Volume 3, No. 2, February 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

OPTIMIZED APPROACH FOR PROVISIONING VPN IN THE HOSE MODEL WITH QOS

S.M.Krishna Ganesh*[1], T.Ashok Kumar[2]
*Department of Computer Science and Engineering, St.Joseph University in Tanzania, Dar Es Salaam, Tanzania, EastAfrica
krishnaganeshsm@gmail.com[1]
[2] Department of Computer Science and Engineering, St.Joseph University in Tanzania, Dar Es Salaam, Tanzania, East Africa
ashoktvpkumar@gmail.com[2]

Abstract: Virtual private networks provide an encrypted connection between a user's distributed sites over a public network. Existing studies on quality of service deals with bandwidth in hose model. In this paper an Enhanced Hose model is used to specify the bandwidth and link utilization between the end points. We introduce a VPN tree algorithm (KDSVT) that is capable of computing all possible cost and link utilization of VPN Tree. Based on this algorithm we introduce a novel algorithm that can achieve a cost optimized and link utilization VPN tree (KCDVT) with low average computational complexity.

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] and [12]. 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 [3]. 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 [5].

ANALYSIS OF DATA

The Enhanced Hose Model:

We model the network as a graph G = (V, E) where V is the set of nodes and E is the set of bidirectional links connecting the nodes. Each link (i, j) is associated with two QoS metrics – the bandwidth capacity Lij and the delay Dij. The delay value of a path is defined as the sum of the delay values of all links along the path. The VPN specification in the hose model includes [7]: (1) A subset of the nodes $P \subseteq V$ corresponding to the VPN endpoints, and (2) for each node i ε P, the associated ingress and egress bandwidths Bi^{in} and Bi^{out} respectively. Note that the terms "ingress" and "egress" are taken with respect to the VPN endpoints. This model can be enhanced to include a delay requirement in two ways: (1) Associate a delay requirement Di with each node i, which specifies the maximum delay from this node to every other node in the VPN, or (2) Group applications that use the VPN into different delay classes characterized by their end-to-end delay requirements that must hold between every pair of end points. We adopt the latter approach in this paper.

Implementing Enhanced Hose Model:

We use |P| source-based trees to realize the hoses, one tree per hose. For a given source based tree T rooted at the VPN endpoint i, we denote by Tv the connected component of T containing node v when link (u, v) is deleted from the tree. In this case, the traffic passing through link (u, v) can only originate from i to the other endpoints in Tv. The traffic that i can send is bounded by Bi^{out}, and the traffic that Tv can receive cannot exceed $\sum B\ j^{in}$, j ε p ∩ TV. Thus the bandwidth reserved for link (u, v) of T is given by CT (u, v) = min (Bi^{out}, $B\sum B\ j^{in}$, j ε p ∩ TV}. Since we are interested in minimizing the total bandwidth reserved for tree T, the problem of computing the optimal source-based tree for endpoint i can be expressed as follows:

Optimal Delay-Constrained Source-Based Tree Problem: Given a set of VPN endpoints P with their associated ingress and egress bandwidths and the delay requirement D, compute a source-based tree T rooted at endpoint i whose leaves are the other VPN endpoints. The objective is to minimize CT while satisfying the delay requirement, delay $(i, j) \leq D$.

Theorem 1: The Optimal Delay-Constrained Source-Based Tree Problem is NP-hard.

RELATED WORKS

System Model and Problem Statement:

System Model:

We adopt some of the notations developed in [5]. A network is modeled as a graph $G = (V, E)$, where V is the set of nodes and E is the set of bidirectional links among the nodes in V. (i, j) and (j, i) are considered as two distinct links. Each link (i, j) is associated with capacity Lij. It is possible that $Lij \neq Lji$. In the hose model, each VPN specification consists of a set of VPN endpoints $P \subseteq V$ and the ingress and egress bandwidths of each of the VPN endpoints. Ingress bandwidth is the maximum amount of traffic a VPN endpoint would receive, while egress bandwidth is the maximum amount of traffic the VPN endpoint would send. For a node $i \varepsilon. P$, the hose ingress and egress bandwidths are both Bi, since we consider the case of symmetric ingress and egress bandwidths only.

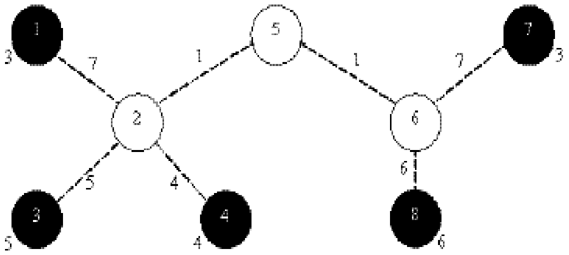

Figure 1: Example network

We use a tree to connect VPN endpoints. Formally, a tree $T = (V_T, E_T)$ is a sub graph of G where $P \subseteq V_T \subseteq V$ and $E_T \subseteq E$. Enough bandwidth has to be reserved on the links of the tree to support the VPN. To facilitate our discussion, we define $T_i^{(i,j)}$ and $T_j^{(i,j)}$, meaning the connected components of T containing nodes I and j respectively after removing (i, j) from T. Refer to the tree in Figure 1 in which dark nodes represent VPN endpoints and light nodes represent other network nodes, $T_2^{(2,5)}$ stands for the connected component consists of nodes 1 to 4 while $T_5^{(2,5)}$ stands for the connected component that is made up of nodes 5 to 8. We denote the set of VPN endpoints on $Ti^{(i,j)}$ and $T_j^{(i,j)}$ as $P_i^{(i,j)}$ and $P_j^{(i,j)}$ respectively. For example, in the tree in Figure 1, $P_2^{(2,5)} = \{1, 3, 4\}$ and $P_5^{(2,5)} = \{7, 8\}$.

We now explain how much bandwidth is needed to be reserved on link (i, j) on T. Link (i, j) has to support the traffic going from $T_i^{(i,j)}$ to $T_j^{(i,j)}$. In other words, it should support the traffic from VPN endpoint a to VPN endpoint b for each a $\sum P_i^{(i,j)}$ and for each b$\sum P j^{(i,j)}$. The maximum amount of traffic that would go through link (i, j) is min $\{\sum_{a \varepsilon P_i^{(i,j)}} B_a,, \sum_{b \varepsilon P j^{(i,j)}} B_b\}$ and this is the bandwidth needed to be reserved on (i, j). We denote this value as CT (i, j). As the ingress and egress bandwidths are symmetric, CT (i, j) = CT (j, i). Refer to the tree in Figure 1 in which the number next to each VPN endpoint represents its ingress or egress bandwidth (B1 = 3,B3 = 5,B4 = 4,B7 = 3,B8 = 6), to find CT (5, 6) = CT (6, 5), we first remove the edge (5, 6) from the tree and then compare the following two values: sum of ingress or egress bandwidths of VPN endpoints on the left of node 5 (3 + 5 + 4 = 12) and sum of ingress or egress bandwidths of VPN endpoints on the right of node 6 (3 + 6 = 9) and pick up the smaller of the two (9). We define the utilization of link (i, j), HT (i, j), to be CT (i, j) / Lij. Refer to the tree in Figure 1 where all edges are of capacity 10, HT (5, 6) = 9/10 = 0.9. We further define HT to be the utilization of the most utilized link on the tree, that is, HT $=\max\{HT (i, j)|(i, j) \varepsilon. E_T\}$.

PROBLEM STATEMENT

We now formally define the VPN routing problem. Given a graph G, a set of VPN endpoints P, and B_i for each $i \varepsilon P$. Compute a VPN tree T that connects all nodes in P with the following properties:

a. HT $(i, j) \leq 1$
b. CT is mini mum among all possible trees.
c. Tree also satisfies the delay requirement.

The first property simply says that the capacity constraint should not be violated. This property makes the problem NP-hard [5]. The second property means that we would like to find an optimal tree that requires the least amount of bandwidth among all possible trees.

PROPOSED METHOD

Algorithm:

Input: Network as graph
Output: Least-delay Least-cost optimized
VPN-Tree
[V-number of VPN nodes
k- Number of shortest path to find between each pair]

Step 1:
For each pair of vertices find the k-Shortest delay path if available. Totally v-1 set of shortest path available
i.e.) {k1} {k2} …{kv-1} set of shortest path and each set having k-paths.

Step 2:
Generate a new tree by taking union of paths taking one from each set of shortest paths. It produces k1 x K2 x… kv-1 number of induced trees. It may contain cycle.

```
Paths = Ø
Path_set = Ø
For each node s ε N
   For each node d ε N – {s}
        Find the shortest path SP between s and d with Dijkstra algorithm
        Path_set = Path_set U {SP}
        n = 1
        While (n < K) and (Path_set ≠ Ø)
                Take the first path p of Path_set
                Path_set = Path_set – {p}
                Paths = Paths U {p}
```

```
                Search L(p)
                While (L(p) ≠ Ø) and (n < K)
                        Take the less-cost link l of L(p)
                        L(p) = L(p) – {l}
                        Remove the link l of the network and search the
                        new shortest path SP' between s and d (Dijkstra)
                        If SP' is found
                                Path_set = Path_set U {SP'}
                                n = (n +1)
                        End If
                        Reinsert l in the network
                End While
        End While
   End For
```

Step 3:
For each tree perform the cycle detection algorithm to detect the cycle. If cycle is found delete the tree from the list.

Step 4:
Compute the cost required for each tree.

Step 5:
The tree with the low cost is selected as the Least-delay Least-cost optimized VPN tree.

Step 6: End

KDSVT Algorithm

K-shortest path algorithm (KSP)

Input: source node (s) and destination node (d)

Output: k-shortest path between node s and t

End For

Figure 2: K-shortest path algorithm

KDSVT algorithm computes the K-delay satisfied VPN tree. The algorithm finds a k-delay satisfied path between source vertexes, which is a VPN endpoint, to all other VPN Endpoints.

Consider there are N VPN Endpoints in the network, KSDVT algorithm find N k-delay satisfied paths. Then a tree is constructed by taking one path from each set of paths. If cycle is detected it may be removed by using spanning tree algorithm. In total it generates k^N VPN Tree. It gives sufficiently large collection of VPN Tree that makes easier for selecting the VPN tree that optimize the cost in terms of bandwidth.

KCDVT Algorithm:

KCDVT algorithm applies the ingress-egress bandwidth measurement algorithm on all the VPN Trees generated by the KDSVT algorithm. The bandwidth measurement in hose model is as specified in the System model description. The tree which having low bandwidth requirement is selected as the final VPN tree that satisfies delay and bandwidth requirement. It can find a cost optimized and delay satisfied VPN tree (KCDVT) with low average computational complexity. Its computational complexity increases with k. We can minimize its computational complexity by adaptively minimizing k.

INTERMEDIATE RESULT

To measure how effective our KCDVT Algorithm is, we conduct simulations. We generate two different sizes of topology for testing. For each size, we generate 1000 random topologies based on the WAXMAN model. For each topology, |P| VPN endpoints are randomly picked up. Two different network topologies were used to compare the different provisioning algorithms. The Waxman topology consists of 50 nodes as shown in Figure 3. All the links are bidirectional. A subset of the nodes in the network acts as the ingress–egress pairs. In the Waxman topology, four ingress–egress pairs are considered, which are (0, 12), (4, 8), (3, 1), and (4, 14). The second topology based on the Barabasi method expands the topology by inserting additional links to increase the connectivity. This topology, called Barabasi topology (BA), consists of 50 nodes and is presented in Figure 4.

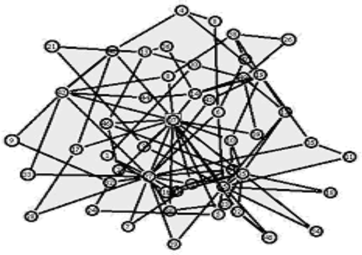

Figure 3: Waxman Model: Nodes-50

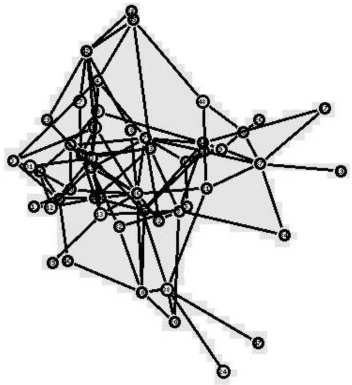

Figure 4: Barabasi Model: Nodes-50

PERFORMANCE EVALUATION

The cost and link utilization of the VPN tree that satisfies the bandwidth generated by the KCDVT algorithm. The cost of the KCDVT is less than BFS algorithm and the link utilization of KCDVT is more than BFS algorithm. The cost is increasing in proportion with the number of nodes.

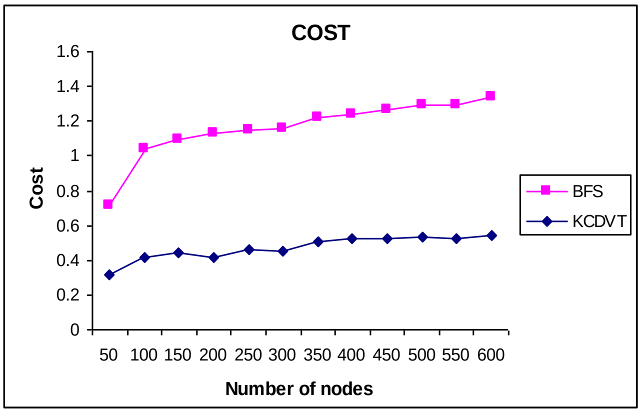

Figure 5.1 VPN cost

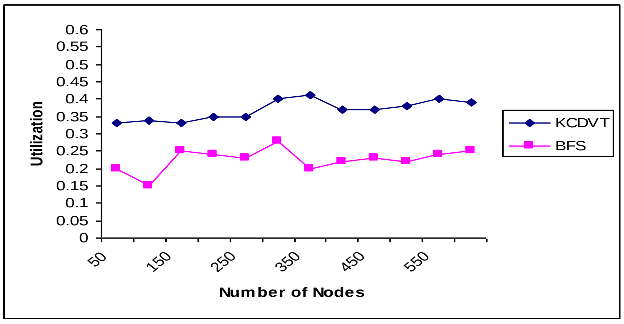

Figure: 5.2 VPN link utilization

CONCLUSION

In this paper, novel algorithms for provisioning VPNs in the hose model are designed. KCDVT connected VPN endpoints using a tree structure and attempted to optimize the total bandwidth reserved on edges of the VPN tree that satisfies the delay requirement. 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, KCDVT proposed algorithm for computing the optimal tree. In future there are still a number of issues relating to hose-model VPNs. For example: (1) the problem of fitting failure restoration mechanisms into KCDVT (2) Provisioning for the Asymmetric VPNs.

ACKNOWLEDGMENT

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 Rev.Fr.Dr.Arulraj Founder, DMI Group of institutions, East Africa and India, Dr.T.X.A.Ananth, Director, DMI group of institutions, East Africa and India,Mr.Ignatius Herman, Director(Academic),DMI group of institutions, East Africa and India and Dr.V.KrishnanPh.D,Principal,DMI.St.Joseph College of Engg & Technology, Tanzania for allowing us to do the research work internally. Also we acknowledge the support provided by Rev.Sr.Fatima Mary, Vice Principal (Administration), DMI.St.Joseph College of Engg & Technology, Tanzania and Mr.N.Ressel Raj,Vice Principal (Academic), DMI.St.Joseph College of Engg & Technology,Tanzania.We thank our friends and collegues for their support and encouragement

REFERENCES

[1]. H. Liang, O. Kabranov, D. Makrakis, and L. Orozco-Barb's,(2002), MinimalCost 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]. N. G. Duffield, P. Royal, A. Greenberg, P. Mishra, K. K.Ramakrishnan, and J. E. van der Merwe (1999), A Flexible Model for Resource Management in Virtual Private Networks, In Proc. ACM SIGCOMM, 1999. pp. 95-108.

[4]. A. Gupta, A. Kumar, J. Kleinberg, R. Rastogi, and B. Yener (2001), Provisioning a Virtual Private Network: A Network Design Problem for Multicommodity Flow, In Proc. ACM STOC, 2001. pp. 389-398.

[5]. 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.

[6]. 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

[7]. I. Matta, and L. Guo (1999), QDMR: An Efficient QoS Dependent Multicast Routing Algorithm, In Journal of Communications and Networks, Real-time Technology and Applications Symposium, 1999. pp. 213-222.

[8]. 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.

[9]. B. M. Waxman (1988), "Routing of Multipoint Connection", IEEE Journal on Selected Areas in Communications, vol. 6, issue 9, December 1988. pp. 1617-1622.

[10]. X. Yuan (1999), On the Extended Bellman-Ford Algorithm to Solve Two-constrained Quality of Service Routing Problems, in ICCN'99, pp.304-310

[11]. 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

[12]. Li-Der Chout Mao Yuan Hong (2006), Design and Implementation of Two Level VPN Service Provisioning Systems over MPLS Networks, IEEE International Symposium on Computer Networks, 2006, pp 42-48

Short Biodata Of All The Author

Mr. S.M.Krishna Ganesh has completed Masters of Technology Degreein Computer Science and Engineering at Kalasalingam University in the year 2009, Tamil Nadu,India.He is currently working as Lecturer in St. Joseph College of engineering and Technology,Dar Es Salaam, Tanzania, East Africa. He has guided more than 20 projects to final year B.E/B.Tech students and good industry and teaching experience. He has 5 publications to his credit. His areas of interests are Image Processing, Computer Networks, Neural networks and Bioinformatics

Mr. T.Ashok Kumar has completed Masters of Engineering Degree in Computer Science and Engineering at P.S.R. Engineering College under Anna University, Tirunelveli, in the year 2011, Tamil Nadu, India. He is currently working as Lecturer in St. Joseph College of engineering and Technology, Dar Es Salaam, Tanzania, East Africa. He has guided more than 5 projects to final year B.E/B.Tech students and good teaching experience. His areas of interests are Computer Networks, VANET.