

Internet Engineering Task Force (IETF)
Request for Comments: 8413
Category: Informational
ISSN: 2070-1721

Y. Zhuang
Q. Wu
H. Chen
Huawei
A. Farrel
Juniper Networks
July 2018

Framework for Scheduled Use of Resources

Abstract

Time-Scheduled (TS) reservation of Traffic Engineering (TE) resources can be used to provide resource booking for TE Label Switched Paths so as to better guarantee services for customers and to improve the efficiency of network resource usage at any moment in time, including network usage that is planned for the future. This document provides a framework that describes and discusses the architecture for supporting scheduled reservation of TE resources. This document does not describe specific protocols or protocol extensions needed to realize this service.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are candidates for any level of Internet Standard; see [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/rfc8413>.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Problem Statement	4
2.1. Provisioning TE-LSPs and TE Resources	4
2.2. Selecting the Path of an LSP	4
2.3. Planning Future LSPs	5
2.4. Looking at Future Demands on TE Resources	6
2.4.1. Interaction between Time-Scheduled and Ad Hoc Reservations	6
2.5. Requisite State Information	7
3. Architectural Concepts	8
3.1. Where is Scheduling State Held?	8
3.2. What State is Held?	10
3.3. Enforcement of Operator Policy	12
4. Architecture Overview	13
4.1. Service Request	13
4.1.1. Reoptimization After TED Updates	14
4.2. Initialization and Recovery	15
4.3. Synchronization Between PCEs	15
5. Multi-domain Considerations	16
6. Security Considerations	18
7. IANA Considerations	19
8. Informative References	19
Acknowledgements	21
Contributors	21
Authors' Addresses	22

1. Introduction

Traffic Engineering Label Switched Paths (TE-LSPs) are connection-oriented tunnels in packet and non-packet networks [RFC3209] [RFC3945]. TE-LSPs may reserve network resources for use by the traffic they carry, thus providing some guarantees of service delivery and allowing a network operator to plan the use of the resources across the whole network.

In some technologies (such as wavelength switched optical networks) the resource is synonymous with the label that is switched on the path of the LSP so that it is not possible to establish an LSP that can carry traffic without assigning a physical resource to the LSP. In other technologies (such as packet switched networks), the resources assigned to an LSP are a measure of the capacity of a link that is dedicated for use by the traffic on the LSP.

In all cases, network planning consists of selecting paths for LSPs through the network so that there will be no contention for resources. LSP establishment is the act of setting up an LSP and reserving resources within the network. Network optimization or reoptimization is the process of repositioning LSPs in the network to make the unreserved network resources more useful for potential future LSPs while ensuring that the established LSPs continue to fulfill their objectives.

It is often the case that it is known that an LSP will be needed at some specific time in the future. While a path for that LSP could be computed using knowledge of the currently established LSPs and the currently available resources, this does not give any degree of certainty that the necessary resources will be available when it is time to set up the new LSP. Yet, setting up the LSP ahead of the time when it is needed (which would guarantee the availability of the resources) is wasteful since the network resources could be used for some other purpose in the meantime.

Similarly, it may be known that an LSP will no longer be needed after some future time and that it will be torn down, which will release the network resources that were assigned to it. This information can be helpful in planning how a future LSP is placed in the network.

Time-Scheduled (TS) reservation of TE resources can be used to provide resource booking for TE-LSPs so as to better guarantee services for customers and to improve the efficiency of network resource usage into the future. This document provides a framework that describes the problem and discusses the architecture for the

scheduled reservation of TE resources. This document does not describe specific protocols or protocol extensions needed to realize this service.

2. Problem Statement

2.1. Provisioning TE-LSPs and TE Resources

TE-LSPs in existing networks are provisioned using a variety of techniques. They may be set up using RSVP-TE as a signaling protocol [RFC3209] [RFC3473]. Alternatively, they could be established by direct control of network elements such as in the Software-Defined Networking (SDN) paradigm. They could also be provisioned using the PCE Communication Protocol (PCEP) [RFC5440] as a control protocol to communicate with the network elements.

TE resources are reserved at the point of use. That is, the resources (wavelengths, timeslots, bandwidth, etc.) are reserved for use on a specific link and are tracked by the Label Switching Routers (LSRs) at the end points of the link. Those LSRs learn which resources to reserve during the LSP setup process.

The use of TE resources can be varied by changing the parameters of the LSP that uses them, and the resources can be released by tearing down the LSP.

Resources that have been reserved in the network for use by one LSP may be preempted for use by another LSP. If RSVP-TE signaling is in use, a holding priority and a preemption priority are used to determine which LSPs may preempt the resources that are in use for which other LSPs. If direct (central) control is in use, the controller is able to make preemption decisions. In either case, operator policy forms a key part of preemption since there is a trade between disrupting existing LSPs and enabling new LSPs.

2.2. Selecting the Path of an LSP

Although TE-LSPs can determine their paths hop by hop using the shortest path toward the destination to route the signaling protocol messages [RFC3209], in practice this option is not applied because it does not look far enough ahead into the network to verify that the desired resources are available. Instead, the full length of the path of an LSP is usually computed ahead of time either by the head-end LSR of a signaled LSP or by Path Computation Element (PCE) functionality that is in a dedicated server or built into network management software [RFC4655].

Such full-path computation is applied in order that an end-to-end view of the available resources in the network can be used to determine the best likelihood of establishing a viable LSP that meets the service requirements. Even in this situation, however, it is possible that two LSPs being set up at the same time will compete for scarce network resources, which means that one or both of them will fail to be established. This situation is avoided by using a centralized PCE that is aware of the LSP setup requests that are in progress.

Path selection may make allowance for preemption as described in [Section 2.1](#). That is, when selecting a path, the decision may be made to choose a path that will result in the preemption of an existing LSP. The trade-off between selecting a less optimal path, failing to select any path at all, and preempting an existing LSP must be subject to operator policy.

Path computation is subject to "objective functions" that define what criteria are to be met when the LSP is placed [[RFC4655](#)]. These can be criteria that apply to the LSP itself (such as the shortest path to the destination) or to the network state after the LSP is set up (such as the maximized residual link bandwidth). The objective functions may be requested by the application requesting the LSP and may be filtered and enhanced by the computation engine according to operator policy.

2.3. Planning Future LSPs

LSPs may be established "on demand" when the requester determines that a new LSP is needed. In this case, the path of the LSP is computed as described in [Section 2.2](#).

However, in many situations, the requester knows in advance that an LSP will be needed at a particular time in the future. For example, the requester may be aware of a large traffic flow that will start at a well-known time, perhaps for a database synchronization or for the exchange of content between streaming sites. Furthermore, the requester may also know for how long the LSP is required before it can be torn down.

The set of requests for future LSPs could be collected and held in a central database (such as at a Network Management System (NMS)): when the time comes for each LSP to be set up, the NMS can ask the PCE to compute a path and can then request the LSP to be provisioned. This approach has a number of drawbacks because it is not possible to determine in advance whether it will be possible to deliver the LSP since the resources it needs might be used by other LSPs in the

network. Thus, at the time the requester asks for the future LSP, the NMS can only make a best-effort guarantee that the LSP will be set up at the desired time.

A better solution, therefore, is for the requests for future LSPs to be serviced at once. The paths of the LSPs can be computed ahead of time and converted into reservations of network resources during specific windows in the future. That is, while the path of the LSP is computed and the network resources are reserved, the LSP is not established in the network until the time for which it is scheduled.

There is a need to take into account items that need to be subject to operator policy, such as 1) the amount of capacity available for scheduling future reservations, 2) the operator preference for the measures that are used with respect to the use of scheduled resources during rapid changes in traffic demand events, or 3) a complex (multiple nodes/links) failure event so as to protect against network destabilization. Operator policy is discussed further in [Section 3.3](#).

2.4. Looking at Future Demands on TE Resources

While path computation, as described in [Section 2.2](#), takes account of the currently available network resources and can act to place LSPs in the network so that there is the best possibility of future LSPs being accommodated, it cannot handle all eventualities. It is simple to construct scenarios where LSPs that are placed one at a time lead to future LSPs being blocked, but where foreknowledge of all of the LSPs would have made it possible for them all to be set up.

If, therefore, we were able to know in advance what LSPs were going to be requested, we could plan for them and ensure resources were available. Furthermore, such an approach enables a commitment to be made to a service user that an LSP will be set up and available at a specific time.

A reservation service can be achieved by tracking the current use of network resources and also having a future view of the resource usage. We call this Time-Scheduled TE (TS-TE) resource reservation.

2.4.1. Interaction between Time-Scheduled and Ad Hoc Reservations

There will, of course, be a mixture of resource uses in a network. For example, normal unplanned LSPs may be requested alongside TS-TE LSPs. When an unplanned LSP is requested, no prior accommodation can be made to arrange resource availability, so the LSP can be placed no better than would be the case without TS-TE. However, the new LSP can be placed considering the future demands of TS-TE LSPs that have

already been requested. Of course, the unplanned LSP has no known end time and so any network planning must assume that it will consume resources forever.

2.5. Requisite State Information

In order to achieve the TS-TE resource reservation, the use of resources on the path needs to be scheduled. The scheduling state is used to indicate when resources are reserved and when they are available for use.

A simple information model for one piece of the scheduling state is as follows:

```
{
  link id;
  resource id or reserved capacity;
  reservation start time;
  reservation end time
}
```

The resource that is scheduled could be link capacity, physical resources on a link, buffers on an interface, etc., and could include advanced considerations such as CPU utilization and the availability of memory at nodes within the network. The resource-related information might also include the maximal unreserved bandwidth of the link over a time interval. That is, the intention is to book (reserve) a percentage of the residual (unreserved) bandwidth of the link. This could be used, for example, to reserve bandwidth for a particular class of traffic (such as IP) that doesn't have a provisioned LSP.

For any one resource, there could be multiple pieces of the scheduling state, and for any one link, the timing windows might overlap.

There are multiple ways to realize this information model and different ways to store the data. The resource state could be expressed as a start time and an end time (as shown above), or it could be expressed as a start time and a duration. Multiple reservation periods, possibly of different lengths, may need to be recorded for each resource. Furthermore, the current state of network reservation could be kept separate from the scheduled usage, or everything could be merged into a single TS database.

An application may make a reservation request for immediate resource usage or to book resources for future use so as to maximize the chance of services being delivered and to avoid contention for

resources in the future. A single reservation request may book resources for multiple periods and might request a reservation that repeats on a regular cycle.

A computation engine (that is, a PCE) may use the scheduling state information to help optimize the use of resources into the future and reduce contention or blocking when the resources are actually needed.

Note that it is also necessary to store the information about future LSPs as distinct from the specific resource scheduling. This information is held to allow the LSPs to be instantiated when they are due, and use the paths/resources that have been computed for them, and also to provide correlation with the TS-TE resource reservations so that it is clear why resources were reserved, thus allowing preemption and handling the release of reserved resources in the event of cancellation of future LSPs. See [Section 3.2](#) for further discussion of the distinction between scheduled resource state and scheduled LSP state.

Network performance factors (such as maximum link utilization and the residual capacity of the network), with respect to supporting scheduled reservations, need to be supported and are subject to operator policy.

3. Architectural Concepts

This section examines several important architectural concepts to understand the design decisions reached in this document to achieve TS-TE in a scalable and robust manner.

3.1. Where is Scheduling State Held?

The scheduling state information described in [Section 2.5](#) has to be held somewhere. There are two places where this makes sense:

- o in the network nodes where the resources exist; or,
- o in a central scheduling controller where decisions about resource allocation are made.

The first of these makes policing of resource allocation easier. It means that many points in the network can request immediate or scheduled LSPs with the associated resource reservation, and that all such requests can be correlated at the point where the resources are

allocated. However, this approach has some scaling and technical problems:

- o The most obvious issue is that each network node must retain the full time-based state for all of its resources. In a busy network with a high arrival rate of new LSPs and a low hold time for each LSP, this could be a lot of state. Network nodes are normally implemented with minimal spare memory.
- o In order that path computation can be performed, the computing entity normally known as a Path Computation Element (PCE) [RFC4655] needs access to a database of available links and nodes in the network (as well as the TE properties of said links). This database is known as the Traffic Engineering Database (TED) and is usually populated from information advertised in the IGP by each of the network nodes or exported using BGP Link State (BGP-LS) [RFC7752]. To be able to compute a path for a future LSP, the PCE needs to populate the TED with all of the future resource availability: if this information is held on the network nodes, it must also be advertised in the IGP. This could be a significant scaling issue for the IGP and the network nodes, as all of the advertised information is held at every network node and must be periodically refreshed by the IGP.
- o When a normal node restarts, it can recover the resource reservation state from the forwarding hardware, from Non-Volatile Random-Access Memory (NVRAM), or from adjacent nodes through the signaling protocol [RFC5063]. If the scheduling state is held at the network nodes, it must also be recovered after the restart of a network node. This cannot be achieved from the forwarding hardware because the reservation will not have been made, could require additional expensive NVRAM, or might require that all adjacent nodes also have the scheduling state in order to reinstall it on the restarting node. This is potentially complex processing with scaling and cost implications.

Conversely, if the scheduling state is held centrally, it is easily available at the point of use. That is, the PCE can utilize the state to plan future LSPs and can update that stored information with the scheduled reservation of resources for those future LSPs. This approach also has several issues:

- o If there are multiple controllers, then they must synchronize their stored scheduling state as they each plan future LSPs and they must have a mechanism to resolve resource contention. This is relatively simple and is mitigated by the fact that there is ample processing time to replan future LSPs in the case of resource contention.

- o If other sources of immediate LSPs are allowed (for example, other controllers or autonomous action by head-end LSRs), then the changes in resource availability caused by the setup or tear down of these LSPs must be reflected in the TED (by use of the IGP as is already normally done) and may have an impact on planned future LSPs. This impact can be mitigated by replanning future LSPs or through LSP preemption.
- o If the scheduling state is held centrally at a PCE, the state must be held and restored after a system restart. This is relatively easy to achieve on a central server that can have access to non-volatile storage. The PCE could also synchronize the scheduling state with other PCEs after restart. See [Section 4.2](#) for details.
- o Of course, a centralized system must store information about all of the resources in the network. In a busy network with a high arrival rate of new LSPs and a low hold time for each LSP, this could be a lot of state. This is multiplied by the size of the network measured both by the number of links and nodes and by the number of trackable resources on each link or at each node. This challenge may be mitigated by the centralized server being dedicated hardware, but there remains the problem of collecting the information from the network in a timely way when there is potentially a very large amount of information to be collected and when the rate of change of that information is high. This latter challenge is only solved if the central server has full control of the booking of resources and the establishment of new LSPs so that the information from the network only serves to confirm what the central server expected.

Thus, considering these trade-offs, the architectural conclusion is that the scheduling state should be held centrally at the point of use and not in the network devices.

3.2. What State is Held?

As already described, the PCE needs access to an enhanced, time-based TED. It stores the Traffic Engineering (TE) information, such as bandwidth, for every link for a series of time intervals. There are a few ways to store the TE information in the TED. For example, suppose that the amount of the unreserved bandwidth at a priority level for a link is B_j in a time interval from time T_j to T_k ($k = j+1$), where $j = 0, 1, 2, \dots$.

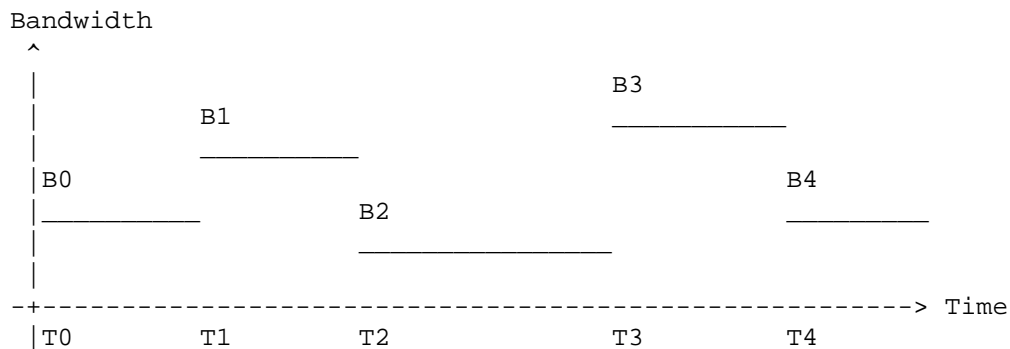


Figure 1: A Plot of Bandwidth Usage against Time

The unreserved bandwidth for the link can be represented and stored in the TED as $[T0, B0]$, $[T1, B1]$, $[T2, B2]$, $[T3, B3]$, ... as shown in Figure 1.

But it must be noted that service requests for future LSPs are known in terms of the LSPs whose paths are computed and for which resources are scheduled. For example, if the requester of a future LSP decides to cancel the request or to modify the request, the PCE must be able to map this to the resources that were reserved. When the LSP (or the request for the LSP with a number of time intervals) is canceled, the PCE must release the resources that were reserved on each of the links along the path of the LSP in every time interval from the TED. If the bandwidth that had been reserved for the LSP on a link was B from time $T2$ to $T3$ and the unreserved bandwidth on the link was $B2$ from $T2$ to $T3$, then B is added back to the link for the time interval from $T2$ to $T3$ and the unreserved bandwidth on the link from $T2$ to $T3$ will be seen to be $B2 + B$.

This suggests that the PCE needs an LSP Database (LSP-DB) [RFC8231] that contains information not only about LSPs that are active in the network but also those that are planned. For each time interval that applies to the LSP, the information for an LSP stored in the LSP-DB includes: the time interval, the paths computed for the LSP satisfying the constraints in the time interval, and the resources (such as bandwidth) reserved for the LSP in the time interval. See also Section 2.3

It is an implementation choice how the TED and LSP-DB are stored both for dynamic use and for recovery after failure or restart, but it may be noted that all of the information in the scheduled TED can be recovered from the active network state and from the scheduled LSP-DB.

3.3. Enforcement of Operator Policy

Computation requests for LSPs are serviced according to operator policy. For example, a PCE may refuse a computation request because the application making the request does not have sufficient permissions or because servicing the request might take specific resource usage over a given threshold.

Furthermore, the preemption and holding priorities of any particular computation request may be subject to the operator's policies. The request could be rejected if it does not conform to the operator's policies, or (possibly more likely) the priorities could be set/overwritten according to the operator's policies.

Additionally, the Objective Functions (OFs) of computation request (such as maximizing residual bandwidth) are also subject to operator policies. It is highly likely that the choice of OFs is not available to an application and is selected by the PCE or management system subject to operator policies and knowledge of the application.

None of these statements is new to scheduled resources. They apply to stateless, stateful, passive, and active PCEs, and they continue to apply to scheduling of resources.

An operator may choose to configure special behavior for a PCE that handles resource scheduling. For example, an operator might want only a certain percentage of any resource to be bookable. And an operator might want the preemption of booked resources to be an inverse function of how far in the future the resources are needed for the first time.

It is a general assumption about the architecture described in [Section 4](#) that a PCE is under the operational control of the operator that owns the resources that the PCE manipulates. Thus, the operator may configure any amount of (potentially complex) policy at the PCE. This configuration would also include policy points surrounding reoptimization of existing and planned LSPs in the event of changes in the current and future (planned) resource availability.

The granularity of the timing window offered to an application will depend on an operator's policy as well as the implementation in the PCE and goes to define the operator's service offerings. Different granularities and different lengths of prebooking may be offered to different applications.

4. Architecture Overview

The architectural considerations and conclusions described in the previous section lead to the architecture described in this section and illustrated in Figure 2. The interfaces and interactions shown in the figure and labeled (a) through (f) are described in [Section 4.1](#).

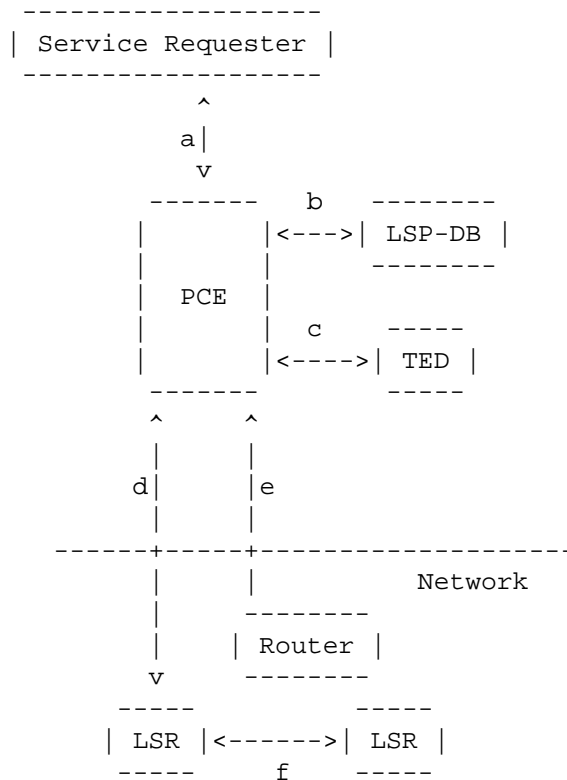


Figure 2: Reference Architecture for Scheduled Use of Resources

4.1. Service Request

As shown in Figure 2, some component in the network requests a service. This may be an application, an NMS, an LSR, or any component that qualifies as a Path Computation Client (PCC). We show this on the figure as the "Service Requester", and it sends a request to the PCE for an LSP to be set up at some time (either now or in the future). The request, indicated on Figure 2 by the arrow (a), includes all of the parameters of the LSP that the requester wishes to supply, such as priority, bandwidth, start time, and end time. Note that the requester in this case may be the LSR shown in the figure or may be a distinct system.

The PCE enters the LSP request in its LSP-DB (b) and uses information from its TED (c) to compute a path that satisfies the constraints (such as bandwidth) for the LSP in the time interval from the start time to the end time. It updates the future resource availability in the TED so that further path computations can take account of the scheduled resource usage. It stores the path for the LSP into the LSP-DB (b).

When it is time (i.e., at the start time) for the LSP to be set up, the PCE sends a PCEP Initiate request to the head-end LSR (d), which provides the path to be signaled as well as other parameters, such as the bandwidth of the LSP.

As the LSP is signaled between LSRs (f), the use of resources in the network is updated and distributed using the IGP. This information is shared with the PCE either through the IGP or using BGP-LS (e), and the PCE updates the information stored in its TED (c).

After the LSP is set up, the head-end LSR sends a PCEP LSP State Report (PCRpt) message to the PCE (d). The report contains the resources, such as bandwidth usage, for the LSP. The PCE updates the status of the LSP in the LSP-DB according to the report.

When an LSP is no longer required (either because the Service Requester has canceled the request or because the LSP's scheduled lifetime has expired), the PCE can remove it. If the LSP is currently active, the PCE instructs the head-end LSR to tear it down (d), and the network resource usage will be updated by the IGP and advertised back to the PCE through the IGP or BGP-LS (e). Once the LSP is no longer active, the PCE can remove it from the LSP-DB (b).

4.1.1.1. Reoptimization After TED Updates

When the TED is updated as indicated in [Section 4.1](#), depending on operator policy (so as to minimize network perturbations), the PCE may perform reoptimization of the LSPs for which it has computed paths. These LSPs may be already provisioned, in which case the PCE issues PCEP Update request messages for the LSPs that should be adjusted. Additionally, the LSPs being reoptimized may be scheduled LSPs that have not yet been provisioned, in which case reoptimization involves updating the store of scheduled LSPs and resources.

In all cases, the purpose of reoptimization is to take account of the resource usage and availability in the network and to compute paths for the current and future LSPs that best satisfy the objectives of those LSPs while keeping the network as clear as possible to support further LSPs. Since reoptimization may perturb established LSPs, it

is subject to operator oversight and policy. As the stability of the network will be impacted by frequent changes, the extent and impact of any reoptimization needs to be subject to operator policy.

Additionally, the status of the reserved resources (alarms) can enhance the computation and planning for future LSPs and may influence repair and reoptimization. Control of recalculations based on failures and notifications to the operator is also subject to policy.

See [Section 3.3](#) for further discussion of operator policy.

4.2. Initialization and Recovery

When a PCE in the architecture shown in Figure 2 is initialized, it must learn the state from the network, from its stored databases, and potentially from other PCEs in the network.

The first step is to get an accurate view of the topology and resource availability in the network. This would normally involve reading the state directly from the network via the IGP or BGP-LS (e), but it might include receiving a copy of the TED from another PCE. Note that a TED stored from a previous instantiation of the PCE is unlikely to be valid.

Next, the PCE must construct a time-based TED to show scheduled resource usage. How it does this is implementation specific, and this document does not dictate any particular mechanism: it may recover a time-based TED previously saved to non-volatile storage, or it may reconstruct the time-based TED from information retrieved from the LSP-DB previously saved to non-volatile storage. If there is more than one PCE active in the network, the recovering PCE will need to synchronize the LSP-DB and time-based TED with other PCEs (see [Section 4.3](#)).

Note that the stored LSP-DB needs to include the intended state and actual state of the LSPs so that when a PCE recovers, it is able to determine what actions are necessary.

4.3. Synchronization Between PCEs

If there is active in the network more than one PCE that supports scheduling, it is important to achieve some consistency between the scheduled TED and scheduled LSP-DB held by the PCEs.

[RFC7399] answers various questions around synchronization between the PCEs. It should be noted that the time-based "scheduled" information adds another dimension to the issue of synchronization

between PCEs. It should also be noted that a deployment may use a primary PCE and then have other PCEs as backup, where a backup PCE can take over only in the event of a failure of the primary PCE. Alternatively, the PCEs may share the load at all times. The choice of the synchronization technique is largely dependent on the deployment of PCEs in the network.

One option for ensuring that multiple PCEs use the same scheduled information is simply to have the PCEs driven from the same shared database, but it is likely to be inefficient, and interoperation between multiple implementations will be harder.

Another option is for each PCE to be responsible for its own scheduled database and to utilize some distributed database synchronization mechanism to have consistent information. Depending on the implementation, this could be efficient, but interoperation between heterogeneous implementations is still hard.

A further approach is to utilize PCEP messages to synchronize the scheduled state between PCEs. This approach would work well if the number of PCEs that support scheduling is small, but as the number increases, considerable message exchange needs to happen to keep the scheduled databases synchronized. Future solutions could also utilize some synchronization optimization techniques for efficiency. Another variation would be to request information from other PCEs for a particular time slice, but this might have an impact on the optimization algorithm.

5. Multi-domain Considerations

Multi-domain path computation usually requires some form of cooperation between PCEs, each of which has responsibility for determining a segment of the end-to-end path in the domain for which it has computational responsibility. When computing a scheduled path, resources need to be booked in all of the domains that the path will cross so that they are available when the LSP is finally signaled.

Per-domain path computation [RFC5152] is not an appropriate mechanism when a scheduled LSP is being computed because the computation requests at downstream PCEs are only triggered by signaling. However, a similar mechanism could be used where cooperating PCEs exchange Path Computation Request (PCReq) messages for a scheduled LSP, as shown in Figure 3. In this case, the service requester asks for a scheduled LSP that will span two domains (a). PCE1 computes a path across Domain 1 and reserves the resources and also asks PCE2 to compute and reserve in Domain 2 (b). PCE2 may return a full path or could return a path key [RFC5520]. When it is time for LSP setup,

PCE1 triggers the head-end LSR (c), and the LSP is signaled (d). If a path key is used, the entry LSR in Domain 2 will consult PCE2 for the path expansion (e) before completing signaling (f).

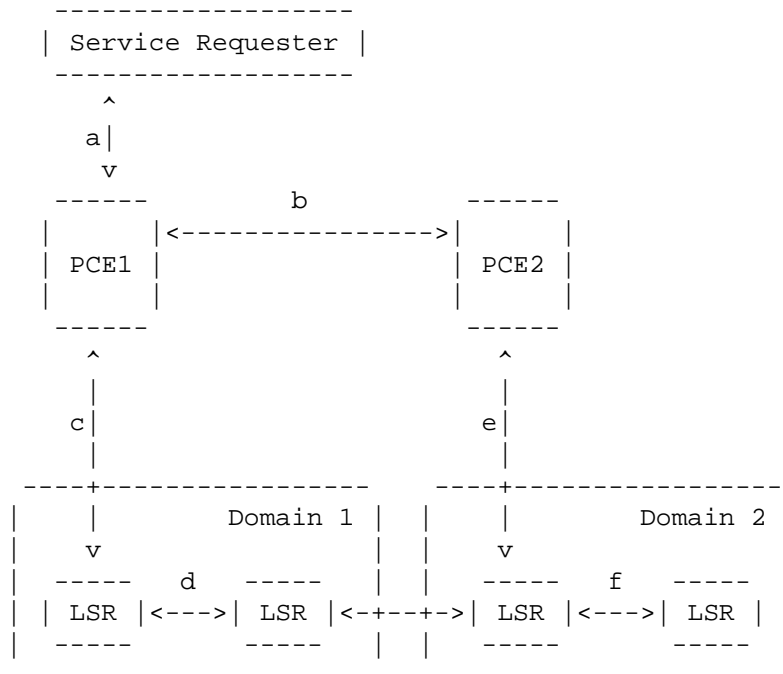


Figure 3: Per-Domain Path Computation for Scheduled LSPs

Another mechanism for PCE cooperation in multi-domain LSP setup is Backward Recursive PCE-Based Computation (BRPC) [RFC5441]. This approach relies on the downstream domain to supply a variety of potential paths to the upstream domain. Although BRPC can arrive at a more optimal end-to-end path than per-domain path computation, it is not well suited to LSP scheduling because the downstream PCE would need to reserve resources on all of the potential paths and then release those that the upstream PCE announced it did not plan to use.

Finally, we should consider hierarchical PCE (H-PCE) [RFC6805]. This mode of operation is similar to that shown in Figure 3, but a parent PCE is used to coordinate the requests to the child PCEs, which then results in better visibility of the end-to-end path and better coordination of the resource booking. The sequenced flow of control is shown in Figure 4.

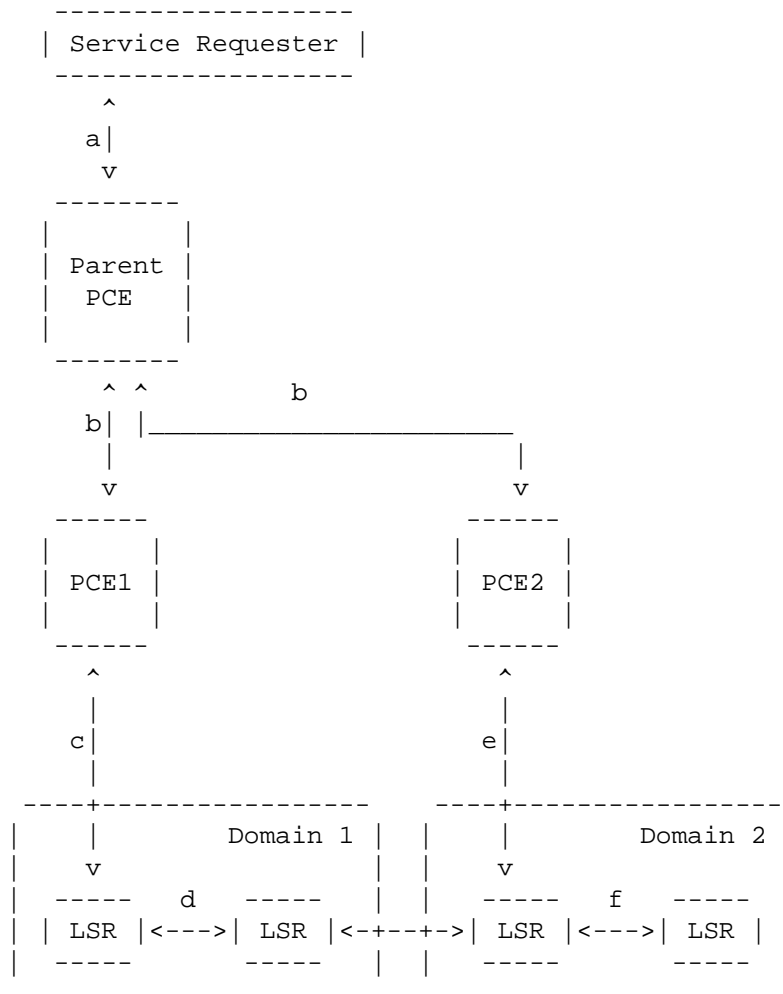


Figure 4: Hierarchical PCE for Path Computation for Scheduled LSPs

6. Security Considerations

The protocol implications of scheduled resources are unchanged from "on demand" LSP computation and setup. A discussion of securing PCEP is found in [RFC5440], and work to extend that security is provided in [RFC8253]. Furthermore, the path key mechanism described in [RFC5520] can be used to enhance privacy and security.

Similarly, there is no change to the security implications for the signaling of scheduled LSPs. A discussion of the security of the signaling protocols that would be used is found in [RFC5920].

However, the use of scheduled LSPs extends the attack surface for a PCE-enabled TE system by providing a larger (logically infinite) window during which an attack can be initiated or planned. That is, if bogus scheduled LSPs can be requested and entered into the LSP-DB, then a large number of LSPs could be launched and significant network resources could be blocked. Control of scheduling requests needs to be subject to operator policy, and additional authorization needs to be applied for access to LSP scheduling. Diagnostic tools need to be provided to inspect the LSP-DB to spot attacks.

7. IANA Considerations

This document has no IANA actions.

8. Informative References

- [AUTOBW] Yong, L. and Y. Lee, "ASON/GMPLS Extension for Reservation and Time Based Automatic Bandwidth Service", Work in Progress, [draft-yong-ccamp-ason-gmpls-autobw-service-00](#), October 2006.
- [DRAGON] National Science Foundation, "The DRAGON Project: Dynamic Resource Allocation via GMPLS Optical Networks", Overview and Status Presentation at ONT3, September 2006, <http://www.maxgigapop.net/wp-content/uploads/The-DRAGON-Project.pdf>.
- [FRAMEWORK-TTS] Chen, H., Toy, M., Liu, L., and K. Pithewan, "Framework for Temporal Tunnel Services", Work In Progress, [draft-chen-teas-frmwk-tts-01](#), March 2016.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", [RFC 3473](#), DOI 10.17487/RFC3473, January 2003, <https://www.rfc-editor.org/info/rfc3473>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", [RFC 3945](#), DOI 10.17487/RFC3945, October 2004, <https://www.rfc-editor.org/info/rfc3945>.

- [RFC4655] Farrel, A., Vasseur, J., 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>>.
- [RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <<https://www.rfc-editor.org/info/rfc5063>>.
- [RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <<https://www.rfc-editor.org/info/rfc5152>>.
- [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>>.
- [RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <<https://www.rfc-editor.org/info/rfc5441>>.
- [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>>.
- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<https://www.rfc-editor.org/info/rfc5920>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <<https://www.rfc-editor.org/info/rfc7399>>.

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", [RFC 7752](#), DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [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>>.
- [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>>.

Acknowledgements

This work has benefited from the discussions of resource scheduling over the years. In particular, the DRAGON project [[DRAGON](#)] and [[AUTOBW](#)], both of which provide approaches to auto-bandwidth services in GMPLS networks.

Mehmet Toy, Lei Liu, and Khuzema Pithewan contributed to an earlier version of [[FRAMEWORK-TTS](#)]. We would like to thank the authors of that document on Temporal Tunnel Services for material that assisted in thinking about this document.

Thanks to Michael Scharf and Daniele Ceccarelli for useful comments on this work.

Jonathan Hardwick provided a helpful Routing Directorate review.

Deborah Brungard, Mirja Kuehlewind, and Benjamin Kaduk suggested many changes during their Area Director reviews.

Contributors

The following person contributed to discussions that led to the development of this document:

Dhruv Dhody
Email: dhruv.dhody@huawei.com

Authors' Addresses

Yan Zhuang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhuangyan.zhuang@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Huaimo Chen
Huawei
Boston, MA
United States of America

Email: huaimo.chen@huawei.com

Adrian Farrel
Juniper Networks

Email: afarrel@juniper.net