

# Topology Design for Service Overlay Networks with Economic and QoS Constraints

(Work in Progress)

Davide Adami, Christian Callegari, Stefano Giordano, Michele Pagano,  
and Teresa Pepe

Dept. of Information Engineering, University of Pisa, Italy  
{d.adami,christian.callegari,s.giordano,m.pagano,teresa.pepe}@iet.unipi.it

**Abstract.** Service Overlay Networks (SON) have emerged as an efficient way to provide end-to-end *Quality of Service* (QoS) support in the Internet, as required by real-time services.

The deployment of a SON is a capital-intensive investment. Thus, minimizing the economic cost is one of the key objectives for the SON operator. Moreover, end-to-end performance must be enhanced, so as to provide innovative value-added services. This paper, addresses the problem of cost and performance optimization of a SON. More specifically, we propose a set of algorithms for the selection of a SON topology, which minimizes the economic cost while taking also into account performance constraints.

**Keywords:** Service Overlay Networks, Topology Optimization, QoS support.

## 1 Introduction

Originally conceived as a network infrastructure for military applications and academic research, the Internet has rapidly turned into a business platform, where private enterprises seek to offer new value-added network services. At the same time, the wide spread of the Internet has brought heterogeneous applications with different QoS requirements, to converge over a single IP-based network infrastructure. However, today's Internet mainly provides a best effort packet delivery service and lacks in the end-to-end QoS support.

In this context, Service Overlay Networks (SONs) have recently emerged as one of the most promising solutions to provide end-to-end QoS [1] [2] [3] [4] [5]. Unlike other network architectures, SONs do not require any change in the underlying network layer. However, from an economic point of view, the deployment of SONs requires a valuable investment.

As clearly shown in several research works [6] [7], the choice of the optimal SON's topology plays a critical role to reduce the overall network costs. Nevertheless, when SONs are used to provide QoS, the topology design problem can not be only faced from an economic point of view, since performance metrics, such as bandwidth and delay, must be also taken into account.

For this reason, in this paper we propose a new approach for the topology design of a SON which aims at minimizing the SON deployment costs, while at the same time meeting multiple QoS performance constraints.

The rest of the paper is organized as follows. Section 2 discusses some related works, whereas Section 3 introduces the problem of the SON topology design. Next, section 4 describes a new approach for the design of a SON by means of algorithms which takes into account economic costs as well as performance metrics. The performance of such algorithms are compared in Section 5 through a large set of simulations. Section 6 concludes the paper with some final remarks.

## 2 Related Works

The use of a SON as a means to provide QoS in the Internet was first introduced in [1]. The authors studied the bandwidth provisioning problem without neglecting economic aspects. Indeed, the key challenge of this research was to provide the optimal amount of bandwidth to support end-to-end QoS-sensitive services while minimizing the bandwidth costs.

In [8] the authors addressed this issue in terms of choosing the number and the location of the overlay nodes to be deployed as well as the reserved bandwidth on each overlay link, while in [2] the authors proposed a framework for the adaptive design of a SON. In more detail, they developed a set of heuristic methods, and compared their performance. Another set of heuristics was proposed in [9] to address the same problem in large-size networks.

## 3 SON Topological Design

Let us we consider an application scenario where a SON, built on top of the traditional IP network, provides end-to-end QoS support to a number of users, who pay the SON operator by means of a service contract. The overlay network consists of overlay nodes (ONs) and a set of end-systems (ESs). The ONs are specialized nodes strategically placed across the ISPs networks by the Overlay Service Provider (OSP). Usually, one or more ONs can be deployed in each domain. Moreover, the number of ONs in an ISP may be increased, so as to improve scalability and fault tolerance. In our scenario, we suppose that:

- the number of ONs to be deployed is set in advance (how many ONs are needed and where to deploy them in certain scenarios also represent interesting research topics, but they are out of the scope of this paper);
- the connections between the ONs are provided by overlay links, usually composed of one or more physical links;
- ESs can access the network using an ON called *access ON*.

In this work, we assume that the connection between the ESs and the access ONs is possible only if they both belong to the same ISP. The OSP buys network services, such as guaranteed bandwidth, from different ISPs and, according to pre-established Service Level Agreements (SLAs), provides bandwidth guarantees connections to the ESs.

## 4 New Approach for the Design of SONs

For the overlay topology selection, we have to solve two distinct problems. First, we have to select the links that connect the ONs (transport links). Second, for each ES, we must select an access ON that allows the ES to access the ON. The total cost of the overlay network is given by the sum of the access costs and the transport costs of the resulting topology. In more detail, given a network composed of  $M$  ESs and  $N$  ONs, our proposed method consists of the following four steps.

**Step 1.** We only consider the ONs. Since the connection between the ONs is provided by the underlying network layer, we can assume that it is always possible to have an overlay link  $(i, j)$  between two distinct ONs. Thus, we start by considering a full-mesh topology for the overlay network, where each link is described by a metric, corresponding to the potential bandwidth that the ISP can make available on it.

**Step 2.** We assume that the ISPs can not always provide all the requested bandwidth. If the bandwidth associated to the link  $(\omega_{ij})$  is less than the bandwidth necessary to provide the required QoS  $(\Omega)$ , we prune the link from the graph corresponding to the network topology. After that, we assign two distinct metrics to each remaining link: economic cost and link delay  $(l_{i,j}$  and  $r_{i,j}$ , respectively).

**Step 3.** We determine a path  $p$  (with cost  $c_{ij}$ ) between each pair  $(i, j)$  of ONs so that the total transport cost and the link delay are minimized. This is a multi-objective optimization problem. It is worth highlighting that solving this problem does not correspond to find an optimal solution for each objective function, separately. Since finding an optimal solution to this multi-objective optimization problem is too complex, in the following we use two distinct approximated methods: **Weight Method** and  **$\varepsilon$ -constraint technique**.

1. **Weight Method.** It is the simplest approach, since it sums the quantities to be minimized (metrics) into a single objective function, pre-multiplying them by an appropriate weight. The weight of a metric is chosen in proportion to the importance of the metric in the problem.
2.  **$\varepsilon$ -constraint Method.** This method corresponds to the classical method of constraint optimization.

In our case we aim to minimize the economic cost while taking into account the constrained delay. Formally, we can write:

$$\text{minimize } \sum_{i=1}^N \sum_{j=1}^N l_{ij} \omega_{ij} \quad \text{subject to } \sum_{(ij) \in p} r_{ij} < \rho \quad (1)$$

**Step 4.** Finally, we have to determine how to connect the ESs to the ONs. In this phase, we assume that the access delays are negligible with respect to the

transport delays. The main idea is to select the access ONs, so as to minimize the total cost. The problem can be formalized as follows:

$$\text{minimize } \sum_{i=1}^M \sum_{k=1}^N w_i \alpha_{ik} y_{ik} + \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^M \sum_{l=1}^N y_{ij} y_{kl} \omega_{ik} c_{jl} + \sum_{j=1}^M \sum_{l=1}^N w_j \alpha_{jl} y_{jl} \quad (2)$$

subject to

$$\sum_{j=1}^N y_{ij} = 1 \quad \text{for } i = 1, \dots, M \quad (3)$$

where  $\alpha_{ij}$  is the access cost for bandwidth unit,  $w_i$  is the requested access bandwidth and  $y_{ij}$  is an indicator function.

$$y_{ij} = \begin{cases} 1 & \text{if } ES_i \text{ is assigned to } ON_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The optimization problem (2) is a variant of the well-known quadratic assignment problem (QAP). Since the QAP is a NP-hard problem, we can not find an exact solution in a polynomial time. Heuristic methods, therefore, are preferable options to solve this problem. There is a large set of heuristic algorithms for solving the QAP. In this work, we take into account two heuristics: the first one consists in the random selection of the access ONs, while the second one performs the selection on the basis of the access costs only, ignoring the transport costs.

In the following, we present six different algorithms we have developed for solving the presented problem. To be noted that step 1 and 2 (pruning phase for satisfying the bandwidth constraints) are common to all these algorithms and are performed as previously described, while the approach used for the step 3 and 4 (transport links selection and access links selection, respectively) differs from algorithm to algorithm. Thus, we detail step 3 and 4 and we present the pseudo-code for each of the algorithms we have proposed. For all the algorithms, the input parameters are:

- $N$  number of ONs
- $\{\omega_{ij}\}$  bandwidth matrix
- $\Omega$  bandwidth constraint
- $\rho$  delay constraint
- $\{l_{ij}\}$  transport cost (per unit of bandwidth)
- $\{r_{ij}\}$  link delay
- $\{R_p\}$  delay of path  $p$ , defined as  $\sum_{(i,j) \in p} r_{ij}$

Moreover, the algorithm output, which is common to all the methods, is given by the overlay network topology with his respective overall cost (given by the sum of transport costs and access costs).

#### 4.1 Modified Dijkstra Random Mode - MDRM

In step 3, for the selection of the transport links this algorithm uses the *Modified Dijkstra algorithm*, which allows to find the  $k$ -shortest (least cost) paths between each pair of ONs. Among these, we select the least cost path, which satisfies the delay constraint.

In step 4, the access nodes are selected in a random mode.

Formally the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then
    prune link  $(i, j) \Rightarrow l_{i,j} = \infty$ 
  end if
  for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Modified Dijkstra algorithm to find  $k$  least-cost paths;
    for  $p = 1 : k$  do
      evaluate path delay for the path  $p$ 
      if  $R_p < \rho$  then
        select the path
        break;
      end if
    end for
  end for
end for
select access nodes in a random mode;
evaluate access cost.

```

#### 4.2 Modified Dijkstra Minimum Access Cost - MDAC

This algorithm is similar to the previous one. Indeed, only step 4 is different: in this case the access nodes are selected, so that the access costs are minimized. Formally, the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then
    prune link  $(i, j) \Rightarrow l_{i,j} = \infty$ 
  end if
  for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Modified Dijkstra algorithm to find  $k$  least-cost paths;
    for  $p = 1 : k$  do
      evaluate path delay for the path  $p$ 
      if  $R_p < \rho$  then
        select the path
        break;
      end if
    end for
  end for
end for

```

**end for**

select the access node which gives minimum access cost;  
evaluate access cost.

### 4.3 Martins Santos Minimum Access Cost - MSMAC

As in the previous algorithm, the selection of the transport links is based on a k-shortest path algorithm, but in this case step 3 is carried out by means of *Martins Santos algorithm* [10], instead of the modified Dijkstra algorithm. Step 4 is the same as in the first case.

Formally the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then
    prune link  $(i,j) \Rightarrow l_{i,j} = \infty$ 
  end if
  for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Martins Santos algorithm to find  $k$  least-cost paths;
    for  $p = 1 : k$  do
      evaluate path delay for the path  $p$ 
      if  $R_p < \rho$  then
        select the path
        break;
      end if
    end for
  end for
end for
select the access node which gives minimum access cost;
evaluate access cost.

```

### 4.4 Elimination

In this algorithm, step 3 is based on the use of the “standard” Dijkstra algorithm for the selection of the least cost path. But, in this case we evaluate the delay related to this path. If the path delay exceeds the imposed bound, the path does not guarantee the required level of QoS, so the path link with the maximum link delay is pruned, from the graph associated to the network topology. Then, if necessary, we iterate this procedure. Step 4 is based on the random selection of the access nodes. Formally, the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then
    prune link  $(i, j) \Rightarrow l_{i,j} = \infty$ 
  end if
  for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Dijkstra algorithm to find least-cost path  $p$ ;
    evaluate path delay for the path  $p$ 
  end for
end for

```

```

if  $R_p > \rho$  then
  prune the link with the maximum link delay and restart from the ap-
  plication of the Dijkstra algorithm
else
  select the path
end if
end for
end for
select access nodes in a random mode;
evaluate access cost.

```

#### 4.5 Additive Metric - AM

In this algorithm, step 3 is based on the Weight Method. The idea is to construct an unique metric which takes into account both cost and delay. This metric is obtained with a linear combination of the two metrics. As in the previous algorithm, step 4 is based on a random selection of the ONs. Formally the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then
    prune link  $(i, j) \Rightarrow l_{i,j} = \infty$ 
  end if
  compute the metric  $\varphi = \lambda_1 \cdot \text{economic cost} + \lambda_2 \cdot \text{delay}$ 
  for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Dijkstra algorithm to find least-cost ( $\varphi$ ) path  $p$ ,
    evaluate path delay for the path  $p$ 
    if  $R_p > \rho$  then
      impossible connection,
    else
      evaluate path cost;
    end if
  end for
end for
select access nodes in a random mode;
evaluate access cost.

```

#### 4.6 Multiplicative Metric - MM

As in the previous case, step 3 is based on the computation of an unique metric, but in this case we consider the product between cost and delay. Once again step 4 is based on a random selection of the ONs. Formally, the algorithm can be described as follows:

```

for all  $i$  and  $j \in \{1, \dots, N\}$  do
  if  $\omega_{ij} < \Omega$  then

```

```

    prune link  $(i, j) \Rightarrow l_{i,j} = \infty$ 
end if
compute the metric  $\varphi = \text{economic cost} \cdot \text{delay}$ 
for all pair of ONs,  $i$  and  $j \in \{1, \dots, N\}$  do
    use Dijkstra algorithm to find least-cost ( $\varphi$ ) path  $p$ ,
    evaluate path delay for the path  $p$ 
    if  $R_p > \rho$  then
        impossible connection,
    else
        evaluate path cost;
    end if
end for
end for
select access nodes in a random mode;
evaluate access cost.

```

## 5 Simulation Results

In this section we present the results of the simulation carried out to compare the performance of the proposed algorithms, focusing at first on the description of the network scenario and then on the discussion of the achieved results.

In our simulation, we considered a network composed of  $M$  ESs, where  $M$  can assume the uniformly distributed values  $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$  and  $N = 200$  ONs. Moreover, we assumed that the underlay network is formed by 38 ASs and each ES is located in a different AS. We also supposed that an ES can be connected to a given access ON, only if they belong to the same AS and that each ES is only connected to a single access ON.

From an economic point of view, we assumed that the transport costs between the ONs, as well as the access costs, are drawn from an uniform distribution in the range  $[5,50]$  monetary unit.

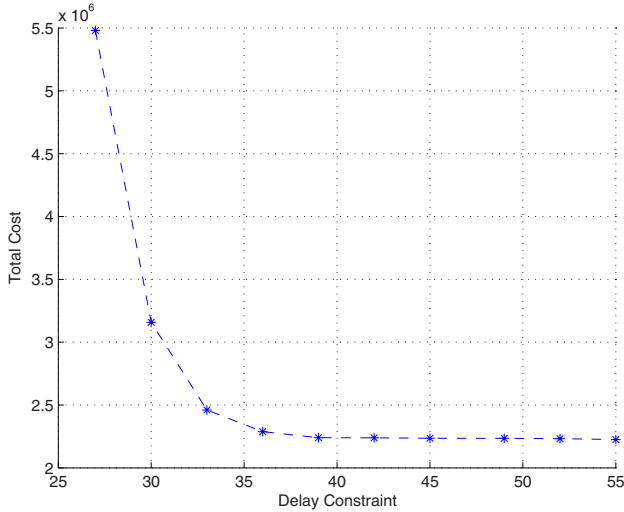
Instead, the reserved bandwidth is uniformly distributed in the range  $[10,20]$  Mbps, while the link delays are uniformly distributed in the range  $[3,15]$  ms.

The simulations have been conducted varying the service requests, namely varying the number of ESs, as well as the delay and bandwidth constraints. In more detail, we have considered the following settings:

- scenario 1: the number of ESs is fixed to 100, the delay constraint can assume the uniformly distributed values  $\{27, 30, 33, 36, 39, 42, 45, 48, 51, 54\}$  ms, and the bandwidth constraint is equal to 13 Mbps;
- scenario 2: the number of ESs is variable, the delay constraint is fixed to 36ms, and the bandwidth constraint is equal to 13Mbps.

It is worth noticing that in all the simulations all the ESs are guaranteed to be connected to an access ON. Moreover, we have decided not to vary the number of ONs, since we consider the situation where the overlay backbone is given by the OSP.





**Fig. 1.** Topology cost and delay threshold

The performance parameter taken into account to compare the proposed algorithm is given by the overall network cost, given by the sum of the access costs and transport costs. The obtained results represent the average value, computed over 100 independent simulations.

At first, we have evaluated (scenario 1) the dependence of the network costs on the given delay constraints. For sake of brevity, we only show the results related to one algorithm (i.e., MDRM), since all the algorithms present a similar trend.

In more detail, figure 1 plots the network costs versus the delay constraint. As expected, the overall costs significantly depend on the last constraint. Indeed, we have an increase of the network costs when the delay constraint becomes tighter. This result is not unexpected, since a tighter bound could imply the rejection of low cost paths.

Regarding the second scenario, we have evaluated the network costs achieved by the different algorithms, when varying the number of ESs. In figure 2, we propose a comparison among the different solutions. Since all the algorithms behave almost equally, in figure 3 we show a zoomed version of the previous figure, from which we can better analyze the algorithms behavior. It emerges that MM and AM algorithms have the highest costs demonstrating they are less suitable to simultaneously take into account both economic cost and delay in the minimization problem.

It is important to highlight that due to the possibility to assign different values to the weights  $(\lambda_1, \lambda_2)$ , in the AM algorithm, it is possible to achieve lower costs by giving more importance to the economic costs, as shown in figure 4. This solution corresponds, in fact, to only take into consideration the economic

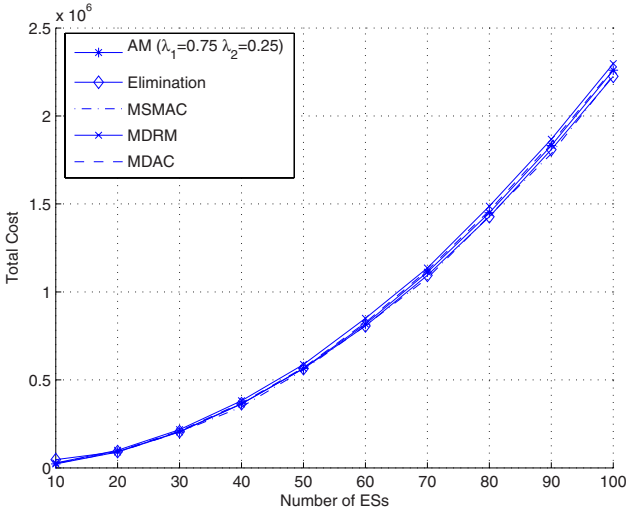


Fig. 2. Algorithms Comparison

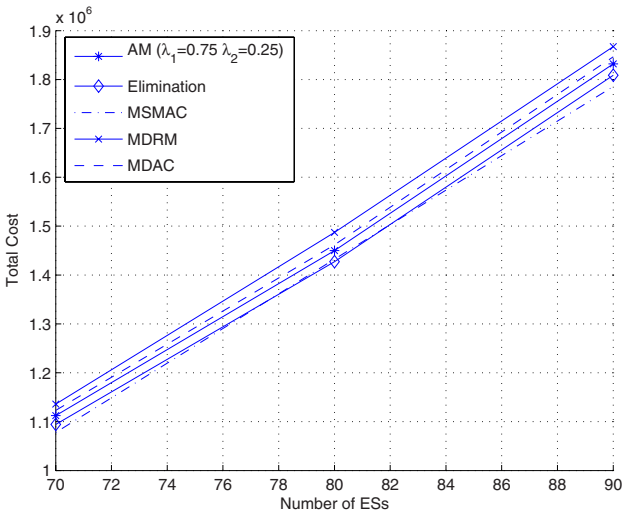


Fig. 3. Algorithms Comparison (zoomed version)

metric (case  $\lambda_1 = 1, \lambda_2 = 0$ ) and so the metric could not be able to guarantee the requested QoS.

On the contrary, K-shortest path based algorithms and the Elimination algorithm allow the creation of low cost topology, while simultaneously taking into account the delay constraint.

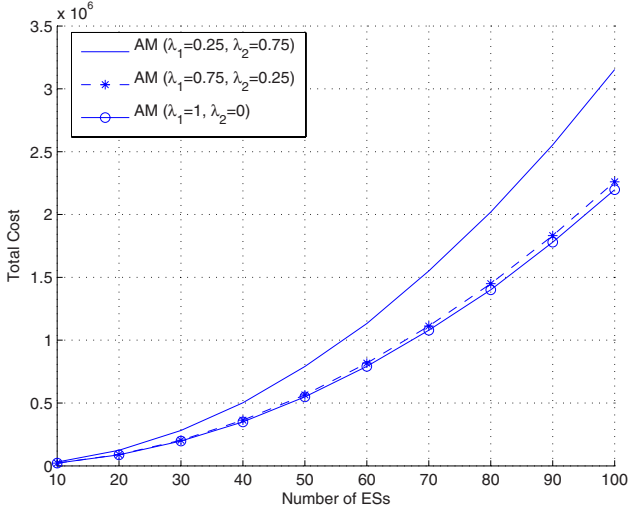


Fig. 4. AM algorithm performance when varying the weights

Table 1. Normalized Execution Time

	Normalized Execution Time
MDRM	4
MDAC	4
MSMAC	23
Elimination	8
AM	1
MM	2

Among these algorithms the best solution is given by the use of the MSMAC. Indeed, it achieves the best performance, in terms of network costs, and at the same time satisfies QoS constraints.

Nevertheless, it is important to also consider that this algorithm presents the highest execution time, as shown in table 1. This is due to the fact that it requires an iterated use of the Dijkstra algorithm up to finding the path that satisfies the delay constraint (Martin Santos algorithm).

Thus, also taking into account the computational costs, it emerges that the best algorithms are those based on the use of the Dijkstra algorithm, since they achieve good performance with low execution time.

## 6 Conclusion

In this paper we have proposed several algorithms to solve the topology design problem for SONs. The presented methods take into account both the economic aspects as well as several QoS constraints.

In more detail our aim has been to choose the access ONs and transport links which optimize costs and performance. To be noted that this optimization problem does not have an optimal solution, since it is a NP-hard problem. Thus, our solutions are based on several heuristics.

The performance analysis has shown that the best performance are achieved by the MSMAC algorithm, even though it has a very high computational cost.

## References

1. Duan, Z., Zhang, Z.L., Hou, Y.T.: Service overlay networks: Slas, qos, and bandwidth provisioning. *IEEE/ACM Trans. Netw.* 11(6), 870–883 (2003)
2. Tran, H.T., Ziegler, T.: A design framework towards the profitable operation of service overlay networks. *Comput. Netw.* 51(1), 94–113 (2007)
3. Lakshminarayanan, K., Stoica, I., Balakrishnan, H., Katz, R.: OverQoS: Offering Internet QoS Using Overlays. In: 1st HotNets Workshop, Princeton, NJ (October 2002)
4. Li, Z., Mohapatra, P.: QRON: QoS-aware Routing in Overlay Networks. *IEEE Journal on Selected Areas in Communications* 22(1), 29–40 (2004)
5. Cui, J.H., Gokhale, S.: Towards Integrated Provisioning of QoS Overlay Network. University of Connecticut, Storrs, CT 06296
6. Li, Z., Mohapatra, P.: The impact of topology on overlay routing service. In: Proceedings of IEEE INFOCOM (2004)
7. Li, Z., Mohapatra, P.: On investigating overlay service topologies. *Computer Network* 51, 54–68 (2007)
8. Capone, A., Elias, J., Martignon, F.: Optimal design of service overlay networks. In: IEEE Telecommunication Networking Workshop on QoS in Multiservice IP Networks, February 2008, pp. 46–52 (2008)
9. Vieira, S.L., Liebeherr, J.: Topology design for service overlay networks with bandwidth guarantee. In: IEEE International Workshop on Quality of Service, IWQoS, pp. 211–220 (2004)
10. Martins, E.Q., Pascoal, M.M., Santos, J.L.: Deviation algorithms for ranking shortest paths. *International Journal of Foundations of Computer Science* 10, 247–261 (1999)