On the Cost of Virtual Private Networks

Reuven Cohen, *Senior Member, IEEE,* and Gideon Kaempfer

***Abstract*—A virtual private network (VPN) is a private data network that uses a nonprivate data network to carry traffic between remote sites. An "Intranet VPN" establishes network layer connectivity between remote Intranet sites by creating an IP overlay network over the nonprivate network, using various tunneling mechanisms. There are two approaches for establishing such tunnels: a "CPE-based approach" and a "network-based approach." In the first approach, tunnels are established only between the CPE devices, whereas in the second approach tunnels are also established between the routers of the core nonprivate network. In this paper we address the problem of determining a CPE-based and a network-based layout of VPN tunnels while taking into account two factors: the cost of the links over which the VPN tunnels are established and the cost of the core routers that serve as end points for the VPN. We define related graph algorithm problems, analyze their complexity, and present heuristics for solving these problems efficiently.**

I. Introduction

THE INTERNET has become a popular low-cost backbone infrastructure. Its universal reach has led many companies to consider constructing a secure virtual private network (VPN) over the public Internet. Essentially, a VPN is a private data network that uses a nonprivate data network to carry its traffic. VPNs offer an alternative to the traditional leased line or frame relay networks by utilizing an established public network. The most ubiquitous, least expensive nonprivate data network is the Internet, which is the perfect foundation for a VPN. The challenge in designing a VPN is often to provide the security of the traditional private self-administered corporate network over the nonprivate backbone.

There are several possible VPN applications [5]. The most popular are the "access VPN" and the "Intranet/Extranet VPN." An access VPN allows remote corporate users to have on demand connectivity into their corporate Intranets through *ad hoc* tunnels. In the past, remote access has been established through the telephone network, with users setting up PPP connections over expensive telephone circuits. However, a new protocol called L2TP [2] allows to set up PPP connections over the Internet, or any other packet switched public network, using compulsory or voluntary L2TP tunnels.

The other common VPN type is Intranet/Extranet VPN. An Intranet/Extranet VPN links the network of a headquarter office to the networks of remote branches and potentially to the networks of suppliers, partners, customers, and other communities of interest. Two of the main requirements of an Intranet/Extranet VPN are [5]:

1) Support for opaque packet transport: The traffic carried within a VPN may have no relation to the traffic on the core network. For example, the VPN network may use private IP addressing, unrelated to that of the core network on which the traffic is transported.
2) Support for data security: In general, VPN users require the same level of data security they have within their private networks. Most recent VPN implementations are converging on the use of IPSec [7] for this purpose.

This paper concentrates upon the first issue: establishing opaque network layer connectivity between the various sites of an Intranet/Extranet VPN. The main mechanism for establishing such connectivity is the creation of tunnels. There is a range of choices for such tunnels, including IP tunnels (using IP-over-IP [11], IPsec, or GRE) [6], ATM VCs, and MPLS.

Generally, there are two approaches for establishing such tunnels: the "CPE-based approach" and the "network-based approach." In a CPE-based approach, tunnels are established only between the CPE devices (mainly border routers). In a network-based approach, tunnels are also established between the routers of the core network. Though the CPE-based approaches are more simple, mainly from the perspective of the ISP operating the core network, for scalability and economic reasons, network-based solutions for VPNs are preferred.

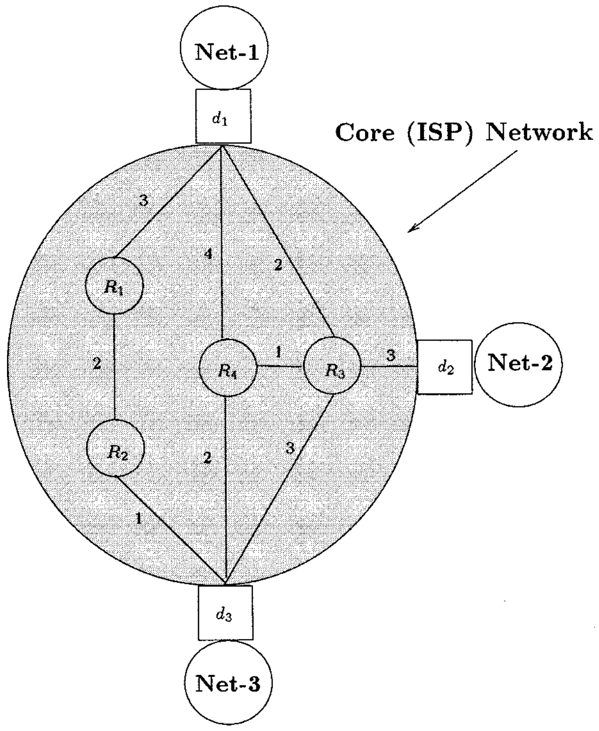
To better understand the trade-off between the two approaches, consider Fig. 1, where a network example and the associated graph are depicted. In this example, there are three corporate networks, referred to as Net-1, Net-2, and Net-3, that need to be connected by a VPN across a core network. The three networks are connected to the core network by means of three border routers: d_1, d_2, and d_3, respectively. In the associated graph, each of these networks is represented by its border router. The number associated with every network edge represents the cost of setting up a VPN tunnel across that link. Three possible VPN configurations for the example in Fig. 1 are shown in Fig. 2. Fig. 2(a) demonstrates the CPE-based case, where the core routers do not support VPN. In such a case, the tunnels are established between the border routers of the edge networks: d_1, d_2, and d_3. The cost of the VPN in this example is 11: 3 for using the link between d_3 and R_3 plus 2 for using the link between d_1 and R_3 plus 2*3 for using the link between d_1 and R_3 twice. In Fig. 2(b), the networks routers support VPN. Hence, the routers can serve as the end points of tunnels, and the cost of the VPN is reduced to 10 $(2+4+1+3)$. In Fig. 2(c), the networks routers also support VPN. However, here R_3 rather than R_4 is employed as an end point of the VPN tunnels, and the cost of the VPN is reduced to 8.

From this example, it is clear that when the core routers are capable of functioning as end points of VPN tunnels, the routing cost for the VPN can be reduced. However, this does not come

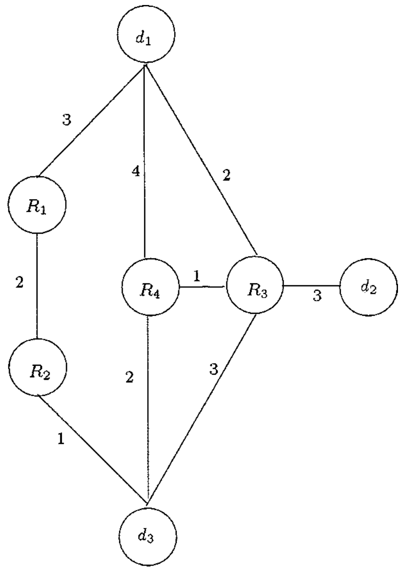
Manuscript received May 14, 1998; revised August 25, 1998 and October 1, 1999; approved by IEEE/ACM Transactions on Networking Editor M. Ammar.

The authors are with the Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel.

Publisher Item Identifier S 1063-6692(00)10927-6.



(a)



(b)

Fig. 1. (a) A network example. (b) The associated graph.

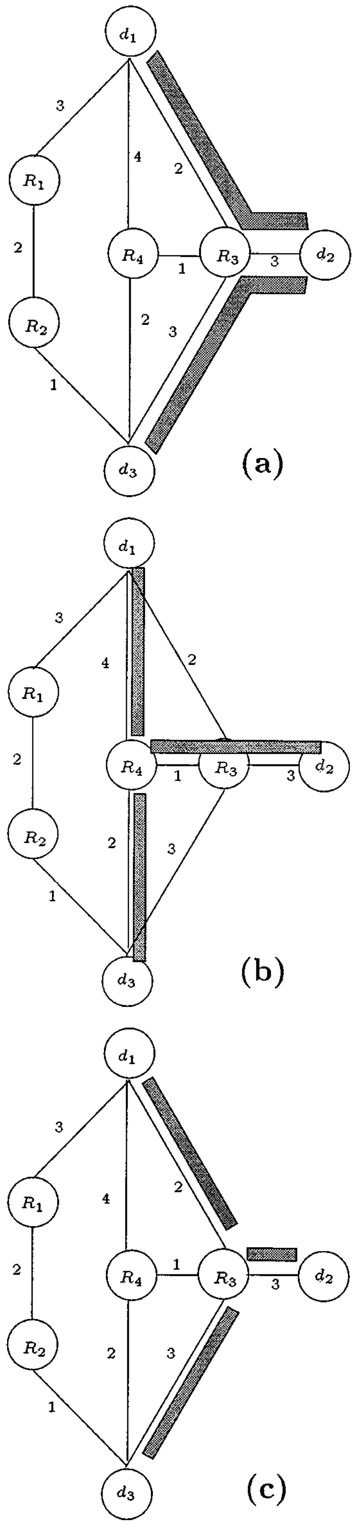


Fig. 2. Three possible VPN configurations for the example in Fig. 1.

with no cost because each VPN tunnel terminated at a core router is associated with management and memory complexity on that router. To understand the reason for this, consider the VPN in Fig. 2(b). Suppose that a local host of d_1 sends a packet to a local host of d_3. Upon receiving the packet, border router d_1 sends it over the tunnel to R_4. Core router R_4 might serve as a VPN router of several VPNs. Therefore, upon receiving the packet it needs to determine the VPN to which the packet belongs. This can be done based upon the identity of the tunnel over which the packet has been received, or the address of the source host. Then, R_4 needs to get the routing table associated with this specific VPN and to locate in this table the entry associated with the destination host. This implies that R_4 needs to maintain a routing table for every VPN for which it serves as a tunnel end point. An immediate consequence is that this table needs to be updated, most likely by means of a routing protocol like RIP or OSPF. An independent instance of such a protocol has to be executed for every VPN by the core routers that function as end points of the tunnels forming the VPN. Therefore,

R_4 also needs to participate in the routing protocol associated with every VPN it serves.

In this paper, we address the problem of determining a layout of VPN tunnels while taking into account two factors: the cost of the links over which VPN tunnels are established, and the cost of the core routers that serve as end points for the tunnels. We define related graph algorithm problems, analyze their complexity, and presents several heuristics for solving these problems.

The rest of the paper is organized as follows. In Section II, we define the problem of determining a VPN layout while limiting the usage of core routers. This problem will be referred to as the Minimum Cost VPN Problem (MC-VPN). We then show that this problem is both NP-hard and hard to approximate. We show that the same results hold for another version of this problem, called Minimal Active Set VPN (MAS-VPN). In Section III, we present two heuristics for solving the MC-VPN problem. Though the presented algorithms achieve no strict theoretic approximation ratios, in practice they perform very respectably. This is shown by simulation results in Section IV. Finally, Section V concludes the paper.

II. Minimum Cost VPN Problem

The layout of VPN tunnels can be optimized to satisfy different optimization parameters, like bandwidth, survivability, minimum hops between source and destination pairs, etc. However, one of the most important factors is usually minimum cost. We assume that when a tunnel is established over a link, it encounters a cost which is associated with this link. Though a strict association between a tunnel and links does not exist when tunnels are established using a connectionless mechanism like IP-over-IP, it is expected that future VPNs will be established over connection-oriented tunnels using technologies like ATM, MPLS, or WDM.

Our model allows each link to have a different cost on every direction. Therefore, the core network is represented by a *directed* graph.[1] The cost of each link on every direction is determined based on administrative considerations of the core network operator (ISP) and on the bandwidth allocated over that link, explicitly or implicitly, for the VPN tunnel. Generally, overloaded links are expected to be more costly than underloaded links. Since network links have different capacity on each direction,[2] and different load on each direction, we believe that a directed graph reflects more accurately the behavior of a real network.

In the following, the routers connecting the remote networks to the core network (d_1, d_2, and d_3 in Fig. 1) are referred to as "the border routers" whereas the other routers (R_1–R_4 in Fig. 1) are referred to as "the core routers." The nodes that function as end-points of tunnels, like R_4 in Fig. 2(b), are referred to as "active core nodes." By definition, every border router is an active node. When the routers of the core network do not support VPN, the VPN paths (tunnels) would have to start and end only at the border routers. In such a case, the group of active nodes is equal to the group of border nodes, as in Fig. 2(a).

As already indicated, in this work we concentrate upon the establishment of the MC-VPN. An immediate consequence is that the VPN tunnels form a tree, because in any circle of VPN tunnels one tunnel can be removed in order to reduce the cost of the layout without affecting connectivity. Since our core network is represented by a directed graph, there are several options to set up a minimum-cost tree spanning a group of nodes. In order to define a unique solution, we assume that one of the branch networks connected by the VPN is the corporate headquarter where most of the corporate servers are located, and that the majority of traffic is sent from the headquarter to the other branches. Therefore, we will seek for a directed tree of VPN tunnels rooted at a headquarter border node and spanning the rest of the border nodes.

Assuming that using a core router as an active node incurs a price, our goal is to find a minimal cost set of tunnels/paths forming a logical tree rooted at the headquarter border router and spanning the other border routers, such that the cost of the active nodes used is below a given bound. A formal definition of this problem is as follows:

Problem 1: The Minimum Cost VPN problem (MC-VPN).

Given a directed graph $G(V, E)$, an edge weight function c: $E \rightarrow \mathcal{R}^+$, a vertex weight function w: $V \rightarrow \mathcal{R}^+$, a bound on the available "funds" F for active core nodes, a group $M \subseteq V$ of border nodes, and a root (headquarter border node) $s \in M$, find a set of directed paths T and a set of active nodes A that minimize $\sum_{p \in T} \sum_{e \in p} c(e)$ such that $\sum_{v \in A \setminus M} w(v) \leq F$ where

1) $\forall p \in T$ the endpoints of p are vertices in A;
2) $\forall v \in M \backslash s$ there exists a sequence of one or more directed paths in T that leads from s to v.

The solution to MC-VPN induces a *logical* directed tree rooted at s and spanning $M \setminus s$. This is a "real" spanning tree in an induced graph $G'(A, E')$ where an edge between two vertices in A exists if and only if a path between these vertices exists in the original graph G. Note, that a solution to MC-VPN may actually induce a subgraph that is *not* a tree in G.

Our cost model is motivated by the work performed in [5]. This work discusses the possible VPN topologies, and the trade-off between the cost of routing, the cost of tunnel set-up and maintenance, and the cost of activating internal nodes as end points of VPN tunnels. The design of a VPN is relatively simple if the main target is to minimize the usage of bandwidth. In such a case the VPN should consist of the collection of "shortest paths" from the headquarter node to each of the other sites of the VPN. However, in many networks today, and in most of the future networks, the cost of operation and maintenance of virtual topologies is larger than the cost of the bandwidth. Our model considers such a network. It therefore does not aim in minimizing the cost of bandwidth usage, but the cost of operation and maintenance of the virtual topology. The weight associated with every link indicates in our model the cost of building and maintaining a VPN tunnel over this

[1]Throughout the paper we have several figures and examples where for simplicity of presentation we use an undirected graph to model the network. In these cases it is assumed that every edge has the same cost on both directions.

[2]This is especially true for the access links, connecting the border routers to the core networks, since in many cases these link use technologies like ADSL, cable modem, or wireless local loop (WLL), that allocate much more bandwidth on the "downstream channel" (toward the user side) than on the "upstream channel" (toward the core network side).

link, regardless of the volume of traffic the link actually carries for each VPN. Such a cost includes the need to set up a tunnel, to maintain the tunnel (e.g., forwarding the "KEEP-ALIVE" messages exchanged between the two end points) and to hold tunnel information (e.g., the VCI/VPI value in the case of an ATM-based tunnel, or the LSP in the case of an MPLS-based tunnel). This cost is incurred for each tunnel, and therefore a link that participates in several tunnels of the same VPN encounters this overhead for each tunnel independently of the other tunnels. This gives rise to our requirement that $\sum_{p \in T} \sum_{e \in p} c(e)$ has to be minimized.

The cost of tunnel maintenance incurred by the intermediate nodes of a tunnel is different than the cost incurred by the tunnel end points. This is due to the following two reasons.

1) These nodes need to do more extensive tunnel maintenance work. For example, the need to have a timer that dictates when a KEEP-ALIVE message has to be sent over the tunnel.
2) These nodes need to maintain not only the tunnel but also the VPN itself. For instance, node R_4 in Fig. 2(b) needs to run a routing protocol (see [5] and [10] for a particular example) and to maintain routing tables in order to determine how to route a packet received from d_1 whose destination is d_2 or d_3. Node R_4 also needs a special logic in order to determine the VPN to which a received packet belongs to (e.g., using a "VPN identifier" as proposed in [3]).

Following this assumption, where the cost of tunnel maintenance incurred by the intermediate nodes of a tunnel is different from the cost incurred by the tunnel end points, we distinguish in our model between $\sum_{p \in T} \sum_{e \in p} c(e)$ and $\sum_{v \in A \setminus M} w(v)$, and seek for a solution that takes into account both the cost of the links and the cost of activating internal routers as end points of a VPN tunnel.

Although MC-VPN assumes that every core router has VPN capability, this assumption may be easily bypassed by assigning infinite costs to nodes that do not have such capability. In order to seek for a layout that use no active core router, the fund F can be set to 0.

MC-VPN imposes no restriction on the active group A, except that this group must include all border routers. Note that the cost of active nodes within the group of border routers is not considered as part of the VPN cost, since the border nodes must be active, and their cost is a constant factor in any feasible solution.

If the funds F is infinite, MC-VPN reduces to the classic Steiner tree problem (STP) [4], since there is no motivation left for reusing edges. In this case, a Steiner tree may be transformed into a valid solution by decomposing it into paths with endpoints either at the source, destinations, or fork nodes of the tree. On the other hand, when the cost of activating[3] some of the core routers is greater than 0, MC-VPN penalizes a solution that uses the same edge several times by multiplying the edge cost by the number of times it is used. For example, in Fig. 3(a), a graph is depicted where s is the root (headquarter) node and every vertex except x is in the set M of border nodes. Assume that all edges in the graph have an equal cost of 1 on both directions and all vertices have a weight of 0 except for x which weighs 2. The routing depicted has a cost of 7, since the edge (s, x) is used twice: for the tunnel (path) $s \rightarrow x \rightarrow d_4$ and for the tunnel $s \rightarrow x \rightarrow d_5$. However, in this solution no core router is activated as a VPN node. If the available funds F for using active nodes allowed activating x, i.e., $F \geq 2$, it would be better to split these paths [Fig. 3(b)], creating a routing with a cost of 6.

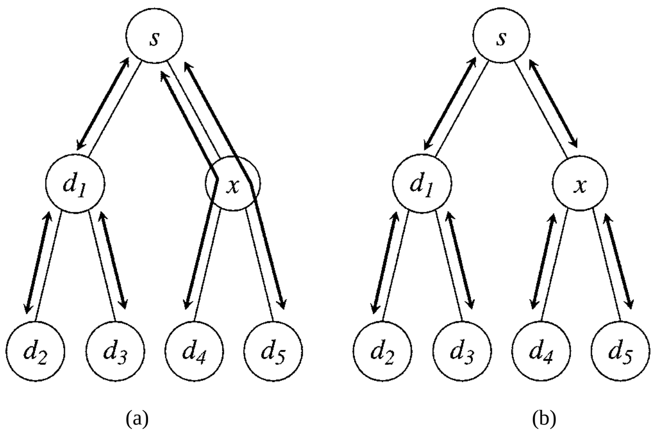


Fig. 3. Reuse of edges in solutions of MC-VPN. (a) x cannot be activated; routing cost is 7. (b) x can be activated; routing cost is 6.

The balance between the node and edge cost function may depend heavily on the details of the core network implementation, such as the processing power of core routers, their memory resources, the cost of setting up and taking down VPN tunnels between core routers, the cost of executing a routing protocol for every VPN, and many other factors. However, as shown in the following, even if either of these costs becomes negligible, the complexity of MC-VPN remains NP-hard.[4]

Theorem 1: MC-VPN is NP-hard, and also hard to approximate.

Proof: This follows from the relation of MC-VPN to the directed STP. In [9] it is shown that the version of STP on directed graphs is NP-hard as well as hard to approximate in the sense that an approximation algorithm achieving an approximation ratio lower than $O(\log n)$ does not exist unless $DTIME(n^{poly \log(n)}) \supseteq NP$ (this assumption is believed to be as unlikely as $P = NP$). We note that MC-VPN reduces to the directed STP when the funds F available for active nodes are infinite. Therefore, the same hardness results apply to MC-VPN as well. □

Suppose now that the only goal is to minimize the number of active nodes. This gives rise to the following problem definition.

[3]A core router is said to be activated when it becomes an end point of a tunnel.

[4]A problem that is NP-hard is believed not to be solvable by any efficient algorithm, i.e., an algorithm that always terminates within a "reasonable" period of time. If an algorithm could be developed that would solve an NP-hard problem efficiently, this would mean that a great variety of intractable problems for which no efficient algorithm is known could be solved efficiently too. For a more formal definition, see [4].

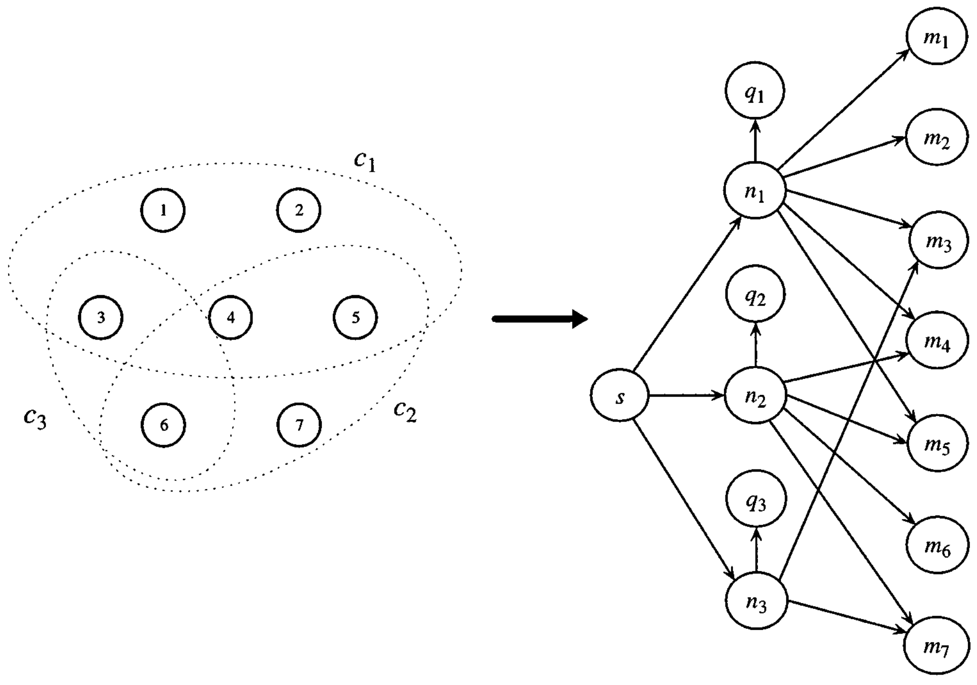


Fig. 4. Reduction from Minimum Set Cover (MSC) to MAS-VPN.

Problem 2: The Minimal Active Set VPN (MAS-VPN).

Given a directed graph $G(V, E)$, a group $M \subset V$ of border nodes, and a headquarter border node $s \in M$, find a tree T rooted at s and spanning M with a set of fork nodes A such that $|A \backslash M|$ is minimal.

MAS-VPN and MC-VPN are closely related. MAS-VPN may be viewed as a version of MC-VPN where the cost of all edges is zero, the cost of all nodes is one, and the goal is to minimize the funds used for activating core routers. The difference between these two problems is that MC-VPN allows a solution to use every edge more than once.

Theorem 2: MAS-VPN is NP-hard, and also hard to approximate.

Proof: We show that MAS-VPN is NP-hard by reducing the minimum set cover (MSC) problem to it. MSC is known to be NP-complete , and is defined as follows:

Let C be a collection of subsets of a finite set S, and $K \leq |C|$ a positive integer. Does C contain a cover for S of size K or less, i.e., a subset $C' \subseteq C$ with $|C'| \leq K$ such that every element of S belongs to at least one member of C'?

Given the input tuple (C, S, K) for MSC, construct the following directed graph $G(V, E)$ (see Fig. 4):

$$\begin{aligned} V &\triangleq \{n_i | c_i \in C\} \cup \{m_i | i \in S\} \cup \{q_i | c_i \in C\} \cup \{s\} \\ E &\triangleq \{(s, n_i) | n_i \in V\} \cup \{(n_i, m_j) | j \in c_i\} \\ &\quad \cup \{(n_i, q_i) | c_i \in C\} \end{aligned}$$

Define a group of border nodes $M \triangleq \{m_i | i \in S\} \cup \{q_i | c_i \in C\} \cup \{s\}$. Let s be the headquarter node. It will now be shown that a solution to MSC exists if and only if a solution to MAS-VPN exists where $|A \backslash M| \leq K$.

Let C' be a solution to MSC ($|C'| \leq K$). Define the following solution to MAS-VPN on G:

$$\begin{aligned} A &\triangleq \{n_i | c_i \in C'\} \cup \{m_i | i \in S\} \cup \{s\} \\ T &\triangleq \{(s, n_i) | c_i \in C'\} \cup \{(n_i, m_j) | c_i \in C', j \in c_i\} \\ &\quad \cup \{(n_i, q_i) | c_i \in C'\} \cup \{(s, n_i), (n_i, q_i) | c_i \in C \backslash C'\} \end{aligned}$$

Clearly, the above defined T and A comprise a solution to MAS-VPN although the size of T may still be reducible. Thus, $|A \backslash M| = |\{n_i | c_i \in C'\}| = |C'| \leq |K|$.

To prove the other direction, let T and A be a solution to MAS-VPN on G such that $|A \backslash M| \leq K$. Define the following solution to MSC:

$$C' \triangleq \{c_i | (n_i, m_j) \in T\}.$$

C' covers S since by definition T must reach every node m_j through some edge (n_i, m_j), and by construction, if $(n_i, m_j) \in T$ then $j \in c_i$. The only potential fork nodes in T are the nodes n_i. By definition, T must contain all edges of the form (n_i, q_i), since these are the only edges entering the nodes q_i. Thus, every node n_i which has an outgoing edge of the form (n_i, m_j) must be a fork node in T. Thus, $|C'| = |\{c_i | (n_i, m_j) \in T\}| \leq |A| = |A \backslash M| \leq K$.

By using the same construction in the above proof, it can be shown that every approximation algorithm for MAS-VPN achieving an approximation ratio of K on the size of the set $A \backslash M$ can be used to achieve the same approximation ratio on MSC. For MSC, it is known that unless $DTIME(n^{poly \log(n)}) \supseteq NP$, no algorithm can achieve a better approximation ratio than $K = O(\log n)$ [8]. Therefore, under the same assumption, MAS-VPN is at least as hard to approximate. $\square$

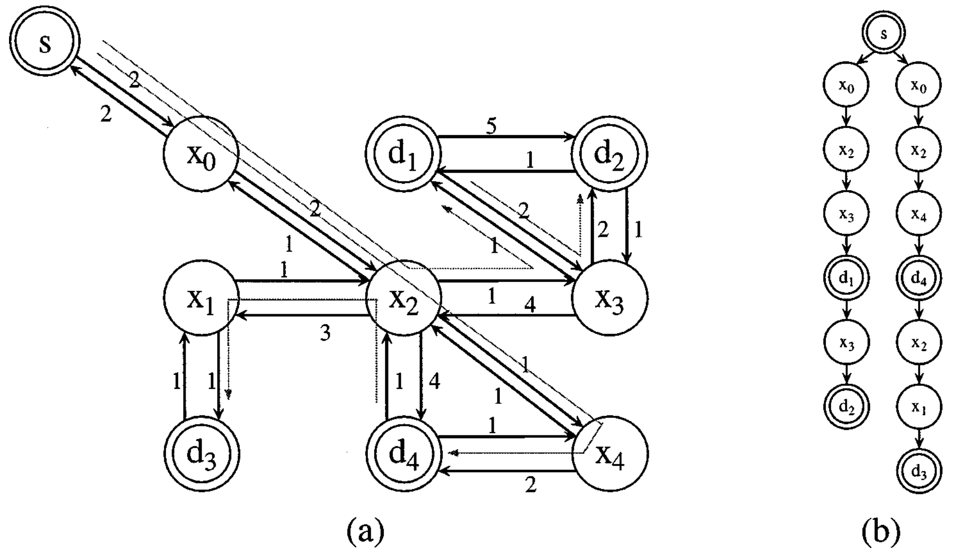


Fig. 5. Example for ASPH algorithm.

III. Algorithms for MC-VPN

As proved in Section II, MC-VPN is NP-hard and therefore a polynomial algorithm that finds an optimal solution is unlikely to exist. We therefore concentrate on heuristics that approximate the optimal solution. Two basic approximation algorithms for MC-VPN are presented. Both algorithms use the same general approach. They first construct a CPE-based solution, that ignores the ability to use core routers as end points of VPN tunnels. Then, they try to improve the solution by spending the funds F on active core routers in strategic points. In Section II, it was shown that MC-VPN is not only NP-hard, but also likely to be hard to approximate. Therefore, the presented algorithms achieve no strict theoretic approximation ratios. However, in practice they perform very respectably.

A. Building a CPE-Based Solution to MC-VPN

The first step in our approximation algorithms is to build a CPE-based solution for MC-VPN that does not use any core router. Such a solution is a set of paths with endpoints exclusively within the group M of border routers.

We present two algorithms for the construction of such a solution. In [9], the SCTF algorithm for STP on directed graphs was proposed. The first proposed algorithm, called the active shortest path heuristic (ASPH), is a slightly altered SCTF algorithm. The algorithm starts at the root (headquarter) node s and covers in each iteration of the algorithm exactly one new node from the group M of border routers. The covered node is the node nearest to one of the previously covered border router nodes, and the shortest path between these two nodes is added to the solution. A similar procedure is employed by the *shortest path heuristic* (SPH) [12], except that in our algorithm, ASPH, all paths added to the solution must begin and end at border router nodes. This implies that some edges might be used more than once.

An example of this phase of the algorithm is depicted in Fig. 5. The headquarter node is s and the border router group is $M = \{d_1, d_2, d_3, d_4\}$. The set of paths constructed in this example, listed in the order they are found, is $\{(s, x_0, x_2, x_3, d_1), (d_1, x_3, d_2), (s, x_0, x_2, x_4, d_4), (d_4, x_2, x_1, d_3)\}$. Note that these paths are *not* disjoint. For example, the edge (x_0, x_2) is used by two of the above paths. The resulting solution is shown in Fig. 5(b).

ASPH as presented above may be implemented to run in $O(|M|(|E|+|V|\log|V|))$ time by using an efficient procedure for the shortest path routine. A formal description of ASPH appears in the following:

```
Algorithm 1: Active Shortest Path
Heuristic (ASPH).
1) Find a CPE-based solution to MC-VPN:
   $X \leftarrow \{s\}$, $T \leftarrow \Phi$.
   While $M \backslash X \neq \Phi$ do
   a) Find the shortest $p$ path
      connecting a vertex $x \in X$ to
      a vertex $v \in M \backslash X$.
   b) $T \leftarrow T \cup \{p\}$
   c) $X \leftarrow X \cup \{v\}$
2) $A \leftarrow M$
3) Call OPTIMIZE(T, A).
4) $(T, A)$ is the desired solution.
```

The second algorithm we present for building a CPE-based solution to MC-VPN is the active double tree heuristic (ADTH). Many algorithms for solving the STP are known in the literature (i.e., [1], [9], [12]). ADTH can use any of these algorithms as a basis for constructing a solution. As an example, we use the SPH algorithm [12]. Fig. 6(a) shows the spanning tree found by SPH in the network of Fig. 5(a). Note that starting with a tree implies that some fork nodes may not belong to the group M of border routers. Thus, the second phase of ADTH transforms the tree found by the initial phase (SPH, in our case) into a set of paths with endpoints exclusively within M. The paths are constructed by touring the tree in a depth-first manner. This tour is broken into subpaths such that every time a vertex belonging to the group of border nodes is passed, the previous subpath is ended and a new one is initiated. Unneeded paths, that lead to border routers already reached by previous paths, are discarded.

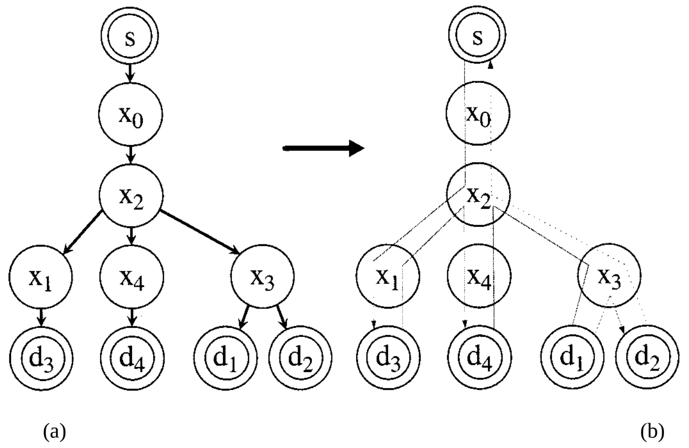


Fig. 6. ADTH algorithm before optimization. (a) Spanning tree. (b) Set of paths.

The paths created from the tree are depicted in Fig. 6(b). Paths discarded are represented in this figure by dotted lines.

The time complexity of the ADTH algorithm which makes use of SPH as the basic Steiner tree algorithm is the same as that of ASPH.

B. Activating Core Routers

After a valid CPE-based solution for MC-VPN has been constructed by either of the above algorithms, it is the responsibility of the node activation optimization procedure to eliminate multiple use of edges wherever this is possible. The target of this procedure is to revise the CPE-based solution such that every node and edge is used once at the most if this can be achieved by activating core routers with the available funds F.

A possible optimization procedure is described hereafter. This procedure examines all core router nodes in the solution with more than one incoming edge. The order in which the nodes are activated is intended to maximize the profit of node activation, defined as the sum of edge costs saved as the result of activation. A core router node is considered a candidate for activation if its cost is less than the remaining funds F. A candidate node is activated if the ratio between the profit and cost of its activation is the largest among the ratios achieved by all other candidate nodes. When a node is activated, the optimization procedure must ensure that the remaining incoming path still connects the headquarter node s to the new active node, so that the correctness of the solution is preserved. If at some stage of the algorithm, an activated node remains with only one outgoing edge, i.e., it becomes a simple intermediate node on some path, it is deactivated, and its cost is returned to the available funds.

As an example, we describe how the proposed local optimization procedure optimizes the initial solution constructed by ASPH as depicted in Fig. 5(b) and given as a reference also in Fig. 7(a). Assume that the weight of all nodes in the considered network is 1, and that the available funds are $F \triangleq 2$. The initial set of candidate nodes is $\{x_0, x_2, x_3\}$. The profit of their individual activation is $\{2, 5, 2\}$, respectively. For example, activating x_2 would enable removing two partial paths reaching it: $s \rightarrow x_0 \rightarrow x_2$ and $d_4 \rightarrow x_2$. Because x_2 must remain reachable from s, it is not possible to remove both partial paths reaching

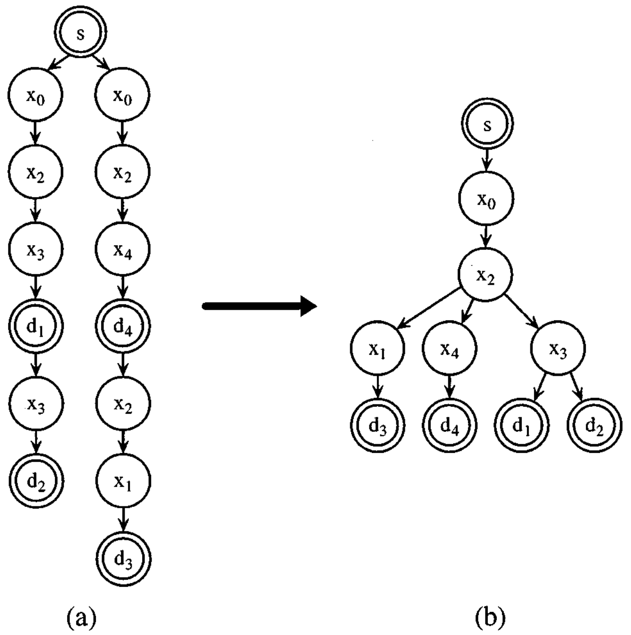


Fig. 7. The ASPH solution before and after optimization.

x_2 from the direction of x_0 which are the most expensive paths reaching it. Since all nodes are assumed to have the same costs, the profit/cost ratio of x_2 is the largest and it is the first candidate node to be activated. As a result, the above mentioned partial paths can be removed from the solution. After x_2 is activated, x_0 is no longer a candidate for activation since its incoming degree becomes 1. Therefore, the only remaining candidate node is x_3. The optimized solution created after x_3 is activated too is depicted in Fig. 7(b).

The order in which nodes are activated is not necessarily optimal, but is a heuristic frequently used for similar problems such as the Knapsack problem [4]. For instance, in the above example, assume that x_0, x_2 and x_3 have a cost of 1, 3 and 4 respectively, and that the available funds are $F \triangleq 3$. In such a case, x_0 is the first (and only) node to be activated since its profit/cost ratio, 2, is larger than the ratios of x_2 and x_3, which are 5/3 and 2/4 respectively. However, in this example, activating x_2 would result in a better solution.

An alternative heuristic, which performs better in the last case activates the candidate nodes starting at the node furthest away from the headquarter node s and gradually nears s until the funds are exhausted. This heuristic, which we call the reverse-BFS heuristic, was tested in simulations and performs on the average slightly worse than the profit/cost heuristic.

The formal description of the node activation optimization procedure appears in the following. Its time complexity is $O(|V|^2|E|^2)$, a value which can be reduced by a less naive implementation that updates cost evaluations incrementally.

```
**Procedure 1** OPTIMIZE(T, A):
1) Let $\delta_{\rm in}(v)$ be the number of incoming
   edges of $v$ in $T$.
   Let $C$ be the set of candidate nodes
   defined as the nodes for which $\delta_{\rm in}(v) \geq 2$
   and $w(v) \leq F$.
```

```
2) While $C$ is not empty do
   a) For every vertex $w \in A$, calculate the
      set of father vertices $fathers(w)$
      that is defined as the set of
      vertices on the single path from $s$ to
      $w$ in $T$.
   b) For every node $v \in C$ calculate the
      following:
      Let $P(v)$ be the set of paths in
      $T$ passing through $v$.
      For every path $p \in P(v)$ let $c(p)$ be the
      cost of the edges from the beginning
      of $p$ until $v$.
      Let $S(v) \triangleq \{p | p \in P(v)$ and
      $v \notin fathers(start(p))\}$ be the set of paths
      entering $v$ that end paths connecting
      $s$ to $v$.
      Let $p_{\min}(v)$ be a path from the set of
      paths $S$ such that $\forall p' \in S: c(p_{\min}(v)) \le c(p')$.
      By the definition of $S$,
      disconnecting $v$ from all incoming
      paths except for $p_{\min}(v)$ will still
      leave $v$ connected to $s$.
      Let $r(v) \triangleq w(v) / \sum_{p \in P(v) \setminus \{p_{\min}(v)\}} c(p)$ be
      the cost/profit ratio of node $v$.
   c) Let $x$ be a vertex in $C$ for which
      $\forall v \in C: r(x) \le r(v)$. $x$ has the lowest
      cost/profit ratio of all candidate
      nodes. Therefore, this is the node
      selected for activation.
   d) For every $p \in P(x) \setminus \{p_{\min}(x)\}$ do
      i) $T \leftarrow T \setminus p$.
      ii) Let $p_{tail}$ be the tail of $p$
          beginning at $x$. $T \leftarrow T \cup \{p_{tail}\}$.
   e) Break $p_{\min}(x)$ at $x$:
      $T \leftarrow T \setminus p_{\min}(x)$
      $T \leftarrow T \cup \{p_{\min_{head}}(x), p_{\min_{tail}}(x)\}$.
   f) Update $A, F, C$:
      Add $x$ to the active node set:
      $A \leftarrow A \cup \{x\}$.
      Update the funds $F$: $F = F - w(x)$
      If for any node $u \in A$, $\delta_{\text{out}}(u) = 1$,
      remove $u$ from $A$, unite the paths
      reaching $u$ and $F = F + w(u)$.
      Recalculate $C$: Remove any node $v$ for
      which $w(v) > F$ or $\delta_{\text{in}}(v) = 1$ and add any
      node $v$ for which $w(v) \le F$ and
      $\delta_{\text{in}}(v) \ge 2$.
```

IV. Simulation Results

In the previous section, two algorithms for approximating MC-VPN were introduced. In this section, the performance of these algorithms is evaluated on random graphs. Although real-world networks are not completely random in their structure, we believe the simulation results predict quite accurately the behavior of the proposed algorithms in a realistic setting.

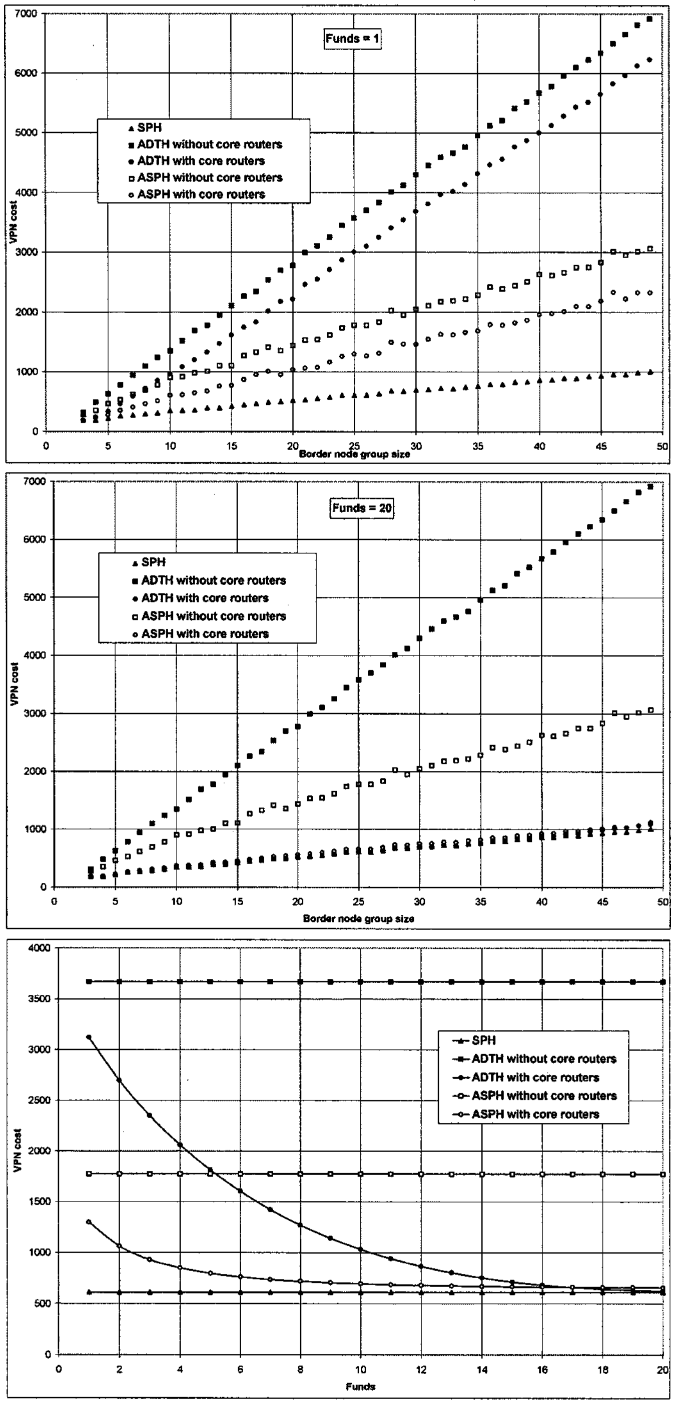


Fig. 8. VPN cost for various funds.

The algorithms were tested on random directed graphs generated as follows. First, a graph with 50 vertices and 150 edges was created, where every possible edge in the graph was selected with equal probability. The weights of these edges were initialized at random values uniformly distributed between 2 and 22. Antisymmetric edges were assigned equal weights. This part of the graph was intended to model the backbone of a large network. Another 50 vertices were connected to this graph, one to every backbone vertex. The connection was made by one edge from the backbone to the new vertex with a weight generated as described above, and one edge from the new vertex to the backbone with a weight randomly distributed between 20 and 220. This connection was intentionally made asymmetrical, thereby modeling end-user connections such as ADSL, cable modem, or satellite links.

On every graph, the algorithms were run on every border router group M size between 3 and 49, including the headquarter node. These groups were reselected at random for every graph and every group size from the set of end-user vertices. The weight of border routers was assumed to be equal to 1 for all nodes. The funds available for using core routers ranged from 1 to 20 (i.e., a maximum of 20 nodes were allowed to be activated). The results were averaged over 50 different graphs. Confidence intervals for several points were calculated and found to be within the 90%.

The algorithms were compared to SPH, the well-known approximation algorithm for the STP [12]. This algorithm constructs solutions to the STP that are at most twice as expensive as the optimal solution and is known to be even better on the average [12]. SPH does not find a valid solution for MC-VPN, since it constructs a spanning tree and not a set of paths. However, *the cost of this tree is a lower bound on the cost of an optimal solution to MC-VPN, since it ignores the bound F on the available funds for active nodes.* It serves as a benchmark since calculating the optimal solution is an NP-hard problem (see Section II).

The VPN cost achieved by the algorithms is depicted in Fig. 8. In this figure, the VPN costs of ASPH and ADTH before and after the node activation optimization procedure are shown for various available funds. When no funds for active nodes are available, ASPH is around three times worse than SPH, and ADTH performs six times worse than SPH. As the available funds increase, the routing cost reduces until ADTH achieves the same routing cost as SPH (due to the fact that it is based on SPH), whereas ASPH remains about 7% worse than SPH. When the results are examined more closely, it appears that the relative benefit of core routers is independent of the border router group size.

In Fig. 9, the number of active nodes used by ADTH and ASPH averaged over the various tested funds is depicted. ASPH, which is a more "path-oriented" algorithm, uses fewer active nodes than the "tree-oriented" ADTH. This is most prominent for border router group sizes around 40% of the number of nodes in the graph. When funds are abundant, ADTH uses on the average at least 10% more active nodes than ASPH.

Although MC-VPN is NP-hard, on the average, both ADTH and ASPH achieve close to optimal performance. ASPH is significantly better than ADTH when the available funds F are scarce. When funds are abundant, ADTH is slightly better.

V. Conclusion

In this paper, we have addressed the problem of determining a layout of VPN tunnels for an Intranet VPN, while taking into account two factors: the cost of the links over which a VPN tunnels is established and the cost of activating core routers as end points of VPN tunnels. We have defined two related graph algorithm problems: the MC-VPN problem and the MAS-VPN problem. We have proved that both problems are not only NP-hard, but

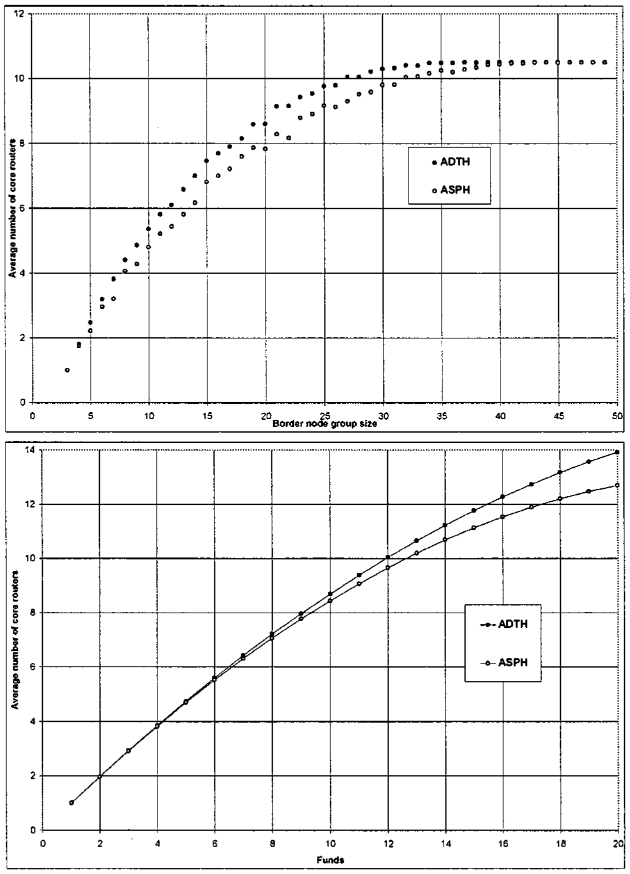


Fig. 9. Number of core routers used.

also hard to approximate in the sense that an approximation algorithm achieving an approximation ratio lower than $O(\log n)$ is unlikely to exist.

We have presented two heuristics that approximate the optimal solution for MC-VPN (which is more general than MAS-VPN): ADTH and ASPH. Both algorithms use the same general approach. They first construct a CPE-based solution, that ignores the ability to use core routers as active nodes. Then, they try to improve the solution by spending the funds F on activating core routers in strategic points. The algorithms have been tested using simulations. Their results have been compared to the results produced by a well known approximation algorithm for the STP, which does not find a valid solution for MC-VPN but can serve as a benchmark. Although MC-VPN is NP-hard, on the average, both ADTH and ASPH were shown to achieve close to optimal performance.

References

[1] K. Bharath-Kumar and J. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. 31, pp. 343–351, Mar. 1983.
[2] A. Valencia *et al.*, "Layer two tunneling protocol (L2TP),", Internet Draft, Oct. 1998.
[3] B. Fox and B. Gleeson, "Virtual private networks identifier,", RFC-2685, Sept. 1999.
[4] M. Garey and D. Johnson, *Computers and Intractability*. San Francisco, CA: Freeman, 1979.
[5] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, "A framework for IP-based virtual private networks,", RFC-2764, Feb. 2000.

[6] S. Hanks, T. Li, D. Farinacci, and P. Tranina, "Generic routing encapsulation,", RFC-1701, Oct. 1994.

[7] S. Kent and R. Atkinson, "Security architecture for the Internet Protocol,", RFC-2401, Nov. 1998.

[8] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *J. Assoc. Comput. Mach.*, vol. 41, no. 5, pp. 960–981, 1994.

[9] S. Ramanathan, "An algorithm for multicast tree generation in networks with asymmetric links," in *Proc. INFOCOM*, San Francisco, CA, Mar. 1996, pp. 337–344.

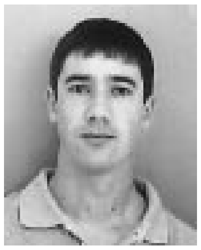
[10] E. Rosen and Y. Rekhter, "BGP/MPLS VPNs,", RFC-2547, Mar. 1999.

[11] W. Simpson, "IP in IP tunneling,", RFC-1853, Oct. 1995.

[12] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Math. Japonica*, vol. 24, pp. 573–577, 1980.

Reuven Cohen (M'93–SM'99) received the B.Sc., M.Sc. and Ph.D. degrees in computer science from the Technion, Israel Institute of Technology, Haifa, Israel, in 1986, 1988 and 1991, respectively.

From 1991 to 1993, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, working on protocols for ATM and high speed LANs. Since 1993, he has been with the Department of Computer Science, Technion, Israel Institute of Technology, and he has also consulted for numerous companies, including Hewlett-Packard and ECI Telecom. His most recent work focuses on the design and evaluation of routing, multicast and transport protocols in networks, and on technologies for broadband access networks.



Gideon Kaempfer received the B.Sc. degree in electrical engineering and the M.Sc. degree in computer science, both from the Technion, Israel Institute of Technology, Haifa, Israel, in 1995 and 1998, respectively. His Master's thesis was on the design of efficient algorithms and protocols for multicast routing.

In 1998, he co-founded Charlotte's Web Networks, a vendor of Terabit routers. He is currently the CTO of this company.