes the PCC to send a Path Computation Request (PCReq) to the PCE. The PCReq contains a number of mandatory objects and potentially some optional objects describing the endpoints and characteristics/constraints of the path.

Table 1 shows a non-exhaustive list of mandatory/optional objects that may be present in a path computation message exchange.

Table 1. Path computation messaging objects

OBJECT	DESCRIPTION
Request Parameters (RP)	Contains a Request-ID number, which together with the source IP address of the PCE/PCC uniquely identifies the path computation request. Contains a flags field to indicate priority, re-optimization of an existing LSP, bi-directionality and strict/loose options.
Stateful Request Parameters (SRP)	Contains an SRP-ID number that increments for every operation (initiation, update) on a given PCEP session and is used to correlate between update requests sent by the PCE and the error reports and state reports sent by the PCC. The SRP object can also include a Path-Setup-TLV if the path setup cannot be unambiguously inferred from the ERO or any other object/TLV.
End-points	Used to specify the source/destination IP address of the path for which a path computation is requested.
LSP	Contains fields to uniquely identify the target LSP (PLSP-ID, Symbolic Path Name), the operation to be performed on the LSP and LSP delegation. Can also contain an LSP-Error TLV or RSVP Error Spec TLV if the LSP transitions to a non-operational state or fails to establish.

OBJECT	DESCRIPTION
Bandwidth	Specifies the requested bandwidth for the computation request or the existing bandwidth in the case of a re-optimization request.
Metric	Used in a path computation request to indicate a bound on the path cost or used in a re-optimization request to indicate the existing metric for which an optimization should be attempted.
Reported-Route (RRO)	Used to report the route of a path for which a re-optimization is desired.
LSP Attributes (LSPA)	Specifies various RSVP-TE LSP attributes that should be taken into account by the PCE during path computation. Encoding is identical to an RSVP-TE SESSION ATTRIBUTE object.
Include Route (IRO)	Optionally used to indicate a number of specified network elements that the path must traverse.
Synchronization Vector (SVEC)	Used by the PCC to request the synchronization of a set of independent path computation requests. Contains flags to request link, node or SRLG diversity.
Load-Balancing	If no path for bandwidth X can be found by a PCE, the PCC may request a set of LSPs so that the sum of their bandwidth is equal to X, and then load-balance across them.
Explicit-Route (ERO)	Used by the PCE to provide the computed path. Encoding is identical to an ERO object in RSVP-TE.
Segment-Routing ERO (SR-ERO)	SR-TE path containing one or more node or adjacency-SIDs encoded in SR-ERO sub-objects.
No-Path	Used by the PCE in response to an unsuccessful path computation request. Nature of Issue (NI) or NO-PATH-VECTOR TLV can be used to report the reason for the negative reply.

The PCE server takes the path characteristics signaled by the PCC in the PCReq message and attempts to compute a path with the relevant constraints. The PCE server then responds with a PC Reply (PCRep) message, which can be positive or negative.

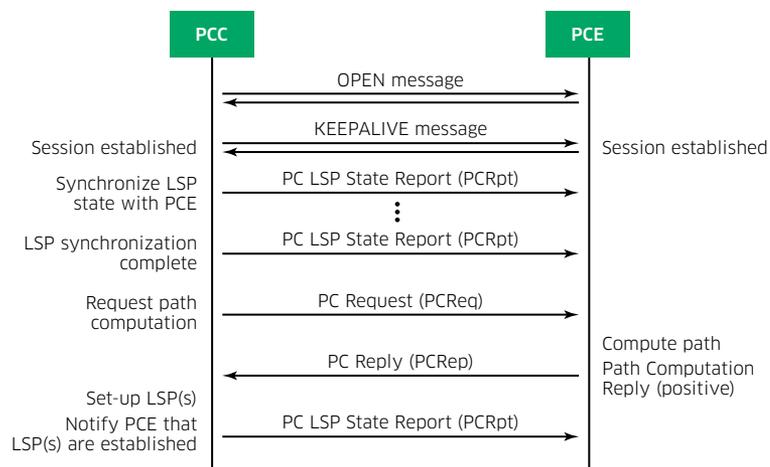
If the response is positive, the PCRep contains details about the path that should be used through the network in the form of an SR- ERO containing a segment list reflecting the relevant Node- and Adjacency-SIDs that the path should take. The PCC instantiates the LSP using the path defined by the PCE.

After the path is established, the PCE notifies the PCC using another PCRpt message containing the operational status of the LSP.

Figure 7 shows the passive stateful PCE process.

If for any reason the PCE is unable to compute a path, the PCRep message is negative. Upon receipt of a negative PCRep message the PCC may resend a modified PCReq with relaxed constraints.

Figure 7. Passive stateful PCE server



Active stateful PCE

An active stateful PCE is an extension of the passive stateful PCE in which the PCE uses a delegation mechanism to update LSP parameters in PCCs that have delegated control of some or all of their LSPs to it. This delegation enables the PCE to make unsolicited updates to existing LSPs.

During the OPEN message exchange, support for this delegation of control is indicated by setting an LSP-Update-Capability flag (U-flag) to 1 in the Stateful-PCE-Capability TLV. If the U-flag is set to 1 by a PCC, this indicates that the PCC allows modification of LSP parameters. If the U-flag is set to 1 by a PCE, this indicates that the PCE is capable of updating LSP parameters.

After the PCEP session is established, the PCC goes through the same process of synchronizing its current LSP state with the PCE server with the difference that each LSP that the PCC wants to delegate to the PCE has a delegate-bit set to 1 in the corresponding LSP object. The delegation gives the PCE server control for updating that LSP although LSP state ownership is retained by the PCC. A PCC can revoke this delegation at any time during the lifetime of the LSP. Equally, a PCE can return LSP delegation at any time.

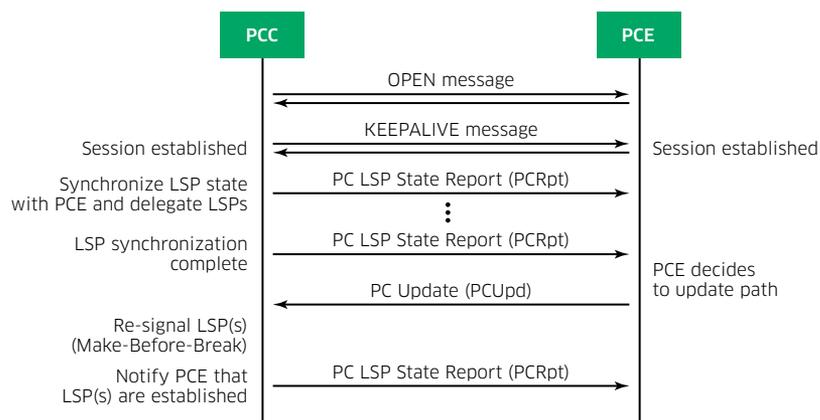
After the LSP state synchronization and delegation is complete, if the PCE decides to update an LSP it sends a Path Computation Update (PCUpd) message to the PCC. The PCUpd message includes a number of objects to identify the particular LSP and any updates to that LSP, such as a new bandwidth or a new SR-ERO through the network

On receipt of the PCUpd, the PCC re-initiates the LSP in a make-before-break manner.

After the LSP is modified, the PCC sends a PCRpt to the PCE to notify it of the status of the LSP. If the update was successful, the LSP object contains an indication that the LSP is operationally up. If the update was unsuccessful, the LSP object contains an LSP-Error TLV with the relevant error code.

Figure 8 shows the active stateful PCE process.

Figure 8. Active stateful PCE server



For a non-delegated LSP or prior to the LSP being delegated to the PCE, the PCReq/RCRep exchange between PCC and PCE (used for passive stateful PCE) is still valid.

PCE-initiated LSPs

In the stateful PCE mechanisms already described, the LSPs are always initiated by the PCC. This process is well-suited to environments in which LSP placement is reasonably static.

The IETF draft *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*¹³ specifies extensions to the stateful PCE model that allow for setup, maintenance and teardown of PCE-initiated LSPs without the need for configuration on the PCC.

The ability for a PCE to dynamically trigger the creation/teardown of LSPs based on application demand is intended to foster an agile WAN-based Software-Defined Networking (SDN) architecture in which intelligence in the SDN controller can determine when and where to establish paths. This can be the establishment of a new path with certain constraints between two endpoints and subsequent teardown when it is no longer required or the dynamic adjustment of an existing path, such as a timed bandwidth increase/decrease.

During the OPEN message exchange, support for PCE-initiated LSPs is indicated by setting the LSP-Instantiation-Capability flag (I-flag) to 1 in the Stateful-PCE-Capability TLV. If the I-flag is set to 1 by a PCC, this indicates that the PCC allows instantiation of an LSP by a PCE. If the I-flag is set to 1 by a PCE, this indicates that the PCE will attempt to instantiate LSPs. To support PCE-initiated LSPs, both parties must have the flag set to 1.

After the PCEP session is established, the PCC synchronizes its current LSP state with the PCE server, delegating control of LSPs to the PCE using the same mechanism of setting the delegate-bit in the corresponding LSP object.

After the LSP state synchronization and delegation are complete, if the PCE decides to update an existing LSP it follows the same process previously described for active stateful PCE.

If the PCE decides to create a new LSP it sends a Path Computation Initiate (PCInit) message to the PCC containing, at a minimum:

- An LSP object with a special PLSP-ID value of 0 and a Symbolic-Path-Name TLV used to correlate the PCC-assigned PLSP-ID and the LSP
- An SRP object containing an SRP-ID-number used to correlate initiation requests sent from the PCE and state/error reports sent by the PCC
- An ERO object (RSVP-TE LSP) or SR-ERO object (SR-TE LSP) defining the hops of the path
- An Endpoint object containing the source/destination addresses (required only for RSVP-TE)

¹³ IETF, Internet Draft PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model, December 2, 2013. <https://tools.ietf.org/html/draft-ietf-pce-pce-initiated-lsp-00>

If the PCInit attributes are acceptable and the PCC is able to establish the LSP, it responds with a PCRpt message containing:

- An LSP object containing the PCC-assigned PLSP-ID and using the Symbolic-Path-Name passed by the PCE in the PCInit message. The delegate-bit is set to 1 to delegate control of the LSP to the PCE, and a new Create-flag (C-flag) is set to 1 to indicate that this LSP was created through a PCInit message. Three operational bits are used to signal the status of the LSP, one of Down, Up, Active, Going down or Going up.
- An SRP object reflecting the SRP-ID-number sent in the PCInit message from the PCE

If any of the mandatory objects are missing, or if the PCInit message is unacceptable for any other reason, or if the LSP cannot be established, the PCC responds with a PC Error (PCErr) message containing the value code of the relevant error type.

After the PCE decides that the LSP is no longer required, LSP deletion is performed by sending a PCInit message with an LSP object carrying the PLSP-ID of the LSP to be removed and an SRP object with a Remove-flag (R-flag) set to 1. The PCInit message instructs the PCC to remove the LSP and any associated state. After removal, the PCC sends a PCRpt to notify the PCE of the LSP removal.

Figure 9 shows the PCE-initiated LSP process.

Figure 9. PCE-initiated LSPs

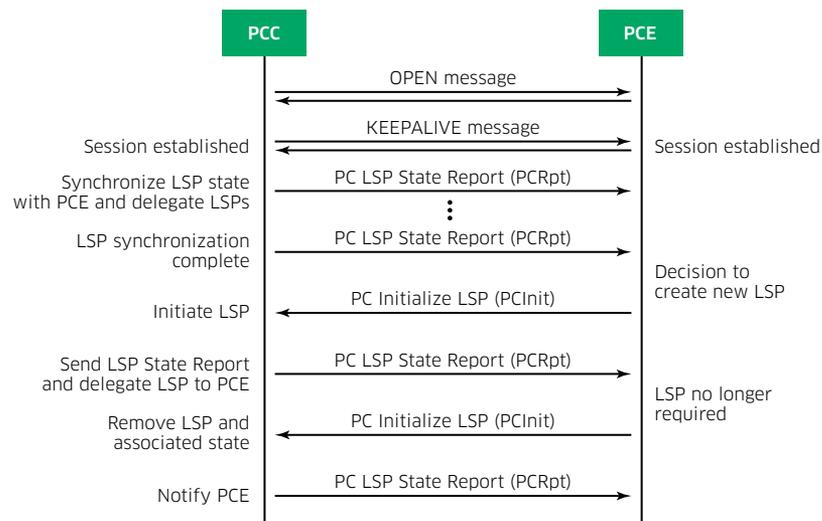


Table 2 provides a summary of the capabilities of stateless and stateful PCE servers.

Table 2. PCE server comparison

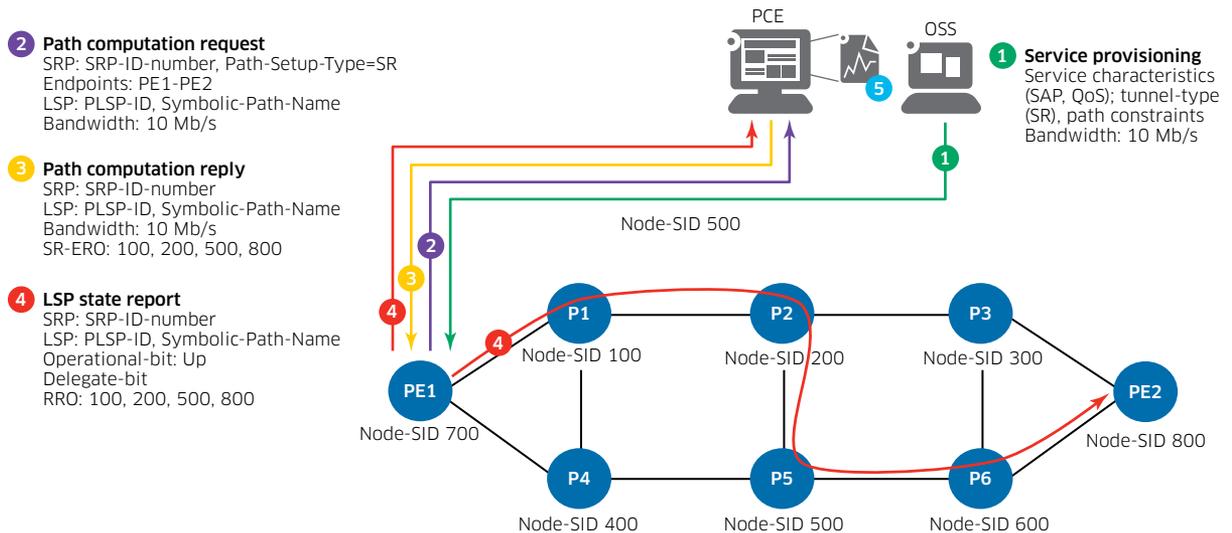
CAPABILITY	STATELESS PCE	PASSIVE STATEFUL PCE	ACTIVE STATEFUL PCE
Strict synchronization between PCE server and network state	No	Yes	Yes
Allows for LSP re-optimization	No	No	Yes
Allows for PCE-initiated LSP creation and removal	No	No	Yes

End-to-end path placement

Figure 10 shows an example of the end-to-end process of path placement using SR-TE and PCE. The process is as follows.

1. Service provisioning: The Operations Support System (OSS) layer provides the service provisioning data to the ingress router (PE1). This data includes service characteristics, the tunnel setup type and any constraints that should be applied to the tunnel to deliver the service requirements.
2. Path computation request: Having received the service layer characteristics, if a tunnel that meets the required service criteria does not exist, the ingress router functions as a PCC and sends a PCReq to the PCE server. In this example, the only path constraint is a request for a bandwidth of 10 Mb/s (requested in the service characteristics received from the OSS layer).
3. Path computation reply: The PCE computes the path and returns a PCRep with a positive response. The PCRep includes the SR-ERO, which contains a segment list of the calculated hops through the network.
4. LSP state report: The ingress router activates the LSP by composing a segment list and notifies the PCE server by sending a PCRpt message indicating that the LSP is up. The PCC also delegates control of the LSP to the PCE server. In the Figure 10 example, the PCE server is an active stateful PCE server.
5. The PCC continues to monitor network state, including this LSP and all other LSPs. The PCE server updates and optimizes the LSP path placement as required.

Figure 10. SR-TE and PCE example



In the example shown in Figure 10, the ingress router, PE1, functions as the PCC. The OSS layer provides the ingress router with the service characteristics, which triggers the ingress router to query the PCE server for a path calculation. This is not, however, the only possible architecture. It is equally feasible for the OSS layer to function as a PCC and query the PCE server directly. After the OSS layer has a computed path returned from the PCE, the path can be passed to the ingress router together with the service provisioning characteristics.

Disjointness and path profiles

As previously described, the use of Anycast-SIDs can provide a loose form of disjointness in dual-plane topologies. However, the disjointness cannot be guaranteed if the calculating entity simply follows the shortest IGP path to the Anycast-SID. To guarantee disjointness between Service M and Service N, some extensions are required to the PCEP protocol. These extensions are called path-profiles and are defined in the IETF draft *PCE Path Profiles*¹⁴.

A path profile represents a list of path parameters or policies that a PCEP speaker may present to a peer to influence path computation. They essentially allow for bundling of commonly used path characteristics into a profile/template that is referenced using a profile identifier. The method through which path-profile attributes are made known to the PCC/PCE peers is out-of-band and not within the scope of the path-profile extensions.

Path profiles are carried in a new object called the Path-Profile object. The Path-Profile object contains one or more Path-Profile-ID-TLVs, each referencing a Path-Profile ID, together with an optional Extended ID field. The Path-Profile ID indicates the algorithm or metric(s) used to calculate the path placement.

The IETF draft makes no assertions about how the Extended-ID field is used, but a common anticipated usage is that the optional Extended-ID will be used to indicate an argument for the algorithm specified in the Path Profile ID to reduce the overall number of required path-profiles. For example:

- Profile 1: {default_igp_cost(igp metric); bandwidth(B)}
- Profile 2: {required_igp_cost(isis-te-metric, one-way latency); latency(< = L)}

By signaling a Path-Profile object with Path-Profile-TLVs containing {Path-Profile ID = 1, Extended ID = 10000000} and {Path-Profile ID = 2, Extended ID = 200}, the path computation would be for a path of 10 Mb/s bandwidth routed using IS-IS TE metrics with a one-way latency threshold of below 200 mSec.

To guarantee disjointness, consider a path profile as follows:

- Profile 3: {disjoint_to(GroupID); default_igp_cost(min cost)}

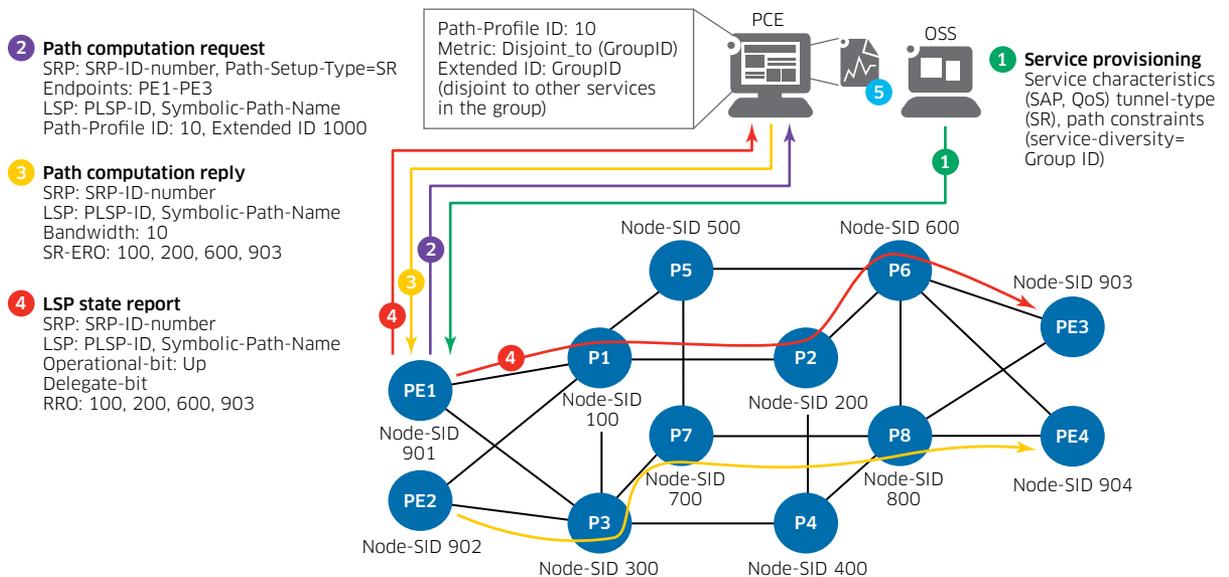
By signaling a Path-Profile object with a Path-Profile-TLV containing {Path-Profile ID = 3, Extended ID = 5000}, the path computation request is specifying that any computed path be completely disjoint from group 5000. Because the PCE is a centralized or hierarchically centralized function with knowledge of all computed paths, it is possible to ensure that the requested path computation is disjoint (providing that a disjoint path exists). A similar method can be used to achieve bi-directionality of a service.

¹⁴ IETF, Internet Draft *PCE Path Profiles*, October 22, 2013. <https://tools.ietf.org/html/draft-alvarez-pce-path-profiles-00>

Figure 11 shows an example of the end-to-end process of path placement using path profiles to guarantee disjointness. The process is as follows.

1. Service provisioning: The OSS layer provides the service provisioning characteristics to the ingress PE. (For clarity, only PE1 is shown, but the same service characteristics would also be provided to the other ingress PEs). The service provisioning characteristics include the tunnel setup type and an indication of the group-ID to which this service belongs.
2. Path computation request: Having received the service layer characteristics, if a tunnel that meets the required service criteria does not exist, the ingress router functions as a PCC and sends a PCReq to the PCE server. In this example, the PCReq includes a path-profile of {Path-Profile ID = 10, Extended ID = 1000}.
3. Path computation reply: The PCE computes the path and returns a PCRep with a positive response. If Service M from PE1-PE3 and Service N from PE2-PE4 both signal a path-profile of {Path-Profile ID = 10, Extended ID = 1000}, the PCE ensures that their respective paths are disjoint from each other. The PCRep includes the SR-ERO, which contains a segment list of the calculated hops through the network.
4. LSP state report: The ingress router instantiates the LSP by imposing a segment list and notifies the PCE server by sending a PCRpt message indicating that the LSP is up. The PCC also delegates control of the LSP to the PCE server. In the Figure 11 example, the PCE server is an Active Stateful PCE Server.
5. The PCC continues to monitor the network state, including this LSP and all other LSPs. The PCE server updates and optimizes the LSP path placement as required.

Figure 11. Disjointness with SR-TE, PCE and path profiles



For PCC-initiated LSPs, the PCC may include a Path-Profile when sending a PCReq message. For PCE-initiated LSPs, a PCE may include a Path-Profile object when sending a PCInit message. If multiple Path-Profile-TLVs are present, the PCC/PCE must process the identifiers in the Path-Profile object in the order in which they are received.

Topology discovery

Overview

To make informed decisions about where to place paths, the PCE needs to understand the network topology. In a multi-area network, this requires learning the topology of each area and correlating it to compute an optimal path from end to end.

To understand the network topology, a PCE (or one or more devices at the front end of the PCE) can passively tap into the IGP and listen to area-wide/level-wide IGP updates. This approach is suitable for smaller networks with a single area/level or a small number of areas/levels. However, if the network has a large number of areas/levels, this approach can present some challenges, both the logistical challenge of connecting separately to each area/level and also the requirement to scale sufficiently to correlate and manage the Link-State Databases from each of the areas/levels.

Link state information

For a network with a large number of areas or levels, link state information distribution using Border Gateway Protocol (BGP-LS) provides a mechanism by which link state and traffic engineering information can be collected from networks and shared with external devices through BGP. This is achieved using new Link State Network Layer Reachability Information (NLRI) that allows for advertisement of VPN and non-VPN link, node and IPv4/IPv6 topology prefix information.

The IETF draft *North-Bound Distribution of Link-State and TE Information using BGP*¹⁵ defines four types of link-state NLRI:

- Node NLRI
- Link NLRI
- IPv4 Topology Prefix NLRI
- IPv6 Topology Prefix NLRI

A new BGP attribute, the BGP-LS attribute, is an optional non-transitive attribute used to carry additional link/node/prefix parameters and attributes. The BGP-LS attribute can be included with any link-state NLRI.

Table 3. Link-state NLRI

TYPE OF LINK-STATE NLRI	DESCRIPTION	BGP-LS ATTRIBUTE
Node	The Node NLRI consists of a Local Node Descriptor field with a number of Sub-TLVs containing information such as the Local Node Descriptor, AS number, BGP-LS Identifier, OSPF Area-ID and IGP Router ID.	Contains a number of Node Attribute TLVs that can be included with a Node NLRI. These include IS-IS Area Identifier TLVs to indicate all areas that the router belongs to, a Node Name TLV, a local IPv4/IPv6 Router-ID TLV used to indicate any auxiliary Router-IDs that the IGP might be using and a Node Flag Bits TLV to indicate information such as Overload-bit, Attached-bit or External-bit.

¹⁵ IETF, Internet Draft *North-Bound Distribution of Link-State and TE Information using BGP*, September 21, 2011. <https://tools.ietf.org/html/draft-gredler-idr-ls-distribution-00>

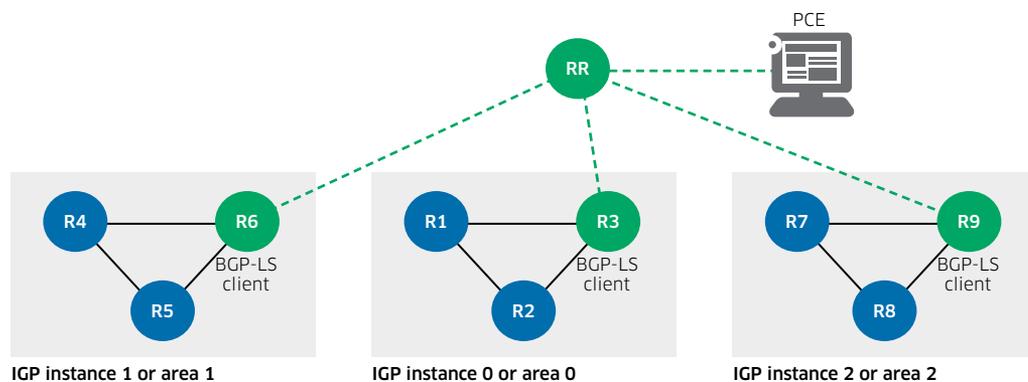
TYPE OF LINK-STATE NLRI	DESCRIPTION	BGP-LS ATTRIBUTE
Link	The Link NLRI consists of a Local and Remote Node Descriptor field and a Link Descriptors field for each routing protocol that it runs over a given link. The Local/Remote Node Descriptor fields contain the same Sub-TLVs as the Node NLRI for each end of the advertised link. The Link Descriptors field contains a number of TLVs describing the link, such as IPv4/IPv6 interface address, IPv4/IPv6 neighbour address and link local/remote identifiers.	Contains a number of Link Attribute TLVs that can be included with a Link NLRI. These include the IPv4/IPv6 Local/Remote Router-ID and TLVs to indicate traffic engineering information such as maximum reservable link bandwidth, unreserved bandwidth, IGP Metric, TE Default Metric and Shared Risk Link Group.
IPv4/IPv6 Prefix	The IPv4/IPv6 Topology Prefix NLRIs follow the same format as Node and Link NLRIs and consist of a Local Node Descriptor field and a Prefix Descriptor field. The Local Node Descriptor contains the same Sub-TLVs as the Node NLRI for each advertised prefix. The Prefix Descriptor field contains a number of TLVs uniquely identifying an IPv4/IPv6. The only mandatory Prefix Descriptor TLV is the IP Reachability Information TLV, which contains one prefix originally advertised to the IGP by a node. The IPv4/IPv6 Prefix NLRI may optionally contain a Multi-Topology Identifier TLV or an OSPF Route Type TLV.	Contains a number of Prefix Attribute TLVs that can be used when advertising IPv4/IPv6 Topology Prefix NLRIs. These include a Route Tag TLV to carry the original IGP Route Tag, a Prefix Metric TLV to carry the metric of the prefix as known in the IGP and an OSPF Forwarding Address TLV.

The IETF draft *BGP Link-State extensions for Segment Routing*¹⁶ extends BGP-LS to include additional BGP-LS attribute TLVs that allow for encoding of SR information such as Prefix-SID, Adjacency-SID, SID/Label binding and SR capabilities. With this information, it is possible for an external controller to collect the segment information and construct a segment stack (segment list) that could be imposed at an ingress PE to determine an egress PE and a given path through the network.

Figure 12 shows an example of how BGP-LS might be used for topology discovery by a centralized PCE server. In this example, each IGP area (instance) has a BGP-LS-capable BGP speaker peering with a Route-Reflector through which network topology information is advertised to a centralized PCE server.

While BGP-LS can be used to advertise network topology information, there may also be a requirement for the PCE server(s) and/or other OSS-layer entities to understand topology information at a service-level context. Therefore, as part of the network discovery process there may also be a requirement to run additional protocols such as NetConf/YANG and/or SNMP to obtain this level of information.

Figure 12. Using BGP-LS for topology discovery



¹⁶ IETF, Internet Draft *BGP Link-State extensions for Segment Routing*, February 14, 2014. <https://tools.ietf.org/html/draft-gredler-bgp-ls-segment-routing-extensions-00>

Egress Peer Engineering

The BGP-LS specification was further extended in the IETF draft *Segment Routing Egress Peer Engineering BGPLS Extensions*¹⁷ to encompass Egress Peer Engineering (EPE). Egress Peer Engineering enables an egress PE to advertise BGP exit point topology information such as its peers, interfaces and neighboring AS numbers together with some newly-defined segment identifiers, known generically as BGP peering segments or BGP peering SIDs:

- A Peer Node segment (PeerNode SID) for each of the egress PEs external peers
- A Peer Adjacency segment (PeerAdj SID) for each recursive interface to a multi-hop EBGP peer; two IP interfaces to an external AS with static routes over each interface to a multi-hop BGP peer would result in load-balancing across each IP interface
- A Peer Set segment (PeerSet SID) for each common set of peers, such as those belonging to the same peer Autonomous System

After an EPE-capable egress PE has advertised its BGP peering SIDs, it installs each SID with a next (pop) operation in the data plane but does not execute an IP route lookup for forwarded packets.

After the BGP peering SID information is learned by an external northbound controller (called an EPE Controller), it is possible for the EPE Controller to construct a segment stack for an ingress PE that determines the path through the network and the egress PE and also determines the specific external interface to use to reach a given destination.

At a minimum, the segment list imposed at an ingress PE requires two segments: the Node-SID of the egress PE followed by the BGP Peering segment for the chosen egress PE peer or peering interface.

There is also a requirement at the ingress PE to implement dynamic mapping of traffic flows to the relevant SR tunnels (with the segment lists reflecting the required egress peering point). Mapping is implemented using a form of Policy Based Routing (PBR) and is driven by some form of centralized SDN controller using a protocol such as NetConf/YANG, BGP Flowspec or OpenFlow.

Using EPE to traffic engineer at egress peering points can be either coarse or granular. An example of coarse traffic engineering is 80 percent of traffic to Peer M and 20 percent of traffic to Peer N. An example of granular traffic engineering is a series of prefix/length entries to varying peering SIDs at varying egress PEs. Whether traffic engineering is coarse or granular, EPE provides a powerful toolset for ISPs to traffic engineer their external interfaces.

¹⁷ IETF, Internet Draft *Segment Routing Egress Peer Engineering BGPLS Extensions*, March 9, 2015. <https://tools.ietf.org/id/draft-previdi-idr-bgpls-segment-routing-epe-02.txt>

Conclusion

Segment Routing (SR) is emerging as a new tunneling protocol with attributes that are desirable to network operators. Unlike LDP and RSVP, it requires no MPLS control plane machinery and imposes no changes to the MPLS data plane. Segment Routing also enables source routing without requiring any midpoint or tail-end state. The only state required is held at the ingress PE, which makes SR significantly more scalable than RSVP-TE.

The ability to source route without scaling concerns has re-energized the PCE architecture, which had previously focused purely on RSVP-TE LSP placement. With the relevant extensions to the PCE protocol, SR coupled with PCE can provide multiple benefits to network operators:

- It provides the ability to optimize LSP path placement across multi-area/multi-level networks. Without PCE, the head-end router calculating the path cannot see the end-to-end topology.
- By monitoring network utilization and optimizing LSPs where appropriate, it can ensure that expensive network capacity is effectively utilized.
- By employing a PCE hierarchy that has network-wide visibility of all calculated paths, it enables delivery of performance-engineered paths at the service level, including characteristics such as disjointness and bi-directionality. This delivery could potentially have been achieved previously using RSVP-TE but not in large multi-area networks and not with scalability.
- By allowing the PCE to dynamically trigger the creation and teardown of LSPs based on application demand without the need for pre-configuration on the PCC, SR with PCE paves the way for a more agile WAN-based SDN architecture in which intelligence in the SDN controller determines where and when to establish paths. This agility and flexibility enable new products to be introduced such as bandwidth calendaring and bandwidth on demand.

Although SR (in conjunction with ELI/EL) has the capability to impose a deep label stack, which may present hardware challenges for existing routers, it is anticipated that most (if not all) requirements will be met using loose (shortest path) hops and/or a series of loose hops combined with a restricted number of strict (Node-SID, Adj-SID) hops.

In summary, SR with PCE has the potential to optimize existing network infrastructure, increase the options available to existing services and potentially create new service offerings.

Alcatel-Lucent is a market leader and technology innovator, with years of experience developing world-class MPLS platforms and comprehensive end-to-end MPLS managed solutions. Alcatel-Lucent IP/MPLS-based service routing and switching products offer network operators the flexibility, scalability and feature sets required by next-generation applications.

Acronyms

Adj-SID	Adjacency-SID	PCInit	Path Computation Initiate
BGP-LS	Border Gateway Protocol - Link State	PCRep	Path Computation Reply
CSPF	Constrained Shortest Path First	PCReq	Path Computation Request
ECMP	Equal Cost Multi-Path	PCRpt	Path Computation Report
EL	Entropy Label	PE	provider edge
ELC	Entropy Label Capability	PHP	Penultimate Hop Popping
ELI	Entropy Label Indicator	PLSP-ID	PCEP-specific LSP Identifier
E-Line	Ethernet-Line	RSVP	Resource Reservation Protocol
EPE	Egress Peer Engineering	RSVP-TE	RSVP - Traffic Engineering
ERO	Explicit Route Object	SDH	Synchronous Digital Hierarchy
IGP	Interior Gateway Protocol	SDN	Software Defined Networking
ILM	Incoming Label Map	SID	Segment Identifier
IPFIX	IP Flow Information Export	SNMP	Simple Network Management Protocol
IS-IS	Intermediate System-Intermediate System	SPF	Shortest Path First
LDP	Label Distribution Protocol	SR	Segment Routing
LFIB	Label Forwarding Information Base	SRGB	Segment Routing Global Block
LSA	Link State Advertisements	SR-ERO	Segment Routing Explicit Route Object
LSP	Label Switched Path	SRLG	Shared Risk Link Group
LSR	Label Switched Router	SR-TE	Service Routing - Traffic Engineering
MPLS	Multi-Protocol Label Switching	TCP	Transmission Control Protocol
NHLFE	Next-Hop Label Forwarding Entry	TED	Traffic Engineering Database
NHRI	Next-Hop Label Forwarding Entry	TLV	Type Length Value
NLRI	Network Layer Reachability Information	VPN	Virtual Private Network
OSPF	Open Shortest Path First	VPRN	Virtual Private Routed Network
OSS	Operations Support System	VPWS	Virtual Private Wire Service
PCC	Path Computation Client	WAN	Wide Area Network
PCE	Path Computation Element	XML	EXtensible Markup Language
PCEP	PCE Protocol	YANG	Yet Another Next Generation (data modeling language)

References

1. IETF. Internet Draft: *BGP Link-State extensions for Segment Routing*. February 14, 2014. <https://tools.ietf.org/html/draft-gredler-bgp-ls-segment-routing-extensions-00>
2. IETF. Internet Draft: *Entropy labels for source routed stacked tunnels*. February 14, 2014. <https://tools.ietf.org/html/draft-kini-mpls-spring-entropy-label-00>
3. IETF. Internet Draft: *North-Bound Distribution of Link-State and TE Information using BGP*. September 21, 2011. <https://tools.ietf.org/html/draft-gredler-idr-ls-distribution-00>
4. IETF. Internet Draft: *PCE Path Profiles*. October 22, 2013. <https://tools.ietf.org/html/draft-alvarez-pce-path-profiles-00>

5. IETF. Internet Draft: *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*. December 2, 2013. <https://tools.ietf.org/html/draft-ietf-pce-pce-initiated-lsp-00>
6. IETF. Internet Draft: *PCEP Extensions for Segment Routing*. October 26, 2014. <https://tools.ietf.org/html/draft-ietf-pce-segment-routing-00>
7. IETF. Internet Draft: *PCEP Extensions for Stateful PCE*. February 28, 2012. <https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-00>
8. IETF. Internet Draft: *Segment Routing Architecture*. June 6, 2014. <https://tools.ietf.org/html/draft-filsfils-spring-segment-routing-03>
9. IETF. Internet Draft: *Segment Routing Egress Peer Engineering BGP Extensions*. March 9, 2015. <https://tools.ietf.org/id/draft-previdi-idr-bgp-segment-routing-epe-02.txt>
10. IETF. Internet Draft: *Segment Routing with MPLS data plane*. November 24, 2014. <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-mpls-00>
11. IETF. Internet Draft: *Signaling Entropy Label Capability Using Interior Gateway Protocols*. September 6, 2013. <https://tools.ietf.org/html/draft-xu-mpls-el-capability-signaling-igp-00>
12. IETF. RFC 4655: *A Path Computation Element (PCE)-Based Architecture*. August 2006. <https://tools.ietf.org/html/rfc4655>
13. IETF. RFC 5440: *Path Computation Element (PCE) Communication Protocol*. March 2009. <https://tools.ietf.org/html/rfc5440>
14. IETF. RFC 6391: *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*. November 2011. <https://tools.ietf.org/html/rfc6391>
15. IETF. RFC 6790: *The Use of Entropy Labels in MPLS Forwarding*. November 2012. <https://tools.ietf.org/html/rfc6790>