A Queueing Model Framework of PCE-based Inter-area Path Computation

Juanjuan Yu¹, Yue He¹, Kai Wu¹, Marco Tacca¹, Andrea Fumagalli¹, and Jean-Phillippe Vasseur²

¹The OpNeAR Laboratory, Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
E-mails:{jxy051000,yxh052000,kxw016500,mtacca,andreaf}@utdallas.edu
²Cisco Systems Inc.
E-mail:jvasseur@cisco.com

Abstract—Path computation elements (PCE’s) are used to compute end-to-end paths across multiple areas. Multiple PCE’s may be dedicated to each area to provide sufficient path computation capacity and redundancy. An open problem is to which PCE to send the path computation request. This problem may be a non trivial problem if PCE’s have uneven processing capacities.

This paper presents a queueing model based on product form to estimate the latencies in path computation while accounting for the arrival rate of path computation requests. The model is used to find the PCE selection policy to minimize the overall expected latencies in path computation.

Simulation studies demonstrate that the use of the simplistic product form approach yields reasonable approximations that are within up to 15% of the simulation results at practical offered loads.

Keywords—PCE, PCEP, MPLS traffic engineering, load-balancing, inter-area

I. Introduction

Today’s Internet has witnessed an increasing demand of mission-critical and real-time multimedia services. Service providers are required to achieve bandwidth optimization and strict quality of service guarantees. Multi protocol label switching (MPLS) traffic engineering (TE) was introduced to meet these requirements. A fundamental building block of MPLS-TE is constrained-based path computation, which requires a TE database (TED) to represent the available network resources, and computes shortest paths based on a sub-graph derived from the TED by pruning the links not satisfying the given constraints.

Most of today’s TE solutions have been developed in the context of single domain, where a domain is a collection of network elements that share a common sphere of address management or path computational responsibility, such as an interior gateway protocol (IGP) area, an autonomous system (AS) or a generalized MPLS (GMPLS) region. In recent years the networking community has expressed an increasing interest in extending the scope of MPLS-TE to cope with multi domain scenarios. End-to-end inter-domain constrained path computation presents some challenges:

1) Optimality: the ability to maximize network utilization across multiple domains.
2) Scalability: the ability to minimize the required network resources and communication overhead in achieving the computation.
3) Responsiveness: the timeliness in finishing the computation.
4) Confidentiality: the capability to confine topology and resource information within confidentiality boundary among service providers.
5) Correctness: the ability to avoid failures or crankback signalling upon setting up the computed path.
6) Computation Capability: the capability in achieving CPU-intensive computation such as multi-constrained or multi-cast path computations.

The IETF recently formed the path computation element (PCE) [1] working group to address these challenges. The PCE working group has specified new protocols and path computation architectures so as to provide efficient solutions to inter-domain constrained path computation, along with other topics (such as multi-constrained optimization and multi-layer TE).

The PCE architecture singles out the path computation functionality within the control plane. A PCE is a network entity with certain path computation capabilities that processes requests from client entities, or path computation clients (PCC’s). PCE’s from different domains can communicate with each other to exchange information needed to perform inter-domain path computation. The PCE solution performs better than native per-domain solution (path computation on the fly per domain entry) in both optimality and correctness without introducing extra complexity [2].

The PCE architecture offers a number of services, e.g., handling of multi-priority requests for path computation, partial path computations performed in parallel by multiple PCE’s, and dynamic redirection of path computation requests based on real-time PCE utilization and estimated waiting time to be serviced. Each of these services will undoubtedly have an impact on the overall network effectiveness and performance, both in steady state and under stress conditions, e.g., in the presence of network element failures. It is thus important to perform detailed modeling and extensive simulation campaigns.
not only of the scheduling policy inside a PCE, but also of the overall network architecture when the use of multiple PCE’s in tandem is required. The outcome resulting from this modeling effort is expected to enable and facilitate better designs of PCE schedulers and PCE related protocol extensions.

In this paper the authors model the PCE architecture as a network of queues, servicing path computation requests. Simplicity is one advantage of the proposed queueing model, as its product form derivation is based on the assumption that request arrivals form a Poisson process. Yet, the accuracy of the model is 15% or better at practical offered load values. The model is general enough to characterize architectures with uneven PCE’s computational capacities, and may be useful in designing wise PCE selection policies to minimize the overall expected latencies in path computation. In particular, the inter-area scenario is chosen as case study. Since only one IGP is involved in inter-area case, the simulation is relatively easier compared to more sophisticated scenarios such as the inter-domain case.

Studies on PCE architectures include [2], [3], [4], [5], [6], [7]. In [2], PCE-based multiple-domain path computation deployment is compared with per-domain path computation in several performance metrics including path cost, CAC (call admission control) protocol failures and crankback signaling. In [3], the cooperation between PCE’s in computing end-to-end across multiple domains is treated as a multistage decision problem, where each decision stage corresponds to selecting one of multiple entry points. For the case of a large set of entry points at each stage, a subset could be chosen considering the blocking probability that the path can be computed if a certain entry point is chosen. In [4], disjoint QoS multi-domain path are computed using PCE architecture with topology aggregation. Various aggregation schemes are proposed and compared. In [5], a hierarchical PCE architecture is proposed to compute multi-domain disjoint backup paths, where an inter-domain PCE delegates path computation requests to lower level PCE’s for selected domains. In [6], a PCE containing multi-layer TED is proposed to limit amount of the exchanged information through PCE communication protocol (PCEP) in case of multi-layer multi-domain path computation. In [7], path setup delay is investigated for PCE-based inter-layer path computation for both centralized PCE and distributed PCE architectures.

II. DISTRIBUTED PCE ARCHITECTURE

This section presents a brief description of the PCE architecture, and defines the problem of properly selecting the subset of PCE’s to perform the distributed computation of end-to-end paths.

The distributed PCE architecture is shown in Fig. 1. Each PCE maintains a TED covering certain area(s). To compute a path across the TED area, the PCE communicates with another PCE that has information outside its TED area. PCE’s communicate in a client/server fashion, same as PCC communicates with PCE, i.e., PCC sends its path computation request to a neighboring PCE, which in turn sends the request to a downstream neighboring PCE. The computation performed by the downstream PCE is returned to the upstream PCE, which in turn returns the complete path computation to PCC. The protocol used for these requests and responses is called PCE communication protocol, or PCEP.

The border routers are typically ideal positions where PCE might reside. The border routers maintain a TE database through routing protocols with TE extensions. In multi-area scenario, an area border router (ABR) may be selected to be a PCE that maintains TE information of both areas of the ABR borders. In this case, the TED area is the combination of two physical areas (domains). The nodes at the boundary of a TED area are called TED domain boundary nodes (BN). BN’s function as entry points for neighboring domains or areas. They are typically also border routers and hence can be PCE’s of neighboring domains. Note that there could be one or more PCE’s sharing the same TED area. For instance, all ABR’s bordering area A and area 0 might be PCE’s for these two areas, thus providing redundancy and increased computational capacity.

Assuming that all border routers are also PCE’s, a 3-area (including backbone area) network and a 2-AS (connected by inter-AS links) network share the same PCE network architecture, i.e., two sets of PCE’s with two intersecting TED areas. Therefore the queuing performance study of PCE’s over multi-area network is a good start to understand more general multi-domain scenarios.

A multi-area path can be computed with a recursive and backward path computation procedure as follows [8]. An inter-area path typically goes from the source area through the backbone area into the destination area. A PCC wishing to establish an inter-area (TE) LSP across the backbone sends a path computation request to one of the (neighboring) PCE’s within its own area. Upon reception of the request, the PCE relays the path computation request to a downstream (neighboring) PCE residing in the destination area of the LSP’s destination. The destination PCE first computes all the constrained shortest paths that originate at entry points (the TED domain BN’s of the upstream PCE) in that area and terminate at the destination node. The set of these paths forms a virtual shortest path tree
(VSPT) rooted at the destination. The computed VSPT is sent to the upstream PCE (in the upstream area), which in turn computes another VSPT using its own area information and the received VSPT combined. As the upstream PCE is in the source area, it completes the distributed path computation procedure by finding the shortest constrained end-to-end path. The end-to-end path is passed onto the source PCC for further use (bandwidth reservation along the path).

Note that detailed topology and resource related information in a given area are handled by the local PCE’s and do not need to be passed onto PCE’s that belong to other areas. Both confidentiality and scalability are then guaranteed.

A. PCE Selection and Scheduling Strategy

While waiting for the VSPT from the downstream PCE, a PCE may continue to process other concurrent requests. Requests for inter-area routing are thus performed in pipeline thanks to the cooperation among PCE’s. A careful scheduling of the related path computation tasks is needed to yield overall high throughput of inter-domain path computation requests. The PCE scheduler is not within the scope of this study. For simplicity, first in first out scheduling is used at PCE’s.

When sending a request, a PCC must choose one of the neighboring PCE’s to contact. In the presence of multiple neighboring PCE’s, the PCC must carefully select which one to contact to avoid unnecessary bottlenecks and late responses. Likewise, the upstream PCE must carefully select one of the downstream neighboring PCE’s for the same reason. The analytical model presented in Section III may be used to design a wise PCE selection policy.

B. Path Setup

Upon reception of the whole computed path, the PCC sends out the path setup request through a signaling protocol such as RSVP-TE. The computed path may be either an explicitly strict path, which includes all the links and nodes over all the areas or a loose path, which includes only entry nodes of other areas. In the latter case, the border router at the entry point must expand the path segment upon receiving the setup signaling message. Note that since a multi-area path is computed using the same TED information as when the path is setup (considering the short latency of path computation), the probability of call admission control (CAC) protocol failure is greatly reduced compared to other solutions. The RSVP-TE dynamics are not within the scope of this study, and bandwidth reservation attempts are assumed to be carried out successfully in all cases.

III. Queueing Model for PCE Based Inter-Area Path Computation

This section presents a queueing model that can be used to evaluate the latency and load at each PCE. As shown in Fig. 2, the model assumes that each PCC is directly connected to all the PCE’s in the area. The above assumption is considered acceptable, as inter-area routing of PCEP packets is carried out taking advantage of the IGP protocol. Therefore, at the $k^{th}$ PCE in area $i$ ($PCE_{ik}$), 5 types of incoming path computation requests can be identified:

- type 1: path computation requests where the source PCC is in area $i \neq 0$ and the destination is in area 0.
- type 2: path computation requests where the source PCC is in area 0 and the destination is in area $i \neq 0$.
- type 3: path computation requests where the source PCC is in area $i \neq 0$ and the destination is in area $j \neq i$, and the PCEP packet is coming from the PCC, i.e., $PCE_{ik}$ is the first PCE visited by the PCEP packet.
- type 4: path computation requests where the source PCC is in area $j \neq i$ and the destination is in area $i \neq 0$, i.e., $PCE_{ik}$ is the second visited PCE by the PCEP packet.
- type 5: path computation requests where the source PCC is in area $i \neq 0$ and the destination is in area $j \neq 0, i$, and the PCEP packet is coming from a PCE in area $j$, i.e., $PCE_{ik}$ is the third and last PCE visited by the PCEP packet.

In order to capture the nature of the flow of path computation requests, the model introduces in each area one block as shown in Fig. 3. Each block represents requests from area $i$ to area $j \neq i$.

The following notation is used.

- $M$: number of areas, not including the backbone area (area 0);
- $K_i$: the number of PCE’s in area $i \neq 0$, i.e., the number of PCE’s connecting area $i$ and area 0;

1Notice that a PCE in area $i \neq 0$ is in area 0 as well. For simplicity, in order to identify which area a PCE belongs to, only area $i$ is used in the paper.
• $\lambda_{i,j}$: arrival rate of path computation requests characterized by source in area $i$ and destination in area $j \neq i$
• $PCE_i$: $k^{th}$ PCE in area $i$
• $p_{i,k}^{(0)}$: probability that a PCC chooses to send a path computation request that has neither the source, nor the destination in area $i$, and $PCE_i$ as the first PCE;
• $p_{i,k}^{(0,0)}$: probability that a PCC in the backbone area (area 0) chooses $PCE_{i,k}$ as the first PCE for a path computation request characterized by the destination in area $i$;
• $p_{i,k}^{(m,j,j)}$: probability that $PCE_{j,m}$ chooses $PCE_{i,k}$ when sending a PCEP packet towards one of the PCE’s in area $i$;
• $\mu_{i,k}^{(1)}$: service rate of $PCE_{i,k}$ when processing a path computation request that has neither the source, nor the destination in area $0$, and $PCE_{i,k}$ is the first PCE in the path computation procedure;
• $\mu_{i,k}^{(2)}$: service rate of $PCE_{i,k}$ when processing a path computation request that has neither the source, nor the destination in area $0$, and $PCE_{i,k}$ is the second PCE in the path computation procedure;
• $\mu_{i,k}^{(3)}$: service rate of $PCE_{i,k}$ when processing a path computation request that has neither the source, nor the destination in area $0$, and $PCE_{i,k}$ is the third and last PCE in the path computation procedure;
• $\mu_{i,k}^{(4)}$: service rate of $PCE_{i,k}$ when processing a path computation request that has either the source, or the destination in area $0$;

With the defined notation, at each $PCE_{i,k}$, it becomes possible to express the values for the incoming rates for each of the 5 types of path computation requests as follows:

$$\lambda_{i,k}^{(1)} = \lambda_{i,0} \cdot p_{i,k}^{(0,0)}$$  \hspace{1cm} (1)

$$\lambda_{i,k}^{(2)} = \lambda_{0,i} \cdot p_{i,k}^{(0,0)}$$  \hspace{1cm} (2)

$$\lambda_{i,k}^{(3)} = \sum_{j=1,i+1}^{i+M} \lambda_{i,j} \cdot p_{i,k}^{(j,j)}$$  \hspace{1cm} (3)

$$\lambda_{i,k}^{(4)} = \sum_{j=1,i+1}^{i+M} \sum_{m=1,K_j} \lambda_{j,m} \cdot p_{j,m,k}^{(j)} \cdot p_{i,k}^{(j)}$$  \hspace{1cm} (4)

$$\lambda_{i,k}^{(5)} = \sum_{j=1,i+1}^{i+M} \sum_{m=1,K_j} \sum_{k=1,K_i} \lambda_{j,m} \cdot p_{j,m,k}^{(j)} \cdot p_{i,k}^{(j)} \cdot p_{i,k}^{(j,m,i)}$$  \hspace{1cm} (5)

After deriving the value of the arrival rates for all types of incoming path computation requests, it is possible to calculate the utilization $\rho_{i,k}$ for $PCE_{i,k}$ as:

$$\rho_{i,k} = \frac{\lambda_{i,k}^{(1)} + \lambda_{i,k}^{(2)} + \lambda_{i,k}^{(3)} + \lambda_{i,k}^{(4)} + \lambda_{i,k}^{(5)}}{\mu_{i,k}^{(1)} + \mu_{i,k}^{(2)} + \mu_{i,k}^{(3)} + \mu_{i,k}^{(4)} + \mu_{i,k}^{(5)}}$$

$$= \frac{\lambda_{i,k}^{(1)} + \lambda_{i,k}^{(2)} + \lambda_{i,k}^{(3)} + \lambda_{i,k}^{(4)} + \lambda_{i,k}^{(5)}}{\mu_{i,k}^{(1)} + \mu_{i,k}^{(2)} + \mu_{i,k}^{(3)} + \mu_{i,k}^{(4)} + \mu_{i,k}^{(5)}}$$

$$+ \frac{\sum_{j=1,i+1}^{i+M} \sum_{m=1,K_j} \lambda_{j,m} \cdot p_{j,m,k}^{(j)} \cdot p_{i,k}^{(j)} \cdot p_{i,k}^{(j,m,i)}}{\mu_{i,k}^{(2)} + \mu_{i,k}^{(3)} + \mu_{i,k}^{(4)} + \mu_{i,k}^{(5)}}$$

$$+ \frac{\sum_{j=1,i+1}^{i+M} \sum_{m=1,K_j} \sum_{k=1,K_i} \lambda_{j,m} \cdot p_{j,m,k}^{(j)} \cdot p_{i,k}^{(j)} \cdot p_{i,k}^{(j,m,i)}}{\mu_{i,k}^{(2)} + \mu_{i,k}^{(3)} + \mu_{i,k}^{(4)} + \mu_{i,k}^{(5)}}$$  \hspace{1cm} (6)

Assume that the arrival process is Poisson and all service times are deterministic, i.e., every PCE behaves as an M/D/1 system. Even though the product form does not represent an exact solution in this case, its application yields a viable approximation, as extensively studied in [9]. By applying the product form, the following statistics are derived.

The average number of jobs in $PCE_{i,k}$ is:

$$N_{i,k} = \rho_{i,k} \cdot (2 - \rho_{i,k}) \cdot \frac{1}{2(1 - \rho_{i,k})}$$  \hspace{1cm} (7)

The average job time (queueing plus service) in $PCE_{i,k}$ is:

$$T_{i,k} = \frac{N_{i,k}}{\rho_{i,k}} \cdot \frac{2 - \rho_{i,k}}{2(1 - \rho_{i,k})}$$  \hspace{1cm} (8)

IV. UNEVEN PCE’S: BALANCING THE AVERAGE JOB TIME

When reliability is a concern, multiple PCE’s in the same area provide a solution. However, if the PCE’s associated with area $i$ have different computing power, i.e., service rate, it is not trivial for PCC’s in area $i$ as well as PCE’s in other areas to decide which PCE in area $i$ should be used. This section provides a criterion that can be used to make such decision under the following assumptions:

• the decision is based on minimizing the maximum average job time that a path computation request has to experience;
• there are only 2 PCE’s in each area;
• the system is modeled assuming that the product form introduced in the previous section is an acceptable approximation.

Fig. 4 shows the model used to obtain a solution. All flows of path computation requests are combined together. Then, each job, i.e., path computation request, is randomly assigned to either $PCE_{i,1}$ with probability $p_{i,1}$ or to $PCE_{i,2}$ with probability $p_{i,2}$. Service rates at each PCE are independent of the type of path computation request. Without loss of generality it is assumed that $\mu_{i,1} \geq \mu_{i,2}$. The following notation is used:

• $X_{i,1}^2$ is the second moment of the service time at $PCE_{i,1}$
• $X_{i,2}^2$ is the second moment of the service time at $PCE_{i,2}$.

The overall arrival rate $\Lambda$ of path computation requests to area $i \neq 0$ is:

$$\Lambda = \sum_{i=1}^{2} [\lambda_{i,1}^{(1)} + \lambda_{i,1}^{(2)} + \lambda_{i,1}^{(3)} + \lambda_{i,1}^{(4)} + \lambda_{i,1}^{(5)}]$$

$$= \rho_{i,1} \cdot \Lambda + \rho_{i,2} \cdot \Lambda$$  \hspace{1cm} (9)
The objective is to set:

\[ T_{i,1} = T_{i,2} \] (10)

Using the Pollaczek-Khinchin (P-K) formula \([10]\), (10) becomes:

\[
\frac{1}{\mu_{i,1}} + \frac{p_{i,1} \cdot \Lambda \cdot X_{i,1}^2}{2 \cdot (1 - \frac{p_{i,1} \cdot \Lambda}{\mu_{i,1}})} = \frac{1}{\mu_{i,2}} + \frac{p_{i,2} \cdot \Lambda \cdot X_{i,2}^2}{2 \cdot (1 - \frac{p_{i,2} \cdot \Lambda}{\mu_{i,2}})}
\] (11)

Combining (9) and (11) a second degree polynomial is obtained:

\[ A \cdot (p_{i,1} \cdot \Lambda)^2 + B \cdot (p_{i,1} \cdot \Lambda) + C = 0 \] (12)

where:

\[
A = 2 \cdot \mu_{i,2} - 2 \cdot \mu_{i,1} + \overline{X_{i,1}}^2 \cdot \mu_{i,1} \cdot \mu_{i,2}^2
\] (13)
\[
B = 2 \cdot \mu_{i,2}^2 - 4 \cdot \mu_{i,1} \cdot \mu_{i,2} + 2 \cdot \mu_{i,1} \cdot \Lambda + 2 \cdot \mu_{i,2}^2
\] (14)
\[
C = \overline{X_{i,2}}^2 \cdot \mu_{i,1}^2 \cdot \mu_{i,2}^2 + 2 \cdot \mu_{i,1} \cdot \mu_{i,2} \cdot \Lambda
\] (15)

Solving equation (12) the values for \(p_{i,1}\) and \(p_{i,2}\) can be found, respectively.

If \(A \neq 0\), the two solutions are:

\[
p_{i,1}(1) = -\frac{B + \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A \cdot \Lambda}
\] (16)
\[
p_{i,2}(1) = 1 - p_{i,1}(1)
\] (17)

and

\[
p_{i,1}(2) = -\frac{B - \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A \cdot \Lambda}
\] (18)
\[
p_{i,2}(2) = 1 - p_{i,1}(2)
\] (19)

Then, assuming that the queueing system is stable, the solution is:

- if \(p_{i,1}(1) > 0\) and \(p_{i,2}(1) > 0\)

\[
p_{i,1} = p_{i,1}(1)
\] (20)
\[
p_{i,2} = p_{i,2}(1)
\] (21)

- if \(p_{i,1}(2) > 0\) and \(p_{i,2}(2) > 0\)

\[
p_{i,1} = p_{i,1}(2)
\] (22)
\[
p_{i,2} = p_{i,2}(2)
\] (23)

- otherwise

\[
p_{i,1} = 1
\] (24)
\[
p_{i,2} = 0
\] (25)

V. Conclusion

Path computation element (PCE) enables scalable solutions for end-to-end multi-area TE path computation. In a distributed PCE architecture, PCE’s communicate with each other through the PCEP protocol to service path computation requests. The selection of the next hop PCE to relay path computation request plays a key role in path computation latency. A careful load balancing technique can improve the overall request throughput and latency.

In this paper a queueing model is presented to evaluate the latency and load at each PCE in this regard. A product form solution is proposed based on the assumption that path computation requests form a Poisson arrival process. Based on this analytical framework, a load balancing strategy is devised with the objective to achieve an average latency that is even across the PCE pair serving the same area.

Both the analytical framework and the load balancing strategy were validated by two levels of simulation [9]: one based on a network of queues and the other based on the full implementation of the MPLS-TE and PCEP protocol suite. The extensive simulation results in [9] show that the proposed product form yields good approximations for most practical networks, where the simulation based results differ from the model estimated values by at most 15% in practical load cases. In addition, the proposed load balancing strategy is found to effectively reduce the overall expected path computation latency.

REFERENCES