Topology Design for Service Overlay Networks with Bandwidth Guarantees

Sibelius Lellis Vieira
Department of Computer Science
Catholic University of Goias
Goiania, Brasil 74605–010
Email: sibelius@ucg.br

Jörg Liebeherr
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
Email: jorg@cs.virginia.edu

***Abstract*— The Internet still lacks adequate support for QoS applications with real-time requirements. In great part, this is due to the fact that provisioning of end-to-end QoS to traffic that traverses multiple autonomous systems (ASs) requires a level of cooperation between ASs 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, we address the problem of the topological synthesis of a service overlay network, where endsystems and nodes of the overlay network (provider nodes) are connected through ISPs that supports bandwidth reservations. We express the topology design problem as an optimization problem. Even though the design problem is related to the (in general NP-hard) quadratic assignment problem, we are able to show that relatively simple heuristic algorithms can deliver results that are sometimes close to the optimal solution.**

I. Introduction

Supporting Quality-of-Service (QoS) in the Internet remains a challenging task, albeit various efforts in the last decade to enhance the basic best effort service. An important reason for the lack of QoS deployment is the Internet's own structure, which is based on a large number of independently operated networks (autonomous systems or ASs) [8], where peering points provide the connection of separate autonomous systems of the Internet into one cooperating infrastructure [12]. The economics of peering make the provisioning of end-to-end QoS unlikely. Whereas most peering agreements are bilateral contracts between ASs at peering points, end-to-end QoS is a cooperative effort of all ASs 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 [21].

To overcome these issues, 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 the business practices [15]. Using overlay networks, network services have been proposed that address the needs of applications for fault-tolerance [2], multicast communication [7], security [13], file sharing [9] and QoS [21].

We consider a framework where a value-added overlay network that sits on top of an infrastructure of ISPs, called

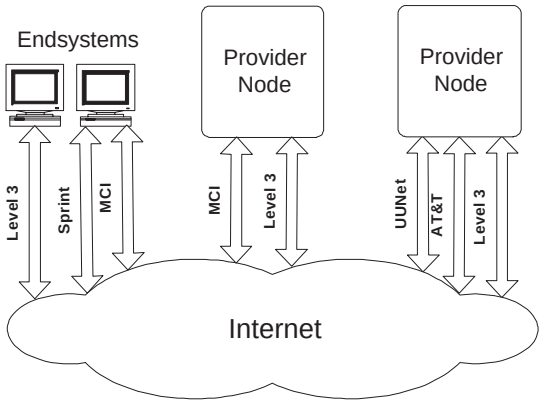

Fig. 1. Endsystems and Provider nodes.

QoS Provider Network or simply *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. Likewise, an endsystem can access a given provider node if both are connected to the same ISP. In Figure 2, we illustrate the relationship of endsystems, provider nodes, and ISPs. As a network that is based on services provided by ISPs, the provider network 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 ISP 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

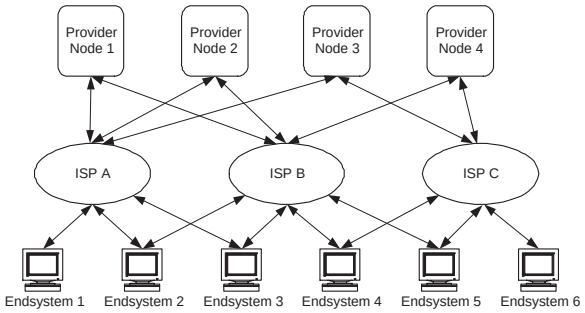

Fig. 2. Relationships between provider nodes, endsystems, and ISPs. An endsystem and provider node or two provider nodes can have a link in the provider network if they have access to a common ISP.

of provider nodes connected to a common ISP, such that all endsystems can exchange traffic over a path of provider nodes. As an example, in Figure 3, we present a feasible provider network topology that corresponds to the set of endsystems and provider nodes of Figure 2.

In this paper, we present a methodology that can guide the topological design of a provider overlay network. To our knowledge, the problem of devising good topologies for service overlay networks has not been studied before. The purpose of this work is to design a provider network which minimizes the cost of the provider network for interconnection of provider nodes and access of endsystems. We formulate the provider network topology as the solution to an optimization problem. We show that this optimization problem has linear and quadratic terms. 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 [6]. In addition, we are able to show that, in some special cases, optimum solutions can be obtained even for larger networks.

Overlay Networks have received a great deal of attention lately, since they facilitate the implementation and deployment of new services. Several models for application-layer overlays have emerged, generally aimed at providing services tailored to specific applications, such as multicasting, content delivery or peer-to-peer file sharing [1], [3], [7]. However, a review of the related work indicates that topological design questions have been given only little attention. A service overlay network (SON) provides generic overlay services that can be used for a variety of applications [2], [15]. Service overlay networks were proposed as a means to provide value-added services, including end-to-end QoS, based on user requests [8]. The architecture is based on services gateways connected by the underlying network domain with bandwidth guarantees. The design goal of a SON is to provision adequate bandwidth to support end-to-end QoS services and satisfy traffic demands while minimizing the bandwidth cost to the SON provider. The cost issues are related to bandwidth costs and penalty costs. The latter is incurred when a QoS violation occurs. The authors assumed the existence of a SON topology and of routes between SON nodes and do not address the topological design aspects. The OverQoS approach [21] also proposes a value-added service based on ISP infrastructure that is aimed at statistical guarantees. Here, the overlay network provides enhanced services that bound the loss rate experienced by overlay traffic, without specific consideration to cost and topological aspects. QUEST [10] is another overlay network that has been proposed to address QoS provisioning, as well as other services. For example, QUEST addresses the management of QoS provisioning for composed services based on individual service requests. QUEST also assumes that a directed graph representing a service overlay network topology is given. There is a large body of works on structured overlays, e.g., [17], [19], [23], which build an overlay network as a graph that implements an abstract data structure, e.g., a tree, a hypercube or a distributed hash table. Structured overlays are popular choices for file sharing and multicasting overlay networks. However, the objectives and design issues of structured overlays are very different from those of the overlay networks considered here. A commercial service overlay network that is closely related to our work is Internap [11]. The main difference to our work is that access for endsystems is provided by the provider network, and not by the ISPs. We note that our topology design approach can be extended to apply to the assumptions made by the Internap overlay network.

The problem formulation in this paper makes a number of assumptions that govern the relationship between customers, the provider network, and ISPs. Arguably a strong assumption is that the cost of sending traffic with bandwidth guarantees across an ISP is proportional to the amount of reserved bandwidth. These and other assumptions can be relaxed, e.g., by considering flat-rate pricing, which, however, may result in a different problem formulation. The contribution of this paper is that it poses the topological design of service overlay networks as a research problem, and, for a specific pricing structure, shows that appropriate algorithms can construct effective solutions with relatively small computational overhead.

The remainder of this paper is structured as follows. In Section II we formulate the parameters of the topology design problem and state the topology synthesis as a solution to an optimization problem. In Section III we consider conditions under which the optimization problem can be easily solved. In Section IV we present heuristic algorithms that can solve the optimization problem for general networks. We also show how grouping of geographically close endsystems into clusters can further reduce the effort of designing the topology of the provider network. In Section V we validate our methods in numerical experiments. We present brief conclusions in Section VI.

II. Formulation of the Topology Design Problem

In this section we formulate the topology design for a provider network 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, we generate a provider network topology, as shown in Figure 3, such that the resulting topology minimizes

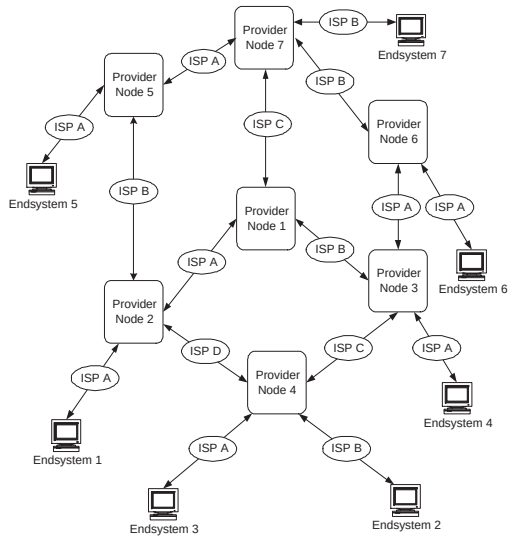

Fig. 3. Solution to the topology of the provider network.

a given cost metric. The cost metric is chosen to reflect the cost to the provider network. We consider a network with M endsystems and N provider nodes. We refer to the ith endsystem as ES_i and to the jth provider node as PN_j. The basic notation is presented in Table I.

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 α_{ij} for reserving a unit of bandwidth (e.g., a Mbps) from endsystem ES_i to provider node PN_j. This cost is referred to as *access cost*. If there is no ISP to which both ES_i and PN_j are connected to, then ES_i cannot be assigned to PN_j, and we set the access cost to $\alpha_{ij} = \infty$. If the same ES_i and PN_j can be connected by more than one ISP, then α_{ij} represents a connection through the ISP with minimal cost. Hence, α_{ij} implies the selection of an ISP to connect ES_i to PN_j.

Provider nodes are connected to each other through ISPs. We say that 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 PN_i to PN_j is l_{ij}. We refer to this cost as the *transport cost*. If PN_i and PN_j are not connected to the same ISP, we set $l_{ij} = \infty$. If two provider nodes can be connected by more than one ISP, then l_{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. We assume that the amount of bandwidth reserved for the between endsystems is given by a reservation matrix $\Omega = \{\omega_{ij}\}$, where ω_{ij} is the bandwidth that is reserved for the traffic from ES_i to ES_j, and we have $\omega_{ii} = 0$. 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

TABLE I

BASIC NOTATION.

ES_i	Endsystem i
PN_j	Provider node j
M	Number of endsystems
N	Number of provider nodes
y_{ij}	0-1 decision variable that indicates if ES_i is assigned to PN_j
α_{ij}	Access cost (per unit of traffic) for traffic from ES_i to PN_j
l_{ij}	Transport cost (per unit of traffic) for traffic on the transport link between PN_i and PN_j
b_{ij}	Cost of least-cost route (per unit of traffic) for traffic between PN_i and PN_j
ω_{ij}	Reserved bandwidth for traffic from ES_i to ES_j
Ω_i	Reserved bandwidth from ES_i to all other endsystems

$\Omega_i = \sum_{j=1}^{M} \omega_{ij}$ denote the total bandwidth reserved for traffic generated at ES_i.

To obtain a provider network topology as shown in Figure 3, we must solve two problems. First, for each endsystem we must select a provider node that carries the traffic between the endsystem and the provider network. Second, we must select transport links between provider nodes so that the provider nodes can relay the traffic between the endsystems. The total cost of the provider network are the costs 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 three steps. In the first step, we only consider provider nodes and their transport links, and 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 PN_n and PN_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 l_{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 [18]. Let us denote by r_{nm} the least-cost route between PN_n and PN_m, and let us write '$(ij) \in r_{nm}$' if the transport link between PN_i and PN_j is part of this route. The cost of the least-cost route per unit of reserved bandwidth between PN_n and PN_m, denoted by b_{nm}, is given by $b_{nm} = \sum_{(ij)\in r_{nm}} l_{ij}$.

In the second step, we determine 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. For an illustration we refer to Figure 3. Suppose that we have determined that ES_1 will be connected to PN_2 and that ES_4 will be connected to PN_3. Then the total access and transport costs incurred by traffic between ES_1 and ES_4 is given by $\omega_{14}\alpha_{12} + \omega_{14}b_{23} + \omega_{14}\alpha_{43}$. Assuming that the least-cost route between PN_2 and PN_3 is $PN_2 \rightarrow PN_4 \rightarrow PN_3$, we have $b_{23} = l_{24} + l_{43}$. To express the assignment

of endsystems to provider nodes as an optimization problem, we now introduce 0-1 decision variables y_{ij}. We set $y_{ij} = 1$ if ES_i is assigned to PN_j, and $y_{ij} = 0$ otherwise. Now we can state the total cost of the provider network as an objective function. The formulation of the optimization problem is as follows:

$$\textit{Minimize} \quad \sum_{i=1}^{M}\sum_{k=1}^{N}\Omega_i\alpha_{ik}y_{ik} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{N} y_{ij}y_{kl}\omega_{ik}b_{jl} + \sum_{j=1}^{M}\sum_{l=1}^{N}\Omega_j\alpha_{jl}y_{jl}$$
$$\textit{subject to} \quad \sum_{j=1}^{N} y_{ij} = 1 \quad \textit{for} \quad i = 1,\ldots,M \qquad (1)$$

The first term in the objective function expresses the total access cost of traffic entering the provider network. The second term expresses the transport cost between provider nodes. The third term expresses the total access cost for traffic leaving the provider network. The side condition ensures that each endsystem is assigned to exactly one provider node.

The optimization problem in Eqn. (1) is a variant of the well-known quadratic assignment problem (QAP) [16]. 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 our 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.

In the third and final step we construct the provider network topology based on the outcome of the optimization. We eliminate all provider nodes that have no endsystem assigned to it and that are not on a least-cost route between two provider nodes that are connected to endsystems. When a provider node is excluded, so are all the transport link incidents to the node. In Figure 3, we show a provider network topology in which PN_1 has no endsystem assigned to it. If PN_1 is not on the least-cost path between any of the other provider nodes, it will be eliminated from the provider network topology. Assuming that PN_1 is part of the least-cost path from PN_3 to PN_7, but not part of any other least-cost path, the transport link between PN_1 and PN_2 is excluded.

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 we show 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.

III. Endsystem-node Assignment as a Matrix-combination Problem

We now express the optimization problem as an equivalent matrix-combination problem. As we will see, 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, we can 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 $\Omega = \{\omega_{ij}\}$, $B = \{b_{ij}\}$ and $\alpha = \{\alpha_{ij}\}$ 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 $y_{il} = 1$. 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 ES_i we have $\alpha_{ik} = \infty$, then $u(i) = k$ is not part of any feasible solution.

Now consider the following algebraic manipulation of the terms for the access cost in Eqn. (1):

$$\sum_{i=1}^{M}\sum_{k=1}^{N}\Omega_i\alpha_{ik}y_{ik} = \sum_{i=1}^{M}\Omega_i\sum_{k=1}^{N}\alpha_{ik}y_{ik} \qquad (2)$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{M}\omega_{ij}\sum_{k=1}^{N}\alpha_{ik}y_{ik} \qquad (3)$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{M}\omega_{ij}\sum_{k=1}^{N}\alpha_{ik}y_{ik}(\sum_{l=1}^{N}y_{jl}) \qquad (4)$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{M}\omega_{ij}(\sum_{k=1}^{N}\sum_{l=1}^{N}\alpha_{ik}y_{ik}y_{jl}) \qquad (5)$$

Equality in Eqn. (2) follows since Ω_i does not depend on k. Using $\Omega_i = \sum_{j=1}^{M}\omega_{ij}$ gives Eqn. (3). The side condition $\sum_{l=1}^{N} y_{jl} = 1$ in Eqn. (3) leads to Eqn. (4). Finally, by readjusting terms we arrive at Eqn. (5).

With Eqn. (5), we can rewrite the optimization problem for the topology of the provider network from Eqn. (1) as follows:

$$\textit{Minimize} \sum_{i=1}^{M}\sum_{j=1}^{M}\omega_{ij}\sum_{k=1}^{N}\sum_{l=1}^{N}(\alpha_{ik} + b_{kl} + \alpha_{jl})y_{ik}y_{jl}$$
$$\textit{subject to} \quad \sum_{j=1}^{N} y_{ij} = 1 \quad for \quad i = 1,..,M \qquad (6)$$

We know that for a given value of i and j, there is a value of k and l such that $y_{ik}y_{jl} = 1$. Let us assume that $u(i)$ and $u(j)$ are such that $y_{iu(i)}y_{ju(j)} = 1$. Then, with the constraints that exactly one provider node is assigned to each endsystem, we must have $y_{ik}y_{jl} = 0$ for $k \neq u(i)$ and $l \neq u(j)$. Hence, the objective function in Eqn. (6) can be rewritten as

$$Z(\underline{u}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\omega_{ij}(\alpha_{iu(i)} + b_{u(i)u(j)} + \alpha_{ju(j)}) \; . \qquad (7)$$

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. (7) is no simpler than the original problem. However, there are special cases when the relationship between matrices B and α, representing, respectively, the transport and access cost, lead to a problem with only linear complexity.

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

(C1) $b_{ij} \leq b_{ik} + b_{kj}$ for all $i, j, k \leq N$.

(C2) $\alpha_{ij} \geq \alpha_{iv_i} + b_{v_ij}$ for all $i \leq M$ and $j, v_i \leq N$.

In our setting, condition (C1) always holds since the elements in matrix B 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 ES_i is minimized by assigning it to the provider node with lowest cost, namely PN_{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 α and B are such that conditions (C1) and (C2) are satisfied, where $\alpha_{iv_i} = \min_j\{\alpha_{ij}\}$, for i, j.

Proof. We can write the objective function as follows: $Z(\underline{u}) = \omega_{11}(\alpha_{1u_1}+b_{u_1u_1}+\alpha_{1u_1})+\ldots+\omega_{1M}(\alpha_{1u_1}+b_{u_1u_M}+\alpha_{Mu_M})+ \ldots + \omega_{M1}(\alpha_{Mu_M} + b_{u_Mu_1} + \alpha_{1u_1}) + \ldots + \omega_{MM}(b_{Mu_M} + \alpha_{u_Mu_M} + b_{Mu_M})$. Using condition (C2) and the fact that $b_{ii} = 0$ for all i, we get that $Z(\underline{u}) \geq \omega_{11}(2\alpha_{1v_1} + 2b_{v_1u_1}) + \omega_{12}(\alpha_{1v_1} + \alpha_{2v_2} + b_{v_1u_1} + b_{v_2u_2} + b_{u_1u_2}) + \ldots + \omega_{MM}(2\alpha_{Mv_M} + 2b_{v_Mu_M})$. After some manipulation using condition (C1), this give us $Z(\underline{u}) \geq \omega_{11}(2\alpha_{1v_1} + 2b_{v_1u_1}) + \ldots+\omega_{12}(\alpha_{1v_1}+\alpha_{2v_2}+b_{v_1v_2})+\ldots+\omega_{MM}(2\alpha_{Mv_M}+2b_{v_Mu_M})$. As $Z(\underline{u_I}) = \omega_{11}(2\alpha_{1v_1}) + \ldots + \omega_{12}(\alpha_{1v_1} + \alpha_{2v_2} + b_{v_1v2}) + \ldots + \omega_{MM}(2\alpha_{Mv_M})$, we see that $Z(\underline{u}) \geq Z(\underline{u_I})$, for any u. This proves our claim. □

IV. Heuristic Solution Approaches

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. We expect that practical provider network topologies must be solved for thousands of endsystems and provider nodes. However, exact solutions of the quadratic assignment problems can been obtained only for problem sizes with at most 30 endsystems and provider nodes [6]. Thus, to solve the provider network design problem for larger networks, we 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 [6]. These methods use an initial solution and iteratively attempt to improve the solution by performing a local search. The solutions found by these heuristics may be a suboptimal local minimum. We select simulated annealing as heuristic algorithm, since it has been shown to perform well for quadratic assignment problems [4], [20]. We also discuss a simple heuristic that performs the assignment based solely on the access costs, and refer to this method as the greedy strategy. Finally, we show how to further reduce the complexity of the topology design problem by grouping geographically close endsystems into clusters, and by assigning all endsystems in a cluster to the same provider node.

A. Simulated Annealing

Simulated annealing draws an analogy between problems from statistical physics and combinatorial problems. Particularly, simulated annealing emulates the crystallization process of cooling metal, the annealing process, in a thermal equilibrium [14]. 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 $e^{\frac{E_j - E_k}{t}}$. 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 perturbation do not decrease the energy level.

In Figure 4, we present the simulated annealing algorithm to the provider network topology problem. 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 (in our case, 100) 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. We refer to the values of the objective functions for the initial, new, best and current solutions as Z_0, Z_{new}, Z_{best} and Z_{cur}, respectively. From the current solution S_{cur} we 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. Assuming the change is such that the endsystem ES_k is chosen and assigned to provider node PN_l, we can change the objective function through the expression $\Delta(Z_{new}, Z_{cur}) = \sum_{j=1,j\neq k}^{M} \omega_{kj}(\alpha_{kl} + b_{lu_j} - \alpha_{ku_k} - b_{u_ku_j}) + \sum_{i=1,i\neq k}^{M} \omega_{ik}(\alpha_{kl} + b_{u_il} - \alpha_{ku_k} - b_{u_iu_k})$. A new solution is accepted if $\Delta(Z_{new}, Z_{cur})$ is negative.

Simulated Annealing Algorithm (S_0, Z_0, t_0) ;
begin
$S_{best} \leftarrow S_{cur} \leftarrow S_0$
$Z_{best} \leftarrow Z_{cur} \leftarrow Z_0$
$t = t_0$
repeat
for $i = 1$ **to** Rep_{max} **do**
$S_{new} \leftarrow SEARCH(S_{cur})$
if $\Delta(Z_{cur}, Z_{new}) < 0$
$S_{new} \leftarrow S_{cur}$
if $Z_{new} < Z_{best}$
$S_{best} \leftarrow S_{new}$
$Z_{best} \leftarrow Z_{new}$
endif
else
if $e^{-\Delta(Z_{new}, Z_{cur})/t} > Rand(0,1)$
$S_{new} \leftarrow S_{cur}$
endif
endif
endfor
$t \leftarrow r_c \cdot t$
until no changes in the objective function
return S_{best}, Z_{best}
end

Fig. 4. Simulated Annealing.

In such a case, we also check 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. A uniformly distributed random number $Rand(0,1)$ is generated to decide whether the new solution given by $Z_{new} = Z_{cur} + \Delta(Z_{new}, Z_{cur})$ is accepted.

The process to decrease the temperature uses a so-called geometric schedule, in which the temperature decreases in a geometric progression ($t = r_c \cdot t$ with $0 < r_c < 1$). We adopted a value of 0.9 for r_c. 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 we reach a temperature where no further improvements to the objective function can be found. At this points, the algorithm terminates and yields the solution S_{best} and the value of the objective function Z_{best}.

B. Greedy Algorithm

Motivated by the special case in Section III, we now present a simple algorithm for assigning endsystems to provider nodes, referred to as greedy assignment or greedy algorithm. Here, we simply assign each endsystem the provider node with lowest access cost, thereby ignoring the transport cost when performing the assignment. Using the notation from Section III, if we choose v_i such that the access cost α_{iv_i} is the smallest access cost, that is, $\alpha_{iv_i} = min_j\{\alpha_{ij}\}$, then the greedy algorithm assigns to endsystem i the provider node v_i.

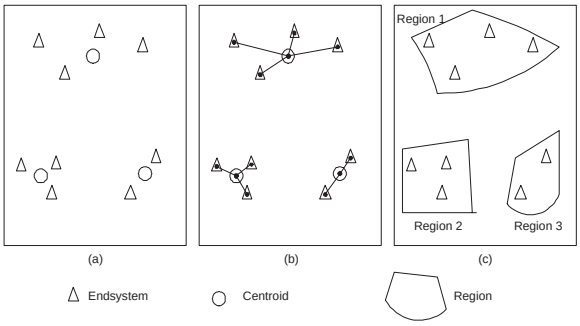

Fig. 5. Illustration of region assignment: (a) position of endsystems and centroids, (b) assignment of endsystems to closest centroid, and (c) resulting assignment of regions.

The greedy strategy can be computed with linear complexity. The algorithm performs well when the access cost per unit of reserved bandwidth is larger than the transport cost. Under the special case considered in Section III, when conditions (C1) and (C2) hold, the greedy assignment yields the optimal solution. Note that, with condition (C2), the lowest cost from endsystem ES_i to any provider node PN_j is attained when ES_i is assigned to PN_{v_i}. The optimality of the assignment follows from the fact that if ES_i and ES_k were assigned to PN_{u_i} and PN_{u_k} respectively, the path cost from ES_i to ES_k would be $\alpha_{iu_i} + b_{u_i u_k} + \alpha_{ku_k}$. By condition (C2), $\alpha_{iu_i} \geq \alpha_{iv_i} + b_{v_i u_i}$ and $\alpha_{ku_k} \geq \alpha_{kv_k} + b_{v_k u_k}$ and by condition (C1), $b_{v_i u_i} + b_{u_i u_k} + b_{v_k u_k} \geq b_{v_i v_k}$. Thus, the path from ES_i to ES_k has minimal cost if ES_i and ES_k are assigned to provider nodes using the greedy strategy.

Our numerical data will show that, even if condition (C2) does not hold for all provider nodes, that is $\alpha_{ik} \leq \alpha_{iv_i} + b_{v_i k}$ for some k, the solution provided by the greedy algorithm can still provide good results.

C. Clustering Algorithm for Endsystems

We can further reduce the complexity of the endsystems assignment by clustering endsystems into groups, and assign complete groups of endsystems to provider nodes. We will refer to a cluster of endsystems as a *region*. All endsystems in the same region will be assigned the same provider node. Thus, instead of solving the optimization problem for all endsystems, we solve the topology problem by assigning regions to provider nodes. Our clustering algorithm exploits the geographical location of endsystems in the sense that endsystems that are geographically close are likely to be assigned to the same region. We assume that each endsystem ES_i has Cartesian coordinates (r_i, s_i), derived, for example, from the longitude and latitude information of an endsystem. The clustering algorithm also accounts for the traffic load of endsystems. Endsystems with a high traffic load are given more consideration when clusters are formed.

We use a *k-means* clustering algorithm [5] to assign endsystems into regions. The algorithm takes a number of M endsystems, with position (r_i, s_i) for ES_i and traffic load Ω_i for

TABLE II

ADDITIONAL NOTATION FOR CLUSTERING.

R	Number of regions
R_i	Region i
x_{ij}	0-1 decision variable that indicates if G_i is assigned to PN_j
c_{ij}	Access cost (per unit of traffic) of assigning G_i to PN_j
a_{ij}	Reserved bandwidth for traffic from G_i to G_j
A_i	Reserved bandwidth for the traffic from G_i to all other regions

each endsystem ES_i $i = 1, \ldots, M$ and the number of desired regions, R. As output, the algorithm generates R cluster centers, called centroids, and an assignment of endsystems to each centroid. Initially, the algorithm randomly chooses initial positions for the R centroids. Then, the algorithm assigns to each endsystem the closest centroid, resulting in an initial cluster assignment. For this assignment, the algorithm computes for each cluster a new position of the centroid. If R_k is the set of endsystems assigned to the kth centroid, then the new position of the centroid $(\overline{r}_k, \overline{s}_k)$ is calculated as follows:

$$\overline{r}_k = \frac{\sum_{i \,:\, ES_i \in R_k} r_i \cdot \Omega_i}{\sum_{i \,:\, ES_i \in R_k} \Omega_i}$$

$$\overline{s}_k = \frac{\sum_{i \,:\, ES_i \in R_k} s_i \cdot \Omega_i}{\sum_{i \,:\, ES_i \in R_k} \Omega_i}$$

The new position of a centroid is weighted by the amount of reserved bandwidth generated by the endsystems assigned to this centroid. After establishing the new centroid position, we re-associate each endsystem with a region, by again assigning to each endsystem the closest centroid. Then, we recalculate the position of each centroid as before. This re-association is repeated until the algorithm converges. At this time, we have established a membership for each endsystem and a centroid for each cluster. In Figure 5 we graphically illustrate the clustering process.

After the assignment of endsystems to regions we can formulate a revised optimization problem for the topology design. The revised problem assigns regions to a provider node, where the access cost of a region is determined from the access costs of the endsystems assigned to the region. We use 0-1 decision variables x_{ij} to indicate if region R_i is assigned to provider node PN_j. The total reserved bandwidth from region R_i to PN_j takes into account the access cost of the endsystem in that region and is be given by

$$c_{ij} = \frac{\sum_{k \,:\, ES_k \in R_i} \alpha_{kj} \Omega_k}{\sum_{k \,:\, ES_k \in R_i} \Omega_k}$$

The bandwidth reserved for the traffic from region R_i to region R_j is referred to as a_{ij}, where $a_{ij} = \sum_{k \,:\, ES_k \in R_i} \sum_{l \,:\, ES_l \in R_j} \omega_{kl}$. The total reserved bandwidth from R_i is given by $A_i = \sum_{j=1}^{R} a_{ij}$. With this notation, which is summarized in Table II, we can express the optimization problem for regions as follows:

$$\textit{Minimize} \quad \sum_{i=1}^{R}\sum_{k=1}^{N} A_i c_{ik} x_{ik} + \sum_{i=1}^{R}\sum_{j=1}^{N}\sum_{k=1}^{R}\sum_{l=1}^{N} x_{ij} x_{kl} a_{ik} b_{jl} + \sum_{j=1}^{R}\sum_{l=1}^{N} A_j c_{jl} x_{jl}$$

$$\textit{subject to} \quad \sum_{j=1}^{N} x_{ij} = 1 \quad \textit{for} \quad i = 1, \ldots, R \qquad (8)$$

This assignment problem has complexity $O(R^N)$. If the access and transport cost have a relationship as discussed in Section III, then the problem can be solved with linear complexity. Otherwise, the heuristic algorithms presented earlier in this section can be used to find approximate solutions.

V. NUMERICAL EVALUATION

In this section we evaluate the approaches for creating topologies for service overlay networks. In the evaluation, we attempt to answer the following questions:

- How closely do the presented heuristic algorithms, i.e., simulated annealing and the greedy algorithm approximate the optimal solution?
- What is the cost sensitivity of the algorithms with respect to the number of provider nodes?
- What is the impact of the clustering algorithm on the cost of the service overlay network?

For our evaluation we generate a network of provider nodes using the Georgia Tech Internetwork Topology Model (GT-ITM) [22]. We produce a random graph choosing the 'Pure Random' model that represents the connectivity between provider nodes. (Note that we do not use GT-ITM to simulate the underlying Internet topology.) An edge between two provider nodes in the random graph indicates that two provider nodes have at least one common ISP.

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, l_{ij} for provider nodes PN_i and PN_j, is drawn from a uniform distribution in the range $[5, 50]$ (for some arbitrary cost metric), and $l_{ij} = \infty$ if GT-ITM does not insert an edge between provider nodes PN_i and PN_j. Unless stated otherwise, the access cost α_{ij} of endsystems ES_i to provider node PN_j is also drawn from a uniform distribution in the range $[5, 50]$. We assume that each endsystem can be connected to one or several number of provider nodes. In our 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 ω_{ij} that are uniformly distributed in the range $[10, 20]$ Mbps. When we show the total cost of a provider network generated in this fashion, we present the average value of 100 cost calculations, where in each calculation we reassign the access costs according to the given uniform distribution. In our experiments, we consider

TABLE III

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

networks with up to $N = 100$ provider nodes and up to $M = 100$ endsystems.

A. Evaluation of the Heuristics Algorithms

First we will evaluate the performance of the simulated annealing heuristic and the greedy algorithms by comparing them to the results of the exact solution of Eqn. (1). For smaller networks, we can solve Eqn. (1), e.g., using branch-and-bound methods or similar techniques. If the network is large, we can determine an exact solution only for the special cases discussed in Section III.

We compare the minimum cost according to Eqn. (1) with the results obtained by simulated annealing 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 III, we compare simulated annealing with the optimal solution. The results for simulated annealing are shown for different values of the repetition factor Rep_{max}. The first column of Table III gives the value of the repetition factor. The second column gives the average deviation of simulated annealing results from the optimal solution, averaged over 100 repetitions of the experiment. The third column depicts how often, among the 100 repetitions, 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 the optimal solution, and even finds the minimum value of the objective function in some cases.

Next, we consider larger networks. Here, a comparison with the optimal solution is possible only when conditions (C1) and (C2) from Section III are satisfied. For this experiment we generate networks with between 10 and 100 provider nodes and endsystems (with $M = N$). We again use 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 ES_i, we randomly select one provider node PN_{v_i} and draw the value α_{iv_i} randomly from the interval $[5, 50]$. Then, for all other provider nodes, we set $\alpha_{ij} = \alpha_{iv_i} + b_{v_i j}$, thereby enforcing that condition (C2) holds.

For these networks, we now compare the simulated annealing algorithm with the optimal solution (Recall, that the greedy algorithm from Subsection IV.B is guaranteed to find the optimal solution when (C2) holds.) We present the comparison in terms of the repetition factor Rep_{max} needed to get within 1% of the optimal cost value. The results are shown in Figure 6. The figure shows that the simulated annealing algorithm is able to get within 1% of the minimum in all cases. The size of the repetition factor needed to get close to the optimum increases linearly with the size of the network. We remark that it has been pointed out elsewhere [4] 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.

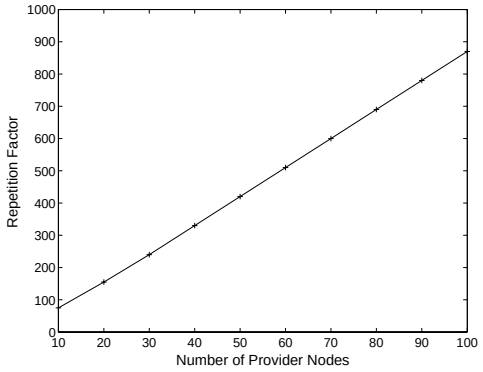

Fig. 6. Repetition factor required by simulated annealing to get within 1% of the optimal solution.

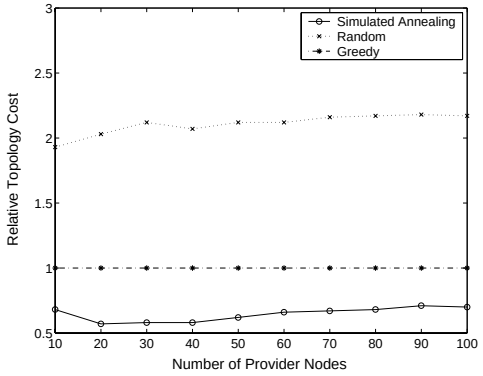

Fig. 7. Comparison of the topology cost of heuristic algorithms (Edge probability set to $P = 0.1$).

B. Comparing the Performance of the Heuristic Algorithms

Next, we consider networks where condition (C2) does not hold, and where obtaining the topology with minimum cost is not possible. Here, we compare the cost of the overlay topology computed by the simulated annealing and the greedy algorithms. As a benchmark, we also include the results of a random assignment of endsystems to provider nodes. The random assignment can be seen as a lower bound for any assignment strategy.

We vary the number of endsystems and provider nodes between 10 and 100, where we set $M = N$. The networks are generated as described at the beginning of this section. We show results for networks where the edge probabilities are given by $P = 0.1$, 0.5 or 0.9. Further, we select $p_\alpha = 0.9$;

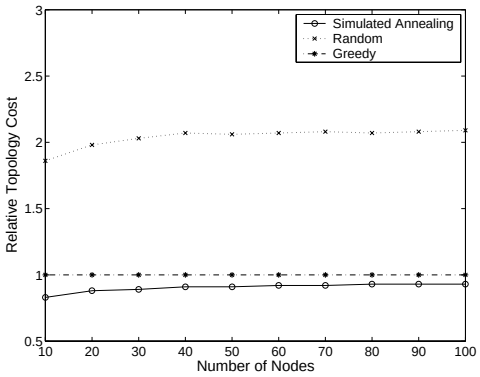

Fig. 8. Comparison of the topology cost of heuristic algorithms (Edge probability set to $P = 0.5$).

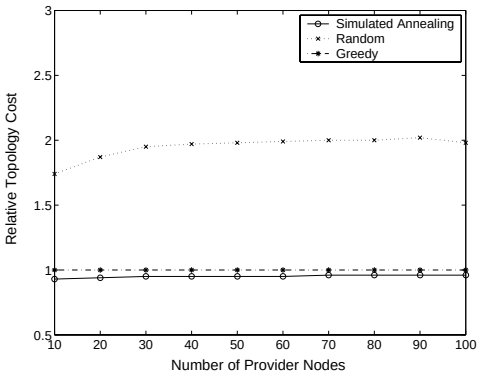

Fig. 9. Comparison of the topology cost of heuristic algorithms (Edge probability set to $P = 0.9$).

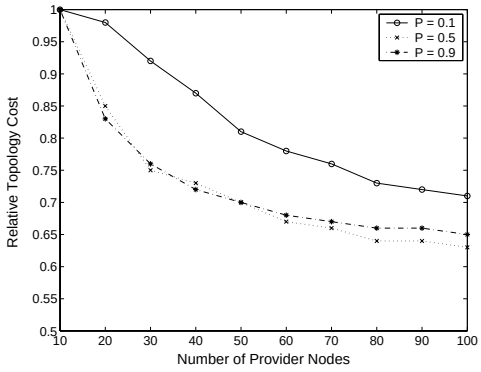

Fig. 10. Relation between topology cost and number of provider nodes for $p_\alpha = 0.9$.

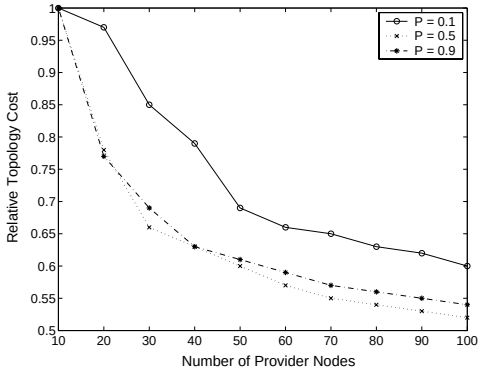

Fig. 11. Relation between topology cost and number of provider nodes for $p_\alpha = 0.5$.

thus, each endsystem has finite access costs to 90% of the provider nodes. The access costs are selected as follows. For each endsystem ES_i, we randomly select one provider node PN_{v_i} and select α_{iv_i} randomly from the interval $[5, 50]$. Then, for all other provider nodes $j \neq v_i$ we randomly select values α_{ij} (for $i = 1, 2, , \ldots, M$), also from the interval $[5, 50]$, where we enforce that $\alpha_{iv_i} \leq \alpha_{ij} \leq \alpha_{iv_i} + b_{v_i j}$ holds. Enforcing the additional condition makes sure that condition (C2) never holds.

The results are shown in Figures 7, 8, and 9. We depict cost values that are normalized by the results obtained with the greedy algorithm. We call this normalized cost the *relative topology cost*. Thus, the greedy algorithm always has a relative topology cost equal to one. A value of two indicates that the cost of the topology is twice of that obtained by the greedy algorithm.

The results in Figures 7–9 show that simulated annealing and the greedy algorithm provide similar results for $P = 0.5$ and $P = 0.5$, but simulated annealing outperforms the greedy algorithm by a factor of 1.5 for $P = 0.1$. Overall, the results of simulated annealing are generally better than the greedy algorithm. We observe that the gain provided by our heuristic algorithms over a random assignment is generally in the order of a factor of two. The results are not sensitive to the size of the network. We note that the good performance of the greedy algorithm is surprising, since, by design, the greedy heuristic is not expected to perform well if condition (C2) does not hold.

C. Impact of the Number of Provider Nodes

Next, we use the simulated annealing algorithm to investigate some properties of the provider network. In particular, 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 network of provider nodes are generated as described in the beginning of this section: the reservation matrix has coefficients that are uniformly distributed in the range [10,20] and the access costs are given by a uniform distribution in the range [5,50].

The results are presented in Figure 10 and Figure 11. 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 10 we select $p_\alpha = 0.9$ and in Figure 11, $p_\alpha = 0.5$. 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 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

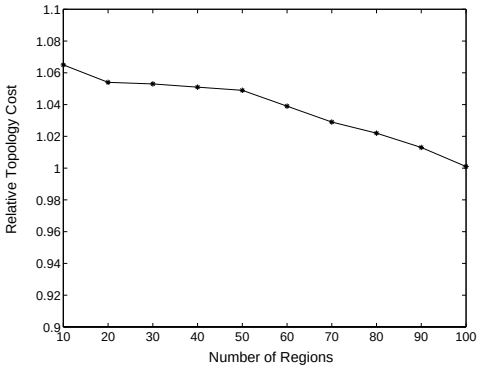

Fig. 12. Relation between topology cost and number of regions.

endsystem assignment may be able to achieve a smaller access cost, thereby decreasing the total cost of the topology.

D. Impact of Clustering

Here we use the simulated annealing algorithm to evaluate the effect of clustering on the cost of the provider network. The objective of the clustering algorithm, presented in Subsection IV.C, is to simplify the construction of the provider network by substituting the individual assignment of endsystems to provider nodes by a collective assignment of endsystems, that are part of the same region.

In this experiment, we employ a network of 100 endsystems and 10 provider nodes. We use the GT-ITM to generate the location of 100 endsystems and a network with 10 provider nodes. The costs of access links are uniformly distributed in the range [5,50] and the reservation matrix has uniformly distributed coefficients in the range [10,20]. We vary the number of regions from 10 to 100 and calculate the cost of the provider network in each region.

We present the results in Figure 12, where we plot the relative topology cost versus the number of regions. The topology cost is presented relative to the cost of the provider network without clustering. We observe that the cost tends to decrease as we increase the number of regions (and decrease the amount of clustering). The results seem to corroborate the idea that clustering increases the cost, but since the cost does not increase significantly, we conclude that clustering may be a viable strategy.

VI. 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, we showed 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 showed that the topology design problem may have only linear complexity. We presented a number of heuristic algorithms that can construct a topology 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 numerical data showed that a very simple greedy algorithm provides good results.

The results presented in this paper depend on a number of assumptions on the underlying network. Particularly, we assume that the cost of purchasing bandwidth guarantees from an ISP is proportional to the amount of reserved bandwidth. A different cost structure may give different results and may require a different solution approach.

Acknowledgment

This work is supported in part by the National Science Foundation through grant ANI-0085955 and by CAPES/Brazil. We would like to thank Jianping Wang for numerous discussions.

References

[1] Akamai, Inc. *http://www.akamai.com*.
[2] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek and R. Morris. Resilient Overlay Network. In *Proc. 18th ACM SOSP 2001*, Banff, October 2001.
[3] R. Braynard, D. Kostic, A. Rodriguez, J. Chase and A. Vahdat. Opus: An Overlay Peer Utility Service. *Proc. IEEE OpenArch'02*, June 2002.
[4] R.E. Burkard and F. Rendl. A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems. *European Journal of Operational Research*, 17:169-14, 1983.
[5] R. Cahn. Wide Area Network Design: Concepts and Tools for Optimization. Morgan Kaufmann, 1998.
[6] E. Cela. *The Quadratic Assignment Problem*. Kluwer Academic Publishers, 1998.
[7] Y. Chu, S.G. Rao and H. Zhang. A Case for End System Multicast. In *Proc. ACM Sigmetrics*, June, 2000.
[8] 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.
[9] Gnutella. *http://gnutella.wego.com*.
[10] 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.
[11] Internap Network Services Corporation. *http://www.internap.com*.
[12] International Telecommunication Union. IP-based networks: Pricing of telecommunication services. Final report, January 2003.
[13] A. Keromytis, V. Misra and D. Rubenstein. Secure Overlay Networks. *Proc. SIGCOMM'02*, pp.61-72, 2002.
[14] S. Kirkpatric, C. Gelatt and M. Vecchi. Optimization by Simulated Annealing. *Science*, 220:291-307, 1983.
[15] 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.
[16] M. Padberg and M.P. Rijal. *Location, Scheduling, Design and Integer Programming*. Kluwer Academic Publishers, 1996.
[17] S. Ratnasamy, P. Francis, M. Handley and R. Karp. A Scalable Content-Addressable Network. In *Proc. ACM SIGCOMM*, 2001.
[18] T. Robertazzi. *Planning Telecommunication Networks*. John Wiley & Sons, 1999.
[19] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Conference on Distributed System Platforms*, Germany, November 2001.
[20] S. Sofianopoulou. Simulated annealing applied to the process allocation problem. *European Journal of Operational Research*, 60:327-334, 1992.
[21] L. Subramanian, I. Stoica, H. Balakrishnan and R.H. Katz. OverQoS: Offering Internet QoS using Overlays. In *Proc. HotNet-I Workshop*, October 2002.
[22] E. W. Zegura. GT-ITM: Georgia Tech Internetwork topology models (software). *http://www.cc.gatech.edu/project*, 1996.
[23] B. Zhao, J. Kubiatowicz and A. Joseph. An Infrastructure for Fault-tolerant Wide-area Location and Routing. Report No. UCB/CSD 01-1141, Computer Science Division, University of California, 2001.