A Queueing Model for PCEP (Path Computation Element Protocol)

Juanjuan Yu, Yue He, Kai Wu, Marco Tacca, Andrea Fumagalli, J.P Vasseur

Technical Report UTD/EE/13/2008

September 2008

A Queueing Model for PCEP (Path Computation Element Protocol)

Juanjuan Yu[1], Yue He[1], Kai Wu[1], Marco Tacca[1], Andrea Fumagalli[1], and Jean-Phillippe Vasseur[2]

[1]The 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
[2]Cisco 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 which PCE should be chosen to send the path computation request to, that may be a non trivial problem if PCE's have uneven processing capacities.**

This paper presents a product form queueing model 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 average expected latencies in path computation. The model is validated against two simulation benchmarks obtained using OPNET, i.e., a network of queues and the multi protocol label switching with traffic engineering (MPLS-TE) network running the PCE communication protocol (PCEP).

The study shows that the use of product form yields approximations that are up to 15% at practical offered loads. Moreover, the PCE selection policy derived under the product form assumption is showed to be effective in minimizing the overall expected latencies in path computation.

***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 multicast path computations.

To address these challenges, the IETF recently formed the PCE (path computation element) [1] working group. 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 could 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

This research is supported in part by NSF Grant No. CNS-0435393.

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 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 report, the authors model the PCE architecture as a network of queues, servicing path computation requests. In particular, the inter-area scenario is chosen as case of study without loss of generality. Since only one IGP is involved in inter-area case, the simulation is relatively easier compared to more sophisticated scenarios such as the inter-as case. 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. To validate the proposed analytical model based on product form, both a network of queue simulator and a real world packet based simulator are developed on OPNET Modeler [3]. To the authors' best knowledge, the latter is the first simulator of PCE architecture at the packet level.

[2], [4], [5], [6], [7], [8] studied PCE architecture. In [2], PCE-based multiple-domain path computation deployment is compared with per-domain path computation in several performance metrics including path cost, CAC failures and crankback signaling. In [4], 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 entry points. For case of large set of entry points at each stage, a subset could be chosen considering the blocking probability that the path can be computed if certain entry point is chosen. In [5], disjoint QoS multi-domain path are computed using PCE architecture with topology aggregation. Various aggregation schemes are proposed and compared. In [6], 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 [7], A PCE containing multi-layer TED is proposed to limit amount of exchanged information through PCEP in case of multi-layer multi-domain path computation. In [8], 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.

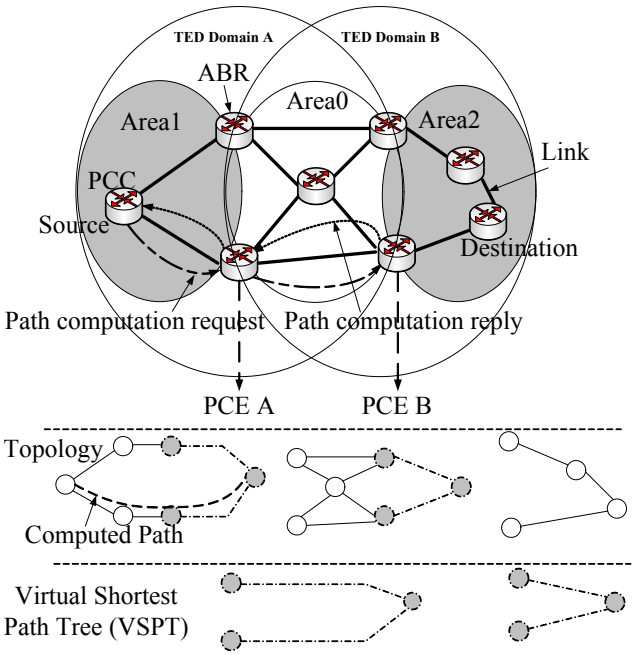

Fig. 1. PCE's inter-area path computation

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 queueing performance study of PCE's over multi-area network is a good start to understand more general multi-domain scenarios. Without loss of generality, the terms area and domain are used inter-changeably from this point unless otherwise noted.

A multi-area path can be computed with a recursive and backward path computation procedure as follows [9]. 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 then used.

When sending a request, PCC must choose one of the neighboring PCE's to contact. In the presence of multiple neighboring PCE's, 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, 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) 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

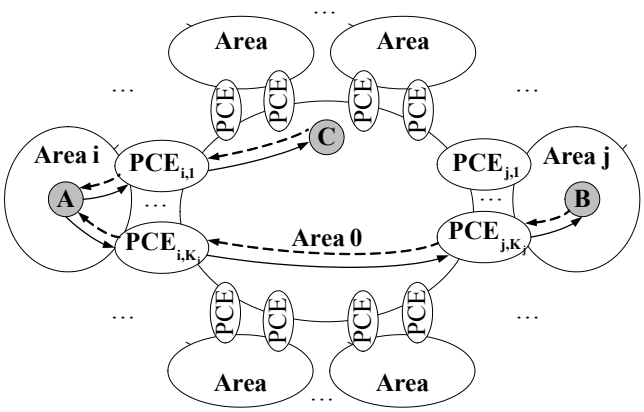

Fig. 2. PCE architecture model

This section presents an 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 ($PCE_{i,k}$[1]), 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 $j \neq 0, i$, and the PCEP packet is coming from the PCC, i.e., $PCE_{i,k}$ is the first PCE visited by the PCEP packet.
- type 4: path computation requests where the source PCC is in area $j \neq 0$ and the destination is in area $i \neq 0, j$, i.e.,$PCE_{i,k}$ 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., $PCE_{i,k}$ 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 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);

[1]Notice that a PCE in area $i \neq 0$ is in area 0 as well. For simplicity, in order to identify to which area a PCE belongs, only area i is used in the paper.

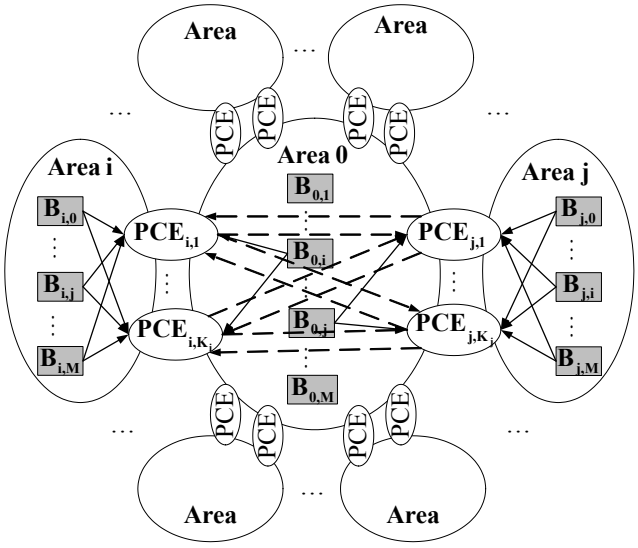

Fig. 3. block based analytical model

- K_i is the number of PCE's in area $i \neq 0$, i.e., the number of PCE's connecting area i and area 0;
- $\lambda_{i,j}$: arrival rate of path computation requests characterized by source in area i and destination in area $j \neq i$
- $PCE_{i,k}$: k^{th} PCE in area i
- $p_{i,k}^{(i,j)}$: probability that a PCC chooses to send a path computation request that belongs to the flow between areas i and j to $PCE_{i,k}$ as the first PCE;
- $p_{i,k}^{(0,i)}$: 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;
- $p_{i,k}^{(m,j,i)}$: probability that $PCE_{j,m}$ chooses $PCE_{i,k}$ when sending a PCEP packet towards one of the PCE's in area i;
- $\mu_{i,k}^{(2,1)}$: service rate of $PCE_{i,k}$ when processing a path computation requests that have neither the source, nor the destination in area 0 and $PCE_{i,k}$ is the first PCE in the path computation procedure;
- $\mu_{i,k}^{(2,2)}$: 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;
- $\mu_{i,k}^{(2,3)}$: service rate of $PCE_{i,k}$ when processing a path computation requests that have neither the source, nor the destination in area 0 and $PCE_{i,k}$ is the third and last PCE in the path computation procedure;
- $\mu_{i,k}^{(1)}$: service rate of $PCE_{i,k}$ when processing a path computation requests that have 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. The rate of type 1 requests at $PCE_{i,k}$ is:

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

The rate of type 2 requests $PCE_{i,k}$ is:

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

The rate of type 3 requests $PCE_{i,k}$ is:

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

The rate of type 4 requests $PCE_{i,k}$ is:

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

The rate of type 5 requests $PCE_{i,k}$ is:

$$\lambda_{i,k}^{(5)} = \sum_{j=1...i-1,i+1...M} \sum_{m=1...K_j} \lambda_{i,j} \cdot p_{i,k}^{(i,j)} \cdot p_{j,m}^{(i,k,j)} \cdot p_{i,k}^{(j,m,i)} \tag{5}$$

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:

$$\begin{aligned}
\rho_{i,k} &= \frac{\lambda_{i,k}^{(1)} + \lambda_{i,k}^{(2)}}{\mu_{i,k}^{(1)}} + \frac{\lambda_{i,k}^{(3)}}{\mu_{i,k}^{(2,1)}} + \frac{\lambda_{i,k}^{(4)}}{\mu_{i,k}^{(2,2)}} + \frac{\lambda_{i,k}^{(5)}}{\mu_{i,k}^{(2,3)}} \\
&= \frac{\lambda_{i,0} \cdot P_{i,k}^{(i,0)} + \lambda_{0,i} \cdot P_{i,k}^{(0,i)}}{\mu_{i,k}^{(1)}} + \frac{\sum_{j=1...i-1,i+1...M} \lambda_{i,j} \cdot P_{i,k}^{(i,j)}}{\mu_{i,k}^{(2,1)}} \\
&+ \frac{\sum_{j=1...i-1,i+1...M} \sum_{m=1...K_j} \lambda_{j,i} \cdot P_{j,m}^{(j,i)} \cdot P_{i,k}^{(j,m,i)}}{\mu_{i,k}^{(2,2)}} \\
&+ \frac{\sum_{j=1...i-1,i+1...M} \sum_{m=1...K_j} \lambda_{i,j} \cdot P_{i,k}^{(i,j)} \cdot P_{j,m}^{(i,k,j)} \cdot P_{i,k}^{(j,m,i)}}{\mu_{i,k}^{(2,3)}}
\end{aligned} \tag{6}$$

If the service rates are all equal, i.e.,:

$$\mu_{i,k}^{(1)} = \mu_{i,k}^{(2,1)} = \mu_{i,k}^{(2,2)} = \mu_{i,k}^{(2,3)} = \mu_{i,k} \tag{7}$$

the previous equation for $\rho_{i,k}$ can be simplified and it becomes:

$$\begin{aligned}
\rho_{i,k} &= \frac{\lambda_{i,k}^{(1)} + \lambda_{i,k}^{(2)} + \lambda_{i,k}^{(3)} + \lambda_{i,k}^{(4)} + \lambda_{i,k}^{(5)}}{\mu_{i,k}} \\
&= \frac{1}{\mu_{i,k}} \cdot [\lambda_{i,0} \cdot p_{i,k}^{(i,0)} + \lambda_{0,i} \cdot p_{i,k}^{(0,i)} \\
&+ \sum_{j=1...i-1,i+1...M} \lambda_{i,j} \cdot p_{i,k}^{(i,j)} \\
&+ \sum_{j=1...i-1,i+1...M} \sum_{m=1...K_j} \lambda_{j,i} \cdot p_{j,m}^{(j,i)} \cdot p_{i,k}^{(j,m,i)} \\
&+ \sum_{j=1...i-1,i+1...M} \sum_{m=1...K_j} \lambda_{i,j} \cdot p_{i,k}^{(i,j)} \cdot p_{j,m}^{(i,k,j)} \cdot p_{i,k}^{j,m,i)}]
\end{aligned} \tag{8}$$

In case the arrival process of the path computation requests is a Poisson process and the service time follows an exponential distribution, then, each $PCE_{i,k}$ follows the behavior of an M/M/1 system, and the solution of the system can be obtained using the product form. Therefore the probability of having n jobs, i.e., path computation requests, at $PCE_{i,k}$ is:

$$p_{i,k}^{(n)} = \rho_{i,k}^{n} \cdot (1 - \rho_{i,k}) \tag{9}$$

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

$$N_{i,k} = \frac{\rho_{i,k}}{1-\rho_{i,k}} \tag{10}$$

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

$$T_{i,k} = \frac{\rho_{i,k}}{\lambda_{i,k} \cdot (1-\rho_{i,k})} \tag{11}$$

The average number of customers in queue at $PCE_{i,k}$ is:

$$N^Q_{i,k} = \frac{\rho^2_{i,k}}{1-\rho_{i,k}} \tag{12}$$

And the average waiting time for a job in $PCE_{i,k}$ is:

$$W_{i,k} = \frac{\rho_{i,k}}{\mu_{i,k} - \lambda_{i,k}} \tag{13}$$

In case the arrival process is Poisson, but the service time is deterministic, i.e., each PCE behaves as an M/D/1 system, the product form is not the exact solution. However, as demonstrated in section V, it can be a viable approximation. As a result of the approximation of applying the product form, the following statistics can be derived.

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

$$N_{i,k} = \frac{\rho_{i,k} \cdot (2-\rho_{i,k})}{2 \cdot (1-\rho_{i,k})} \tag{14}$$

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

$$T_{i,k} = \frac{N_{i,k}}{\lambda_{i,k}} = \frac{1}{\lambda_{i,k}} \cdot \frac{\rho_{i,k} \cdot (2-\rho_{i,k})}{2 \cdot (1-\rho_{i,k})} \tag{15}$$

The average number of customers in queue at $PCE_{i,k}$ is:

$$N^Q_{i,k} = \lambda_{i,k} \cdot W_{i,k} \tag{16}$$

And the average waiting time for a job in $PCE_{i,k}$ is:

$$W_{i,k} = T_{i,k} - \frac{1}{\mu_{i,k}} \tag{17}$$

IV. Uneven PCE's: Balancing the Average Job Time

When reliability is a concern, multiple PCE's 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 wait;
- 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 $p_{i,2}$.

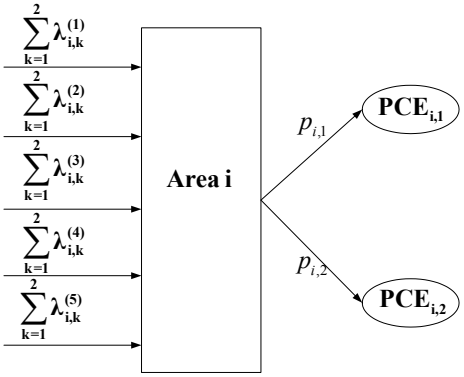

Fig. 4. Model used to balance the job time

Service rates at each PCE are independent of the type of path computation request as in equation 7. Without loss of generality it is assumed that $\mu_{i,1} \geq \mu_{i,2}$. The following notation is used:

- $\overline{X^2_{i,1}}$ is the second moment of the service time at $PCE_{i,1}$
- $\overline{X^2_{i,2}}$ is the second moment of the service time at $PCE_{i,2}$.

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

$$\begin{aligned}\Lambda &= \sum_{k=1}^{2} [\lambda^{(1)}_{i,k} + \lambda^{(2)}_{i,k} + \lambda^{(3)}_{i,k} + \lambda^{(4)}_{i,k} + \lambda^{(5)}_{i,k}] \\ &= p_{i,1} \cdot \Lambda + p_{i,2} \cdot \Lambda \end{aligned} \tag{18}$$

While the objective is to set:

$$T_{i,1} = T_{i,2} \tag{19}$$

Using the Pollaczek-Khinchin (P-K) formula [10], equation (19) becomes:

$$\frac{1}{\mu_{i,1}} + \frac{p_{i,1} \cdot \Lambda \cdot \overline{X^{(2)}_{i,1}}}{2 \cdot (1 - \frac{p_{i,1} \cdot \Lambda}{\mu_{i,1}})} = \frac{1}{\mu_{i,2}} + \frac{p_{i,2} \cdot \Lambda \cdot \overline{X^{(2)}_{i,2}}}{2 \cdot (1 - \frac{p_{i,2} \cdot \Lambda}{\mu_{i,2}})} \tag{20}$$

Combining equations (18) and (20) a second degree polynomial is obtained:

$$A \cdot (p_{i,1} \cdot \Lambda)^2 + B \cdot (p_{i,1} \cdot \Lambda) + C = 0 \tag{21}$$

where:

$$\begin{aligned} A =& 2 \cdot \mu_{i,2} - 2 \cdot \mu_{i,1} + \overline{X^2_{i,2}} \cdot \mu_{i,1} \cdot \mu^2_{i,2} \\ & - \overline{X^2_{i,1}} \cdot \mu^2_{i,1} \cdot \mu_{i,2} \end{aligned} \tag{22}$$

$$\begin{aligned} B =& 2 \cdot \mu^2_{i,1} - 4 \cdot \mu_{i,1} \cdot \mu_{i,2} + 2 \cdot \mu_{i,1} \cdot \Lambda + 2 \cdot \mu^2_{i,2} \\ & - 2 \cdot \mu_{i,2} \cdot \Lambda - \overline{X^2_{i,2}} \cdot \mu_{i,1} \cdot \mu^2_{i,2} \cdot \Lambda - \overline{X^2_{i,2}} \cdot \mu^2_{i,1} \cdot \mu^2_{i,2} \\ & - \overline{X^2_{i,1}} \cdot \mu^2_{i,1} \cdot \mu^2_{i,2} + \overline{X^2_{i,1}} \cdot \mu^2_{i,1} \cdot \mu_{i,2} \cdot \Lambda \end{aligned} \tag{23}$$

$$\begin{aligned} C =& \overline{X^2_{i,2}} \cdot \mu^2_{i,1} \cdot \mu^2_{i,2} \cdot \Lambda + 2 \cdot \mu^2_{i,1} \cdot \mu_{i,2} \\ & - 2 \cdot \mu^2_{i,1} \cdot \Lambda - 2 \cdot \mu_{i,1} \cdot \mu^2_{i,2} + 2 \cdot \mu_{i,1} \cdot \mu_{i,2} \cdot \Lambda \end{aligned} \tag{24}$$

Solving equation (21) the value for $p_{i,1}$ and $p_{i,2}$ can be found. If $A \neq 0$, there are two solutions, the first is:

$$p_{i,1}(1) = \frac{-B + \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A \cdot \Lambda} \tag{25}$$

$$p_{i,2}(1) = 1 - p_{i,1}(1) \tag{26}$$

and the second is:

$$p_{i,1}(2) = \frac{-B - \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A \cdot \Lambda} \tag{27}$$

$$p_{i,2}(2) = 1 - p_{i,1}(2) \tag{28}$$

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

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

$$p_{i,1} = p_{i,1}(1) \tag{29}$$

$$p_{i,2} = p_{i,2}(1) \tag{30}$$

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

$$p_{i,1} = p_{i,1}(2) \tag{31}$$

$$p_{i,2} = p_{i,2}(2) \tag{32}$$

- otherwise

$$p_{i,1} = 1 \tag{33}$$

$$p_{i,2} = 0 \tag{34}$$

In the special case where the service and the arrival process follow an exponential distribution, i.e., M/M/1 system, the product form is not an approximation. As a result the term A in equation (22) is zero and the result can be further simplified as follows:

- if $\Lambda > \mu_{i,1} - \mu_{i,2}$

$$p_{i,1} = \frac{\Lambda + \mu_{i,1} - \mu_{i,2}}{2 \cdot \Lambda} \tag{35}$$

$$p_{i,2} = \frac{\Lambda + \mu_{i,2} - \mu_{i,1}}{2 \cdot \Lambda} \tag{36}$$

- if $\Lambda < \mu_{i,1} - \mu_{i,2}$

$$p_{i,1} = 1 \tag{37}$$

$$p_{i,2} = 0 \tag{38}$$

V. Model Validation and Performance

The proposed analytical framework is validated against two simulation experiments:

- simulation of network of queues, and
- simulation of MPLS-TE protocol suite.

Both simulators are realized using OPNET. The following assumptions are made.

- each area (except area 0) hosts two PCE's, i.e., $K_i = 2, i = 1, \cdots, M$, This is the simplest PCE architecture in which PCE selection is required when requesting path computation in every area.
- The computational (service) time at PCE is deterministic. While this assumption may be overly too simplistic, it allows one to validate the analytical model using a service time other than random with exponential distribution, without requiring any elaborated assumption on the computational time distribution at the PCE.

Simulation of network of queues.

The layout of the queues in this simulator is same as the one described in the analytical product form model (Section III). Each PCE is modeled as an M/D/1 queue,and path computation requests are stored in the queue and serviced first in first out. Newly generated requests enter the network of queues forming a Poisson arrival process. The request's transient time between queues is set to zero. The sequence of queues which are visited by each request is chosen as follows.

Case 1: memory.

The source and destination areas of each newly generated request (job identity) are chosen randomly and uniformly (excluding area 0). The job identity is maintained (memory) throughout the job's lifetime in the network of queues, i.e., the sequence of queues visited by the job is one to one with the sequence of PCE's that are visited by the corresponding request in the real world network. At completion of service at the last queue of the sequence, the job departs from the network of queues.

Case 2: memoryless.

In this case the source area of the newly generated job is chosen randomly and uniformly. The first queue visited by the job is one of the two queues which represent the PCE's hosted by the source area. The queue is chosen using probabilities $p_{i,k}^{(i,j)}$. Upon departure from this queue, the job identity is lost (memoryless), and a new identity is given to the job. The new identity is chosen according to the distribution of jobs departing from the real world PCE, thus respecting the flow values from and in to every queue in the network. Similarly, upon departure from any subsequent queue, the job identity is chosen anew. The average number of queues (PCE's) visited by the job before departing from the network of queues is 3, as in the real world network.

The memory and memoryless cases are used to determine the effectiveness on results due to the independence approximation made in the product form. The advantage of having both simulations is that this will enable to better understand the effect of the use of the approximation of using the product form solution and its impact, i.e., is it the approximation of memoriless operation that introduces the largest discrepancy between model and simulation or is it the product form in the case of M/G/1?

Simulation of protocol suite.

In this second simulator, the complete MPLS-TE protocol suite along with PCEP is simulated, accounting for line propagation delay of control packets.

The following assumptions are used.

- All areas (except area 0) have 8 active PCC's.
- PCC's in area 0 are not active, i.e., they are neither source nor destination nodes for any LSP.
- All PCE's are not active PCC's.
- Traffic is uniform and symmetric, i.e., one LSP is created from every active PCC to every other active PCC (full bidirectional mesh). Consequently, both intra-area and inter-area LSP's are modeled.

- Each active PCC generates multiple path computation requests, one for each (inter-area) LSP originating at that PCC destined to reach another active PCC in a different area.
- Data bandwidth availability is assumed to be sufficient to prevent race conditions from taking place when reserving the LSP bandwidth. Thus, only one request is sent for each LSP to be reserved.
- The generation time of each request at PCC is uniformly and independently chosen over the simulation time duration. The simulation time duration is varied in the study to change the offered load value.
- Processing time at PCC is considered negligible.
- Average cumulative line propagation delay for a request is approximately 90 ms.
- Multiple simulation instances (5) are used to obtain confidence intervals that are below 17% at 95% level.

A. PCE's with Even Computational Capacity

The analytical model is first validated assuming that all PCE's have the same computational capacity when it comes to servicing the received path computation requests, i.e., $\mu_{i,k} = 37.5\text{s}^{-1}$, $\forall i = 1, \ldots, M,\ k = 1, 2$.

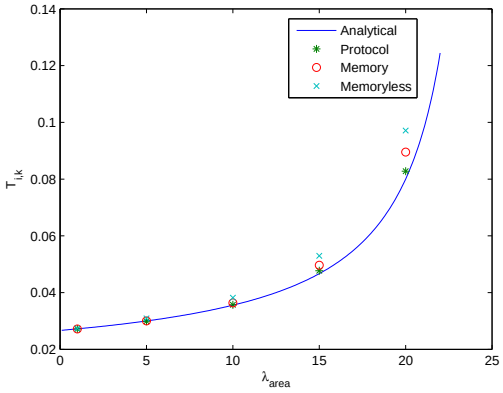

Fig. 5. $T_{i,k}$ for a setup with 10 areas and even computational capacity

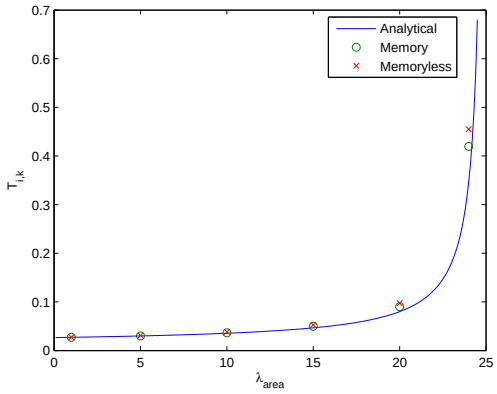

Fig. 6. $T_{i,k}$ for a setup with 50 areas and even computational capacity

Figs. 5 and 6, along with Table I show $T_{i,k}$ versus total arrival rate in each area λ_{area} ($\lambda_{area} = \sum_{k=1}^{2} \lambda_{i,k}^{(3)}$) for two number of areas M, $M = 10, 50$. For $M = 50$ simulation of the MPLS-TE protocol suite was not possible on the available desktops due to the 4 GB memory limitation. At low and medium load values simulation results and analytical model are in agreement. At high load ($\lambda_{area} \geq 15$, $\rho_{i,k} \geq 0.6$), the discrepancy across results can be as large as 21%.

More precisely, the values obtained using the memoryless network of queues are greater compared to the case with memory. The reason of this difference may be that in the memory case each request visits exactly 3 queues and then departs from the network. In the memoryless case, each request visits 3 queues on average, but the actual value may vary from request to request. The non-zero variance of the number of queues visited by each request in the memoryless case may be the cause for the increased values of $T_{i,k}$.

The numeric value of the results obtained using the protocol simulator are smaller compared to the value obtained using the simulator of the network of queues. The reason for this difference may be that in the protocol simulation, the total number of request arrivals is known and finite (i.e., one request for each LSP). Compared to the Poisson arrival process used in the network of queues, the bounded arrivals in the protocol simulator yield lower $T_{i,k}$ values.

The analytical model underestimates $T_{i,k}$ at high load values. The explanation of this result may be found in the product form assumption, which is an approximation when service time is not exponential[2]. The discrepancy of the model is 13% or better at load values below 15.

1) Simulation of protocol suite: This section present results obtained using the simulation of the protocol suite when all PCE's have the same computational power, i.e., service rate. The results shown plot the values for $\lambda_{i,k}$, $\rho_{i,k}$, $N_{i,k}$ $T_{i,k}$ of $PCE_{i,k}$, and L, where L is the latency, i.e., the total time spent in the 3 queues (this does not include propagation time).

This first set of plots is obtained when the network has 3 areas in addition to area 0.

[2]When running the same set of experiments but using random service time exponentially distributed the discrepancy is within the confidence interval of the simulation runs.

TABLE I

ANALYTICAL MODEL VALIDATION FOR QUEUE SIMULATION AND PROTOCOL SIMULATION WITH FULLY SYMMETRIC TRAFFIC IN THE NETWORK.

Area	λ_{area}	Queue Simulation				Protocol Simulation				Analytical		Difference %			
		$N_{i,k}$(M)	Conf %	$T_{i,k}$(M)	Conf %	$N_{i,k}$(pro)	Conf %	$T_{i,k}$(pro)	Conf %	$N_{i,k}$(M)	$T_{i,k}$(M)	$N_{i,k}$(M)	$T_{i,k}$(M)	$N_{i,k}$(pro)	$T_{i,k}$(pro)
10	1	0.041	0.696	0.027	0.022	0.041	0.210	0.027	0.076	0.041	0.027	0.069	-0.111	0.069	0.000
	5	0.225	0.266	0.030	0.118	0.225	0.219	0.030	0.365	0.225	0.030	-0.015	0.043	-0.015	-0.333
	10	0.546	0.243	0.036	0.157	0.536	2.101	0.036	2.084	0.533	0.036	2.347	2.199	0.506	0.281
	15	1.117	0.556	0.050	0.401	1.074	3.646	0.048	3.750	1.050	0.047	6.418	6.311	2.286	2.141
	20	2.686	0.870	0.090	0.729	2.482	5.877	0.083	5.477	2.400	0.080	11.913	11.923	3.417	3.500
50	1	0.041	0.274	0.027	0.020	-	-	-	-	0.041	0.027	-0.107	-0.088	-	-
	5	0.226	0.232	0.030	0.044	-	-	-	-	0.225	0.030	0.253	0.260	-	-
	10	0.548	0.161	0.037	0.067	-	-	-	-	0.533	0.036	2.762	2.596	-	-
	15	1.122	0.316	0.050	0.211	-	-	-	-	1.050	0.047	6.843	6.744	-	-
	20	2.706	0.187	0.090	0.173	-	-	-	-	2.400	0.080	12.750	12.726	-	-
	24	15.110	2.387	0.420	2.298	-	-	-	-	12.480	0.347	21.074	20.998	-	-

Area	λ_{area}	Queue Simulation				Analytical		Difference %	
		$N_{i,k}$(NM)	Conf %	$T_{i,k}$(NM)	Conf %	$N_{i,k}$(NM)	$T_{i,k}$(NM)	$N_{i,k}$(NM)	$T_{i,k}$(NM)
10	1	0.041	1.160	0.027	0.103	0.041	0.027	0.653	0.595
	5	0.232	0.462	0.031	0.226	0.225	0.030	2.973	3.081
	10	0.573	0.482	0.038	0.207	0.533	0.036	7.425	7.295
	15	1.191	0.685	0.053	0.505	1.050	0.047	13.417	13.404
	20	2.915	0.810	0.097	0.688	2.400	0.080	21.460	21.377
50	1	0.041	0.581	0.027	0.055	0.041	0.027	0.473	0.617
	5	0.232	0.313	0.031	0.092	0.225	0.030	3.111	3.052
	10	0.572	0.293	0.038	0.126	0.533	0.036	7.337	7.219
	15	1.191	0.345	0.053	0.259	1.050	0.047	13.443	13.316
	20	2.925	0.663	0.097	0.599	2.400	0.080	21.894	21.835
	24	16.392	2.725	0.455	2.654	12.480	0.347	31.344	31.266

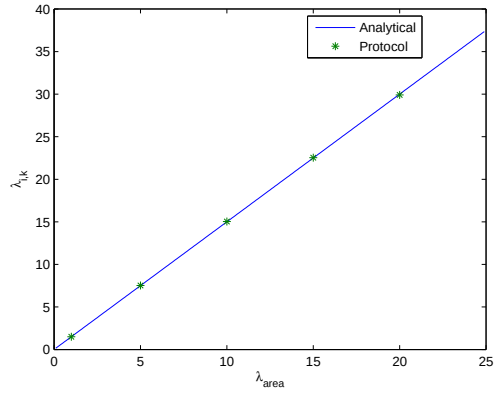

Fig. 7. $\lambda_{i,k}$, 3 areas, even PCE computational capacity

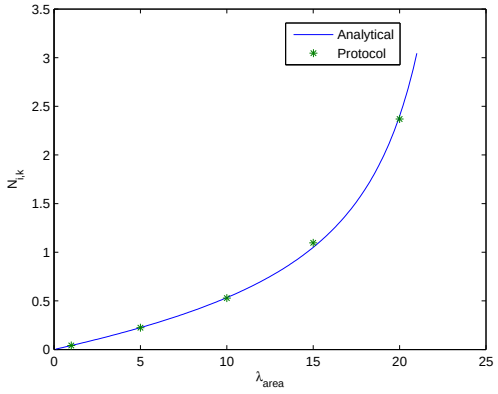

Fig. 9. $N_{i,k}$, 3 areas, even PCE computational capacity

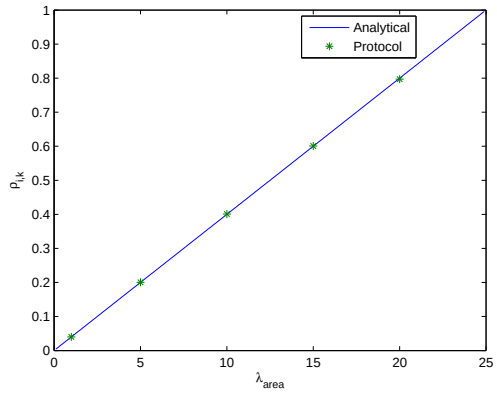

Fig. 8. $\rho_{i,k}$, 3 areas, even PCE computational capacity

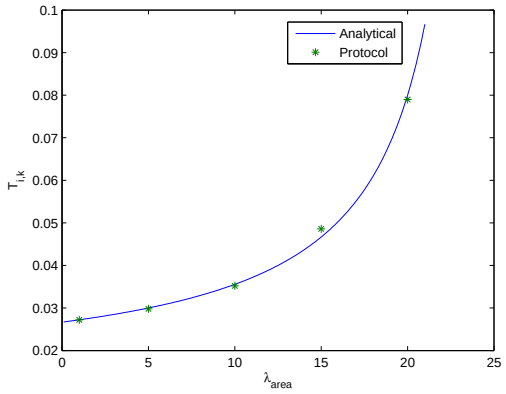

Fig. 10. $T_{i,k}$ 3 areas, even PCE computational capacity

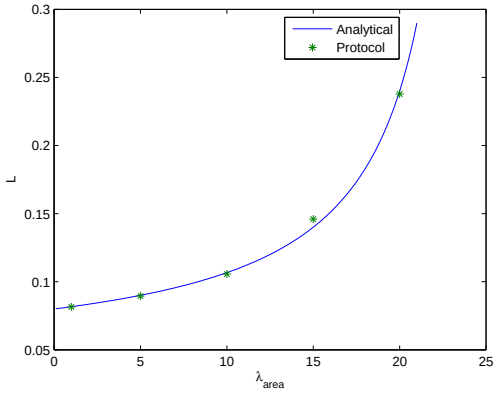

Fig. 11. L, 3 areas, even PCE computational capacity

This next set of plots is obtained with 10 areas in addition to area 0.

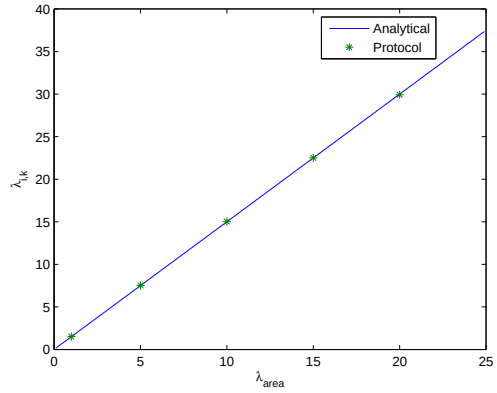

Fig. 12. $\lambda_{i,k}$ 10 areas, even PCE computational capacity

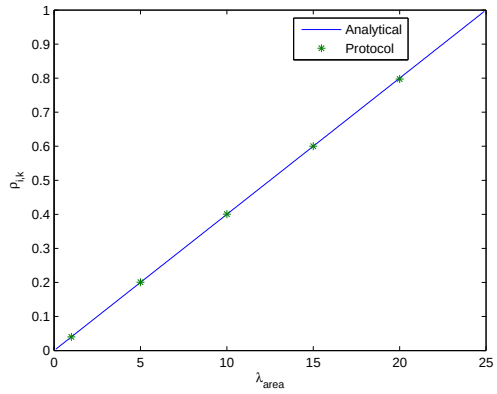

Fig. 13. $\rho_{i,k}$, 10 areas, even PCE computational capacity

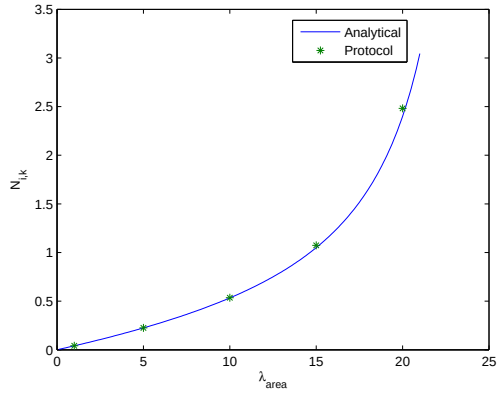

Fig. 14. $N_{i,k}$, 10 areas, even PCE computational capacity

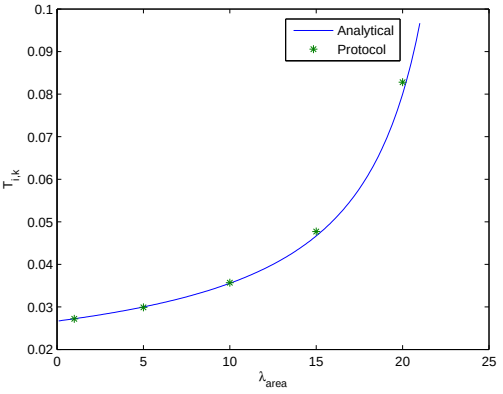

Fig. 15. $T_{i,k}$, 10 areas, even PCE computational capacity

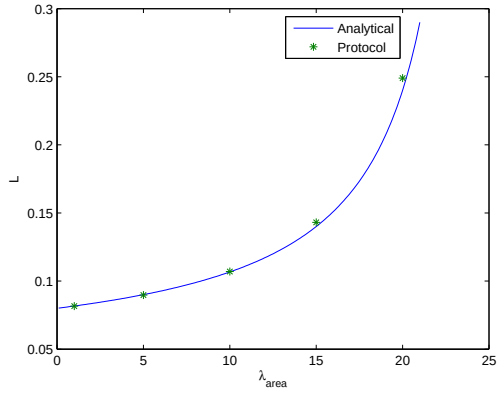

Fig. 16. L, 10 areas, even PCE computational capacity

2) Simulation of network of queues: Case 1: M/D/1 Model. This section reports results for the network of queues based simulation, both for the memory and memoryless cases, when the service time at the PCE is modeled as an M/D/1 system.

Here, the results for memory and memoryless cases are presented on the same chart of $N_{i,k}$ and $T_{i,k}$ in order to compare their similarity.

First, a network that consists of 3 areas in addition to area 0 is considered.

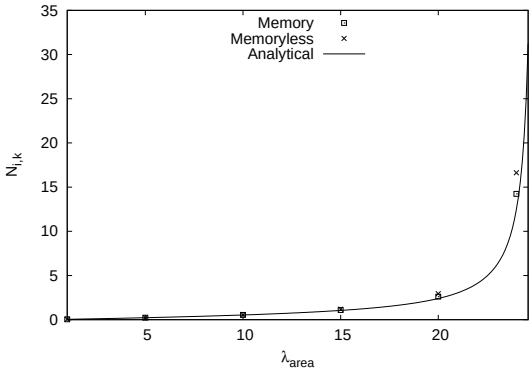

Fig. 17. $N_{i,k}$, 3 areas, even PCE computational capacity

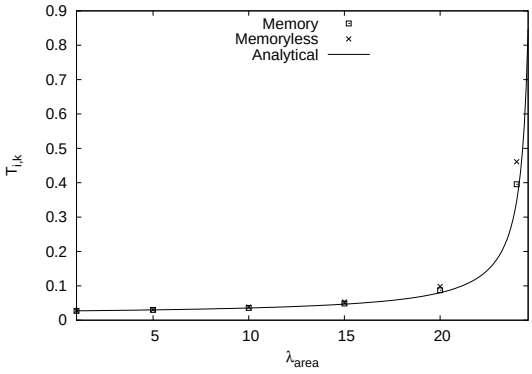

Fig. 18. $T_{i,k}$, 3 areas, even PCE computational capacity

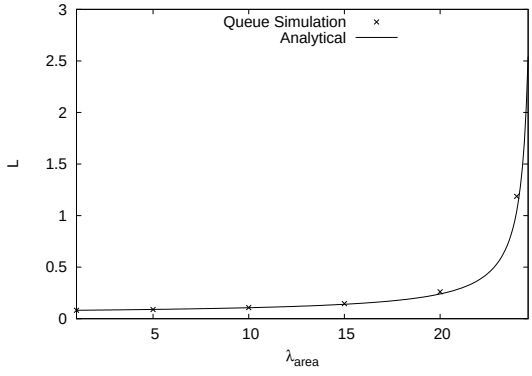

Fig. 19. L, 3 areas, even PCE computational capacity, memory case

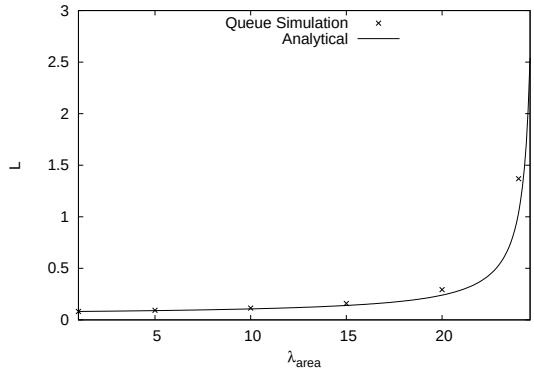

Fig. 20. L, 3 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields very similar results for the 3 areas. And L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

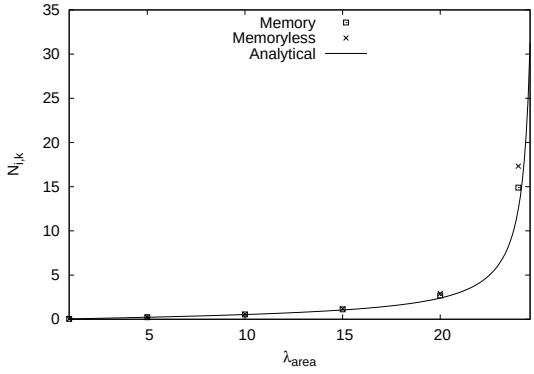

Fig. 21. $N_{i,k}$, 10 areas, even PCE computational capacity

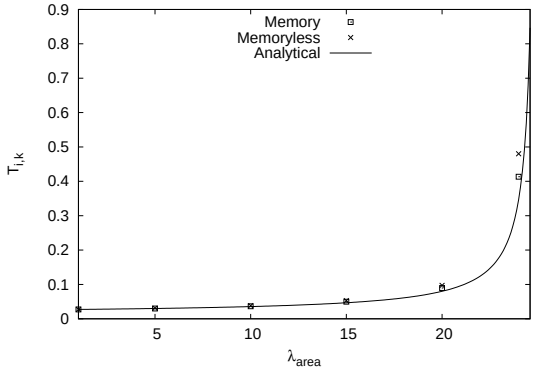

Fig. 22. $T_{i,k}$, 10 areas, even PCE computational capacity

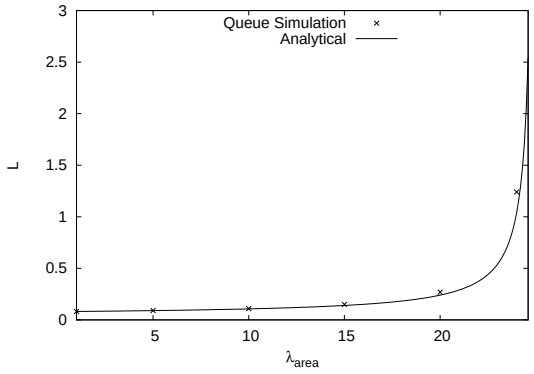

Fig. 23. L, 10 areas, even PCE computational capacity, memory case

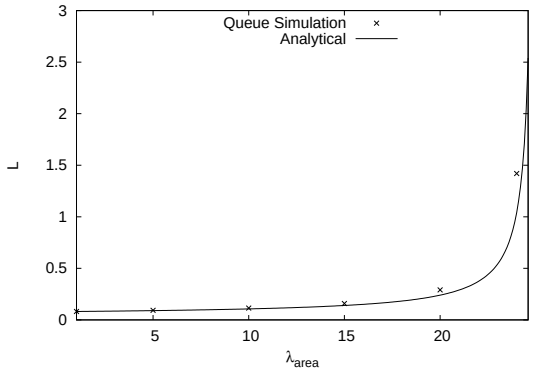

Fig. 24. L, 10 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases still yields similar results for the 10 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 22 areas in addition to area 0 is considered.

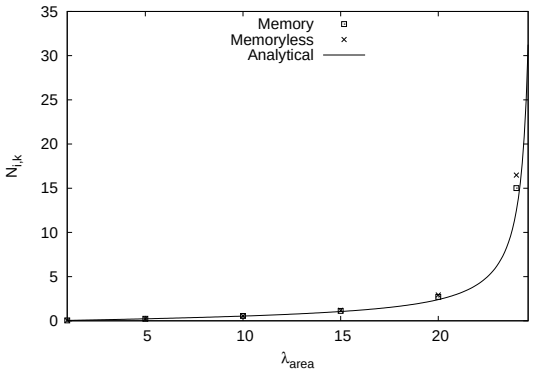

Fig. 25. $N_{i,k}$, 22 areas, even PCE computational capacity

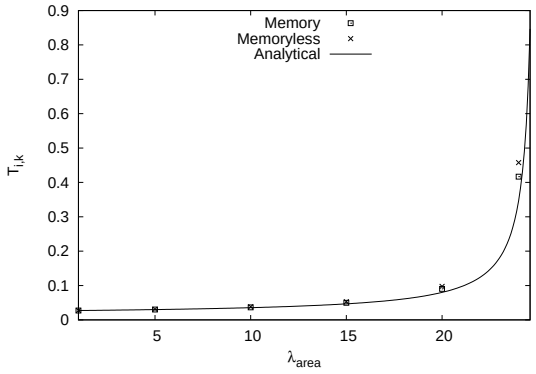

Fig. 26. $T_{i,k}$, 22 areas, even PCE computational capacity

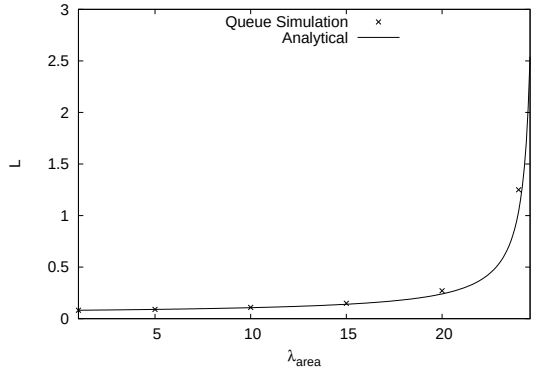

Fig. 27. L, 22 areas, even PCE computational capacity, memory case

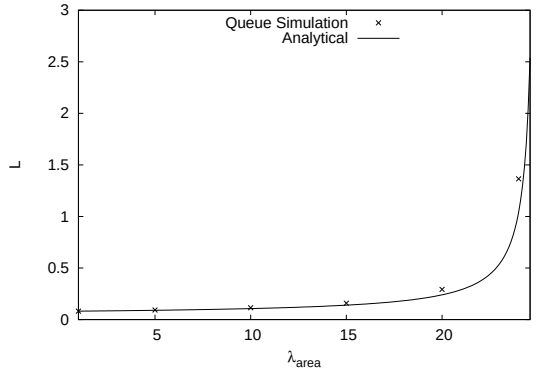

Fig. 28. L, 22 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 22 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 50 areas in addition to area 0 is considered.

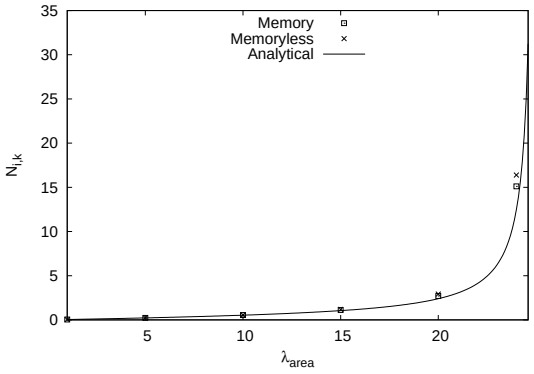

Fig. 29. $N_{i,k}$, 50 areas, even PCE computational capacity

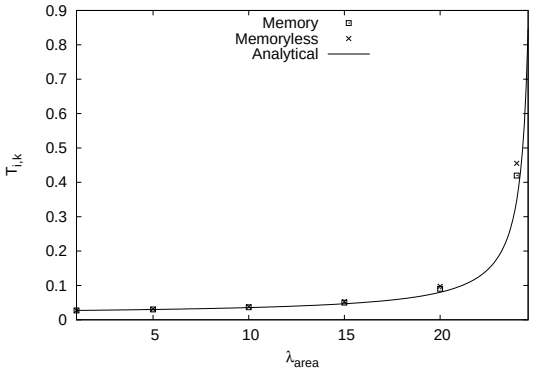

Fig. 30. $T_{i,k}$, 50 areas, even PCE computational capacity

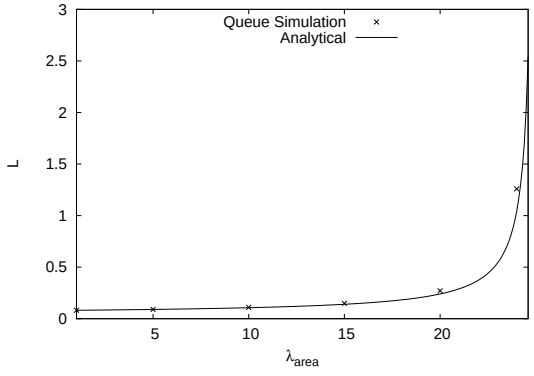

Fig. 31. L, 50 areas, even PCE computational capacity, memory case

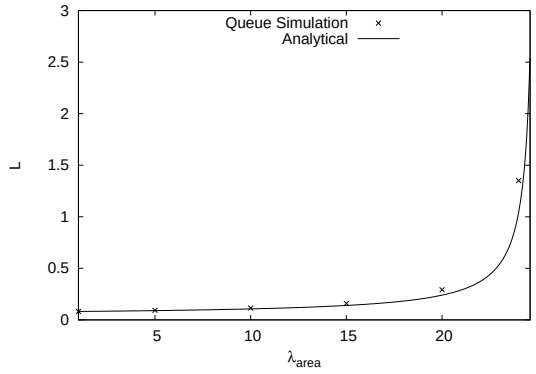

Fig. 32. L, 50 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 22 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Case 2: M/M/1 Model. This section reports results for the network of queues based simulation, both for the memory and memoryless cases, when the service time at the PCE is modeled as an M/M/1 system. First, a network that consists of 3 areas in addition to area 0 is considered.

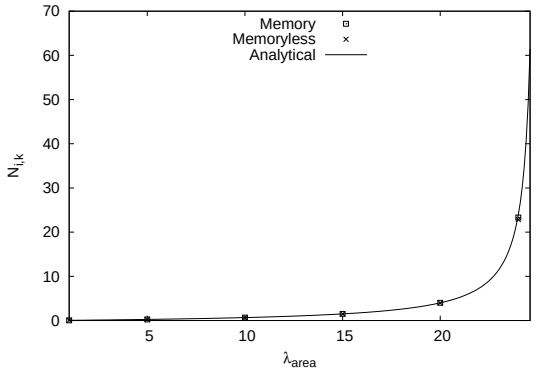

Fig. 33. $N_{i,k}$, 3 areas, even PCE computational capacity

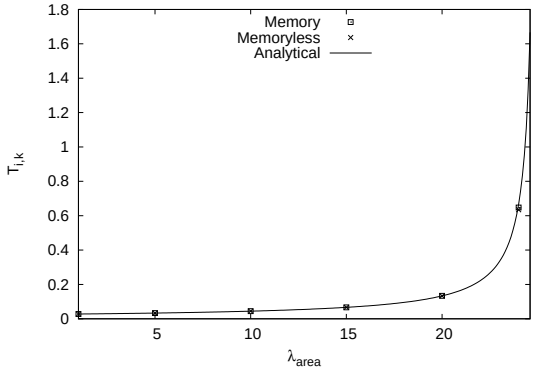

Fig. 34. $T_{i,k}$, 3 areas, even PCE computational capacity

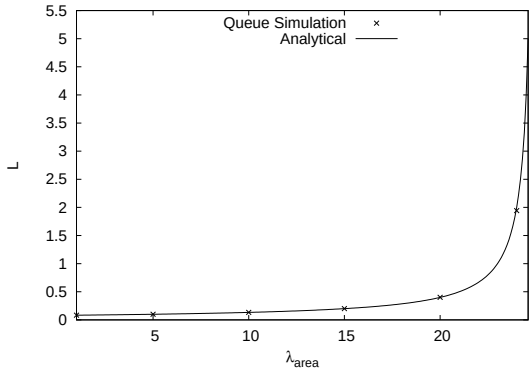

Fig. 35. L, 3 areas, even PCE computational capacity, memory case

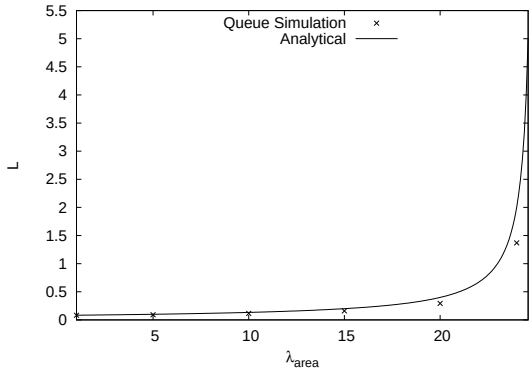

Fig. 36. L, 3 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 3 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

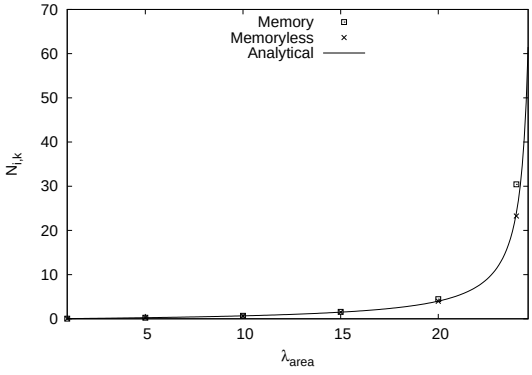

Fig. 37. $N_{i,k}$, 10 areas, even PCE computational capacity

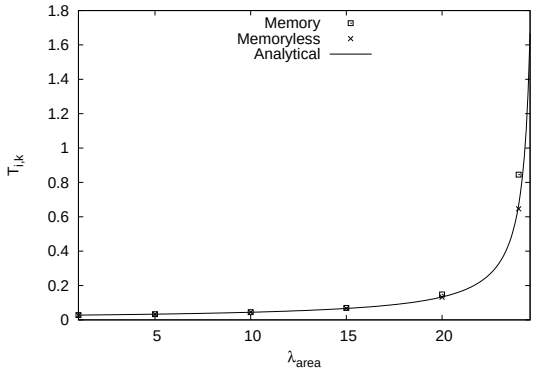

Fig. 38. $T_{i,k}$, 10 areas, even PCE computational capacity

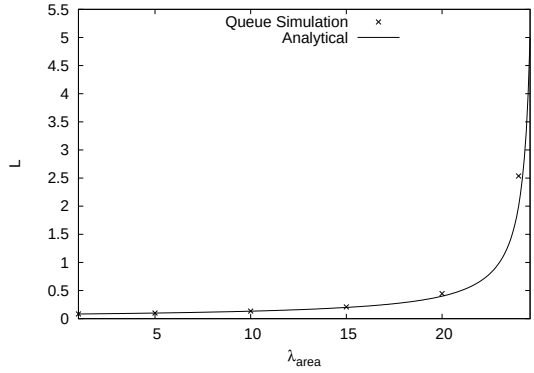

Fig. 39. L, 10 areas, even PCE computational capacity, memory case

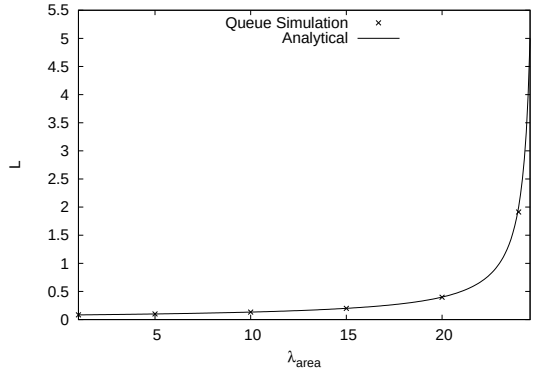

Fig. 40. L, 10 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 10 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 22 areas in addition to area 0 is considered.

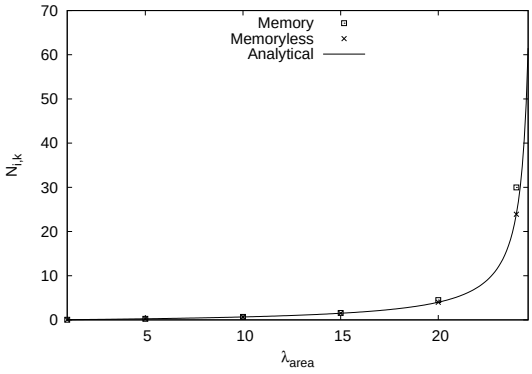

Fig. 41. $N_{i,k}$, 22 areas, even PCE computational capacity

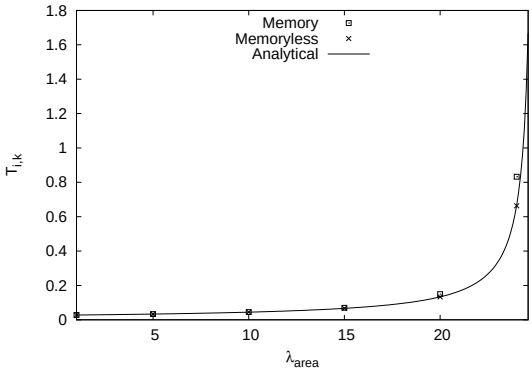

Fig. 42. $T_{i,k}$, 22 areas, even PCE computational capacity

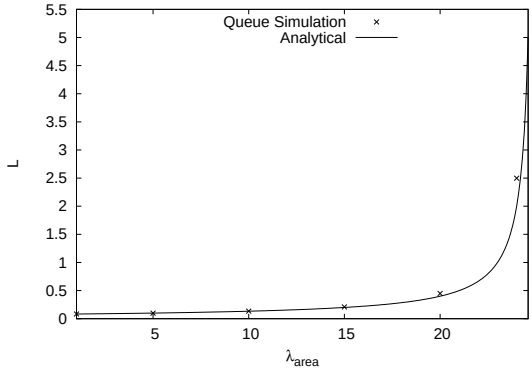

Fig. 43. L, 22 areas, even PCE computational capacity, memory case

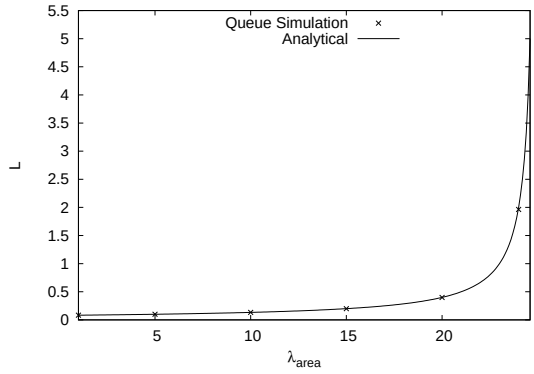

Fig. 44. L, 22 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 22 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

Next, the case with a network consisting of 50 areas in addition to area 0 is considered.

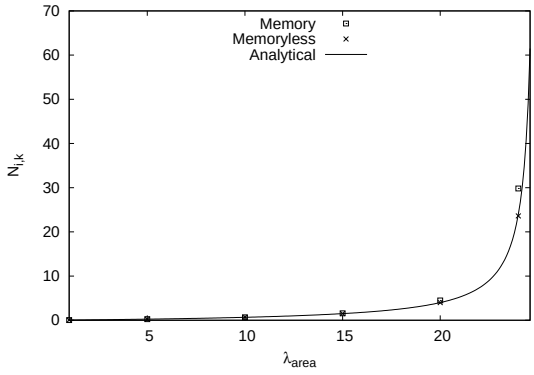

Fig. 45. $N_{i,k}$, 50 areas, even PCE computational capacity

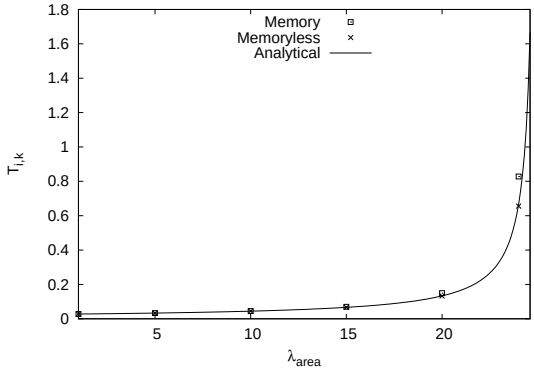

Fig. 46. $N_{i,k}$, 50 areas, even PCE computational capacity

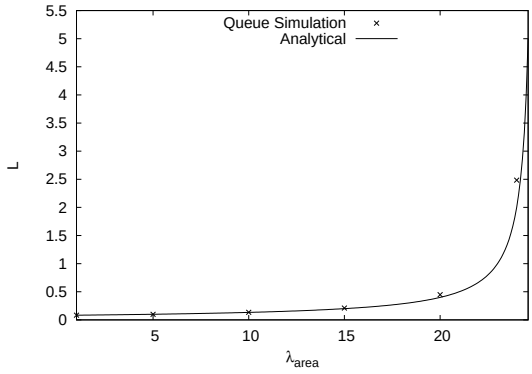

Fig. 47. L, 50 areas, even PCE computational capacity, memory case

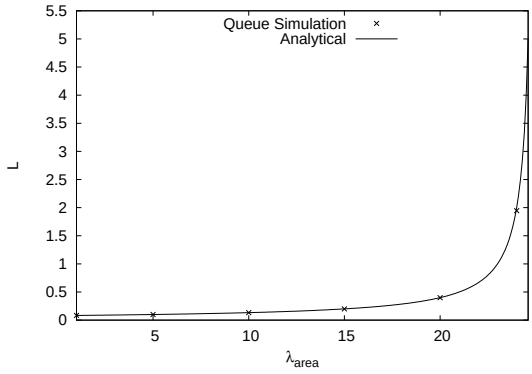

Fig. 48. L, 50 areas, even PCE computational capacity, memoryless case

The $N_{i,k}$ and $T_{i,k}$ figures show that the memory and memoryless cases yields similar results for the 50 areas. L retrieved from memory and memoryless model agrees with our analytical model. This can be used to prove the memoryless case is a successful approximation of the memory case.

B. PCE's with Uneven Computational Capacity

In this section, the results presented, refer to the case in which each area has 2 PCE's, in each area , the PCE's have different computational capacity. In particular, the faster PCE in area i ($PCE_{i,1}$) is characterized by twice the computational capacity of the slower PCE in area i ($PCE_{i,2}$), i.e., $\mu_{i,1} = 2\mu_{i,2} = 37.5\text{s}^{-1}$, $\forall i = 1, \ldots, M$.

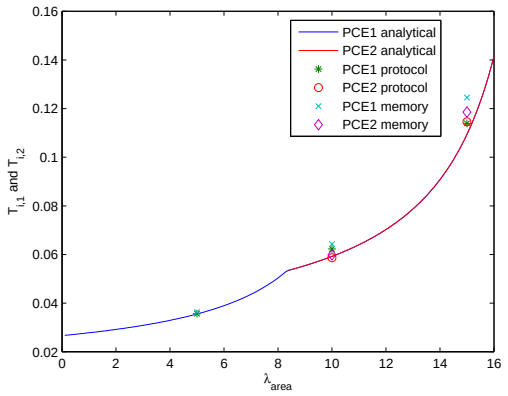

Fig. 49. $T_{i,1} and T_{i,2}$, 10 areas, uneven computational capacity

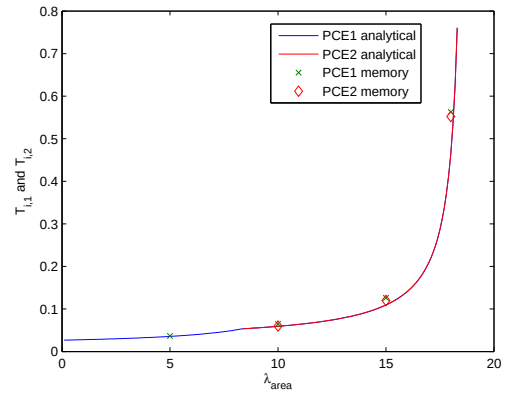

Fig. 50. $T_{i,1} and T_{i,2}$, 50 areas, uneven computational capacity

Figs. 49, 50, and table II report the values of $T_{i,1}$ and $T_{i,2}$ versus the value of λ_{area} for two cases: $M = 10, 50$ areas. As already mentioned, for $M = 50$ simulation of the MPLS-TE suite was not possible on the available desktops due to their memory limitation. At low and medium load values simulation results and analytical model are once again in full agreement. At high load ($\lambda_{area} \geq 15$,$\rho_{i,1} \geq 0.86$, $\rho_{i,2} \geq 0.68$), the discrepancy across the results can be as high as 21%. However, the accuracy of the model is 15% or better at load values below 15. The differences across the analytical model and simulation (queues with memory and protocol) results are consistent with those already observed for the even PCE computational capacity case.

When comparing the time that requests spent at fast and slow PCE's, the PCE selection policy based on the product form assumption yields a well balanced latency. The difference in values between $T_{i,1}$ and $T_{i,2}$ is within the confidence interval values.

For the detailed simulation results,we give the $\lambda_{i,1}$, $\rho_{i,1}$, $N_{i,1}$ and $T_{i,1}$ of $PCE_{i,1}$ and $\lambda_{i,2}$, $\rho_{i,2}$, $N_{i,2}$ and $T_{i,2}$ of $PCE_{i,2}$, and also give packet average latency.

1) Simulation of protocol suite: First, the case with a network consisting of 3 areas in addition to area 0 is considered.

TABLE II

ANALYTICAL MODEL VALIDATION FOR QUEUE SIMULATION AND PROTOCOL SIMULATION WITH LOAD-BALANCING IN THE NETWORK.

Area	λ_{area}	Queue Simulation				Protocol Simulation				Analytical		Difference %			
		$N_{i,1}$	Conf %	$T_{i,1}$	Conf %	$N_{i,1}$(Pro)	Conf %	$T_{i,1}$(Pro)	Conf %	$N_{i,1}$	$T_{i,1}$	$N_{i,1}$	$T_{i,1}$	$N_{i,1}$(Pro)	$T_{i,1}$(Pro)
10	5	0.544	0.508	0.036	0.278	0.535	2.377	0.036	2.275	0.533	0.036	2.098	1.966	0.394	0.281
	10	1.711	1.043	0.064	0.846	1.656	2.355	0.062	2.514	1.576	0.059	8.554	8.615	5.109	5.236
	15	4.023	2.081	0.125	1.828	3.668	17.372	0.114	17.352	3.533	0.109	13.893	13.803	3.836	4.113
50	5	0.547	0.276	0.037	0.124	-	-	-	-	0.533	0.036	2.550	2.528	-	-
	10	1.722	0.485	0.065	0.387	-	-	-	-	1.576	0.059	9.252	9.291	-	-
	15	4.075	0.475	0.126	0.446	-	-	-	-	3.533	0.109	15.346	15.265	-	-
	18	20.502	3.173	0.563	3.105	-	-	-	-	16.906	0.465	21.273	21.206	-	-
Area	λ_{area}	Queue Simulation				Protocol Simulation				Analytical		Difference %			
		$N_{i,2}$	Conf %	$T_{i,2}$	Conf %	$N_{i,2}$(Pro)	Conf %	$T_{i,2}$(Pro)	Conf %	$N_{i,2}$	$T_{i,2}$	$N_{i,2}$	$T_{i,2}$	$N_{i,2}$(Pro)	$T_{i,2}$(Pro)
10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	0.203	0.851	0.060	0.259	0.201	4.700	0.059	1.297	0.201	0.059	0.995	1.182	0.000	-0.845
	15	1.505	0.969	0.119	0.809	1.452	11.188	0.115	11.316	1.390	0.109	8.318	8.410	4.439	4.753
50	5	0	0	0	0	-	-	-	-	0	0	0	0	-	-
	10	0.203	0.282	0.060	0.084	-	-	-	-	0.201	0.059	1.094	1.182	-	-
	15	1.522	0.679	0.120	0.495	-	-	-	-	1.390	0.109	9.541	9.415	-	-
	18	9.730	2.497	0.552	2.366	-	-	-	-	8.180	0.465	18.952	18.902	-	-

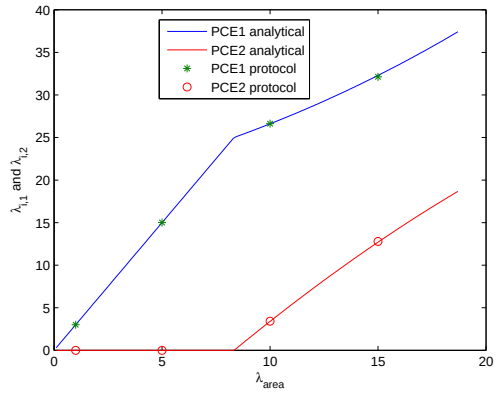

Fig. 51. $\lambda_{i,1}$ and $\lambda_{i,2}$, 3 areas, uneven computational capacity

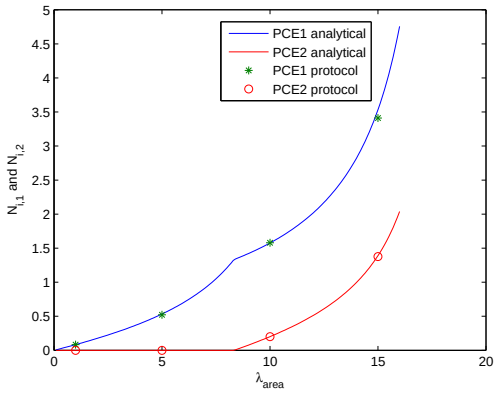

Fig. 53. $N_{i,1}$ and $N_{i,2}$, 3 areas, uneven computational capacity

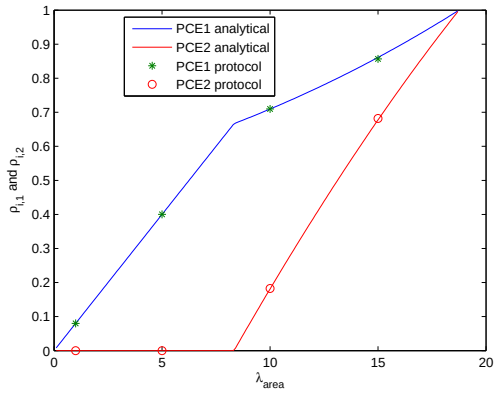

Fig. 52. $\rho_{i,1}$ and $\rho_{i,2}$, 3 areas, uneven computational capacity

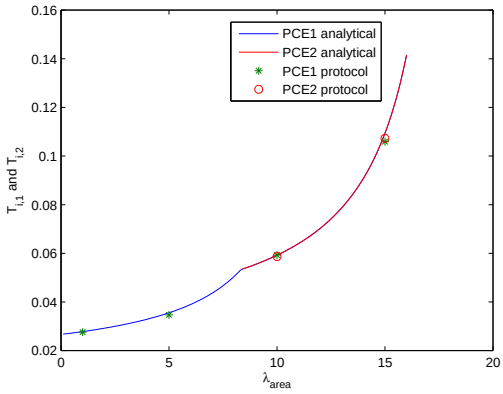

Fig. 54. $T_{i,1}$ and $T_{i,2}$, 3 areas, uneven computational capacity

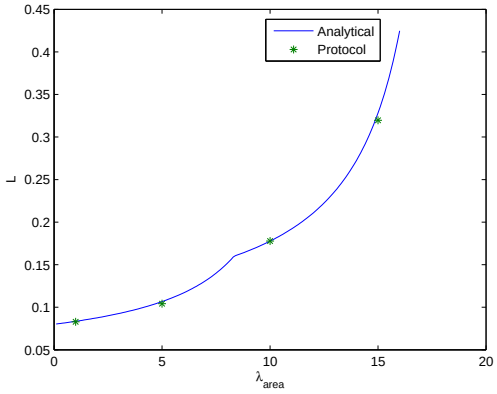

Fig. 55. L, 3 areas, uneven computational capacity

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

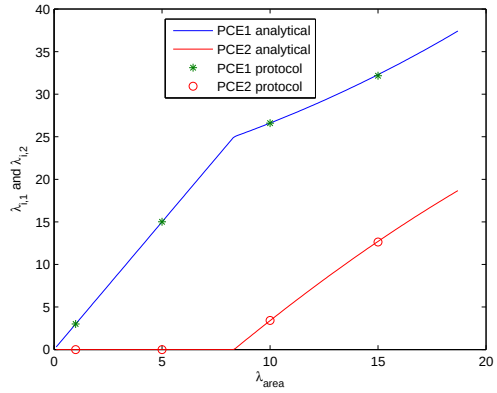

Fig. 56. $\lambda_{i,1}$ and $\lambda_{i,2}$, 10 areas, uneven computational capacity

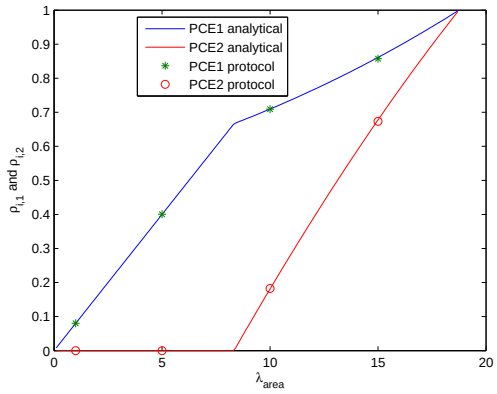

Fig. 57. $\rho_{i,1}$ and $\rho_{i,2}$, 10 areas, uneven computational capacity

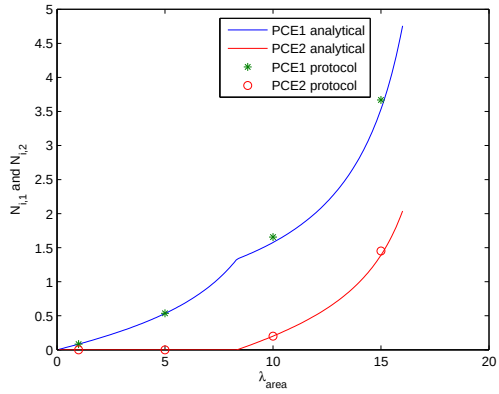

Fig. 58. $N_{i,1}$ and $N_{i,2}$, 10 areas, uneven computational capacity

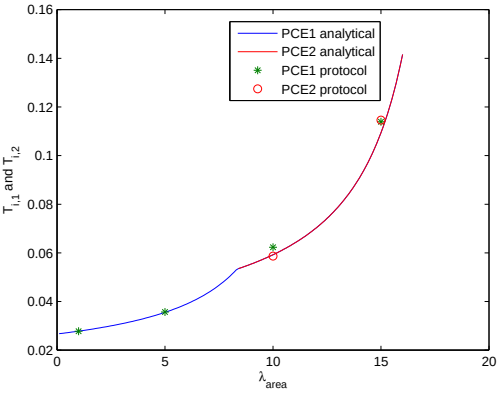

Fig. 59. $T_{i,1}$ and $T_{i,2}$, 10 areas, uneven computational capacity

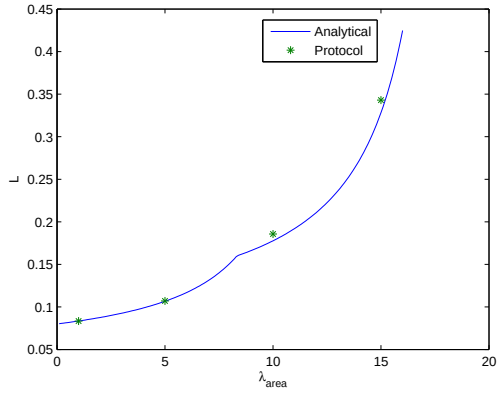

Fig. 60. L, 10 areas, uneven computational capacity

Next, results obtained when the balancing selection scheme is applied are compared to results where no balancing is applied, i.e., the balancing selection is based on minimizing the average expected latencies in path computation, and no balancing selection is based on randomly select PCE (50% and 50% to each PCE). The two schemes are labeled by "Random" and "Bal".

First, the case with a network consisting of 3 areas in addition to area 0 is considered.

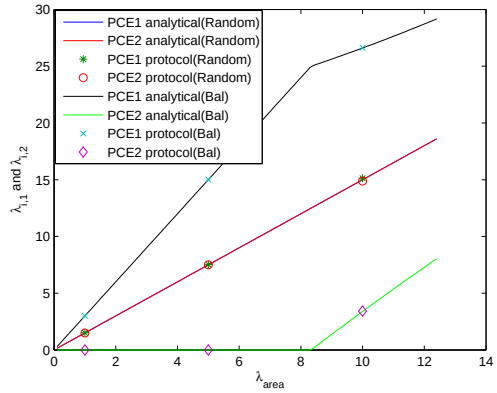

Fig. 61. $\lambda_{i,1}$ and $\lambda_{i,2}$, 3 areas, with and without balancing

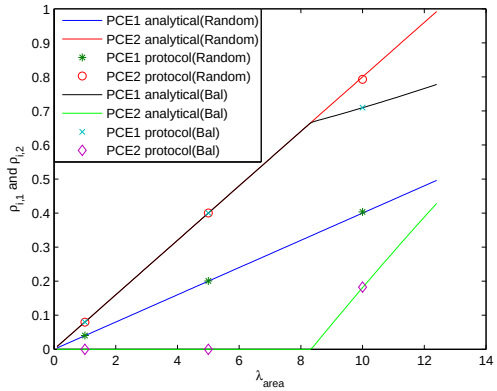

Fig. 62. $\rho_{i,1}$ and $\rho_{i,2}$, 3 areas, with and without balancing

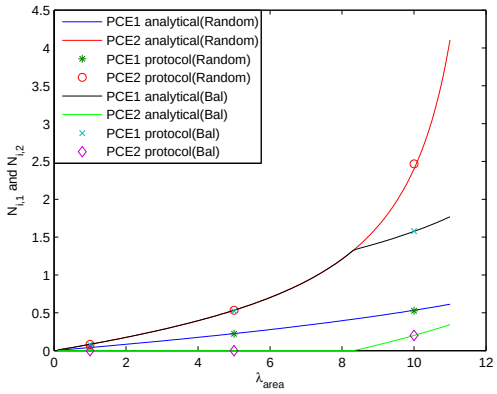

Fig. 63. $N_{i,1}$ and $N_{i,2}$, 3 areas with and without balancing

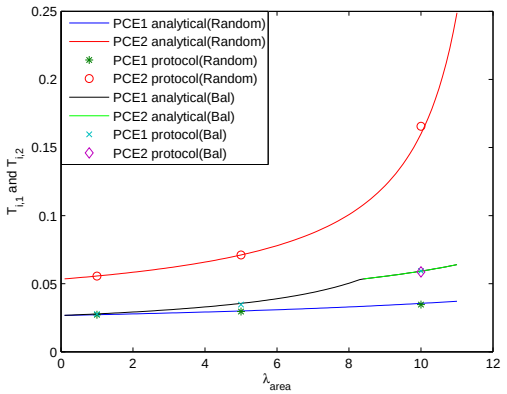

Fig. 64. $T_{i,1}$ and $T_{i,2}$, 3 areas with and without balancing

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

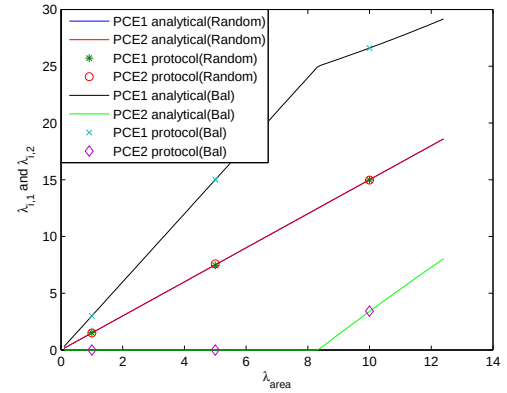

Fig. 65. $\lambda_{i,1}$ and $\lambda_{i,2}$, 10 areas, with and without balancing

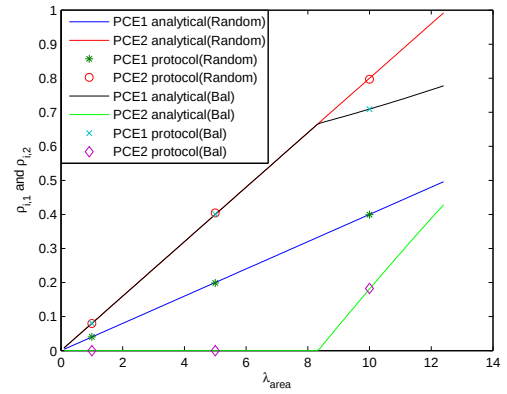

Fig. 66. $\rho_{i,1}$ and $\rho_{i,2}$, 10 areas, with and without balancing

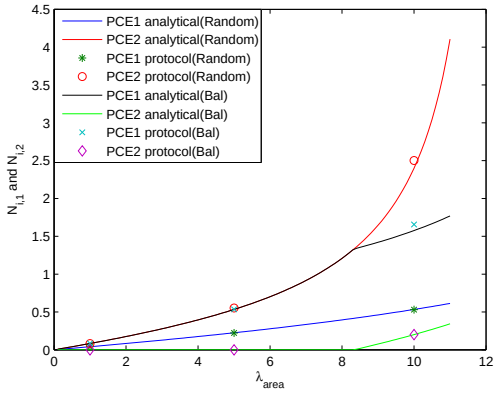

Fig. 67. $N_{i,1}$ and $N_{i,2}$, 10 areas, with and without balancing

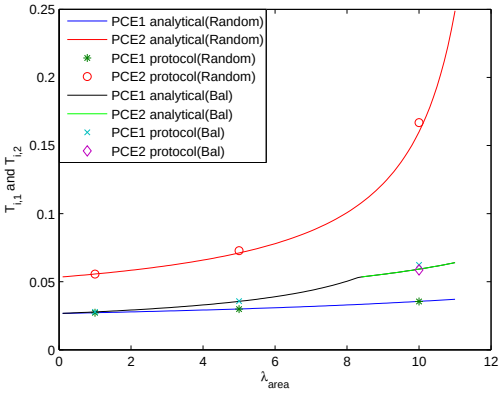

Fig. 68. $T_{i,1}$ and $T_{i,2}$, 10 areas, with and without balancing

2) Simulation of network of queues: Case 1: Fast Queue vs. Slow Queue under load-balancing. Here, only the load-balancing strategy performance between the two uneven PCE queues are compared: For 3 areas network of queues simulation, the $N_{i,1}$, $T_{i,1}$ of $PCE_{i,1}$, and the $N_{i,2}$, $T_{i,2}$ of $PCE_{i,2}$ are given, as well as the figure for packet average latency–L, where L means the total time spent in the 3 queues.

First, the case with a network consisting of 3 areas in addition to area 0 is considered.

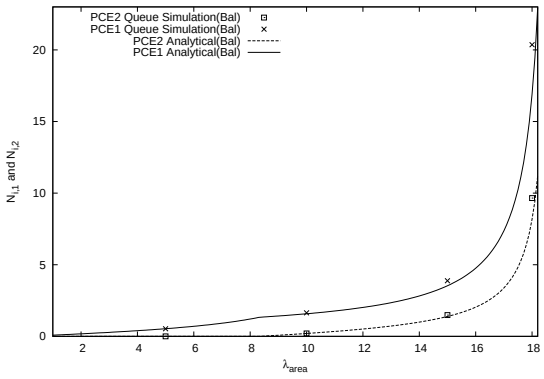

Fig. 69. $N_{i,1}$ and $N_{i,2}$, 3 areas, uneven computational capacity

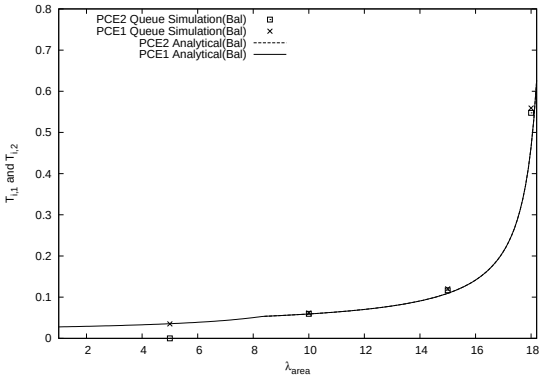

Fig. 70. $T_{i,1}$ and $T_{i,2}$, 3 areas, uneven computational capacity

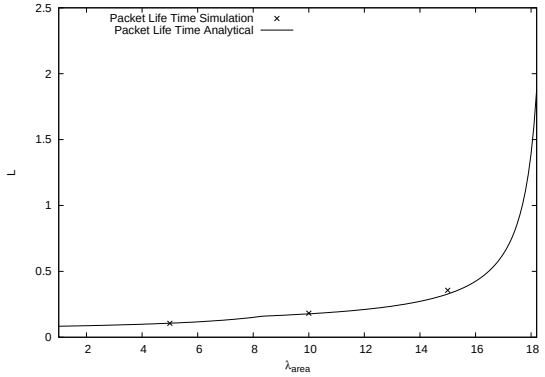

Fig. 71. L, 3 areas, uneven computational capacity

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

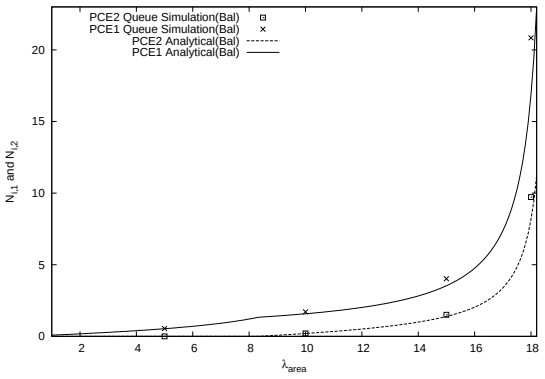

Fig. 72. $N_{i,1}$ and $N_{i,2}$, 10 areas, uneven computational capacity

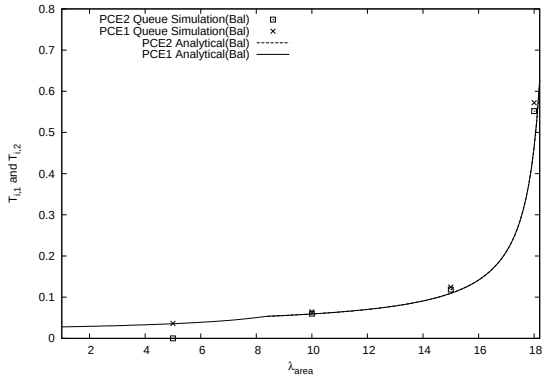

Fig. 73. $T_{i,1}$ and $T_{i,2}$, 10 areas, uneven computational capacity

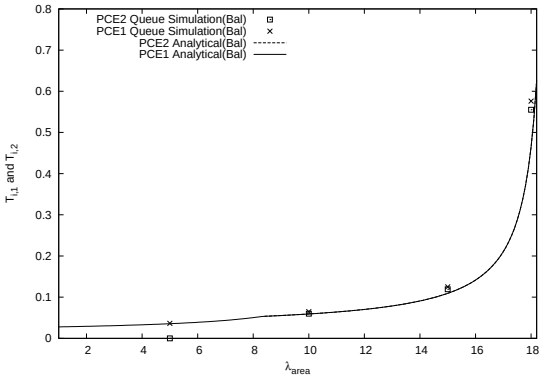

Fig. 76. $T_{i,1}$ and $T_{i,2}$, 22 areas, uneven computational capacity

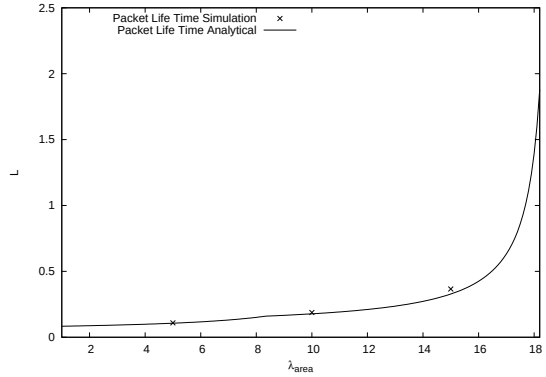

Fig. 74. L, 10 areas, uneven computational capacity

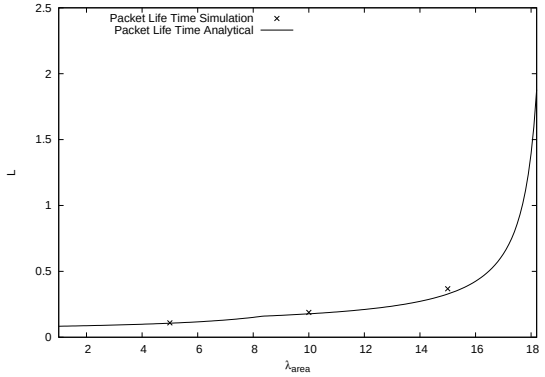

Fig. 77. L, 3 areas, uneven computational capacity

Next, the case with a network consisting of 22 areas in addition to area 0 is considered.

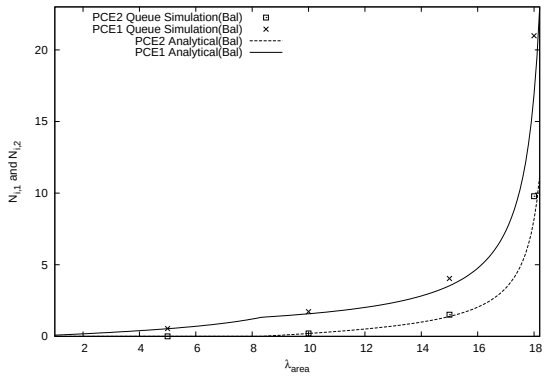

Fig. 75. $N_{i,1}$ and $N_{i,2}$, 22 areas, uneven computational capacity

Next, the case with a network consisting of 50 areas in addition to area 0 is considered.

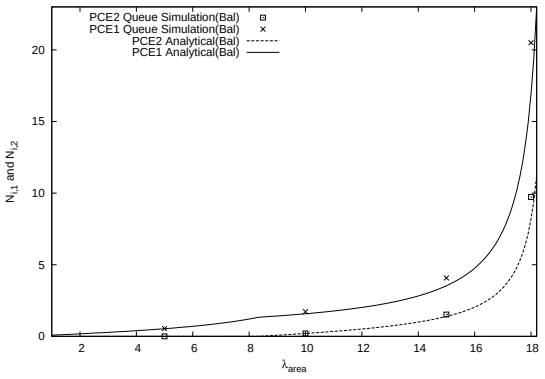

Fig. 78. $N_{i,1}$ and $N_{i,2}$, 50 areas, uneven computational capacity

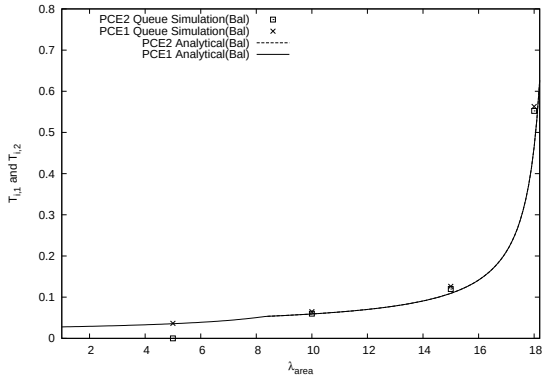

Fig. 79. $T_{i,1}$ and $T_{i,2}$, 50 areas, uneven computational capacity

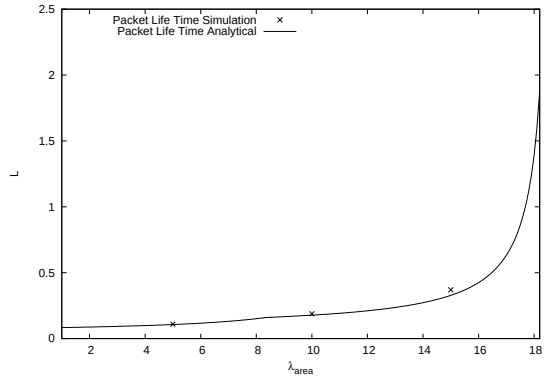

Fig. 80. L, 50 areas, uneven computational capacity

Case 2: load-balancing vs. no load-balancing.

Next, results obtained when the balancing selection scheme is applied are compared to results where no balancing is applied, i.e., the balancing selection is based on minimizing the average expected latencies in path computation, and no balancing selection is based on randomly select PCE (50% and 50% to each PCE). The two schemes are labeled by "Random-Simulation" and "with load-Simulation".

First, the case with a network consisting of 3 areas in addition to area 0 is considered.

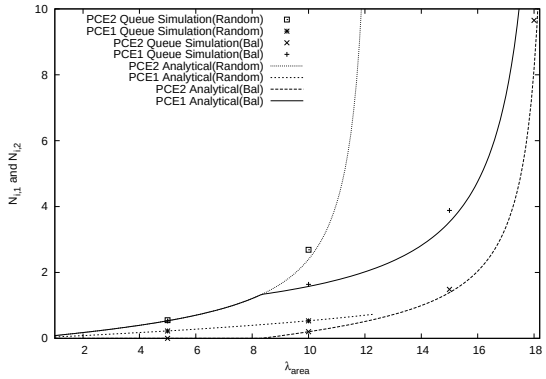

Fig. 81. $N_{i,1}$ and $N_{i,2}$, 3 areas, with and without balancing

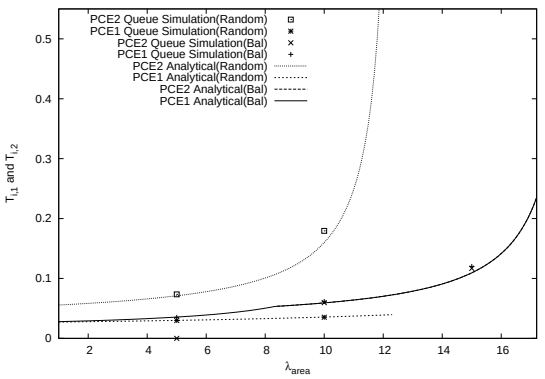

Fig. 82. $T_{i,1}$ and $T_{i,2}$, 3 areas, with and without balancing

Next, the case with a network consisting of 10 areas in addition to area 0 is considered.

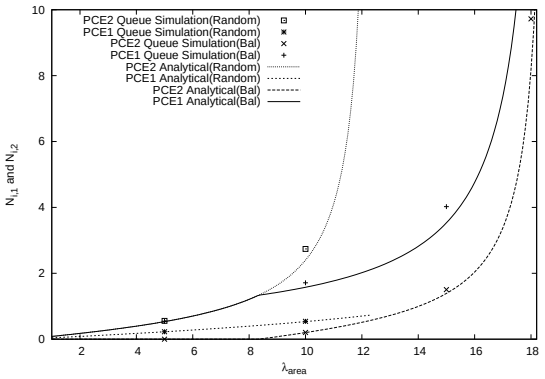

Fig. 83. $N_{i,1}$ and $N_{i,2}$, 10 areas, with and without balancing

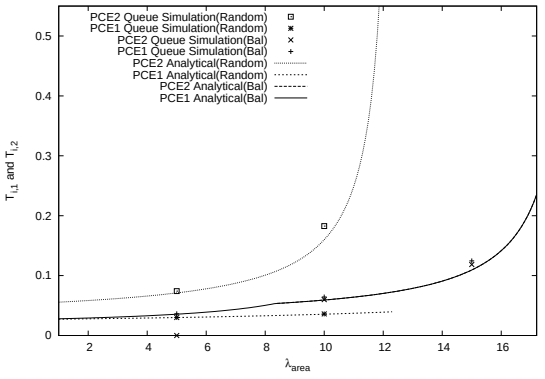

Fig. 84. $T_{i,1}$ and $T_{i,2}$, 10 areas, with and without balancing

Next, the case with a network consisting of 22 areas in addition to area 0 is considered.

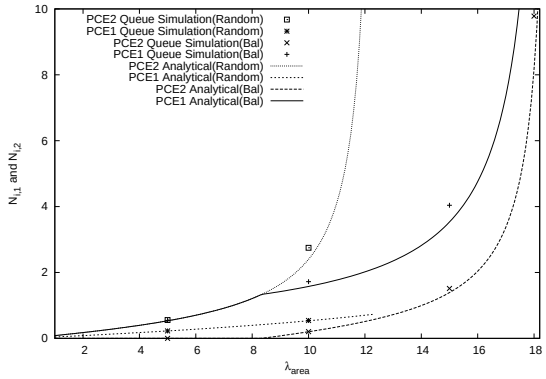

Fig. 85. $N_{i,1}$ and $N_{i,2}$, 22 areas, with and without balancing

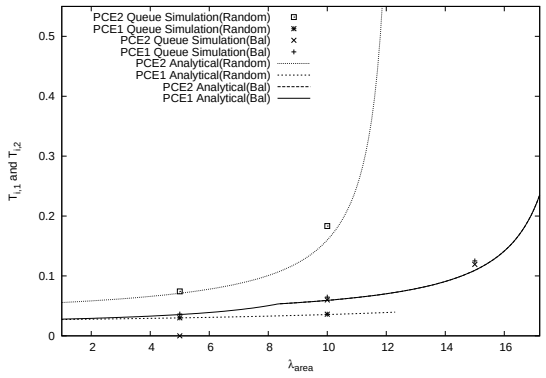

Fig. 86. $T_{i,1}$ and $T_{i,2}$, 22 areas, with and without balancing

Next, the case with a network consisting of 50 areas in addition to area 0 is considered.

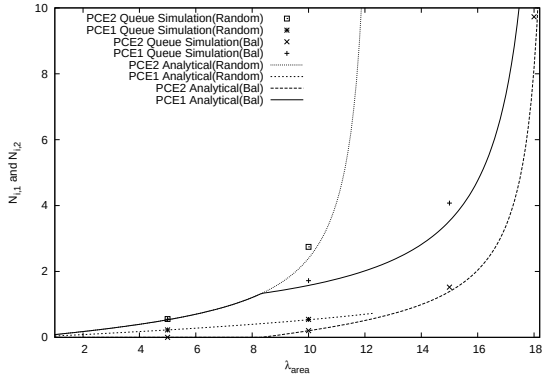

Fig. 87. $N_{i,1}$ and $N_{i,2}$, 50 areas, with and without balancing

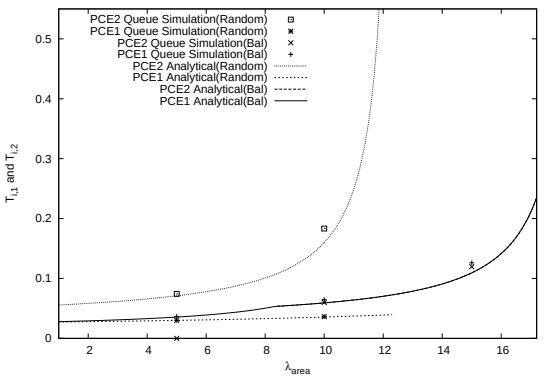

Fig. 88. $T_{i,1}$ and $T_{i,2}$, 50 areas, with and without balancing

VI. Conclusion

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

A queueing model is presented to evaluate the latency and load at each PCE in this regard. A product form solution is obtained with poisson arrival assumption. Based on this analytical framework, a load balancing strategy is proposed with the objective to achieve even average latency among two PCE's in selection.

The framework and the load balancing strategy are validated by two levels of simulation: one with network of queues and the other with full MPLS-TE and PCEP protocol suite. The simulation results show that the product form yields a good approximation for practical networks. In addition, the proposed load balancing strategy effectively reduces the overall expected path computation latency.

VII. Acknowledgments

The authors wish Thanks for the support from OPNET to provide us a professional simulation environment under educational uses and researches.

References

[1] A. Farrel, J. Vasseur, and J. Ash, "A Path Computation Element PCE-Based Architecture," IETF RFC 4655, Aug 2006.

[2] S. Dasgupta, J. de Oliveira, and J.-P. Vasseur, "Path-Computation-Element-Based Architecture for Interdomain MPLS/GMPLS Traffic Engineering: Overview and Performance," *IEEE Network*, vol. 21, no. 4, pp. 511 – 518, July-August 2007.

[3] "OPNET Modeler." [Online]. Available: www.opnet.com

[4] P. Torab and et al, "On Cooperative Inter-Domain Path Computation," in *IEEE Symposium on Computers and Communications, ISCC '06*, June 2006, pp. 511 – 518.

[5] A. Sprintson, M. Yannuzzi, A. Orda, and X. Masip-Bruin, "Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks," in *IEEE International Conference on Computer Communications, INFOCOM 2007*, July 2007, pp. 1820 – 1828.

[6] N. M. T. T. K. Matsuura, H.; Morita, "Hierarchically Distributed PCE for Backup Strategies on an Optical Network," in *International Conference on Communications and Electronics, ICCE '06*, Oct 2006, pp. 135 – 140.

[7] F. Cugini and et al, "Multiple Path Computation Element (PCE) Cooperation for Multi-layer Traffic Engineering," in *Optical Fiber Communication and the National Fiber Optic Engineers Conference, OFC/NFOEC 2007*, March 2007, pp. 1 – 3.

[8] S. Gunreben and F. Rambach, "Assessment and Performance Evaluation of PCE-based Inter-layer Traffic Engineering," in *Optical Network Design and Modeling, ONDM 2008*, March 2008, pp. 1 – 6.

[9] J. Vasseur, R. Zhang, N. Bitar, and J. L. Roux, "A Backward Recursive PCE-based Computation (BRPC) Procedure To Compute Shortest Constrained Inter-domain Traffic Engineering Label Switched Paths," IETF Draft draft-ietf-pce-brpc-09.txt, Apr 2008.

[10] D. Bertsekas and R. Gallager, *Data Networks (2nd Edition)*. Prentice-Hall,Inc.,Englewood Cliffs, 1992.