Using Metaheuristics for Topological Design of Service Overlay Networks

Sibelius Lellis Vieira

Department of Computer Science
Catholic University of Goiás
Goiânia, GO 74605-010, Brasil,
sibelius@ucg.br

Abstract. Adequately supporting applications with Quality-of-service (QoS) requirements is a challenging task, particularly in the Internet. In great part, this is due to the fact that the provisioning of end-to-end QoS to traffic that traverses multiple autonomous systems (ASes) requires a level of cooperation between ASes that is difficult to achieve in the current architecture. Recently, service overlay networks have been considered as an approach to QoS deployment that avoids these difficulties. In this study, the problem of the topological synthesis of a service overlay network is addressed. Endsystems and nodes of the overlay network are connected through ISPs that supports bandwidth reservations. The topology design problem is expressed as an optimization problem. Even though the design problem is related to the (in general NP-hard) quadratic assignment problem, it is shown that relatively simple heuristic algorithms can deliver results that are sometimes close to the optimal solution with little overhead.

1 Introduction

The structure of the Internet is based on a large number of independently operated networks (autonomous systems or ASes) [6], where peering points provide the connection of separate autonomous systems of the Internet into one cooperating infrastructure [9]. The economics of peering make the provisioning of end-to-end Quality of Service (QoS) unlikely, which makes the QoS support in the Internet a challenging task. Most peering agreements are bilateral contracts between ASes at peering points and end-to-end QoS is a cooperative effort of all ASes on an end-to-end path of a flow with service guarantees. Although an ISP (Internet Service Providers) may have an interest in providing QoS guarantees within its own AS, there is a lack of incentives to support similar service guarantees to customers of remote autonomous systems [6].

Overlay networks have been considered as a higher level mechanism that can support new services to users on top of the network-layer infrastructure without requiring changes to the infrastructure or its business practices [11]. In this work, a framework is considered where a value-added overlay network that sits on top of an infrastructure of ISPs, called *provider network*, supports end-to-end QoS guarantees to a collection of subscribers. The provider network consists of *provider nodes* and a set of subscribers, called *endsystems*. Each provider node and endsystem gains access to the Internet using one or more ISPs (see Figure 1). The provider nodes are connected to each other and endsystems are connected to a provider node by ISPs. Two provider nodes can establish a link in the provider network if they are both connected to the same ISP. By the same token, an endsystem can access a given provider node if both are connected to the same ISP. In Figure 2, the relationship of endsystems, provider nodes, and ISPs is illustrated.

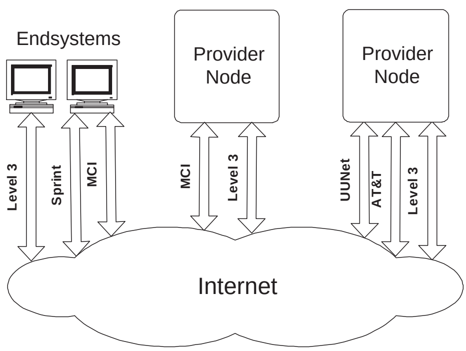

Fig. 1. Endsystems and Provider nodes.

As a network that is based on services provided by ISPs, the provider network is a private network that buys services, such as guaranteed bandwidth, from different ISPs and, according to pre-established agreements, provides bandwidth guarantees to endsystems. The endsystems are connected to the provider nodes through ISPs and these connections are administered by the provider network. Endsystems purchase QoS services from the provider network, which in turn purchases bandwidth guarantees from each ISPs for traffic between provider nodes, as well as for traffic between provider nodes and endsystems.

Given the connectivity of provider nodes and endsystems to a set of ISPs, as shown in Figure 2, the problem of designing a provider network topology consists of assigning each endsystem to one provider node, and in assigning pairs of provider nodes connected to a common ISP, such that all endsystems can exchange traffic over a path of provider nodes.

The purpose of this work is to find a topology for a provider network which minimizes the cost of the provider network for interconnection of provider nodes and access of endsystems. The provider network topology is formulated here as the solution to a 0-1 integer programming problem. Since such problems are, in general, solvable only for small instances or for special cases, we investigate the use of heuristics, such as simulated annealing, to find good solutions to the problem [4]. In addition, it is shown that, in some special cases, optimum solutions can be obtained even for larger networks. As an example, in Figure 3, a feasible provider network topology that corresponds to the set of endsystems and provider nodes of Figure 2 is presented. The contribution of this paper is that it poses the topological design of provider networks as a research problem, and show that appropriate algorithms can construct effective solutions with relatively small computational overhead.

Overlay Networks have received a great deal of attention lately, since they facilitate the implementation and deployment of new services. For instance, service overlay networks (SON) can provide generic overlay services that can be used for a variety of applications [1, 11]. In particular, a service overlay network has been proposed as a means to provide value-added services, including end-to-end QoS, based on user requests [6]. Other approaches, like OverQoS [15] proposes a value-added service based on ISP infrastructure that is aimed at statistical guarantees. QUEST [7] is another overlay network that has been proposed to address QoS provisioning, as well as other services. A review of these related work assumptions indicates, however, that the topological design questions have been given only little attention. A commercial service overlay network that is closely related to the proposed provider network is Internap [8]. One structural difference is that access in Internap is provided by P-NAPs (provider national access points), which are Internap Point-of-presence (PoP), while our approach does not assumes PoPs (access to endsystems is provided by acquiring bandwidth from ISP links). There is also no topological design problem stated for Internap.

The paper is structured as follows. In Section 2 the parameters of the topology design problem are established and the topology synthesis is formulated as a solution to an optimization problem. In Section 3 conditions under which the optimization problem can be easily solved are presented. In Section 4 heuristic algorithms that can solve the optimization problem for general networks are discussed. In Section 5 the methods are validated in numerical experiments and brief conclusions are presented in Section 6.

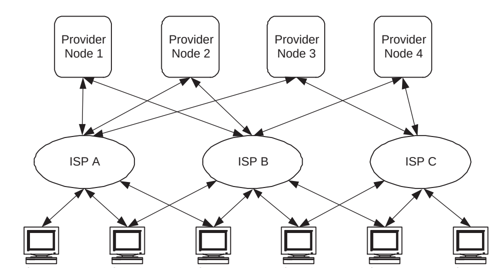

Fig. 2. Relationships between provider nodes, endsystems and ISPs.

2 The Topology Design Problem

In this section the topology design for a provider network is presented in terms of the solution to an optimization problem. The input to the problem is the connectivity between endsystems, provider nodes, and ISPs as shown in Figure 2. With this data, a provider network topology is generated such that the resulting topology minimizes a given cost metric. An example of such topology is shown in Figure 3. The cost metric is chosen to reflect the cost to the provider network. A network with M endsystems and N provider nodes is considered. E_i is refered as ith endsystem and P_j as the jth provider node.

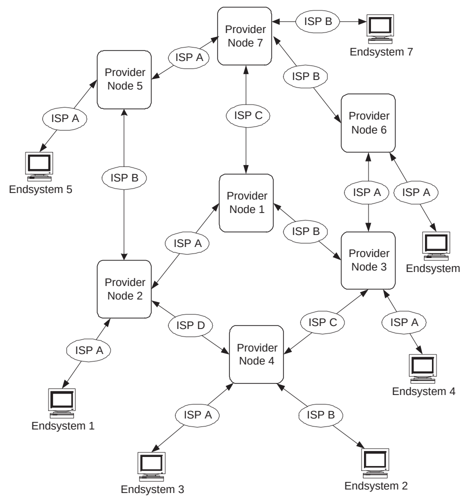

Fig. 3. Solution to the topology of the provider network.

In the provider network considered here, each endsystem is connected to exactly one provider node. An endsystem accesses a provider node using an ISP that is connected to both the endsystem and the provider node. There is a constant cost a_{ij} for reserving a unit of bandwidth (e.g., a Mbps) from endsystem E_i to provider node P_j. This cost is referred to as *access cost*. If there is no ISP to which both E_i and P_j are connected to, then E_i cannot be assigned to P_j, and the access cost to $a_{ij} = \infty$. If the same E_i and P_j can be connected by more than one ISP, then a_{ij} represents a connection through the ISP with minimal cost. Hence, a_{ij} implies the selection of an ISP to connect E_i to P_j.

Provider nodes are connected to each other through ISPs. There is a *transport link* between two provider nodes, if both provider nodes have at least one common ISP. The cost to reserve a unit of bandwidth from P_i to P_j is t_{ij}. This cost is refered as the *transport cost*. If P_i and P_j are not connected to the same ISP, then $t_{ij} = \infty$. If two provider nodes can be connected by more than one ISP, then t_{ij} is the cost through the ISP that incurs the least cost.

The provider network reserves bandwidth on access links and transport links for the traffic between endsystems. It is assumed that the amount of bandwidth reserved for the traffic between endsystems is given by a reservation matrix $B = \{b_{ij}\}$, where b_{ij} is the bandwidth that is reserved for the traffic from E_i to E_j, and we have $b_{ii} = 0$. Clearly, it is desirable to keep the reserved bandwidth close to the actual traffic rate. Thus, the reservation matrix can be estimated based on measurements or predictions. The reservation matrix can vary over time. However, changes to the reservation matrix require to recalculate the provider network topology. We let $B_i = \sum_{j=1}^{M} b_{ij}$ denote the total bandwidth reserved for traffic generated at E_i.

To obtain a provider network topology as shown in Figure 3, two problems must be solved. First, for each endsystem a provider node must be selected that carries the traffic between the endsystem and the provider network. Second, transport links must be selected between provider nodes so that the provider nodes can relay the traffic between the endsystems. The total cost of the provider network is the cost of the access links and the transport links of the resulting topology, weighted by the amount of reserved bandwidth on the links. The objective is to determine a provider network topology such that the total cost is minimized.

The construction of the provider network topology is done in two steps. In the first step, it is only considered provider nodes and their transport links, in order to determine a route between each pair of provider nodes, such that the total transport cost is minimized. These routes are determined independent of the assignments of endsystems to provider nodes and independent of the amount of bandwidth reserved on a route. Given two provider nodes P_n and P_m, the transport cost between the provider nodes is minimized if traffic is sent on the least-cost path connecting the two provider nodes. Hence, a transport link with cost t_{ij} is part of the topology of the provider network if the link is on the least-cost route between some pair of provider nodes [13]. Let us denote by r_{nm} the least-cost route between P_n and P_m, and let us write '$(ij) \in r_{nm}$' if the transport link between P_i and P_j is part of this route. The cost of the least-cost route per unit of reserved bandwidth between P_n and P_m, denoted by l_{nm}, is given by $l_{nm} = \sum_{(ij)\in r_{nm}} t_{ij}$.

In the second step, it should be determined how to connect endsystems to provider nodes. Given that, once this determination is made, traffic between endsystems is taking the least-cost route in the transport network, we have fully determined the transport network. To express the assignment of endsystems to provider nodes as an optimization problem, a 0-1 decision variables x_{ij} is introduced, such that $x_{ij} = 1$ if E_i is assigned to P_j, and $x_{ij} = 0$ otherwise. Now the total cost of the provider network can be stated as an objective function. The formulation of the optimization problem is as follows:

$$\begin{aligned} \textit{Minimize} \ & \sum_{i=1}^{M}\sum_{k=1}^{N} B_i a_{ik} x_{ik} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{N} x_{ij} x_{kl} b_{ik} l_{jl} \\ & + \sum_{j=1}^{M}\sum_{l=1}^{N} B_j a_{jl} x_{jl} \\ & \textit{subject to} \quad \sum_{j=1}^{N} x_{ij} = 1 \quad \textit{for} \ \ i = 1, \ldots, M \end{aligned} \tag{1}$$

The optimization problem in Eqn. (1) is a variant of the well-known quadratic assignment problem (QAP) [12]. In this problem, which is known to be NP-hard, one assigns one item to a resource such that each item is assigned to exactly one resource and each resource has exactly one item assigned to it. In this context, items correspond to endsystems and resources correspond to provider nodes. The difference of our problem to the QAP is that more than one endsystem can be connected to a provider node. Also, it is possible that a provider node has no endsystem assigned to it. The complexity of the overall entire topology construction is dominated by the assignment of endsystems to provider nodes, resulting in a complexity of $O(N^M)$. In the next section it is showed that the assignment of endsystems to provider nodes can be computed efficiently in certain special cases, where the access cost and the transport cost can be related according to a triangular inequality.

3 The Matrix-combination Problem Formulation

The optimization problem can be expressed as an equivalent matrix-combination problem. This representation expresses the combinatorial structure of the problem better than the formulation given in Eqn. (1). By viewing transport and access costs in matrix form, one can more easily identify conditions under which a provider network topology calculation does not require the solution of an NP-hard quadratic assignment problem.

Let us view the parameters of the provider network topology in terms of matrices. Let matrices $B = \{b_{ij}\}$, $L = \{l_{ij}\}$ and $A = \{a_{ij}\}$ represent, respectively, the bandwidth requirements, the transport cost and the access cost. Let u be a mapping of $i \in \{1, .., M\}$ such that $u(i) = j$, where $j \in \{1, .., N\}$. In terms of Eqn. (1), we have $u(i) = l$ if and only if $x_{il} = 1$. Note that a vector $\underline{u} = (u(1), u(2), \ldots, u(M))$, with $u(i) \leq N$ for $i = 1, .., M$ gives a feasible assignment of endsystems to provider nodes. If for some E_i we have $a_{ik} = \infty$, then $u(i) = k$ is not part of any feasible solution.

The objective function in Eqn. (1) can be rewritten as

$$Z(\underline{u}) = \sum_{i=1}^{M} \sum_{j=1}^{M} b_{ij} (a_{iu(i)} + l_{u(i)u(j)} + a_{ju(j)}) \ . \tag{2}$$

It is easily verified that the function $Z(\underline{u})$ is the objective function for the original problem. A minimization over all vectors $\underline{u}$ without side conditions yields a solution to the topology design problem. Note that the side conditions in the original problem are implicitly given via the definition of the $u(i)$'s.

In general, the reformulated optimization in Eqn. (2) is no simpler than the original problem. However, there are special cases when the relationship between matrices L and A, representing, respectively, the transport and access cost, leads to a problem with only linear complexity.

Let us choose v_i such that a_{iv_i} is the smallest value among the a_{ij}, i.e., $a_{iv_i} = \min_j \{a_{ij}\}$. Then, the complexity of solving the optimization problem can be reduced if the following conditions hold:

(C1) $l_{ij} \leq l_{ik} + l_{kj}$ for all $i, j, k \leq N$.
(C2) $a_{ij} \geq a_{iv_i} + l_{v_i j}$ for all $i \leq M$ and $j, v_i \leq N$.

In our setting, condition (C1) always holds since the elements in matrix L are based on the calculation of least-cost paths. Hence, the triangular inequality is enforced by construction. Condition (C2) is satisfied if the cost structure is such that the access cost outweighs the transport cost. In such a scenario, the access cost of endsystem E_i is minimized by assigning it to the provider node with lowest cost, namely P_{v_i}.

Let us now evaluate the objective function $Z(\underline{u_I})$, where u_I is the mapping in which $u(i) = v_i$ for all i. To simplify notation, we will refer to $u(i)$ as u_i.

Lemma 1. *The objective function $Z(\underline{u})$ is minimized for the mapping $u(i) = v_i$, if the matrices A and L are such that conditions (C1) and (C2) are satisfied, where $a_{iv_i} = \min_j\{a_{ij}\}$.*

4 Simulated Annealing Application

If the network does not satisfy conditions (C1) and (C2) given in the previous section, the computational effort to solve the optimization problem precludes the use of exact solution methods in large networks. Exact solutions of the quadratic assignment problem can been obtained only for problem sizes with at most a few dozens of endsystems and provider nodes [4]. Thus, to solve the provider network design problem for larger networks, one needs to resort to heuristic methods. There is a large set of heuristic algorithms for solving combinatorial problems such as our quadratic assignment problem. These include the construction method, improvements method, Tabu search algorithms, simulated annealing and genetic algorithms [2, 4]. These methods use an initial solution and iteratively attempt to improve the solution by performing a local search. The simulated annealing was selected as the heuristic algorithm, since it has been shown to perform very well for quadratic assignment problems [3, 5, 14].

4.1 Simulated Annealing

Simulated annealing draws an analogy between problems from statistical physics and combinatorial problems [10]. The procedure considers a system in thermal equilibrium at some energy level E_k and temperature t. Then, a random perturbation is applied to the system and the corresponding change in the energy is evaluated. If the new energy level E_j is less than E_k, the perturbation is accepted and the system evolves to a new state. If the energy level increases, the system evolves to a new state with a probability that is proportional to $e^{\frac{E_i - E_k}{t}}$. After a reasonable large number of states have been generated and evaluated, the temperature is decreased and new states are generated. As the temperature decreases, the probability of accepting a perturbation that increases the energy of the current state also decreases. The algorithm terminates when further perturbations do not decrease the energy level.

The temperature t, with initial value t_0, is a parameter that controls the evolution of the algorithm. The initial value t_0 is set to a high value and an initial solution, denoted by S_0. We refer to S_{best}, S_{cur} and S_{new} as variables that represent the best solution, the current solution and the new solution obtained in the current iteration, respectively. The values of the objective functions for the initial, new, best and current solutions are refered as Z_0, Z_{new}, Z_{best} and Z_{cur}, respectively. From the current solution S_{cur} one can obtain a new solution S_{new} by performing a local random search through $SEARCH(S_{cur})$. The local search changes the current solution by randomly assigning a new provider node to one randomly chosen endsystem. Given the constraints of the topology design problem, if the endsystem cannot be assigned to the chosen provider node, a new search is performed. A new solution is accepted if $\Delta(Z_{new}, Z_{cur})$ is negative. In such a case, a check is made to see if the best solution Z_{best} can be improved. If $\Delta(Z_{new}, Z_{cur})$ is non-negative, the new solution is accepted with a probability that decreases with the temperature t.

The process to decrease the temperature uses a so-called geometric schedule, in which the temperature at any level k (t_k) decreases in a geometric progression ($t_{k+1} = r_c t_k$ with $0 < r_c < 1$). At each temperature level, a fixed number of solutions are evaluated. The number of solutions evaluated at a temperature level is referred to as *repetition factor*, and denoted by Rep_{max}. This repetition factor should be sufficiently large so that good solutions are found at each temperature level. The process continues until a temperature t_f is reached where no further improvements to the objective function can be found. At this point, the algorithm terminates and yields the solution S_{best} and the value of the objective function Z_{best}.

5 Numerical Evaluation

In this section the approaches for creating topologies for service overlay networks are evaluated, in an attempt to answer the following questions:

- How closely do the presented heuristic algorithms, i.e., simulated annealing approximate the optimal solution and at what processing overhead?
- What is the cost sensitivity of the algorithms with respect to the number of provider nodes?

For the evaluation a network of provider nodes using the Georgia Tech Internetwork Topology Model (GT-ITM) [16] is generated, and a random graph is produced choosing the ‘Pure Random’ model that represents the connectivity between provider nodes. Note that the GT-ITM is used to simulate the provider network potential links, as illustrated in Figure 1 and not the underlying Internet topology. An edge in the graph indicates that two provider nodes share a common ISP. The Pure Random model inserts an edge with probability P, where P is an input parameter, called the *edge probability*. The transport cost between two provider nodes, t_{ij} for provider nodes P_i and P_j, is drawn from a uniform distribution in the range $[5, 50]$ (for some arbitrary cost metric), and $t_{ij} = \infty$

if GT-ITM does not insert an edge between provider nodes P_i and P_j. Unless stated otherwise, the access cost a_{ij} of endsystems E_i to provider node P_j is also drawn from a uniform distribution in the range $[5, 50]$. It is assumed that each endsystem can be connected to one or more provider nodes. In the numerical experiments each endsystem can access a randomly selected sample of $p_\alpha \cdot 100\%$ of the provider nodes, where $0 \leq p_\alpha \leq 1$ is a parameter. The reservation matrix has coefficients b_{ij} that are uniformly distributed in the range $[10, 20]$ Mbps.

5.1 Evaluation of the Heuristics Algorithms

First an evaluation of the performance of the simulated annealing heuristic is made by comparing them to the results of the exact solution of Eqn. (1). For smaller networks, Eqn. (1) can be solved, e.g., using branch-and-bound methods or similar techniques. If the network is large, an exact solution can be found only for the special cases discussed in Section 3.

Table 1. Evaluation of simulated annealing for small networks ($M = N = 9$).

Repetition factor Rep_{max}	Average deviation from minimum (in Percent)	Number of optimal solutions found (Total is 100)
10	6.59%	1
20	4.44%	3
30	1.41%	4
40	0.02%	7
50	0.02%	9

A comparison of the minimum cost according to Eqn. (1) with the results obtained by simulated annealing is made for a small network with $M = 9$ endsystems and $N = 9$ provider nodes, where the networks are generated as described above. For this size, the optimal solution can be computed reasonably quickly. For this experiment, we set $p_\alpha = 1$ and $P = 0.5$.

In Table 1, simulated annealing results are compared with the optimal solutions. The processing overhead is represented by the different values of the repetition factor Rep_{max} and the number of temperature levels. Given t_0 and t_f, respectively the initial and final temperature, the processing overhead can be given by a number of iterations, which is equal to $Rep_{max} log(t_f/t_i)/log(r_c)$. For typical values of $t_0/t_f = 10^4$ and $r_c = 0.95$, that yields 180 temperature levels, and $180 Rep_{max}$ iterations. The first column of Table I gives the values of the repetition factor. The second column gives the average deviation of simulated annealing results from the optimal solution, averaged over 100 runs of the experiment. The third column depicts how often, among the 100 runs, the simulated annealing algorithm found the optimal solution. The results indicate that for a repetition factor larger than 40, simulated annealing gets very close to

Table 2. Evaluation of simulated annealing for large networks networks

Number of endsystems (M)	10	20	30	40	50	60	70	80	90	100
Rep_{max} for $N = M$	200	800	1800	3200	5000	7200	8800	12800	16200	20000
Rep_{max} for $N = 10$	200	400	600	800	1000	1200	1400	1600	1800	2000

the optimal solution, and even finds the minimum value of the objective function in some cases.

Consider now larger networks. Here, a comparison with the optimal solution is possible only when conditions (C1) and (C2) from Section 3 are satisfied. For this experiment, we generate a two sets of data: one with 10 provider nodes and a varying number of endsystems from 10 to 100, taken in steps of 10, and the other with a varying number of both provider nodes and endsystems, from 10 to 100, with $M = N$. We use the parameters $p_\alpha = 1$ and $P = 0.5$. To enforce that access nodes obey condition (C2), we select the access cost different from our description above. For each endsystem E_i, one provider node P_{v_i} is selected and the value a_{iv_i} randomly from the interval $[5, 50]$ is drawn. Then, for all other provider nodes, we set $a_{ij} = a_{iv_i} + l_{v_i j}$, thereby enforcing that condition (C2) holds.

For these networks, we now compare the simulated annealing algorithm with the optimal solution. We present the comparison in terms of the repetition factor Rep_{max} needed to get $X\%$ of the optimal solutions. The results are shown in Table II for $X = 99$. The table shows that the simulated annealing algorithm is able to find optimal solutions 99% of the time, with a repetition factor that increases with the size of the network according to a relationship given by $Rep_{max} = kMN$, with k around 2. Although not shown, we find similar results for different values of k. It can also be observed that the repetition factor increases linearly with the number of endsystems. We remark that it has been pointed out elsewhere [3] that QAP solutions exhibit the same linearity of the repetition factor. As QAP and our topology design problem have structural similarities, we expect to observe the same scaling properties.

5.2 Impact of the Number of Provider Nodes

Next, we investigate the relationship of the cost of the provider network to the number of provider nodes. To that end, we consider a network with 100 endsystems and a varying number of provider nodes, in the range 10 to 100. The results are presented in Figure 4 and Figure 5. Each data point represents 100 repetitions of the simulated annealing algorithm. The relative topology cost is the cost relative to the cost obtained for a topology with 10 provider nodes. In Figure 4 we select $p_\alpha = 0.9$ and in Figure 5, $p_\alpha = 0.5$. The provider nodes have edge probability of $P = 0.1$, $P = 0.5$ and $P = 0.9$. The results indicate that the topology cost is sensitive to the number of provider nodes, showing a tendency to

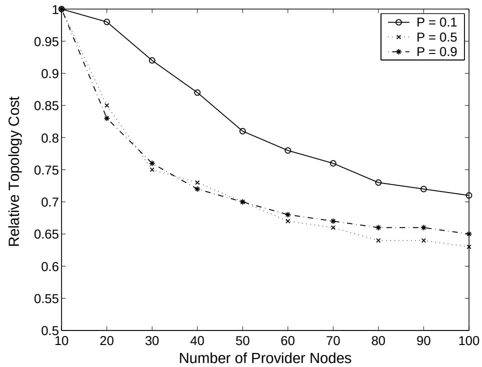

Fig. 4. Relation between topology cost and number of provider nodes for $p_\alpha = 0.9$.

decrease with increasing number of provider nodes. This is explained by the fact that increasing the number of provider nodes has little impact on the transport cost. On the other hand, increasing the number of nodes enlarges the assignment base for endsystems. With more provider nodes to choose from, the endsystem assignment may be able to achieve a smaller access cost, thereby decreasing the total cost of the topology.

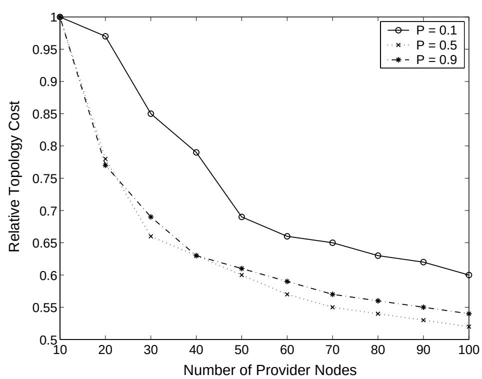

Fig. 5. Relation between topology cost and number of provider nodes for $p_\alpha = 0.5$.

6 Conclusions

This paper addressed the problem of designing a network topology for a service overlay network, which offers value-added services to customers, and which purchases links with bandwidth guarantees from a number of ISPs. Under the assumptions made in this paper, it is shown that the general problem of designing a topology for the service overlay network is NP-hard. In some cases, when the cost structure of the underlying network satisfies specified conditions, we can see that the topology design problem may have only linear complexity.

A number of heuristic algorithms that can construct a topology is presented even if an exact solution is not feasible. The presented numerical results demonstrated that in cases where a comparison with an optimal topology is feasible, the heuristic algorithms are reasonably accurate. The problem formulation here makes a number of assumptions and arguably a strong assumption is that the cost of sending traffic with bandwidth guarantees across an ISP is proportional to the amount of transmitted traffic. Other pricing structures, e.g., a flat-rate pricing, will result in a different problem formulation. These and other assumptions should be relaxed in future work. A comparison with other metaheuristics can also be object of further investigation.

Acknowledgments

We would like to thank PROPE/UCG for the support.

References

1. D.G. Andersen, H. Balakrishnan, M.F. Kaashoek and R. Morris. Resilient Overlay Network. In *Proc. 18th ACM SOSP 2001*, Banff, October 2001.
2. C. Blum and A. Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268-308, 2003.
3. R.E. Burkard and F. Rendl. A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems. *European Journal of Operational Research*, 17:169-14, 1983.
4. E. Cela. *The Quadratic Assignment Problem*. Kluwer Academic Publishers, 1998.
5. D. Connoly. An improved annealing scheme for the QAP. *European Journal of Operational Research*, 46:93-100, 1990.
6. Z. Duan, Z.L. Zhang and Y.T. Hou. Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning. In *Proc. 10th IEEE International Conference on Network Protocols*, Paris, France, November 2002.
7. X. Gu, K. Nahrstedt, R.H. Chang and C. Ward. QoS-Assured Service Composition in Managed Service Overlay Networks. In *Proc. IEEE 23rd International Conference on Distributed Computing Systems*, Providence, May 2003.
8. Internap Network Services Corporation. *http://www.internap.com*.
9. International Telecommunication Union. IP-based networks: Pricing of telecommunication services. Final report, January 2003.
10. S. Kirkpatric, C. Gelatt and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220:291-307, 1983.
11. Z. Li and P. Mohapatra. QRON: QoS-aware Routing in Overlay Networks. *IEEE Journal of Selected Areas of Communications*, 22(1):29-40, January, 2004.
12. M. Padberg and M.P. Rijal. *Location, Scheduling, Design and Integer Programming*. Kluwer Academic Publishers, 1996.
13. T. Robertazzi. *Planning Telecommunication Networks*. John Wiley & Sons, 1999.
14. S. Sofianopoulou. Simulated annealing applied to the process allocation problem. *European Journal of Operational Research*, 60:327-334, 1992.
15. L. Subramanian, I. Stoica, H. Balakrishnan and R.H. Katz. OverQoS: Offering Internet QoS using Overlays. In *Proc. HotNet-I Workshop*, October 2002.
16. E. W. Zegura. GT-ITM: Georgia Tech Internetwork topology models (software). *http://www.cc.gatech.edu/project*, 1996.