Enhanced Cost Optimized VPN Provisioning Algorithm

[1]R.Ravi, [2]Dr. S.RadhaKrishnan,TIFAC-CORE, Arulmigu Kalasalingam College of Engineering

[1]Assistant Professor in Department of CSE,
[2]HOD and Senior Professor in Department of CSE,
Arulmigu Kalasalingam College of Engineering, Anand Nagar, Krishnankovil, 626 190
Virudhunagar Distri t, Tamil Nadu, India.

Summary
A Virtual Private Network (VPN) aims to emulate the services provided by a private network over the shared Internet. The endpoints of VPN are connected using abstractions such as Virtual Channels (VCs). Reliability of an end-to-end VPN connection depends on the reliability of the links and nodes. VPN service providers provide new services with Quality of Service (QoS), guarantees are also resilient to failures. Supporting QoS connections requires the existence of routing mechanisms that computes the QoS paths, where these paths satisfy the QoS constraints. Resilience to failures, on the other hand, is achieved by providing, each primary QoS path, a set of alternative QoS paths, upon a failure of either a link or a node. We aim at to minimize the total bandwidth reserved on the backup edges. The above objectives, coupled with the need to minimize the global use of network resources, imply that the cost of both the primary path and the restoration topology should be a major consideration of the routing process. It turns out that the widely used approach of disjoint primary, restoration paths is not an optimal strategy. Hence, the proposed approximation restoration algorithms construct a restoration topology, and this topology protects a portion of the primary QoS path. This approach guarantees to find a restoration topology with optimal cost which satisfies the QoS constraints.

Keywords

Restoration Schemes, QoS Models, QoS Constraints, Restricted Shortest path, Primary path and Restoration Topology.

1. Introduction

A virtual private network (VPN) is a private data network that makes use of the public Internet [1] to maintain privacy through the use of IP tunneling technology and network security protocols. VPNs can be regarded as a replacement of the expensive private leased lines. The main purpose of a VPN is to provide a company secure communication among multiple sites through the shared Internet. More detailed descriptions of VPNs can be found in [2]. To support a VPN, a service provider has to allocate predetermined paths to connect among customer sites. As customers may want to have bandwidth guaranteed, enough bandwidth has to be reserved on these paths. Therefore, finding appropriate paths and appropriate bandwidth reservation while minimizing the total bandwidth used becomes an important problem to service providers. Two popular models for specifying customer bandwidth requirements have been proposed. They are known as the *pipe model* and the *hose model*. In the pipe model, customers are required to specify the bandwidth they need among each pair of VPN endpoints. In other words, a customer has to know the traffic between each pair of sites in advance and inform the service provider. This model is not very flexible since a customer may not be able to predict the communication patterns between VPN endpoints. Another disadvantage of this model is that the resources reserved for a pair of VPN endpoints cannot be allocated to other traffic flows. Thus, the utilization of Internet resources becomes very inefficient. The hose model was proposed by *Duffield et al.* to solve the problems of the pipe model. In the hose model, VPN customers just need to specify the incoming and outgoing traffic volume of each VPN endpoint (known as *ingress bandwidth* and *egress bandwidth*) instead of between every pair of VPN endpoints. The ingress bandwidth of an endpoint is the capacity required for aggregating the incoming traffic to the endpoint from other endpoints. The egress bandwidth is the capacity required for aggregating the outgoing traffic from the endpoint into the network. In other words, ingress bandwidth specifies the maximum amount of traffic an endpoint would receive per time unit while egress bandwidth specifies the maximum amount of traffic an endpoint would send out per time unit. Detailed examples showing the differences between the pipe model and the hose model can be found in [3].

Present paper describes a new hose-model VPN provisioning algorithm called “Cost Optimized VPN

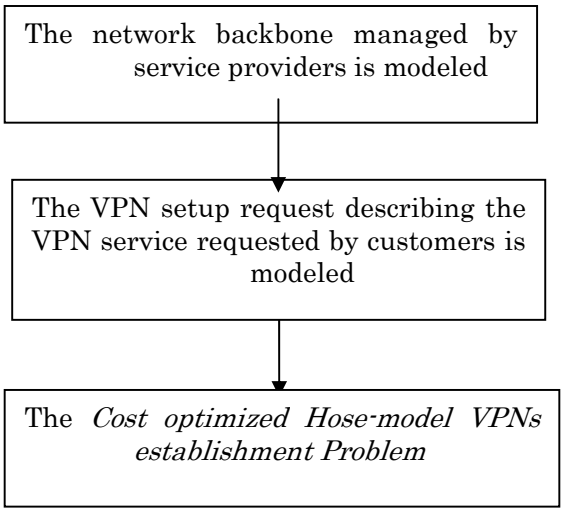
Provisioning Algorithm –*COVPA"* to address this issue.

2. Cost Optimized VPN Provisioning Algorithm –*COVPA*

INPUT:A Network graph G=(N,L),VPN access routers AR=(ar_1,ar_2,….,ar_p) ∑ N residual bandwidth constraints B on L and a VPN setup request Vr=(r_1,r_2,…,r_p)

OUTPUT: A minimum cost VPN tree VT_{MC} for Vr_i on which all leaf nodes are VPN access routers ar_i with r_i>0.

Diagram of Algorithmic approach



PHASE-I: The network backbone managed by service providers is modeled

The network backbone is modeled by an undirected graph G= (*N*, *L*), where *N* and *L* are the set of routers and the set of links, respectively. Let *n* and *m* denote the cardinality of *N* and *L*, respectively. Let *B* be the residual bandwidth of links on *L* and the amount of residual bandwidth on link *l* (*l*Є*L*) is denoted by *B*(*l*). A subset *AR* = {ar_1, ar_2…ar_p} of *N* (*AR* ⊆ *N*) is the set of VPN access routers. Each endpoint e_i of a VPN gains access to VPN service by connecting to a specific VPN access router ar_i in *AR*. In other words, for each endpoint of a VPN, there is a corresponding VPN access router in *AR*.

The elliptic region in figure 1 is an example of network backbone *G*. The round regions (A to G) inside *G* are routers in *N*. The solid lines between two routers are links in *L*. The number beside each link is the amount of residual bandwidth on it (*B* (*l*) =5 for all *l*Є*L* in this figure). The VPN access routers set *AR* = {*A*, *E*, *G*}. The round regions (1, 2 and 3) outside *G* are endpoints (*e1*, *e2* and *e3*, respectively, in our notation) of a VPN, which gain access to VPN service via routers in *AR*. The dotted lines labeled as *path i, j* is the transmission path for VPN traffic between *ei* and ej.

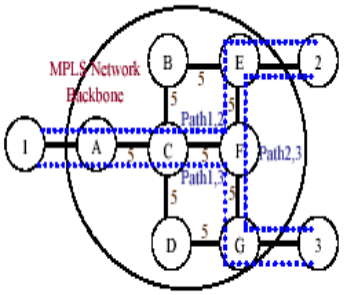


Figure 1. An example of MPLS Network Backbone *G*

PHASE-II: *VPN Setup Request Modeling*

The demands for VPN service of customers is described by VPN setup requests. In this paper consider that the bandwidth requirement of each endpoint *ej* is symmetric. Let *b*(*ej*) denotes the bandwidth requirement of endpoint *ej*, and *Maxr* denote the maximum bandwidth guarantee provided by service providers. The *i*th VPN setup request, denoted by vr_i , describes a VPN that customer's request service provider to establish. Each vr_i is represented by a *p*-tuple vector (r_1, r_2,…, r_p), where *p* is the cardinality of access routers *AR*. The number of nonzero elements in vr_i represents the number of endpoints contained in the corresponding VPN. The value of *j*th element r_j of vr_i represents the bandwidth requirement of endpoint *ej*.

PHASE III: *Cost Optimized Hose- model VPNs Establishment Problem*

The *COVPA* defined in this paper which mainly considers on-line establishment of bandwidth guaranteed point-to-point tunnels. However, in the context of VPN provisioning, the basic unit concerned is a VPN consisting of numerous point-to-point tunnels, rather than a point-to-point tunnel, that makes the problem more challenging. In *COVPN*, service providers manage a network backbone *G* (as described in subsection PHASE-I) on which VPNs are established. The VPN setup requests of customers are sent to VRS (VPN Request Server) by service provider.

In this paper consider the situation where (a) VPN setup requests arrive one by one independently and (b) information about future VPN setup requests is

unknown. This information includes the number of future VPN setup requests, the number of endpoints contained in each VPN setup request, and the bandwidth requirement of each endpoint. The service provider must process each VPN setup request in an on-line manner, the off-line model in not suitable. Upon receiving a VPN setup request vr_i, the service provider triggers the provisioning algorithm to establish a corresponding VPN. The provisioning algorithm performs this task by first choosing a path between each endpoint pair and then allocating bandwidth on each link on the paths. If there is not enough residual bandwidth on the link when the bandwidth is being allocated, vr_i will be rejected. In this paper the *rejection ratio is taken* as the performance metric to compare different hose-model VPN provisioning algorithms. The *rejection ratio* is defined as:

$$\text{Rejection ratio} = \frac{\text{Number of requests rejected}}{\text{Total number of request received}}$$

The optimization goal of provisioning algorithms is to minimize the rejection ratio, which in turn will maximize the number of requests successfully established on the network backbone. Although the main performance metric here focus on rejection ratio.

Other important performance metrics (eg: link utilization and bandwidth allocation efficiency) are also investigate in the experimental simulations. In this paper, assume that service provider uses a server-based strategy for processing VPN setup requests. In server-based strategy, the VPN provisioning algorithm is run on a single entity called *VPN request server* (*VRS*). The *VRS* also keeps the complete link state topology database and is responsible for finding an explicit path for each endpoint pair of a VPN. Then the explicit paths can be setup using a signaling protocol such as RSVP . For computing the explicit paths, the *VRS* needs to know the current network topology and link residual bandwidth. We assume that there exists a link state routing protocol for information acquisition.

3. The Factors Influencing Rejection Ratio

In this case, the links of the network backbone have a finite amount of bandwidth and the service provider needs to establish multiple VPNs on the network backbone on-line. The two most important factors influencing the rejection ratio achieved by provisioning algorithms are: (1) *Bandwidth allocation efficiency* (2) *Load balance mechanism that considers the amount of residual bandwidth on links*. Provisioning algorithms must take the residual bandwidth of links into account, and avoid using links that are thinly spread. This will balance the load on links of *G* and reduce rejection ratio.

4. COVPA

To address the problems described above we propose a new provisioning algorithm called the *Cost Optimized Hose-Model VPN Provisioning Algorithm* (*COVPA*). Both of COVPA and tree routing are tree-based.*COVPA* also adopts VPN tree for establishing each VPN. The cost function of *COVPA* for VPN tree selection is defined as following:

$$\text{Cost}_{\text{COVPA}}(T) = \sum_{1 \le x \le k} \frac{RS(l_x)}{B(l_x)}$$

Where $RS(l_x)$ and $B(l_x)$ represent the amount of bandwidth needed and the amount of residual bandwidth on link *lx*, respectively. The cost function of *COVPA* is inspired by the cost function defined in the routing algorithms for route selection.

When processing a request, *COVPA* tries to find a *VPN tree* that minimizes the cost function defined above. It is clear that the additional cost for using a link *lx* in building a *VPN tree* is proportional to the value of $RS(lx)$ and is reciprocal to the value of $B(lx)$. Therefore, *COVPA* tries to finds a *VPN tree* that has abundant residual bandwidth and only requires a small amount of bandwidth to be allocated to the tree links. As a result, *COVPA* can look after both bandwidth allocation efficiency and load balancing. The pseudo code of *COVPA* is described below

Input: Tree Storage
Output: Cost of COVPA.

1. Two bandwidths are Link Bandwidth and Residual Bandwidth are used.

2. Calculate the normal tree for all VPN routers are connected, which is calculated using BFS using pruning
   cost in Tv to get normal VPN tree, where all paths are satisfied for the VPN request.

3. Compute RS to find the minimum path in PTv (several VPN) select one.

4. Find the cost of COVPA using the formula $\text{Cost}_{\text{COVPA}}(T)$

Figure2: Pseudocode for COVPA

Assume a network graph *G* consisting of *n* nodes. To process a VPN setup request vr_i, C*OVPA* iterates totally *n*

times, once for each $v \in N$. In each iteration, C*OVPA* first finds a candidate *VPN tree* PT_v rooted at v for vr_i, and then computes the amount of bandwidth needed to be allocated to each link l_x of PT_v. Finally the cost value associated with PT_v can be computed. After finding all PT_v ($v \in N$), if there do not exist any PT_v ($v \in N$) on which all links have enough residual bandwidth for allocation, C*OVPA* will reject vr_i. In the case of accepting vr_i, C*OVPA* will return the *VPN tree* with the minimum cost value among all *PTv* ($v \in N$) for vr_i which is denoted by VT_{MC}. In addition, C*OVPA* then allocates bandwidth to each link *lx* of VT_{MC} by performing $B(l_x) = B(l_x)\text{-}RS(l_x)$. To find a candidate *VPN tree PTv* rooted at *v*, C*OVPA* first find a *BFS tree* (breadth first search tree) *Tv* rooted at *v* (by calling Function *BFS_Tree*). *Tv* contains all nodes in *G* and in addition, *Tv* may contain nodes which are not VPN access routers used in vr_i as leaf nodes. Therefore, C*OVPA* prunes *Tv* and obtained a candidate *VPN tree PTv*, on which all leave nodes are VPN access routers used in vr_i (by calling Function *Prune_Tree*). C*OVPA* computes the amount of bandwidth needed for each link *lx* of a *VPN tree T* according to the bandwidth requirement information in vr_i (by calling Function *Compute_RS* in figure 3). To compute the value of $RS(lx)$ ($lx \in T$), we first remove *lx* from *T* and this will partition the *VPN tree* into two sub trees T_x^a and T_x^b. Let $BR\_\ T_x^a$ and $BR\_T_x^b$ denote the accumulated bandwidth requirement of the VPN access routers (endpoints) on T_x^a and T_x^b, respectively. Then $RS(lx)$ is determined by the minimum value of $BR\_\ T_x^a$ and $BR\_T_x^b$. Given a *VPN tree T*, in a normal case, the function *Cost* of C*OVPA* returns the cost value computed by the cost function defined previously. However, where *T* is null (Ø), or there are links on *T* which do not have enough bandwidth for allocation, the function Cost will return ∞.

Function *Compute_RS*(T, vr_i)
Let l_x be the xth link on T.
Let $RS(l_x)$ be the amount of bandwidth needed on l_x with respect to the bandwidth requirement specified in vr_i.
Let T_x^a and T_x^b be the two subtrees obtained by remove l_x from T.

```
1.  for (each l_x in T)
2.  {
3.      Initialize two variable BR_T_x^a, BR_T_x^b to value 0;
4.      For (each element r_j≠0 (1≤j≤p) of vr_i)
5.      {
6.          if(ar_j∈T_x^a) then add r_j to BR_T_x^a
7.          else add r_j to BR_T_x^b
8.      }
9.      RS(l_x) =min(BR_T_x^a, BR_T_x^b);
10. }
```

Figure3: Pseudo code for compute_RS

5. COMPARIION AND ANALYSIS:

We conducted an study to measure the performance of our COVPA and BFS primal–dual algorithms, and compared them with the approach of using a Steiner tree to connect VPN endpoints. The major findings of our study can be summarized as follows.

• The primal–dual algorithm generates VPN trees with the smallest cost for a wide range of ingress/egress bandwidth ratios. It outperforms both the BFS and the Steiner tree algorithms for medium-to-large bandwidth ratios.

• For low ingress/egress bandwidth ratios, the primal–dual algorithms slightly outperform the COVPA algorithm. In many cases, they construct VPN trees that reserve half the bandwidth reserved by Steiner trees.

5.1 Experimental Results

(Designed and developed COVPA Vs BFS and Steiner Tree Routing)

We compared the provisioning cost (that is, the total bandwidth reserved on links of the VPN tree) and the running times of the algorithms for the symmetric as well as the asymmetric bandwidth models. In the study, we examined the effect of varying the following two parameters on provisioning cost: 1) network size 2) number of VPN nodes. Most of the plots in the following subsections were generated by running each experiment five times (with different random networks) and using the average of the cost/execution times for the five repetitions as the final result.

Simulation 1: (Network Size)

Figure.4 depicts the provisioning cost of the BFS and Steiner tree algorithms as the number of network nodes is increased from 10 to 100. VPN endpoints are assigned equal ingress/egress bandwidths and the number of VPN endpoints is set to 10% of the network size. The COVPA algorithm is provably optimal for the symmetric case. Further, unlike the Steiner tree and BFS algorithm which is oblivious to the bandwidths of endpoints, the BFS algorithm does take into account the bandwidth requirements for VPN endpoints. As a result, it outperforms Steiner tree algorithm by almost a factor of 2 for a wide range of node values.

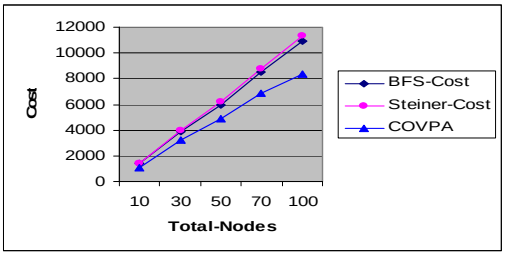


Figure 4 Effect of number of network nodes on performance of algorithms(symmetric)

No of Nodes	BFS-Cost	Steiner-Cost	COVPA-Cost
10	1413	1437	1114
30	3859	3985	3242
50	5997	6207	4876
70	8522	8810	6895
100	11370	11370	8345

Table1 Parameter configuration of Simulation 1

***Simulation 2* (Number of VPN Nodes)**

Similar results are obtained for the COVPA, BFS and Steiner tree approaches for a wide range of VPN node values (see Figure 5). In the experiment, the number of nodes in the network were fixed at 100 and VPN endpoints were assigned symmetric bandwidths.

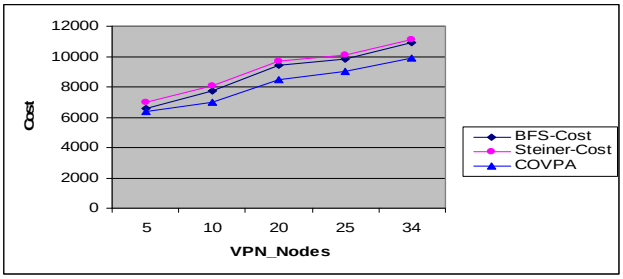


Figure 5 Effect of number of VPN nodes

Nodes	BFS-Cost	Steiner-Cost	COVPA-Cost
10	6550	6961	6345
30	7710	8070	6950
50	9398	9689	8450
70	9858	10122	9010
100	10920	11145	9910

Table 2 : Parameter configuration of Simulation 2

Simulation 3: (Rejection Ratio)

We conduct 15 runs with various number of topology, in each of which, 100 requests are randomly generated. The rejection ratios achieved by the three provisioning algorithms are (see Figure 6). The x-axis represents the run no. and the y-axis represents the *rejection ratio* and average link utilization achieved by each provisioning algorithm in each run. We can see that the rejection ratio achieved by C*OVPA* is much less than that achieved by *BFS* and optimal tree routing.

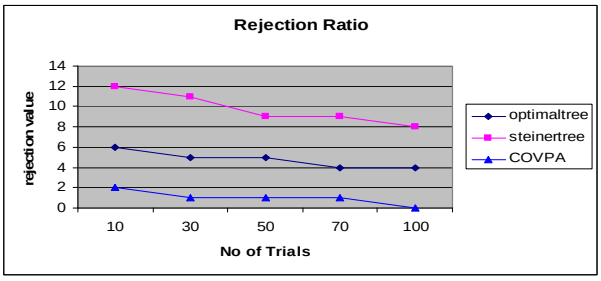


Figure 6 Effect of rejection ratio(symmetric)

No of Nodes	BFS-rejection Ratio	Steiner-rejection ratio	COVPA-rejection ratio
10	6	12	2
30	5	11	1
50	5	9	1
70	4	9	1
100	4	8	0

Table 3 Parameter configuration of Simulation 3

6. CONCLUSIONS

In this research work, the designed novel algorithms for provisioning VPNs in the hose model. COVPA connected VPN endpoints using a tree structure and attempted to optimize the total bandwidth reserved on edges of the VPN tree. The algorithm showed that even for the simple scenario in which network links are assumed to have infinite capacity, the general problem of computing the optimal VPN tree is NP-hard. However, for the special case when the ingress and egress bandwidths for each VPN endpoint are equal, COVPA proposed a breadth-first search algorithm for computing the optimal tree. According the simulation results *COVPA* can indeed reduce the rejection ratio effectively.

Acknowledgement

First of all we thank the almighty for giving us the knowledge and courage to complete the research work successfully. We express our gratitude to our respected Vice Chancellor Dr. Chelliah Thangaraj M.Tech., Ph.D for allowing us to do the research work internally. Also we acknowledge the support provided by TIFAC-CORE Network Engineering. (Department of Science and Technology, Government of India).

References

[1] H. Liang, O. Kabranov, D. Makrakis, and L. Orozco-Barbosa,(2002) "Minimal Cost Design of Virtual Private Networks," in *IEEE Proceedings of the CCECE '02*, 2002, vol.3, pp. 1610 – 1615.

[2] T. Erlebach, M. Ruegg (2004), "*Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing",* INFOCOM 2004., vol.4,pp.2275-2282
[3] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener (2002), "*Algorithms for Provisioning Virtual Private Networks in the Hose Model"*, IEEE/ACM Trans. on Networking, vol.10, issue 4, August 2002. pp. 565-578.
[4] A. Juttner, I. Szabo, and A. Szentesi (2003), "*On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks",* In Proc. INFOCOM 2003.vol 1, pp.386-395
[5] P. P. Mishra, H. Saran(2000), "*Capacity Management and Routing Policies for Voice over IP Traffic"*, *IEEE Network*, vol. 14, no. 2, pp. 20-27, March/April 2000. pp. 20-27.
[6] B. M. Waxman (1988), "*Routing of Multipoint Connection"*, IEEE Journal on Selected Areas in Communications, vol. 6, issue 9, December 1988. pp. 1617-1622.
[7] X. Yuan (1999), "*On the Extended Bellman-Ford Algorithm to Solve Two-constrained Quality of Service Routing Problems"*, in ICCN'99, pp.304-310
[8] Zhanfeng Jia and Pravin Varaiya (2006), "Heuristic Methods for Delay Constrained Least Cost Routing Using K-Shortest-Paths", IEEE TRANSACTIONS On Automatic control, vol. 51, no. 4, April 2006, pp.707-712

The Authors

Mr.R.Ravi is presently working as Assistant Professor & M.E (CSE) coordinator in department of Computer Science and Engineering at A.K.College of Engineering, Krishnankoil, Tamilnadu. He has completed his B.E Computer Science and Engineering from Thiagarajar College of Engineering, Madurai and M.E Computer Science and Engineering from Jadavpur University, Kolkata. Now he is a research scholar in Anna University. He has guided more than 25 M.E./B.E. projects. His field of interests are Network Engineering, Artificial Intelligent, Natural Language Processing, Network Security and Micro processors.

E-mailid:ravi@akce.ac.in,
rraviraj_thiru2003@yahoo.co.in
Phone:(College) 04563 – 289042- Extension
Mobile: 94424 18917.

Dr.S.Radhakrishnan is presently working as a Senior Professor and Head, Department of Computer Science and Engineering at A.K.College of Engineering, Krishnankoil, Tamilnadu. He has completed his M.Tech. and Ph.D. in Biomedical Engineering from Institute of Technology, Banaras Hindu University. He has guided more than 50 M.E./M.Tech. projects and 10 M.Phil. Thesis. Currently five candidates are working for Ph.D. under his guidance. His field of interests are Network Engineering, Computer Applications in medicine and evolutionary computing. He is also serving as Project Director(Network technologies) in TIFAC CORE in Network Engineering at A.K.College of Engineering. He has more than 10 publications to his credit.