Isolation and Resource Efficiency of Virtual Optical Networks

Marc De Leenheer*†, Jens Buysse*, Chris Develder*, and Biswanath Mukherjee†

*Department of Information Technology, Ghent University - IBBT, Belgium

†Department of Computer Science, University of California, Davis, USA

Email: marc.deleenheer@intec.ugent.be

Abstract—**Virtualization of optical networking infrastructure is considered a fundamental technology in the future Internet. A key principle is that virtual networks are isolated to coexist on a shared physical substrate without interference. Although a very attractive proposition for virtual network operators and users, in this paper we demonstrate that realizing complete isolation by partitioning resources is wasteful. Therefore, we propose to group virtual network requests in clusters: within a cluster, bandwidth can be shared, whereas different cluster are properly isolated. Results indicate that intelligent isolation and design of virtual networks can lead to substantial savings of optical network resources compared to a fully isolated approach. Finally, we demonstrate the trade-off between network resource utilization and control plane scalability.**

I. INTRODUCTION

The concept of network virtualization is considered a key technology for the development of the future Internet. As discussed in [1], it can help reduce the current ossification of the Internet, by allowing large-scale experiments of novel protocols and algorithms in a contained environment, namely a virtual network. Virtualization is also a fundamental concept in cloud computing systems, where both IT resources (computational and storage servers) and network resources are offered as virtual devices [2].

Essentially, network virtualization allows sharing of a physical substrate, without interference between deployed virtual networks. This concept is usually referred to as *isolation*, and creates a desirable working environment for virtual network operators and users. For instance, large fluctuations in traffic load, misconfigured devices, malicious users, etc. in one virtual network have no effect on the other virtual networks.

Future Internet architectures are largely based on an optical transport network, employing (D)WDM or Dense Wavelength Division Multiplexing. Here, virtual topologies consisting of lightpaths (i.e. an end-to-end wavelength connection) are introduced and each virtual topology can be managed independently [3]. However, isolation usually leads to an increase in required capacity with respect to the physical optical infrastructure, since each virtual network is allocated its own network resources. Given the coarse bandwidth granularity in current commercial (D)WDM products (each wavelength offers 10, 40 or 100 Gbps), total network capacity my be very high while resource utilization unacceptably low.

In this paper, we propose to cluster virtual network requests and introduce traffic grooming in these clusters. As such, we do not introduce isolation within each cluster. However, full isolation is enforced between different clusters.

The remainder of this paper is organized as follows. First, in Section II we discuss related work, while Section III introduces the problem we aim to solve. The following Section IV introduces a formal description of the network and traffic models we consider, and our solution technique is presented in Section V. Finally, experimental results are presented in Section VI, and our concluding remarks are given in Section VII.

II. RELATED WORK

Network virtualization has seen numerous studies and implementations in the form of Virtual Private Networks (VPNs). These allow connections of pre-determined nodes over a transport network, and can be established in various network layers. For example, Layer 1 VPNs [4] can provision high-bandwidth connectivity between data centers. On the other end of the spectrum, overlay networks are usually implemented on the application layer (Layer 7), and are therefore aimed at providing specific services such as file sharing [6], multicasting or various other goals, including offering Quality of Service (QoS), protecting against Denial of Service attacks and many others. An extensive survey on VPN technologies can be found in [5].

The current interest in architectures for the future Internet has led to substantial research on this topic [1], [6]. In this context, the GEYSERS project [7] is building a novel architecture to provide network operators with an infrastructure composed of several optical network and IT resources in an on-demand fashion. To this end, the physical resources can be partitioned and aggregated to create a virtual infrastructure (VI), which in turn can be controlled by a network operator without interference of other VIs. To control this infrastructure on demand, GEYSERS architecture deploys an enhanced Network Control Plane (NCP+) that can control both network and IT resources. This way, both network and IT resources can be seen as elements of one homogenous set, able to be provisioned on-demand. The NCP+ is based on the Generalized Multi-Protocol Label Switching (GMPLS) solution, considerably extending it to cope in virtual environments. Our analysis is also based on GMPLS, and we discuss relevant details in Section IV-B.

III. PROBLEM STATEMENT

Whereas current physical infrastructure providers (PIPs) usually operate their own networks, this could change so that PIPs could offer virtual infrastructures which they do not operate themselves. Instead, a different entity, the virtual infrastructure operator or VIO, is responsible for the correct operation of this virtual network, without actually owning the physical equipment. This obviously requires isolation between different virtual network infrastructures.

In this paper, we argue that isolation can indeed be a useful tool to keep traffic separate and eliminate interference among virtual networks. However, care should be taken by the physical infrastructure provider to keep his cost model sufficiently low. Some researchers have proposed novel business models whereby physical network operators sell their network infrastructure to one or more virtual network providers. This may indeed lead to novel scenarios and applications, but we argue that respecting individual customer's virtual network request is a wasteful process in terms of resource utilization, and propose to bundle certain requests together in order to optimize physical resource utilization. Furthermore, control plane scalability is shown to be unaffected by this process. The problem we solve in this paper is thus as follows.

Given
- Physical network topology
- Set of virtual network requests, each specified as a traffic matrix.
- The number of isolated virtual networks that should be mapped on the physical topology. Each isolated virtual network is composed of one or more virtual network requests.

Find
- The composition of the isolated virtual networks, i.e. which virtual network requests belong to which isolated virtual network.
- The mapping of the isolated virtual networks on the physical topology.

A small number of isolated virtual networks maximizes the opportunities of statistical multiplexing and as such will lead to the highest resource utilization. However, this will lead to large isolated virtual networks that in turn degrades control plane scalability, since the number of control plane messages is directly influenced by the number of nodes in a network. As such, we will study the trade-off between resource utilization and control plane scalability in this paper.

IV. NETWORK AND TRAFFIC MODELS

A. Physical Network and Requests

The physical topology is assumed to be based on optical networking technology, and is described by directed graph $G = (V, E)$, V the set of nodes and E the set of edges. We are also given a list of virtual network requests r, each request described by a traffic matrix Λ^r where $[\Lambda^r]_{ij} = (\lambda^r_{ij}, \mu^r_{ij})$. Here, λ^r_{ij} represents the arrival rate for virtual network request r between nodes i and $j \in V$, while μ^r_{ij} represents the holding

rate (inverse of the holding time). Obviously, $\rho^r_{ij} = \frac{\lambda^r_{ij}}{\mu^r_{ij}}$ is the load between nodes i and j.

B. GMPLS Control Plane

We assume that each isolated virtual network will run a GMPLS control plane, and we study the scalability in terms of message exchange rate. GMPLS' control plane traffic is in large part generated by two protocols[1]: Open Shortest Path First - Traffic Engineering (OSPF-TE) and Resource Reservation Protocol - Traffic Engineering (RSVP-TE). We discuss these protocols in more detail in the following subsections.

1) OSPF-TE: a widely deployed, intra-domain link-state routing protocol [8]. Its main responsibility is the distribution of Link State Advertisements (LSAs) to create a consistent topological view of the network. It is composed of three protocols, namely:

- The *hello protocol* to allow the discovery of neighbors. Hello packets are exchanged between adjacent routers to establish their existence and proper behavior. GMPLS standardization documents recommend a repeat period of 10 seconds for Ethernet links and 30 seconds for non-broadcast links.
- The *exchange protocol* exists to set up routing adjacencies between neighboring routers.
- The *flooding protocol* ensures LSAs are distributed throughout the network so routers can build their forwarding tables.

The flooding protocol is the single most important factor that influences OSPF-TE's message exchange rate. As indicated in several studies, the contribution of OSPF-TE to total control plane scalability (in terms of message exchanges within the control network), can be reduced significantly by use of appropriate *flooding reduction* techniques [9]–[11]. Moreover, OSPF domains may even be divided into areas to further reduce control plane traffic. As such, we will not consider OSPF-TE message traffic in our results on the control plane scalability.

2) RSVP-TE: the signaling protocol to set up label-switched paths (LSPs) between nodes [12], including various Quality of Service parameters. RSVP-TE traffic consists of the following messages.

- *Path* messages are sent from source to destination and signal the receiver about the upcoming traffic flow.
- *Resv* messages are sent from destination to the source and perform the actual reservation of network resources.
- *PathTear* and *ResvTear* are used to explicitly signal the impending end of the traffic flow and the actual release of network resources, respectively.

It is very important to note that, once reservations are complete, their allocation must be acknowledged over time according to a *soft state* protocol. This implies that both *Path* and *Resv* must be repeatedly refreshed if the LSP should not

---

[1]We do not consider Link Management Protocol (LMP) in the remainder of this paper, as we are mainly interested in control plane scalability in terms of traffic load. Instead, the main focus of LMP is on TE link management.

automatically time out. In our study, we follow the model originally given in [9] to fully capture RSVP-TE traffic (i.e. depends on generated traffic load, connection holding times, and the refresh rate as dictated by the soft state protocol).

V. Solution Technique

Our solution consists of a *two-step algorithm*: first we perform clustering to group individual virtual network requests in groups of virtual networks, after which we calculate the exact topology of each virtual networks based on the aggregate network demand.

A. Virtual Network Clustering

The clustering algorithm should group virtual network request that are most similar. Similarity can be based on criteria such as node activity (i.e. nodes where data traffic is generated/received), link overlap (i.e. virtual network requests that have the highest number of overlapping links in the physical topology), etc. Here we consider only *node activity*, i.e. our objective is to cluster virtual network requests that have the highest number of active nodes in common. In the following, we present our solution based on an Integer Linear Programming (ILP) formulation. To compare the influence of the clustering algorithm on the final network dimensions and operation, we also evaluate a random clustering approach. The clustering algorithms described below will thus cluster the virtual network requests in k clusters, after which each virtual network k is designed based on the aggregate traffic demand matrix $\Lambda^k = \sum_{r \in k} \Lambda^r$.

1) ILP-Based Clustering: The following ILP model can provide optimal results for the clustering of virtual network requests, based on node activity. Assume the virtual network requests are represented as traffic matrices Λ^r, and assume we have converted them to active node matrices A^r as follows: $A_{ij}^r = 1$ if $\lambda_{ij}^r \geq 0, 0$ otherwise. Introduce binary variables $x_k^r = 1$ if virtual network request r is partitioned in cluster k, 0 otherwise. Furthermore, let the binary variables $y_k^i = 1$ if node i is active in cluster k, 0 otherwise. Furthermore, integer variables z_k represent the total number of nodes active in cluster k, while integer variable z is the maximum over all z_k. The objective is to minimize z.

The following constraints must be applied:

- Each cluster contains at least one virtual network request $\forall k : \sum_r x_k^r \geq 1$.
- Each demand belongs to one and only one cluster $\forall r : \sum_k x_k^r = 1$.
- Active nodes are either a source or destination of traffic $\forall i, j, k, r : y_k^i \geq A_{ij}^r x_k^r$ and $\forall i, j, k, r : y_k^i \geq A_{ji}^r x_k^r$.
- Total number of actives nodes per cluster $\forall k : z_k = \sum_i y_k^i$.
- Highest number of active nodes in a cluster $\forall k : z \geq z_k$.

We could also consider the binary linear program, whereby we eliminate variables z_k and z, and introduce objective function $min \sum_k \sum_i y_k^i$. This objective tries to minimize the total number of active nodes over all clusters, but leads to suboptimal results in our scenario. Indeed, results will be composed of 1 cluster containing $r - (k - 1)$ application requests, while the remaining $k - 1$ clusters contain a single application demand. This corresponds to the lowest number of total active nodes over all clusters, as it is the solution where nodes are activated in as little as clusters as possible. Instead, our proposed objective function integrates load-balancing to effectively strike a balance between control plane scalability and network resource utilization.

Also, additional constraints $\forall k : x_k^{r_1} + x_k^{r_2} \leq 1$ may be introduced to enforce isolation between any two application demands r_1 and r_2, which could be necessary based on performance arguments or virtual network operator policy. For instance, two virtual network operators of a competing service may choose not to be deployed in the same cluster. However, we do not consider such cases in this paper.

2) Random Clustering: To demonstrate the need for an intelligent clustering algorithm, we also study results when clustering is performed in a random fashion. However, the clustering is not completely random, as we must ensure that all clusters contain at least one virtual network request. As such, the clustering algorithm was implemented with a round robin strategy, whereby each cluster is, in turn, assigned a randomly chosen virtual network request. Note that this also ensures that each cluster contains more or less an identical number of elements.

B. Virtual Network Design

To study the trade-off between resource utilization and control plane scalability in detail, we propose two alternative design techniques, corresponding to the extremes of the potential objectives. The first minimizes the hop distance in the virtual network, effectively creating a *full mesh* topology between all active nodes. Implementation of this algorithm is straightforward and detailed below.

First, let $\rho = \frac{\lambda}{\mu}$, then $Erl(\rho, W) = \frac{\frac{\rho^W}{W!}}{\sum_{i=0...W} \frac{\rho^i}{i!}}$ represents the Erlang-B function to calculate the blocking probability of a network link under load ρ and having W wavelengths. The inverse function $Erl^{-1}(\rho, B)$ then is used to calculate the number of wavelengths necessary to obtain a target blocking probability B for a link under load ρ [13].

1) Initialize physical link loads to $\rho_l = 0$
2) For each active node pair (i, j) in Λ^k
   a) Perform shortest path routing between i and j on the physical topology
   b) For link l in path: update number of wavelengths $W_l = W_l + Erl^{-1}(\rho_{ij}^k, B)$

We expect this solution to have the lowest number of control plane messages, as every connection only requires signaling over a single link between the source and destination node (Fig. 1(a)). However, as the single hop constraint does not allow sharing of resources in the physical network, the resource utilization will consequently suffer.

On the other hand, the second design technique focuses on *maximizing utilization* (MaxUtil) by exploiting the effects of statistical multiplexing as much as possible (refer to the control

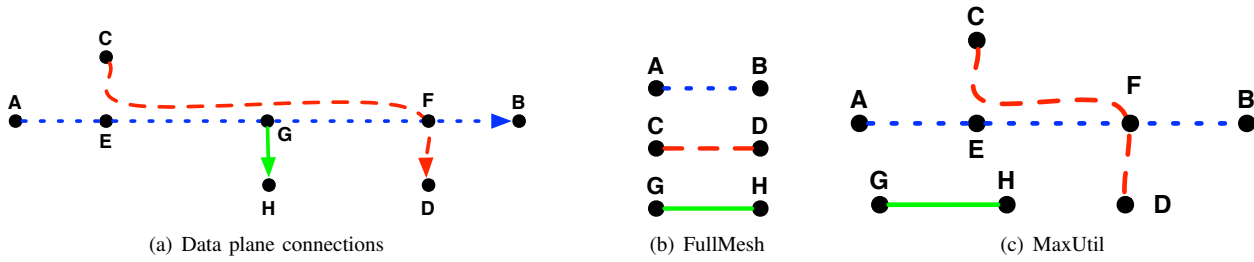(a) Data plane connections   (b) FullMesh   (c) MaxUtil

Fig. 1. Virtual network design approaches FullMesh (b) and MaxUtil (c) for the virtual network traffic matrix shown in (a). Hop distance in the FullMesh topology is always 1, while it depends on the number of shared links in the MaxUtil case (e.g. hop distance 3 for connections AB and CD).
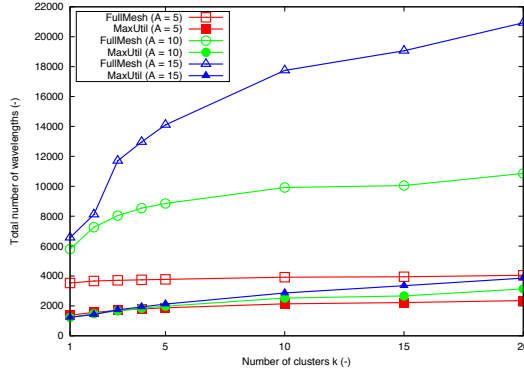

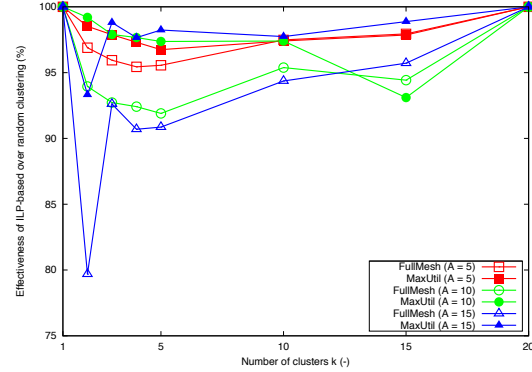Fig. 2. Relatively stable wavelength usage for MaxUtil virtual network design, and fast growth for the FullMesh approach.

Fig. 3. Ratio of total wavelength capacity (= effectiveness) of ILP-based over random clustering.

plane topology shown in Fig. 1(c)). This becomes apparent in step 3(b)ii, where previously used links are re-used to reduce physical wavelength usage and improve wavelength utilization in the virtual network. The algorithm is outlined below and executed for each virtual network demand matrix Λ^k.

1) Initialize physical link costs $c_l = c_l^{max} = Erl^{-1}(\rho_{max}, B)$ where $\rho_{max} = \sum_{ij} \rho_{ij}^k$ equals the total load of the virtual network demand matrix[2]
2) Initialize physical link loads to $\rho_l = 0$
3) For each active node pair (i, j) in Λ^k
   a) Perform shortest path routing between i and j on the physical topology with link costs c_l
   b) For each link l in path
      i) Update current load on link $\rho_l = \rho_l + \rho_{ij}^k$
      ii) Update current cost of link $c_l = c_l^{max} - Erl^{-1}(\rho_l, B)$
4) Calculate the number of wavelengths on each link $W_l = Erl^{-1}(\rho_l, B)$

VI. NUMERICAL RESULTS

The physical topology is the COST239 basic European network, consisting of 28 nodes and 41 bidirectional links [14]. Virtual network requests are generated by random selection out of all physical nodes of A active nodes that will generate traffic; the remaining nodes in the traffic matrix have $\lambda_{ij} = 0$.

---

[2]i.e. the worst case scenario where all traffic in the cluster would be routed through a single physical network link
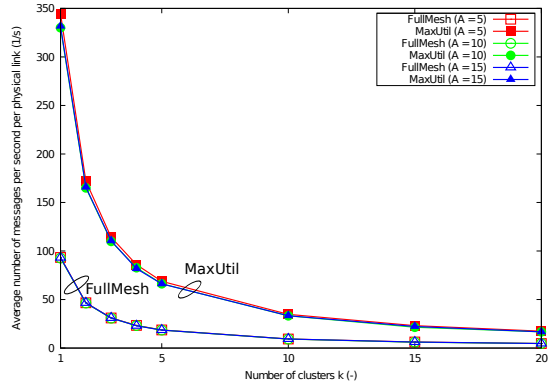

Fig. 4. Convergence of average message exchange rate for large number of virtual network clusters.

The non-zero arrival and holding rates are generated according to an exponential distribution with means $\lambda_{ij} = \frac{1}{A(A-1)}$ and $\mu_{ij} = 1$. To ensure fair comparison, note that the aggregate traffic load is constant over all scenarios by the weighing of the arrival rates with the number of traffic pairs. Finally, 20 virtual network requests are generated, and the target blocking probability is configured to 1%. Each data point in the following graphs is the result of averaging over 20 experiments with randomly generated virtual network requests to ensure statistically relevant results. The refresh rate of the RSVP-TE protocol is configured to $\frac{1}{30}s^{-1}$, the default value proposed in the relevant standards documentation [9]. Note that in the following graphs, a value of $k = 1$ corresponds to

the case where no virtualization is considered, while $k = 20$ implies full isolation, i.e. each virtual network requests is allocated its own network resources.

Figure 2 shows the total number of wavelengths necessary to instantiate a varying number of virtual network clusters k, using the ILP-based clustering algorithm. We observe the relatively slow growth in wavelength usage for the MaxUtil approach, which is in stark contrast to the behavior of the FullMesh virtual network design. Furthermore, the number of active nodes A in each virtual network request has relatively little effect on the MaxUtil approaches.

The effectiveness of the ILP-based clustering algorithm is studied in Fig. 3. Here, we show the ratio of the total number of wavelengths for the ILP-based over random clustering. Obviously, identical results are obtained when $k = 1$ and $k = 20$, i.e. when there is no or full virtualization. Clearly, most cases show that the ILP algorithm requires 5 to 10% less wavelength capacity compared to random clustering. Also, the effectiveness of the random clustering reaches a minimum between $k \in [3, 6]$ clusters, indicating the region where intelligent clustering is most in order. However, the relatively low improvement of ILP-based over random clustering indicates that more advanced clustering should be developed. Indeed, our approach only incorporated node activity of virtual network requests, whereas the potential for, for instance, network grooming was not considered. Also, since we study here completely random virtual network requests, their correlation is limited and as such opportunities for intelligent clustering are most likely rare.

In Fig. 4, the control plane message exchange rate is averaged over all virtual networks. As discussed in Section IV-B we only consider connection signaling traffic (RSVP-TE), as this forms the majority of control plane traffic. This is especially true when introducing flooding reduction techniques for OSPF. Note that the hop count always equals 1 in the FullMesh case, while the distance between any two network nodes in the MaxUtil case depends on the number of intermediate GMPLS controllers (Fig. V-B). Figure 4 shows that both design techniques converge to approximately the same average message exchange rate, although the MaxUtil approaches have a very high control plane load for a small number of virtual networks. The size of the virtual network requests A does not influence the message exchange rates at all.

Note that the average message exchange rate is a hyperbolic function ($\frac{1}{x}$), and thus the total message exchange rate (sum over all virtual networks) remains constant. However, the reduction in control traffic within each cluster indicates that virtualization offers virtual network operators the compelling advantage of control plane scalability.

VII. Conclusions

We presented a clustering and design algorithm to intelligently isolate virtual optical networks. Results indicate that our algorithms can reduce the need for network resources by about 10%. We also studied control plane scalability of virtual networks, and demonstrated the trade-off between resource utilization and control plane traffic. Future work will explore further optimizations in virtual optical network design, and concentrate on introducing time-dependent traffic models.

References

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *IEEE Computer*, vol. 38, no. 4, pp. 34–41, April 2005.

[2] B. Armbrust, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–59, April 2010.

[3] K. Shiomoto, I. Inoue, E. Oki, S. Member, and A. Concepts, "Network virtualization in high-speed huge-bandwidth optical circuit switching network," *IEEE INFOCOM 2008 - IEEE Conference on Computer Communications Workshops*, pp. 1–6, April 2008.

[4] T. Takeda, D. Brungard, D. Papadimitriou, and H. Ould-Brahim, "Layer 1 virtual private networks: driving forces and realization by GMPLS," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 60–67, July 2005.

[5] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, April 2010.

[6] J. Schonwalder, M. Fouquet, G. Rodosek, and I. Hochstatter, "Future Internet = content + services + management," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 27–33, July 2009.

[7] A. Tzanakaki, M. Anastasopoulos, K. Georgakilas, J. Buysse, M. D. Leenheer, C. Develder, S. Peng, R. Nejabati, E. Escalona, D. Simeonidou, N. Ciulli, G. Landi, M. Brogle, E. López, J. F. Riera, J. A. Garcia-espin, P. Donadio, G. Parladori, and J. Jiménez, "Energy Efficiency in integrated IT and Optical Network Infrastructures : The GEYSERS approach," in *INFOCOM 2011 Workshop on Green Communications and Networking*, April 2011, pp. 343–348.

[8] J. Moy, "OSPF Version 2," RFC 2328, April 1998.

[9] S. Salsano, A. Botta, P. Iovanna, M. Intermite, and A. Polidoro, "Traffic engineering with OSPF-TE and RSVP-TE: Flooding reduction techniques and evaluation of processing cost," *Computer Communications*, vol. 29, no. 11, pp. 2034–2045, July 2006.

[10] H. M. Alnuweiri, L.-Y. K. Wong, and A.-K. Tariq, "Performance of New Link State Advertisement Mechanisms in Routing Protocols with Traffic Engineering Extensions," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 151–162, May 2004.

[11] O. Komolafe and J. Sventek, "Impact of GMPLS Control Message Loss," *Journal of Lightwave Technology*, vol. 26, no. 14, pp. 2029–2036, July 2008.

[12] I. Cselényi, G. Fehér, and K. Németh, "Benchmarking of Signaling Based Resource Reservation in the Internet," *Lecture Notes in Computer Science*, vol. 1815, pp. 643–654, May 2000.

[13] S. A. Berezner, A. E. Krzesinski, and P. G. Taylor, "On the Inverse of Erlang's Function," *Journal of Applied Probability*, vol. 35, no. 1, pp. 246–252, March 1998.

[14] S. D. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, and J. Derkacz, "Pan-European Optical Transport Networks: An Availability-based Comparison," *Photonic Network Communications*, vol. 5, no. 3, pp. 203–225, May 2003.