Network Working Group
Request for Comments: 5671
Category: Informational

S. Yasukawa
NTT
A. Farrel, Ed.
Old Dog Consulting

October 2009

Applicability of the Path Computation Element (PCE) to Point-to-Multipoint (P2MP) MPLS and GMPLS Traffic Engineering (TE)

Abstract

The Path Computation Element (PCE) provides path computation functions in support of traffic engineering in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks.

Extensions to the MPLS and GMPLS signaling and routing protocols have been made in support of point-to-multipoint (P2MP) Traffic Engineered (TE) Label Switched Paths (LSPs).

This document examines the applicability of PCE to path computation for P2MP TE LSPs in MPLS and GMPLS networks. It describes the motivation for using a PCE to compute these paths and examines which of the PCE architectural models are appropriate.

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Table of Contents

1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph and of applying computational constraints. The intention is that the PCE is used to compute the path of Traffic Engineered Label Switched Paths (TE LSPs) within Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks.

[RFC4655] defines various deployment models that place PCEs differently within the network. The PCEs may be collocated with the Label Switching Routers (LSRs), may be part of the management system that requests the LSPs to be established, or may be positioned as one or more computation servers within the network.

Requirements for point-to-multipoint (P2MP) MPLS TE LSPs are documented in [RFC4461], and signaling protocol extensions for setting up P2MP MPLS TE LSPs are defined in [RFC4875]. In this

document, P2MP MPLS TE networks are considered in support of various features including layer 3 multicast VPNs [RFC4834], video distribution, etc.

Fundamental to the determination of the paths for P2MP LSPs within a network is the selection of branch points. Not only is this selection constrained by the network topology and available network resources, but it is determined by the objective functions that may be applied to path computation. For example, one standard objective is to minimize the total cost of the tree (that is, to minimize the sum of the costs of each link traversed by the tree) to produce what is known as a Steiner tree. Another common objective function requires that the cost to reach each leaf of the P2MP tree be minimized.

The selection of branch points within the network is further complicated by the fact that not all LSRs in the network are necessarily capable of performing branching functions. This information may be recorded in the Traffic Engineering Database (TED) that the PCE uses to perform its computations, and may have been distributed using extensions to the Interior Gateway Protocol (IGP) operating within the network [RFC5073].

Additionally, network policies may dictate specific branching behavior. For example, it may be decided that, for certain types of LSPs in certain types of networks, it is important that no branch LSR is responsible for handling more than a certain number of downstream branches for any one LSP. This might arise because the replication mechanism used at the LSRs is a round-robin copying process that delays the data transmission on each downstream branch by the time taken to replicate the data onto each previous downstream branch. Alternatively, administrative policies may dictate that replication should be concentrated on specific key replication nodes behaving like IP multicast rendezvous points (perhaps to ensure appropriate policing of receivers in the P2MP tree, or perhaps to make protection and resiliency easier to implement).

Path computation for P2MP TE LSPs presents a significant challenge because of the complexity of the computations described above. Determining disjoint protection paths for P2MP TE LSPs can add considerably to this complexity, while small modifications to a P2MP tree (such as adding or removing just one leaf) can completely change the optimal path. Reoptimization of a network containing multiple P2MP TE LSPs requires considerable computational resources. All of this means that an ingress LSR might not have sufficient processing power to perform the necessary computations, and even if it does, the act of path computation might interfere with the control and

management plane operation necessary to maintain existing LSPs. The PCE architecture offers a way to offload such path computations from LSRs.

2. Architectural Considerations

2.1. Offline Computation

Offline path computation is performed ahead of time, before the LSP setup is requested. That means that it is requested by, or performed as part of, a management application. This model can be seen in Section 5.5 of [RFC4655].

The offline model is particularly appropriate to long-lived LSPs (such as those present in a transport network) or for planned responses to network failures. In these scenarios, more planning is normally a feature of LSP provisioning.

This model may also be used where the network operator wishes to retain full manual control of the placement of LSPs, using the PCE only as a computation tool to assist the operator, not as part of an automated network.

Offline path computation may be applied as a background activity for network reoptimization to determine whether and when the current LSP placements are significantly sub-optimal. See Section 5 for further discussions of reoptimization.

2.2. Online Computation

Online path computation is performed on-demand as LSRs in the network determine that they need to know the paths to use for LSPs. Thus, each computation is triggered by a request from an LSR.

As described in [RFC4655], the path computation function for online computation may be collocated with the LSR that makes the request, or it may be present in a computation-capable PCE server within the network. The PCE server may be another LSR in the network, a dedicated server, or a functional component of a Network Management System (NMS). Furthermore, the computation is not necessarily achieved by a single PCE operating on its own, but may be the result of cooperation between several PCEs.

The remainder of this document makes frequent reference to these different online models in order to indicate which is more appropriate in different P2MP scenarios.

2.2.1. LSR Loading

An important feature of P2MP path computation is the processing load that it places on the network element that is determining the path. Roughly speaking, the load to compute a least-cost-to-leaf tree is the same as the cost to compute a single optimal path to each leaf in turn. The load to compute a Steiner tree is approximately an order of magnitude greater, although algorithms exist to approximate Steiner trees in roughly the same order of magnitude of time as for a least-cost-to-leaf tree.

Whereas many LSRs are capable of simple Constrained Shortest Path First (CSPF) computations to determine a path for a single point-to-point (P2P) LSP, they rapidly become swamped if called on to perform multiple such computations, such as when recovering from a network failure. Thus, it is reasonable to expect that an LSR would struggle to handle a P2MP path computation for a tree with many destinations.

The result of an LSR becoming overloaded by a P2MP path computation may be two-fold. First, the LSR may be unable to provide timely computations of paths for P2P LSPs; this may impact LSP setup times for simple demand-based services and could damage the LSR's ability to recover services after network faults. Secondly, the LSR's processing capabilities may be diverted from other important tasks, not the least of which is maintaining the control plane protocols that are necessary to the support of existing LSPs and forwarding state within the network. It is obviously critically important that existing traffic should not be disrupted by the computation of a path for a new LSP.

It is also not reasonable to expect the ingress LSRs of P2MP LSRs to be specially powerful and capable of P2MP computations. Although a solution to the overloading problem would be to require that all LSRs that form the ingresses to P2MP LSPs be sufficiently high-capacity to perform P2MP computations, this is not an acceptable solution because, in all other senses, the ingress to a P2MP LSP is just a normal ingress LSR.

Thus, there is an obvious solution: off-load P2MP path computations from LSRs to remotely located PCEs. Such PCE function can be provided on dedicated or high-capacity network elements (such as dedicated servers, or high-end routers that might be located as Autonomous System Border Routers - ASBRs).

2.2.2. PCE Overload

Since P2MP path computations are resource-intensive, it may be that the introduction of P2MP LSPs into an established PCE network will cause overload at the PCEs. That is, the P2MP computations may block other P2P computations and might even overload the PCE.

Several measures can be taken within the PCE architecture to alleviate this situation as described in [RFC4655]. For example, path computation requests can be assigned priorities by the LSRs that issue them. Thus, the LSRs could assign lower priority to the P2MP requests, ensuring that P2P requests were serviced in preference. Furthermore, the PCEs are able to apply local and network-wide policy and this may dictate specific processing rules [RFC5394].

But ultimately, a network must possess sufficient path computation resources for its needs and this is achieved within the PCE architecture simply by increasing the number of PCEs.

Once there are sufficient PCEs available within the network, the LSRs may choose between them and may use overload notification information supplied by the PCEs to spot which PCEs are currently over-loaded. Additionally, a PCE that is becoming over-loaded may redistribute its queue of computation requests (using the PCE cooperation model described in [RFC4655]) to other, less burdened PCEs within the network.

2.2.3. PCE Capabilities

An LSR chooses between available PCEs to select the one most likely to be able to perform the requested path computation. This selection may be based on overload notifications from the PCEs, but could also consider other computational capabilities.

For example, it is quite likely that only a subset of the PCEs in the network have the ability to perform P2MP computations since this requires advanced functionality. Some of those PCEs might have the ability to satisfy certain objective functions (for example, least cost to destination), but lack support for other objective functions (for example, Steiner). Additionally, some PCEs might not be capable of the more complex P2MP reoptimization functionality.

The PCE architecture allows an LSR to discover the capabilities of the PCEs within the network at the same time it discovers their existence. Further and more detailed exchanges of PCE capabilities can be made directly between the PCEs and the LSRs. This exchange of PCE capabilities information allows a Path Computation Client (PCC) to select the PCE that can best answer its computation requests.

3. Fragmenting the P2MP Tree

A way to reduce the computational burden of computing a large P2MP tree on a single PCE is to fragment or partition the tree. This may be particularly obvious in a multi-domain network (such as multiple routing areas), but is equally applicable in a single domain.

Consider the network topology in Figure 1. A P2MP LSP is required from ingress LSR A to egress LSRs s, t, u, v, w, x, y, and z. Using a single PCE model, LSR A may request the entire path of the tree and this may be supplied by the PCE. Alternatively, the PCE that is consulted by LSR A may only compute the first fragment of the tree (for example, from A to K, L, and M) and may rely on other PCEs to compute the three smaller trees from K to t, u, and v; from L to w and x; and from M to s, y, and z.

The LSR consulted by A may simply return the first subtree and leave LSRs K, L, and M to invoke PCEs in their turn in order to complete the signaling. Alternatively, the first PCE may cooperate with other PCEs to collect the paths for the later subtrees and return them in a single computation response to PCE A. The mechanisms for both of these approaches are described in the PCE architecture [RFC4655].

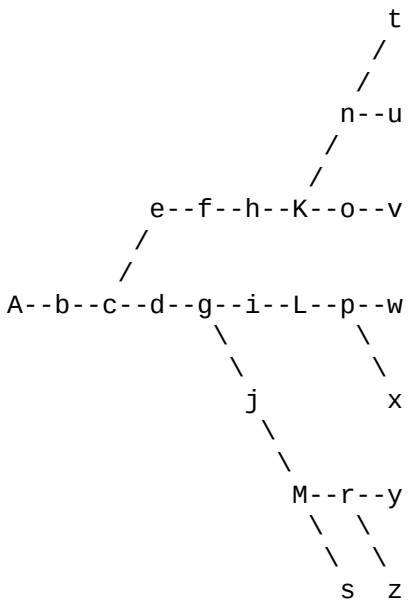

Figure 1: A P2MP Tree with Intermediate Computation Points

A further possibility is that LSRs at which the subtrees are stitched together (K, L, and M in this example) are selected from a set of potential such points using a cooperative PCE technique, such as the Backward Recursive Path Computation (BRPC) mechanism [RFC5441]. Indeed, if LSRs K, L, and M were ASBRs or Area Border Routers (ABRs), the BRPC technique would be particularly applicable.

Note, however, that while these mechanisms are superficially beneficial, it is far from obvious how the first LSR selects the transit LSRs K, L, and M, or how the leaf nodes are assigned to be downstream of particular downstream nodes. The computation to determine these questions may be no less intensive than the determination of the full tree unless there is some known property of the leaf node identifiers such as might be provided by address aggregation.

4. Central Replication Points

A deployment model for P2MP LSPs is to use centralized, well-known replication points. This choice may be made for administrative or security reasons, or because of particular hardware capability limitations within the network. Indeed, this deployment model can be achieved using P2P LSPs between ingress and replication point as well as between replication point and each leaf so as to achieve a P2MP service without the use of P2MP MPLS-TE.

The motivations for this type of deployment are beyond the scope of this document, but it is appropriate to examine how PCE might be used to support this model.

In Figure 2, a P2MP service is required from ingress LSR a to egress LSRs m, n, o, and p. There are four replication-capable LSRs in the network: D, E, J, and K.

When LSR a consults a PCE, it could be given the full P2MP path from LSR a to the leaves, but in this model, the PCE simply returns a P2P path to the first replication point (in this case, LSR D). LSR D will consult a PCE in its turn and determine the P2P LSPs to egress LSRs m and p as well as the P2P LSP to the next replication point, LSR J. Finally, LSR J will use a PCE to determine P2P LSPs to egresses n and o.

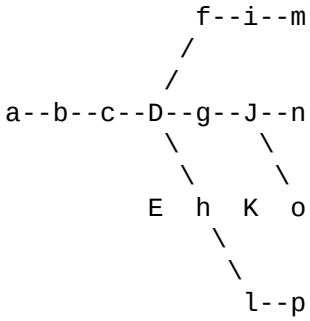

Figure 2: Using Centralized Replication Points

In this model of operation, it is quite likely that the PCE function is located at the replication points, which will be high-capacity LSRs. One of the main features of the computation activity is the selection of the replication points (for example, why is LSR D selected in preference to LSR E, and why is LSR J chosen over LSR K?). This selection may be made solely on the basis of path optimization as it would be for a P2MP computation, but may also be influenced by policy issues (for example, LSR D may be able to give better security to protect against rogue leaf attachment) or network loading concerns (for example, LSR E may already be handling a very large amount of traffic replication for other P2MP services).

5. Reoptimization and Modification

Once established, P2MP LSPs are more sensitive to modification than their P2P counterparts. If an egress is removed from a P2P LSP, the whole LSP is torn down. But egresses may be added to and removed from active P2MP LSPs as receivers come and go.

The removal of an egress from a P2MP LSP does not require any new path computation since the tree can be automatically pruned.

The addition of a new egress to a P2MP LSP can be handled as the computation of an appropriate branch point and the determination of a P2P path from the branch point to the new egress. This is a relatively simple computation and can be achieved by reverse-path CSPF, much as in the manner of some multicast IP routing protocols.

However, repeated addition to and removal from a P2MP LSP will almost certainly leave it in a sub-optimal state. The tree shape that was optimal for the original set of destinations will be distorted by the changes and will not be optimal for the new set of destinations.

Further, as resource availability changes in the network due to other LSPs being released or network resources being brought online, the path of the P2MP LSP may become sub-optimal.

Computing a new optimal path for the P2MP LSP is as simple as computing any optimal P2MP path, but selecting a path that can be applied within the network as a migration from the existing LSP may be more complex. Additional constraints may be applied by the network administrator so that only subsets of the egresses (or subtrees of the P2MP tree) are optimized at any time. In these cases, the computational load of reoptimization may be considerable, but fortunately reoptimization computations may be performed as background activities. Splitting the P2MP tree into subtrees, as described in Section 3, may further reduce the computation load and may assist with administrative preferences for partial tree reoptimization.

Network-wide reoptimization of multiple LSPs [RFC5557] can achieve far greater improvements in optimality within overloaded networks than can be achieved by reoptimizing LSPs sequentially. Such computation would typically be performed offline and would usually require a dedicated processor such as a PCE invoked by the NMS.

6. Repair

LSP repair is necessary when a network fault disrupts the ability of the LSP to deliver data to an egress. For a P2MP LSP, a network fault is (statistically) likely to only impact a small subset of the total set of egresses. Repair activity, therefore, does not need to recompute the path of the entire P2MP tree. Rather, it needs to quickly find suitable new branches that can be grafted onto the existing tree to reconnect the disconnected leaves.

In fact, immediately after a network failure there may be a very large number of path computations required in order to restore multiple P2P and P2MP LSPs. The PCEs will be heavily loaded, and it is important that computation requests are restricted to only the 'essential'.

In this light, it is useful to note that the simple repair computations described in the first paragraph of this section may be applied to achieve a first repair of the LSPs, while more sophisticated reoptimization computations can be deferred until the network is stable and the load on the PCEs has been reduced. Those reoptimizations can be computed as described in Section 5.

7. Disjoint Paths

Disjoint paths are required for end-to-end protection services and sometimes for load balancing. They may require to be fully disjoint (except at the ingress and egress!), link disjoint (allowing common nodes on the paths), or best-effort disjoint (allowing shared links or nodes when no other path can be found).

It is possible to compute disjoint paths sequentially, but this can lead to blocking problems where the second path cannot be placed. Such issues are more readily avoided if the paths are computed in parallel.

The computation of link disjoint P2P paths may be non-trivial and may be the sort of task that an LSR offloads to a PCE because of its complexity. The computation of disjoint P2MP paths is considerably more difficult and is therefore a good candidate to be offloaded to a PCE that has dedicated resources. In fact, it may well be the case that not all P2MP-capable PCEs can handle disjoint path requests and it may be necessary to select between PCEs based on their capabilities.

8. Manageability Considerations

The use of PCE to compute P2MP paths has many of the same manageability considerations as when it is used for P2P LSPs [RFC5440]. There may be additional manageability implications for the size of P2MP computation requests and the additional loading exerted on the PCEs.

8.1. Control of Function and Policy

As already described, individual PCEs may choose to not be capable of P2MP computation, and where this function is available, it may be disabled by an operator, or may be automatically withdrawn when the PCE becomes loaded or based on other policy considerations.

Further, a PCE may refuse any P2MP computation request or pass it on to another PCE based on policy.

8.2. Information and Data Models

P2MP computation requests necessitate considerably more information exchange between the LSR and the PCE than is required for P2P computations. This will result in much larger data sets to be controlled and modeled, and will impact the utility of any management data models, such as MIB modules. Care needs to be taken in the

design of such data models, and the use of other management protocols and data modeling structures, such as NETCONF [RFC4741] and the NETCONF Data Modeling Language (NETMOD), could be considered.

8.3. Liveness Detection and Monitoring

PCE liveness detection and monitoring is unchanged from P2P operation, but it should be noted that P2MP requests will take longer to process than P2P requests, meaning that the time between request and response will be, on average, longer. This increases the chance of a communications failure between request and response and means that liveness detection is more important.

8.4. Verifying Correct Operation

Correct operation of any communication between LSRs and PCEs is exactly as important as it is for P2P computations.

The correct operation of path computation algorithms implemented at PCEs is out of scope, but LSRs that are concerned that PCE algorithms might not be operating correctly may make identical requests to separate PCEs and compare the responses.

8.5. Requirements on Other Protocols and Functional Components

As is clear from the PCE architecture, a communications protocol is necessary to allow LSRs to send computation requests to PCEs and for PCEs to cooperate. Requirements for such a protocol to handle P2P path computations are described in [RFC4657], and additional requirements in support of P2MP computations are described in [PCE-P2MP]. The PCE Communication Protocol (PCEP) is defined in [RFC5440], but extensions will be necessary to support P2MP computation requests.

As described in the body of this document, LSRs need to be able to recognize which PCEs can perform P2MP computations. Capability advertisement is already present in the PCE Discovery protocols ([RFC5088] and [RFC5089]) and can also be exchanged in PCEP ([RFC5440]), but extensions will be required to describe P2MP capabilities.

As also described in this document, the PCE needs to know the branch capabilities of the LSRs and store this information in the TED. This information can be distributed using the routing protocols as described in [RFC5073].

8.6. Impact on Network Operation

The use of a PCE to perform P2MP computations may have a beneficial impact on network operation if it can offload processing from the LSRs, freeing them up to handle protocol operations.

Furthermore, the use of a PCE may enable more dynamic behavior in P2MP LSPs (such as the addition of new egresses, reoptimization, and failure recovery) than is possible using more traditional management-based planning techniques.

9. Security Considerations

The use of PCE to compute P2MP paths does not raise any additional security issues beyond those that generally apply to the PCE architecture. See [RFC4655] for a full discussion.

Note, however, that P2MP computation requests are more CPU-intensive and also use more link bandwidth. Therefore, if the PCE was attacked by the injection of spurious path computation requests, it would be more vulnerable through a smaller number of such requests.

Thus, the use of message integrity and authentication mechanisms within the PCE protocol should be used to mitigate attacks from devices that are not authorized to send requests to the PCE. It would be possible to consider applying different authorization policies for P2MP path computation requests compared to other requests.

10. Acknowledgments

The authors would like to thank Jerry Ash and Jean-Louis Le Roux for their thoughtful comments. Lars Eggert, Dan Romascanu, and Tim Polk provided useful comments during IESG review.

11. References

11.1. Normative References

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

11.2. Informative References

[RFC4461] Yasukawa, S., Ed., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.

[RFC4657] Ash, J., Ed., and J. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol Generic Requirements", RFC 4657, September 2006.

[RFC4741] Enns, R., Ed., "NETCONF Configuration Protocol", RFC 4741, December 2006.

[RFC4834] Morin, T., Ed., "Requirements for Multicast in Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4834, April 2007.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

[RFC5073] Vasseur, J., Ed., and J. Le Roux, Ed., "IGP Routing Protocol Extensions for Discovery of Traffic Engineering Node Capabilities", RFC 5073, December 2007.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.

[RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[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, April 2009.

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, July 2009.

[PCE-P2MP] Yasukawa, S., and Farrel, A., "PCC-PCE Communication Requirements for Point to Multipoint Multiprotocol Label Switching Traffic Engineering (MPLS-TE)", Work in Progress, May 2008.

Authors' Addresses

Seisho Yasukawa
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-Shi, Tokyo 180-8585,
Japan

EMail: yasukawa.seisho@lab.ntt.co.jp

Adrian Farrel
Old Dog Consulting

EMail: adrian@olddog.co.uk