Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing

Thomas Erlebach
Computer Engineering and Networks Laboratory (TIK)
Dept. of Information Technology and Electrical Engineering
ETH Zürich, CH-8092 Zürich, Switzerland
Email: erlebach@tik.ee.ethz.ch

Maurice Rüegg
Remote Sensing Laboratories, Dept. of Geography
University of Zurich – Irchel, Winterthurerstr. 190
CH-8057 Zürich, Switzerland
Email: mrueegg@geo.unizh.ch

***Abstract*— A virtual private network (VPN) provides private network connections over a publicly accessible shared network. Bandwidth provisioning for VPNs leads to challenging optimization problems. In the hose model proposed by Duffield et al., each VPN endpoint specifies bounds on the total amount of traffic that it will send or receive at any time. The network provider must provision the VPN so that there is sufficient bandwidth for any traffic matrix that is consistent with these bounds. While previous work has considered tree routing and single-path routing between the VPN endpoints, we demonstrate that the use of multi-path routing offers significant advantages. On the one hand, we present an optimal polynomial-time algorithm that computes a bandwidth reservation of minimum cost using multi-path routing. This is in contrast to tree routing and single-path routing, where the problem is computationally hard. On the other hand, we present experimental results showing that the reservation cost using multi-path routing can indeed be significantly smaller than with tree or single-path routing.**

I. Introduction

A Virtual Private Network (VPN) is a logical network that is established on top of a physical network in order to provide the behavior of a dedicated network with private lines to the users of the VPN. Two important requirements for a VPN are security and bandwidth guarantees. Security is usually achieved by cryptographic methods and will not be considered further in this paper. Bandwidth guarantees mean that the network provider guarantees to the VPN users that a certain amount of bandwidth will be available to them at any time, unaffected by the traffic sent through the physical network by other users (who are not part of the VPN). Bandwidth guarantees are only possible if a certain amount of bandwidth is explicitly reserved for the VPN on the links of the physical network. From the perspective of the network provider, it is an important optimization problem to satisfy the bandwidth guarantees required by a VPN while minimizing the amount of network resources (link bandwidth) that are reserved for it. In the following, we will be investigating this problem for a specific model of bandwidth reservation, the so-called *hose model*.

The hose model was proposed by Duffield et al. [1] as a flexible and user-friendly model for specifying the bandwidth requirements of a VPN. The idea is to specify for each VPN endpoint v the maximum total bandwidth $b^+(v)$ of traffic that v will send into the network at any time and the maximum total bandwidth $b^-(v)$ of traffic that v will ever receive from the network at any time. It is *not* necessary to specify the exact demands between all pairs of VPN endpoints. Instead, the network capacity reserved for the VPN must be sufficient for *every* possible traffic pattern that is consistent with the b^+ and b^- values. The hose model provides a convenient way for customers to specify their bandwidth requirements, but makes the problem of efficient bandwidth reservation harder than in the traditional model where the customer has to specify pairwise demands for all VPN endpoints.

It is important to distinguish several routing paradigms that can be used to transport VPN traffic through the physical network. With *tree routing*, the VPN endpoints are connected by a Steiner tree and all VPN traffic is sent along the unique path in this tree from sender to receiver. With *single-path routing*, every pair (u, v) of VPN endpoints is assigned a single path $\pi_{u,v}$ for routing the traffic from u to v, but the union of all such paths need not be a tree. With *multi-path routing*, every pair (u, v) is assigned a collection of paths from u to v together with a specification of which fraction of the traffic from u to v should be sent along each of these paths.

The minimum amount of bandwidth that must be reserved for the VPN on each of the links of the physical network is uniquely determined once the routing is specified (see [2] and also Section III). Therefore, the difficult problem is to compute a good routing that minimizes the total amount of necessary bandwidth reservations. Previous work has mainly focussed on the case of tree routing and single-path routing. For the special case of symmetric b values (i.e., $b^+(v) = b^-(v)$ for all VPN endpoints v) it was shown that an optimal tree routing can be computed in polynomial time [3]. For arbitrary b values, however, it is already NP-complete to compute an optimal tree routing [3]. Therefore, approximation algorithms have been proposed for the computation of good tree routings or single-path routings.

A. Our Results

In this paper, we study the bandwidth reservation problem for hose model VPNs under multi-path routing. Our results are as follows. We show that an optimal bandwidth reservation can be computed in polynomial time for multi-path routing. Our algorithm works also for the case where the links of the physical network have bounded capacities, while previous algorithms usually assumed that the available capacity is infinite. The case with finite capacities is known to be NP-hard for tree routing and single-path routing. We also give an example demonstrating that multi-path routing can reduce the bandwidth that needs to be reserved for a VPN. Therefore, multi-path routing has the advantage of reducing the cost of an optimal bandwidth reservation while at the same time admitting efficient algorithms for the computation of optimal reservations. This implies that it may pay off for network providers to implement multi-path routing in order to use their network resources more efficiently for the allocation of hose model VPNs.

Furthermore, we have implemented our algorithm for optimal VPN reservations under multi-path routing. By enforcing integrality constraints for some of the variables in the linear program used by the algorithm, we are able to compute optimal reservations under single-path routing as well (however, with exponential running-time in the worst case). We also implemented known algorithms computing optimal reservations for tree routing. We use these implementations to compare the running-times and reservation costs for the different algorithms and routing models.

B. Previous Work

The hose model was introduced by Duffield et al. in [1]. This triggered a substantial amount of subsequent work on this topic. Kumar et al. [3] studied tree routing in the case where the links of the physical network have infinite capacity. They presented a polynomial algorithm computing the optimal tree routing in the case of symmetric b values and proved that it is NP-hard to compute the optimal tree routing for general b values. For the latter problem, they gave a 10-approximation algorithm. Gupta et al. [4] extend this work and show that the cost of an optimal tree routing is at most twice as large as the cost of an optimal multi-path routing in the case of symmetric b values and infinite link capacities. For the case of symmetric b values and finite link capacities, they show that it is already NP-hard to find any feasible tree routing or single-path routing. For the computation of tree routings in the case of general b values and infinite link capacities, they improve the approximation ratio to 9.002. Swamy and Kumar [5] reduce the ratio further and obtain a 5-approximation algorithm. Gupta et al. [6] consider the problem of computing an optimal single-path routing for general b values and infinite link capacities. They present an algorithm that computes a tree routing of cost at most 5.55 times the cost of an optimal single-path routing. Italiano et al. [7] show that for general b values and infinite link capacities, an optimal tree routing can be computed in polynomial time provided that the sum of all b^+ values equals the sum of all b^- values.

Jüttner et al. [2] study the bandwidth efficiency of the hose model compared to other models such as the pipe model (with explicit reservations for all pairwise demands between VPN endpoints). They define the overprovisioning factor, which represents the additional capacity required for a VPN reservation in the hose model compared to a reservation where the exact traffic matrix is known in advance, and study its dependence on the size and density of the network topology. Italiano et al. [8] consider the problem of making a VPN fault-tolerant for single-edge failures.

C. Paper Outline

The remainder of the paper is structured as follows. Section II provides formal problem definitions and discusses some preliminaries. In Section III, we explain how the amount of bandwidth that needs to be reserved on an edge can be computed efficiently once the routing is determined. This is an important ingredient for our optimal algorithm computing multi-path routings that minimize the required bandwidth reservations, which is presented in Section IV. In Section V, we report our experimental results. Finally, in Section VI, we give our conclusions and discuss future work.

II. Preliminaries

The communication network is modeled as a bidirected graph $G = (V, E)$. A bidirected graph is a graph in which $(u, v) \in E$ implies $(v, u) \in E$. Each edge $e \in E$ might have a per-unit reservation cost c_e and a capacity limit C_e. Often the capacity limit C_e is considered to be infinite. The VPN endpoints are given as a subset $Q \subseteq V$. For each $v \in Q$, two bounds $b^+(v)$ and $b^-(v)$ are given. We assume that all values $b^+(v)$ and $b^-(v)$ are non-negative integers.

A traffic matrix $D = (d_{u,v})_{u,v \in Q}$ is valid if it maps each pair (u, v) of distinct nodes in Q to a non-negative demand d_{uv} such that, for every $v \in Q$, we have

$$\sum_{u \in Q} d_{uv} \le b^-(v) \text{ and } \sum_{u \in Q} d_{vu} \le b^+(v) .$$

We assume $d_{vv} = 0$ for all $v \in Q$.

The problem that we are interested in is reserving bandwidth in the network and routing the traffic between VPN endpoints such that the reserved bandwidth supports every valid traffic matrix. The bandwidth reserved for the VPN on edge $e \in E$ is denoted by x_e. The cost of the reservation is $\sum_{e \in E} c_e x_e$. The goal is to find a valid reservation of minimum cost. The validity of a reservation depends on the routing. We distinguish the following routing models:

- Tree routing: The edges on which bandwidth is reserved ($x_e > 0$) form a tree. The traffic from $u \in Q$ to $v \in Q$ is routed along the unique path from u to v in the tree.
- Single-path routing: For every pair (u, v) of distinct nodes in Q, the traffic from u to v is routed along a single path $\pi_{u,v}$. The path does not depend on the current traffic matrix.
- Multi-path routing: For every pair (u, v) of distinct nodes in Q, the traffic from u to v can be split arbitrarily among several paths. The routing does not depend on the traffic matrix.

In each of the three routing models, we define $f^e_{u,v}$ as the fraction of traffic from $u \in Q$ to $v \in Q$ that is routed through edge $e \in E$. For tree routing and single-path routing, the $f^e_{u,v}$

are either zero or one, but for multi-path routing, they can be arbitrary values in the interval $[0,1]$. Now a bandwidth reservation x is valid if and only if it holds for every valid traffic matrix D and for every edge $e \in E$ that

$$\sum_{u,v \in Q} d_{u,v} \cdot f^e_{u,v} \leq x_e \,.$$

The term $\sum_{u,v \in Q} d_{u,v} \cdot f^e_{u,v}$ expresses the amount of traffic that is routed through e when the current traffic matrix is D.

III. Checking the Validity of a Reservation

In this section, we assume that we are given the b^+ and b^- values as well as a routing for the traffic between VPN endpoints. The routing could be a tree routing, a single-path routing, or a multi-path routing. It is simply specified by the values $f^e_{u,v}$ for all $u, v \in Q$ and $e \in E$. We want to compute the maximum amount of VPN traffic that is sent through an edge e for any valid traffic matrix. Call this value $m(e)$. Knowing the values $m(e)$ for all $e \in E$ is useful for several reasons. First, we can use them to check the validity of a given bandwidth reservation vector $(x_e)_{e \in E}$, because the vector is valid if and only if $x_e \geq m(e)$ holds for all $e \in E$. Second, if we want to compute the optimal x_e values for a given routing, we only have to set $x_e := m(e)$. Finally, the computation of the values $m(e)$ will be needed as a subroutine in our algorithm computing an optimal multi-path routing in Section IV.

As already observed in [2], the value $m(e)$ can be computed for a given edge $e \in E$ by solving a linear program or a min-cost network flow problem. We briefly outline the method in order to make the paper self-contained and because we need to establish that the largest traffic through e can always be achieved with an *integral* valid traffic matrix.

The value $m(e)$ is the maximum VPN traffic through edge e, taken over all valid traffic matrices D. Therefore, the computation of $m(e)$ can be formulated as the following linear program:

$$\begin{aligned} \max \quad & \sum_{u,v \in Q} f^e_{u,v} \cdot d_{u,v} \\ \text{s.t.} \quad & \sum_{v \in Q} d_{u,v} \leq b^+(u), \qquad u \in Q \\ & \sum_{v \in Q} d_{v,u} \leq b^-(u), \qquad u \in Q \\ & d_{uu} = 0, \qquad u \in Q \\ & d_{uv} \geq 0, \qquad u, v \in Q, u \neq v \end{aligned}$$

The constraints of this linear program characterize all valid traffic matrices. The objective function maximizes the traffic through e. The value of the optimal solution is therefore just the value $m(e)$ that we are looking for.

This linear program can be solved as a min-cost flow problem in the graph $H = (V_H, E_H)$ constructed as follows: V_H contains a node s, a node t, and two copies u_1 and u_2 of every node $u \in Q$. E_H contains directed edges from s to every node u_1 (with capacity $b^+(u)$ and cost 0), from u_1 to v_2 for every $u, v \in Q$ (with capacity ∞ and cost $-f^e_{u,v}$), and from every node u_2 to t (with capacity $b^-(u)$ and cost 0). It is easy to see that the set of all valid traffic matrices corresponds to the set of all possible flows from s to t in H: $d_{u,v}$ simply corresponds to the flow on edge (u_1, v_2). The cost of the flow corresponding to traffic matrix D is just the negative of the traffic through e created by D. Consequently, a min-cost flow computation in H yields an optimal solution to the above linear program and therefore also the desired value $m(e)$. Furthermore, the capacities of the edges in H are all integral, therefore there exists also an integral min-cost flow (i.e., with integral flow values on all edges) [9]. This means that there is an integral traffic matrix that creates the maximum possible amount of traffic on edge e. Therefore, we obtain that a bandwidth reservation is valid for all valid traffic matrices if and only if it is valid for all integral traffic matrices.

Summing up this section, we know that for a given routing, we can compute efficiently what the maximum bandwidth requirement of any valid traffic matrix on an edge e is, and an integral traffic matrix that maximizes this requirement can be found with a min-cost flow algorithm. (In the special case of tree routing, it is of course easy to compute the bandwidth requirements directly, without resorting to linear programming or min-cost flow computations [3].)

IV. Optimal VPN Reservation with Multi-Path Routing

Now we are ready to present our polynomial algorithm for computing an optimal bandwidth reservation for multi-path routing. The main idea is to formulate the problem as a linear program. Unfortunately, the linear program will contain an exponential number of inequalities. Nevertheless, we can show that the separation problem (deciding whether any of the inequalities is violated) can be solved efficiently. This implies, using Khachiyan's ellipsoid algorithm (see, e.g., [10]), that an optimal solution to the linear program can be computed in polynomial time.

The variables of the linear program are x_e for $e \in E$ and $f^e_{u,v}$ for $u, v \in Q$, $u \neq v$ and $e \in E$. Let $\mathcal{D}$ be the set of all integral traffic matrices D that are valid for the given b^+ and b^- values. Note that the cardinality of $\mathcal{D}$ is finite, but can be extremely large. Let $\Gamma^+(v)$ and $\Gamma^-(v)$ denote the set of outgoing and incoming edges of vertex v in G, respectively. The linear program can then be stated as follows:

$$\min \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \quad x_e \leq C_e, \qquad e \in E \tag{2}$$

$$\sum_{u,v \in Q} d_{u,v} f^e_{u,v} \leq x_e, \quad D \in \mathcal{D}, e \in E \tag{3}$$

$$\sum_{e \in \Gamma^+(u)} f^e_{u,v} = 1, \qquad u, v \in Q, u \neq v \tag{4}$$

$$\sum_{e \in \Gamma^-(u)} f^e_{u,v} = 0, \qquad u, v \in Q, u \neq v \tag{5}$$

$$\sum_{e \in \Gamma^+(v)} f^e_{u,v} = 0, \qquad u,v \in Q, u \neq v \tag{6}$$

$$\sum_{e \in \Gamma^-(v)} f^e_{u,v} = 1, \qquad u,v \in Q, u \neq v \tag{7}$$

$$\sum_{e \in \Gamma^+(w)} f^e_{u,v} - \sum_{e \in \Gamma^-(w)} f^e_{u,v} = 0, \quad u,v \in Q, u \neq v, w \in V \setminus \{u,v\} \tag{8}$$

$$0 \leq f^e_{u,v} \leq 1, \quad u,v \in Q, u \neq v, e \in E \tag{9}$$

$$x_e \geq 0, \qquad e \in E \tag{10}$$

The objective function (1) minimizes the cost of the reservation. Inequalities (2) ensure that the bandwidth reserved on edge e does not exceed its capacity C_e. The exponentially many inequalities (3), one for each valid integral traffic matrix, ensure that the bandwidth reservation given by the variables x_e is feasible. Equations (4) to (8) are flow conservation constraints that ensure that the variables $f^e_{u,v}$ represent a flow of value 1 from u to v in G. Inequalities (9) and (10) give the bounds for the variables.

As the linear program has too many constraints (3), we cannot afford to construct the whole linear program explicitly. Instead, we show how to check efficiently whether all constraints are satisfied and, if not, how to identify a violated constraint. Constraints (2) and (4)–(10) can be checked directly. To see whether any constraint (3) is violated, we compute the value $m(e)$ for every $e \in E$ with respect to the current values $f^e_{u,v}$ as explained in Section III. If $m(e) \leq x_e$ for all $e \in E$, we know that the solution is feasible. If $m(e) > x_e$ for some $e \in E$, we know that constraint (3) is violated for the traffic matrix creating traffic $m(e)$ on edge e. This traffic matrix is obtained as a result of the min-cost flow computation (see Section III). Therefore, we can solve the separation problem efficiently, and this implies that there is a polynomial algorithm to compute the optimal solution to the above linear program (see [11] for further details about such applications of the ellipsoid method). The variables x_e in the optimal solution represent the bandwidth that has to be reserved on edges e, and the variables $f^e_{u,v}$ represent the fraction of traffic from u to v that is routed through edge e. In order to determine the routing for each pair of VPN endpoints in terms of paths instead of flows, one can then apply a standard flow decomposition method (cf. [9]).

Note that the linear program has no feasible solution in case the capacities C_e are not large enough to allow a valid VPN reservation. In that case, the ellipsoid algorithm will determine that the problem is infeasible. Therefore, we obtain the following theorem.

Theorem 1: For hose-model VPNs with multi-path routing, there is a polynomial algorithm for the problem of checking whether a valid reservation exists and, if so, computing a reservation of minimum cost.

Furthermore, it is easy to observe that if the values of the variables $f^e_{u,v}$ of the linear program are restricted to be either 0 or 1, then the optimal solution to the resulting mixed integer linear program actually gives a single-path routing that minimizes the reservation cost among all possible single-path routings. Therefore, we can use this formulation to compute optimal single-path routings on small instances, by employing a solver for mixed integer linear programs. Of course, we cannot hope to solve large instances with this approach, since integer linear programming is an NP-hard problem and the solver may need excessive running-time to arrive at the optimal solution.

A. Implementation

Although Theorem 1 asserts the existence of a polynomial algorithm for optimal VPN reservations with multi-path routing, the algorithm is not very practical due to the use of the ellipsoid method, which is mainly a theoretical tool for proving polynomial-time solvability. However, a practical implementation with reasonable running-times can be obtained by using a cutting-plane approach. Initially, we construct the linear program without the constraints (3) and use a standard LP solver to compute an optimal solution. Then, as explained above, we check for every edge e of the network whether there is a valid traffic matrix D such that the constraint (3) is violated for e and D. If no constraint is violated, an optimal solution has been obtained. Otherwise, for every edge e for which a violated constraint was found, we add that constraint to the linear program and compute a new optimal solution using the LP solver. This process repeats until either the linear program becomes infeasible (showing that no valid VPN reservation exists) or an optimal solution has been obtained. This implementation might lead to exponential running-times in the worst case, but such worst-case behavior is unlikely. In our experiments we were able to obtain optimal VPN reservations under multi-path routing in reasonable time.

V. Experimental Results

We have implemented the algorithm for computing an optimal VPN reservation with multi-path routing presented in Section IV. Our implementation is in C++ and we use CPLEX 8.1 [12] to solve (integer) linear programs. We also use some data structures and algorithms provided by the C++ library LEDA 4.2.1 [13], in particular an algorithm for min-cost flow. As CPLEX can solve mixed integer linear programs as well, we can simply require integrality of the variables $f^e_{u,v}$ and thus obtain also an implementation of an algorithm (with potentially exponential running-time, however) for the computation of optimal reservations with single-path routing.

In addition, we implemented the polynomial algorithm for optimal reservations with tree routing in the case of infinite capacities and symmetric b values from [3]. We also adapted an implementation from [14] of the integer linear programming approach presented in [3] for the computation of optimal reservations with tree routing in the case of infinite capacities and arbitrary b values. However, we should point out that this implementation is not optimized and explicitly constructs an integer linear program whose size is exponential in the size of the given graph.

With these implementations, we have a suite of programs that allow us to compute optimal hose-model VPN reservations for tree routing, single-path routing, and multi-path routing. Note, however, that the only cases for which algorithms with polynomial running-time are known are tree routing for infinite capacities and symmetric b values [3], and multi-path routing for arbitrary capacities and arbitrary b values (Theorem 1).

The goals of our experiments are as follows. First, we want to confirm our expectation that multi-path routing can reduce the cost of a VPN reservation as compared to tree routing and single-path routing. Second, we would like to demonstrate that our algorithm for the computation of optimal reservations with multi-path routing achieves reasonable running-times in practice. Finally, we hope to gain some insights into a problem left open by [4], [7]: Is it true that for the case of infinite capacities and symmetric b values, there is always a tree routing whose reservation cost is not worse than that of the best single-path routing?

We discuss each of the three questions in turn.

A. *Cost Savings of Multi-Path Routing*

For a given instance of the VPN reservation problem (specified by a graph and the b values of VPN endpoints), we denote by W_T, W_S and W_M the cost of an optimal reservation with tree routing, single-path routing and multi-path routing, respectively. First, we present a concrete instance that shows that the optimal cost of the three different routing models can indeed differ, i.e., the instance has $W_T > W_S > W_M$. The graph and the three optimal reservations are depicted in Fig. 1. Each node v is labeled by the pair $(b^+(v), b^-(v))$. All four nodes are VPN endpoints. Each edge e has per-unit reservation cost $c_e = 1$ and infinite capacity. Each pair of directed edges (u, v) and (v, u) is drawn as a single undirected edge for the sake of simplicity. The reservation cost shown next to an edge between u and v represents the sum of the reservation costs on (u, v) and (v, u).

The optimal reservations were computed with our implementations. In the case of tree routing, the direct edges between q_3 and q_4 are not used at all. The reservation cost on the edges between q_1 and q_2 is 22, because the worst-case traffic pattern given by $d_{12} = 1$, $d_{43} = 10$, $d_{21} = 10$, $d_{34} = 1$ requires 11 units of bandwidth on (q_1, q_2) and on (q_2, q_1). The total cost of the tree routing is 54.

For the optimal single-path routing, all edges are used. To specify the routing, we give an explicit list of the pairs of VPN endpoints whose path is not just a single edge: q_1 sends to q_3 via q_4; q_2 sends to q_1 via q_3 and q_4; q_2 sends to q_4 via q_3; q_3 sends to q_1 via q_4; and q_4 sends to q_2 via q_3. The total cost of the reservation is 46.

In the optimal multi-path routing, every pair (q_i, q_j) of distinct VPN endpoints sends 50% of the traffic from q_i to q_j along one of the two possible paths and the remaining 50% along the other path. In other words, every VPN endpoint splits its traffic for each destination equally among the two possible paths to that destination. The cost of the resulting reservation is 44.

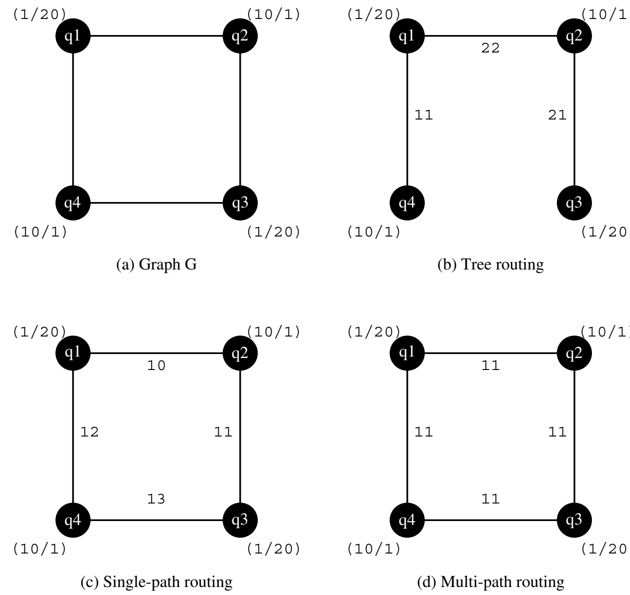

Fig. 1. For the network G shown in (a), tree routing in (b) has cost 54, single-path routing in (c) has cost 46, and multi-path routing in (d) has cost 44.

Therefore, we have shown that even a very small instance with only four nodes can have the property that multi-path routing reduces the reservation cost as compared to single-path and tree routing.

Next, we ran a series of experiments with small instances. We created connected random graphs with n nodes and m edges, where n ranges from 3 to 5 and m ranges from $n-1$ to $n(n-1)/2$. (The case $m = n-1$ means that the graph is in fact a tree; then every VPN reservation must use tree routing, but we include this case anyway for the sake of completeness.) The random graphs were created by calling the LEDA routines `random_simple_undirected_graph` (with parameters n and m) and `Make_Connected`. The reason for choosing small instances was that the running-times of the algorithms for computing optimal tree routings and optimal single-path routings quickly become excessive for larger instances, making it impossible to obtain results for a reasonably sized sample. The per-unit reservation cost c_e was set to 1 for all edges. The number of VPN endpoints was chosen between 3 and n, and the b^+ and b^- values were selected randomly in $\{1, \ldots, 50\}$. Altogether, we obtained 6,200 such instances. Interestingly, for all except two of these instances we had $W_T = W_S$. Therefore, single-path routing improved over tree routing only very rarely. On the other hand, we observed $W_M < W_T$ for roughly 20% of the instances with 3 nodes, 25% of the instances with 4 nodes, and 17% of the instances with 5 nodes. In the cases where the multi-path routing had reduced reservation cost compared to tree routing, the cost reduction was 8.6% on the average.

While such experiments on small instances do not allow direct conclusions about what can be expected for larger instances, we view the results as an indication that multi-path

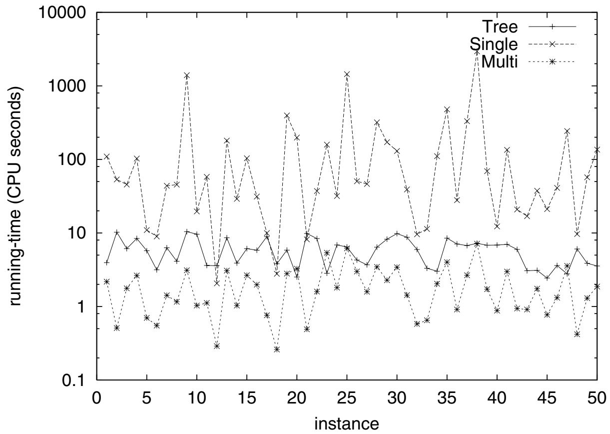

Fig. 2. Running-time of optimal algorithms on 50 random networks with 10 nodes, 20 edges, and 5 VPN endpoints.

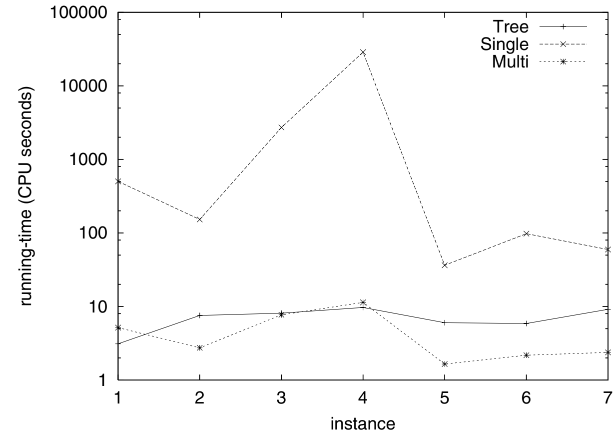

Fig. 3. Running-time of optimal algorithms on 7 random networks with 10 nodes, 20 edges, and 6 VPN endpoints.

routing does indeed have the potential of offering significant bandwidth savings for VPN reservations in the hose model.

B. Running-Time of Optimal Algorithms

We are interested in comparing the running-times required for computing a bandwidth reservation of minimum cost for tree routing, single-path routing and multi-path routing. The algorithm that we employ for computing optimal tree routings transforms the problem into a connected facility location problem as in [3] (where the transformation is used to obtain an approximation algorithm). The latter problem can be formulated as an integer linear program with exponentially many constraints. The implementation that we use constructs this whole integer linear program and then solves it using CPLEX. Better running-times could probably be achieved using a cutting-plane approach, but even then it seems unlikely that the algorithm would be practical on large instances.

All running-times that we report in the following are given in CPU seconds measured on a SunBlade 1000 workstation with two 900 MHz UltraSPARC III CPUs (our code uses only one of the two CPUs) and 2 GB of main memory. As the running-times of the optimal algorithms for tree routing and single-path routing grow very quickly with the size of the network, we present detailed results on the comparison of all three optimal algorithms only for very small instances. Fig. 2 gives the running-times for 50 randomly generated instances with 10 nodes and 20 edges. The per-unit edge reservation costs were set equal to 1 and the edges had infinite capacity. The number of terminal endpoints was set to 5 and the b values were chosen randomly between 1 and 5. As one can see in Fig. 2, the running-time of each of the algorithms varies significantly for different instances of the same size. This indicates that the running-time depends on the topology of the network and on the locations and b values of the VPN endpoints. It can be seen clearly that the algorithm computing optimal multi-path routings is significantly faster than the other two algorithms; note that the y-axis of the plot is in logarithmic scale. For these 50 instances, the average running-time of the three algorithms was 5.87 seconds for tree-routing, 200.15 seconds for single-path routing, and 1.99 seconds for multi-path routing. (As an aside, we note that for 13 of these 50 instances, the optimal multi-path routing had smaller reservation cost than the tree and single-path routings, and for these 13 instances, the maximum cost saving was 28.1% and the average saving was 6.5%.)

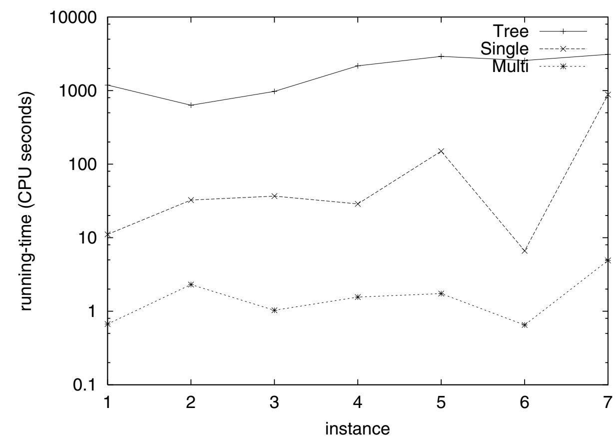

Fig. 4. Running-time of optimal algorithms on 7 random networks with 15 nodes, 30 edges, and 5 VPN endpoints.

The running-times of the three optimal algorithms for seven randomly generated instances with 10 nodes, 20 edges, and 6 VPN endpoints (all other parameters are the same as above) are shown in Fig. 3, and those for seven randomly generated instances with 15 nodes, 30 edges, and 5 VPN endpoints (all other parameters unchanged) are shown in Fig. 4. We observe that the algorithm for optimal single-path routings becomes impractical even for small networks if just one additional VPN endpoint is included (Fig. 3), and the algorithm computing optimal tree routings has excessive running-time already for very moderate values of n (Fig. 4). In both cases, optimal multi-path routings can be computed very quickly by our algorithm.

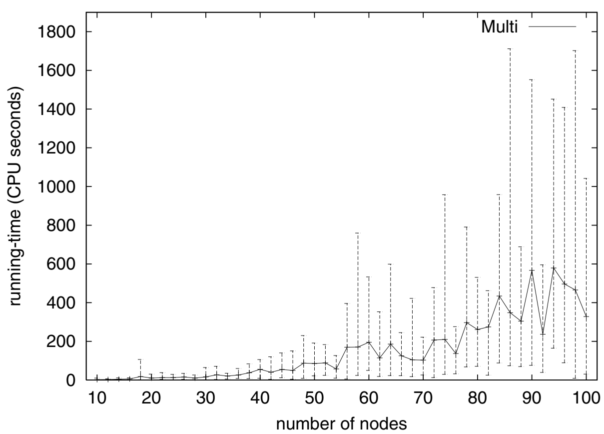

Fig. 5. Running-time of optimal multi-path algorithm for networks with 10 to 100 nodes (5 VPN endpoints).

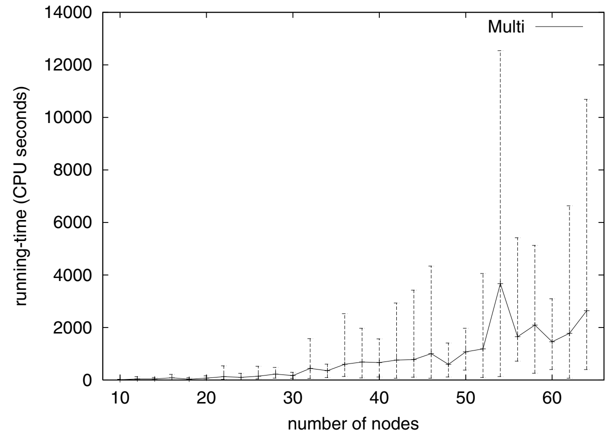

Fig. 6. Running-time of optimal multi-path algorithm for networks with 10 to 64 nodes (7 VPN endpoints).

For larger instance sizes, we ran only our algorithm computing optimal multi-path routings. In Figs. 5 and 6, we present its running-time for instances with n nodes and m edges, where n ranges from 10 to 100 in steps of 2 (Fig. 5) and from 10 to 64 in steps of 2 (Fig. 6). The number of edges is set to $m = 2n$. Edges have per-unit reservation cost 1 and infinite capacity. The number of VPN endpoints is 5 in Fig. 5 and 7 in Fig. 6. The b values are chosen between 1 and 5. The plots show the average running-time on 10 random instances (generated as explained above) for each value of n. The error bars represent the smallest and largest running-time observed in the 10 runs for each value of n. The extent of the error bars indicates again that the running-time can vary significantly for different instances of the same size. However, even the longest running-times we observed were still acceptable: for the instances with 5 VPN endpoints, the longest computations took between a few minutes and half an hour; for the instances with 7 VPN endpoints, all instances except two could be solved in less than two hours, and the two remaining instances took less than four hours. The general growth behavior of the running-time appears to be bounded by a polynomial with moderate degree. Therefore, we think that the algorithm can be applied to reasonably sized instances arising in practice. Since VPNs are usually established for extended time periods, it is worth to invest a moderate amount of computation time for obtaining an optimal VPN reservation at the time the VPN is established.

C. Tree Optimality Hypothesis for Symmetric Bandwidth Requirements

In the context of VPN reservations in the hose model, it is a curious fact that for the case of infinite capacities and symmetric b values, no problem instance is known where the cost of the optimal tree reservation is higher than the cost of an optimal single-path or multi-path reservation. Gupta et al. [4] proved that the cost of the optimal tree routing is at most twice the cost of an optimal multi-path routing in this case (even if the routing is allowed to depend dynamically on the current traffic matrix), but did not give an example showing that the optimal costs of the two models can indeed differ. Therefore, it is not clear whether the factor 2 is necessary or the best possible reservation can in fact be obtained already with tree routing. This open problem was also pointed out by Italiano et al. [7].

In order to shed some light on this issue, we computed optimal tree routings and optimal multi-path routings for more than 100,000 random graphs with 3 to 20 nodes and b values between 1 and 150 (setting $b^+(v) = b^-(v)$ for all VPN endpoints v). We did not find a single instance of a network where tree routing led to a more expensive reservation than multi-path routing. This provides significant empirical support for the hypothesis that for symmetric b values, the optimal tree routing is never worse than the optimal single-path or multi-path routing. Therefore, it seems a promising subject for further theoretical investigations to prove that this hypothesis is true. In fact, a proof that the hypothesis is true for the special case that the given network is a ring network was recently announced by Hurkens, Keijsper and Stougie [15].

VI. Conclusion

In this paper we have studied the problem of bandwidth reservation for virtual private networks in the setting where the requirements are specified according to the hose model. Our main result is a polynomial-time algorithm that computes a multi-path routing minimizing the total cost of the bandwidth reservation. The algorithm applies to symmetric and asymmetric bandwidth requirements and can handle the natural constraint that the capacities of the network links are finite. This should be contrasted with previous work showing that computing optimal reservations under tree routing or single-path routing is NP-hard in general.

Furthermore, we have demonstrated with a concrete example and a set of experiments that bandwidth reservations for multi-path routing are not only easier to compute, but can also achieve reduced reservation costs. Network providers should be aware of these potential benefits and consider employing multi-path routing in order to realize hose-model

VPNs efficiently. Note that multi-path routing could be implemented using MPLS (multi-protocol label switching) [16], for example.

A potential further use of our algorithm for computing optimal multi-path routings could be to combine it with randomized rounding [17] in order to derive a fast approximation algorithm for computing good single-path routings (for networks that do not support multi-path routing). The idea would be to first compute an optimal multi-path routing. If the traffic from u to v is routed along k paths $\pi_1, \ldots, \pi_k$ in this routing, we select one of these k paths randomly, where the probability of each path is equal to the fraction of traffic from u to v that is routed through it. One could hope that the resulting single-path routing achieves a reservation cost that is close to optimal with high probability, at least in cases where the optimal reservation cost is large. Analyzing this approach from a theoretical point of view as well as evaluating it by experiments are interesting directions for further research.

From a theoretical point of view, interesting open problems are to prove or disprove the tree optimality hypothesis for symmetric b values (cf. Section V-C) and to settle the complexity of single-path routing in the case of infinite capacities and arbitrary b values (we suppose that the problem is NP-hard, but to our knowledge no proof for this has appeared in the literature so far). In addition, further considerations about fault-tolerant VPN reservations, building on the work in [8], could lead to interesting results.

References

[1] N. Duffield, P. Goyal, and A. Greenberg, "A flexible model for resource management in virtual private networks," in *ACM SIGCOMM*, 1999.

[2] A. Jüttner, I. Szabó, and Á. Szentesi, "On bandwidth efficiency of the hose resource management model in virtual private networks," in *Proceedings of IEEE INFOCOM*, 2003.

[3] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 565–578, August 2002.

[4] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: A network design problem for multicommodity flow," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, 2001, pp. 389–398.

[5] C. Swamy and A. Kumar, "Primal-dual algorithms for connected facility location problems," in *Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'02)*, ser. LNCS 2462, 2002, pp. 256–270.

[6] A. Gupta, A. Kumar, and T. Roughgarden, "Simpler and better approximation algorithms for network design," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, 2003, pp. 365–372.

[7] G. Italiano, S. Leonardi, and G. Oriolo, "Design of networks in the hose model," in *Proceedings of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE)*. Carleton Scientific, 2002, pp. 65–76.

[8] G. Italiano, R. Rastogi, and B. Yener, "Restoration algorithms for virtual private networks in the hose model," in *Proceedings of IEEE INFOCOM*, 2002.

[9] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice Hall, 1993.

[10] A. Schrijver, *Theory of linear and integer programming*. Chichester: John Wiley & Sons, 1986.

[11] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer-Verlag, 1988.

[12] *CPLEX*, ILOG CPLEX Division, http://www.cplex.com/.

[13] K. Mehlhorn and S. Näher, *LEDA — A Platform for Combinatorial and Geometrical Computing*. Cambridge University Press, 1999.

[14] C. Chhabra, T. Erlebach, B. Stiller, and D. Vukadinović, "Price-Based Call Admission Control in a Single DiffServ Domain," Computer Engineering and Networks Laboratory (TIK), ETH Zürich, TIK-Report 135, May 2002. [Online]. Available: ftp://ftp.tik.ee.ethz.ch/pub/publications/TIK-Report135.pdf.

[15] C. Hurkens, J. Keijsper, and L. Stougie, 2003, personal communication.

[16] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Internet RFC 2702: Requirements for traffic engineering over MPLS," September 1999.

[17] P. Raghavan and C. D. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.