Appl. Math. Inf. Sci. 6 No. 2S pp. 539S-546S (2012)

Applied Mathematics & Information Sciences
An International Journal

A Traffic Engineering Algorithm for Provisioning Virtual Private Networks in the Enhanced Hose Model

Yu-Liang Liu[1] and Der-Jiunn Deng[2]
[1] Department of Computer Science and Information Engineering, Aletheia University, Taiwan
[2] Department of Computer Science and Information Engineering, National Changhua University of Education, Taiwan
Corresponding author: Email: au4377@au.edu.tw

Received July 1, 2011; Revised September 7, 2011; Accepted 15 September 2011
Published online: 1 January 2012

Abstract: A Virtual Private Network is a logical network established on top of a public packet switched network. To guarantee that quality of service requirements, specified by customers, can be met, the network service provider needs to reserve enough resources on the network and allocate/manage them in an optimal way. Traffic engineering algorithms can be used by the Network Service Provider to establish multiple Virtual Private Networks in an optimal way, while meeting customers' Quality of Service requirements. For delay sensitive network applications, it is critical to meet both bandwidth and delay requirements. In contrast to traditional Virtual Private Network Quality of Service models (customer-pipe model and hose model), which focused only on bandwidth requirements, a new model called the enhanced hose model has been proposed, which considers both bandwidth and delay requirements. However, to the best of our knowledge, thus far, traffic engineering problems associated with establishing multiple enhanced hose model Virtual Private Networks have not been investigated. In this paper, we proposed a novel Virtual Private Network traffic engineering algorithm, called the minimum bandwidth-delay cost tree algorithm to address these problems. According to experimental simulations conducted and reported in our paper, the minimum bandwidth-delay cost tree algorithm can indeed achieved better performance (lower rejection ratios) compared to previous algorithms.

Keywords: VPN, QoS, Enhanced Hose Model, Bandwidth and Delay Requirements, Traffic Engineering

1. Introduction

The main feature of a Virtual Private Network (VPN) is the use of public networks to provide services comparable to those of a private network. Traditionally, a private network is established by grouping dedicated leased lines, as the number of endpoints grows, connecting them with dedicated lines becomes increasingly expensive [1]. As a result, VPNs have emerged as a replacement for private networks in recent years.

Recent research issues of interest in the field of telecommunication networks have dealt with introducing a quality of service (QoS) guarantee to VPN services, which is achieved by QoS specifications and resource reservation. QoS specifications are provided by VPN customers, and are often implemented via Service Level Agreements (SLAs). The two most common QoS models for VPNs are: (1) the customer-pipe model [2] and (2) the hose model [3, 4].

Hose model is more flexible for customers in terms of specifying their QoS requirements, compared to customer-pipe model [5]. As such they have inspired some research work around the development of provisioning algorithms to establish a single hose model VPN [5-9]. However, both the customer-pipe and hose models focus only on bandwidth requirements. For delay sensitive applications, such as VoIP [10] and voice/video conferencing [11], meeting both bandwidth and delay requirements are critical. To address this

problem, a new QoS model for VPNs, called the enhanced hose model, has been proposed, which considers both bandwidth and delay requirements while retaining the flexibility of the hose model [12].

In terms of provisioning VPN service, traffic engineering algorithms can be used by a network service provider (NSP), like Chunghwa Telecom, to establish multiple VPNs in an optimal way, while simultaneously meeting customers' QoS requirements. Recently, traffic engineering problems associated with establishing multiple VPNs, maintaining bandwidth guarantees, have been addressed in Ref. [13-16]. However, to the best of our knowledge, until now, traffic engineering problems associated with establishing multiple VPNs, maintaining both bandwidth and delay guarantees, have not been investigated.

In this paper, we review the QoS specifications of the enhanced hose model, and the provisioning of a single enhanced hose model VPN. The online enhanced hose model VPN establishment problem (OEHMVEP) is defined, in which the NSP establishes multiple enhanced hose model VPNs online, on a network backbone. A traffic engineering algorithm, called the minimum bandwidth-delay cost tree algorithm (MBDCTA), is also proposed to address the OEHMVEP. Experimental simulations are reported that compare the performance of the MBDCTA and previous algorithms.

2. The Enhanced Hose Model

The QoS specifications of the enhanced hose model are divided into specifications of bandwidth requirements and delay requirements [12]. The bandwidth requirement specification of this model is the same as that of the hose model (interested readers please refer to Ref. [3, 4]). In terms of delay requirements, VPN customers measure the characteristics of typical applications over the VPN and categorize them into L delay classes: $0 < d_1 < d_2 < \ldots < d_L$. Note that each delay class, d_j ($1 \leqq j \leqq L$) is characterized by its end-to-end delay requirements. Clearly, in this setting, establishing an enhanced hose model VPN is equivalent to establishing a total of L sub-VPNs, each with delay requirement d_j ($1 \leqq j \leqq L$). In the following, for clarity of presentation, we only need to exemplify the establishment of one specific delay class, with its given delay requirement, d.

Figure 1(a) shows a network backbone, on which an enhanced hose model VPN is to be established with 4 endpoints, labeled P_1, P_2, P_3, and P_4. We assume that the delay requirement of the enhanced hose model VPN is d=55. The number beside each endpoint represents its bandwidth requirement. For clarity, we assume that the bandwidth requirements for P_1, P_2, P_3, and P_4 are symmetric [5], that is, for each endpoint in the VPN, the ingress and egress bandwidth requirements are equal, being 8, 7, 5 and 6, respectively. The nodes labeled $N_1 \sim N_6$ are routers ($N_1 \sim N_4$ are VPN access routers) in the network backbone. The solid lines represent the backbone links, and the number beside each backbone link represents the delay incurred when packets are transmitted through it.

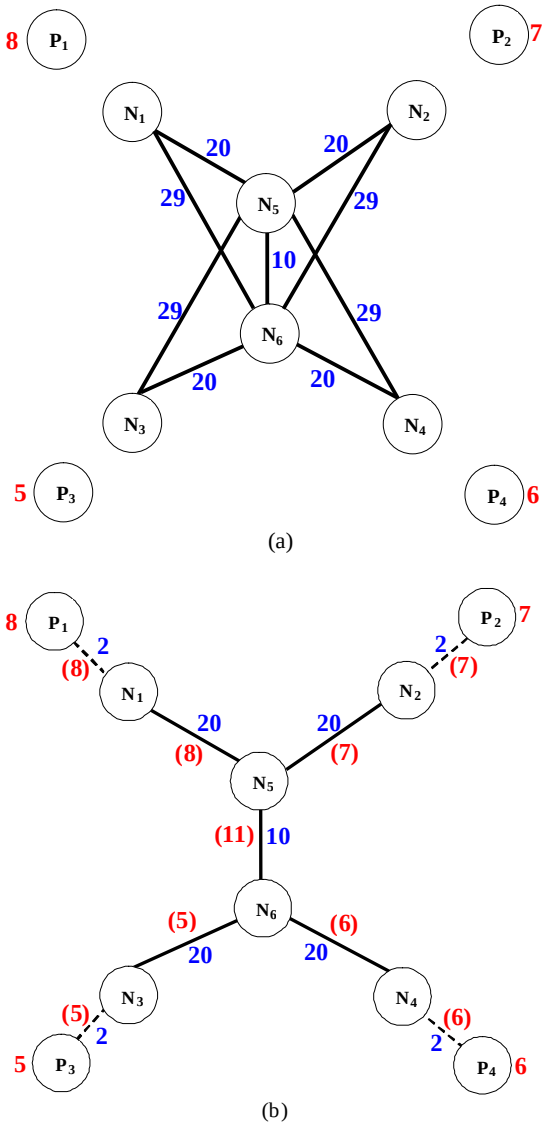

Figure 1. An example of establishing an enhanced hose model VPN.

In Figure 1(b), we show a VPN tree [5] corresponding to an enhanced hose model VPN that meets both the bandwidth and delay requirements. The dotted line between endpoint p_j and VPN access router N_j ($1 \leqq j \leqq 4$) represents the access link. In this example, we assume that the delay incurred when packets transmitted through each access link is 2. The number in parenthesis beside each backbone/access link is the bandwidth reservation required of it. Given a VPN tree and the bandwidth requirements on each endpoint, for the bandwidth reservation required on links of the VPN tree, please refer to Ref. [5]. The end-to-end delay value on the path between each endpoint pair of the VPN is listed in Table 1. Clearly, the maximum end-to-end delay value on the path between any endpoint pair in the VPN tree is 54, and can thus satisfy a delay requirement of d=55.

Table 1. The end-to-end delay value on the path between each endpoint pair of the VPN tree in Figure 1 (b).

Source / destination	P_1	P_2	P_3	P_4
P_1	0	44	54	54
P_2	44	0	54	54
P_3	54	54	0	44
P_4	54	54	40	0

Note that, given a VPN tree, T, there is a unique path between any endpoint pair in T, and the end-to-end delay value of a path is defined as the sum of the delay values of all links along the path. In this paper, we define the diameter of T, $dia(T)$, as the maximum end-to-end delay value on the path between any endpoints pair in T. We also define T such that it can satisfy the delay requirement, d, if $dia(T) \leqq d$.

3. The Traffic Engineering Problem Considered

In this section, we formulate the traffic engineering problem, called the online enhanced hose model VPN establishment problem (OEHMVEP), which is considered in this paper.

3.1 Backbone Network Modeling

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 and D be the set of residual bandwidths and link delay values for L, respectively. Let $B(l)$ and $D(l)$ denote the amount of residual bandwidth and delay value, respectively, on a given link l ($l \in L$). A subset, $AR = \{ar_1, ar_2, \ldots, ar_q\}$, of N ($AR \in N$) is the set of VPN access routers, where q denote the cardinality of AR. Each endpoint, p_j of a VPN, gains access to VPN service by connecting to a specific VPN access router, ar_j in AR. In other words, for each endpoint of a VPN, there is a corresponding VPN access router in AR and there are at most q endpoints contained in an enhanced hose model VPN.

3.2 VPN Setup Request Modeling

The customer demands for an enhanced hose model VPN service are described in terms of VPN setup requests. In this paper, we consider the case where the bandwidth requirements on endpoints are symmetric [5]. Let vr_i denote the ith VPN setup request, from customers, that the NSP is to establish. Each vr_i is represented by a (q+1)-tuple vector, $vr_i=(d, b(p_1), b(p_2),\ldots, b(p_q))$. The first element, d, of vr_i is the delay requirement, and the (j+1)-th element is the bandwidth requirement, $b(p_j)$, of endpoint p_j ($1 \leqq j \leqq q$). For example, the VPN setup request of the enhanced hose model shown in Figure 1 is vr_1=(55, 8, 7, 5, 6).

3.3 Online Enhanced Hose Model VPN Establishment Problem

In this problem, the NSP manages an MPLS network backbone, G, on which enhanced hose model VPNs are established. We consider the situation where (a) VPN setup requests arrive one-by-one, independently, and (b) the NSP does not have a priori knowledge about future VPN setup requests. This knowledge includes the number of future VPN setup requests, the delay requirement and the number of endpoints contained in each VPN setup request, and the bandwidth requirement of each endpoint. In this situation, the NSP must process each VPN setup request in an online manner. Upon receiving a VPN setup request vr_i, the NSP triggers the VPN traffic engineering algorithm to establish a corresponding enhanced hose model VPN. The VPN traffic engineering algorithm performs this task by first choosing a data transmission path between each endpoint pair, and then checking whether both the bandwidth and delay requirements are satisfied. If either of the bandwidth or delay requirements is not satisfied, vr_i will be rejected. We use the rejection ratio as the

performance metrics to evaluate VPN traffic engineering algorithms. The rejection ratio is defined as:

$$\frac{Number\ \ of\ \ requests\ \ rejected}{Total\ \ number\ \ of\ \ requests\ \ received} \quad (1)$$

In this paper, the optimization goal of VPN traffic engineering algorithms is to minimize the rejection ratio, which in turn, will maximize the number of requests successfully established on the network backbone. Note that the authors in Ref. [17-19] also use the rejection ratio as the main performance metric to compare traffic engineering algorithms.

In this problem, we assume that the NSP uses a server-based strategy [20] for processing VPN setup requests. In a server-based strategy, the VPN traffic engineering algorithm is run on a single entity called the VPN request server (VRS). The VRS also maintains the complete link state topology database and is responsible for computing an explicit data transmission path between each endpoint pair of a VPN. The paths can then be setup using a signaling protocol, such as the Resource Reservation Protocol (RSVP) or Constraint-based Routing Distribution Protocol (CR-LDP) [2]. To compute the explicit paths, the VRS needs to know the current network topology, residual bandwidth and delay value on links. We assume that a link state routing protocol for information acquisition exists.

4. The Minimum Bandwidth-Delay Cost Tree Algorithm

In this section, we propose a novel VPN traffic engineering algorithm called the MBDCTA, for the OEHMVEP. The inputs to the MBDCTA are a network graph, *G*, VPN access router set, *AR*, residual bandwidth, *B(l)*, and delay value, *D(l)*, for link *l* ($l \in L$), and a VPN setup request, vr_i. The output of the MBDCTA is a minimum cost VPN tree, VTMC, for vr_i, on which all leaf nodes are VPN access routers in the set *AR*. In Table 2, we provide pseudo code for the MBDCTA. The main idea of the MBDCTA is inspired by the algorithms proposed in Ref. [5, 17].

Let *T* be a VPN tree consisting of *k* links. The *Cost* subroutine of the MBDCTA for VPN tree selection calculates the following:

$$Cost(T) = \sum_{1 \le x \le k} \frac{RS(l_x)}{B(l_x) \times D(l_x)} \quad (2)$$

where $RS(l_x)$, $B(l_x)$ and $D(l_x)$ represent the amount of bandwidth allocation needed, the amount of residual bandwidth and the delay value on the *x*th link, l_x , respectively. Note that, given a VPN tree, *T*, normally, the *Cost* subroutine returns the cost value computed by Eq. (2). However, where *T* is null (Ø) or *T* does not satisfy the delay requirement, *d* ($dia(T)>d$), or there are links on *T* that do not have enough bandwidth for allocation, the *Cost* subroutine will return ∞. Note that, given a VPN tree, *T*, and a VPN setup request, vr_i, the values of $dia(T)$ and $RS(l_x)$ can be computed easily.

Table 2. Pseudo code for MBDCTA.

Minimum Bandwidth-Delay Cost Tree Algorithm (*MBDCTA*)

Input:
1. A Network graph $G=(N,L)$.
2. A VPN access router set $AR = \{ar_1, ar_2, \ldots, ar_q\}$.
3. Residual bandwidth $B(l)$ and delay value $D(l)$, $(l \in L)$.
4. A VPN setup request $vr_i=(d, b(p_1), b(p_2), \ldots, b(p_q))$.

Output:
A minimum cost VPN tree VT_{MC} for vr_i, on which all leaf nodes are nodes in *AR*.

Algorithm:

```
VT_MC :=Ø;
For each v∈N
{
  T_v:= BFS_Tree(G, v);
  PT_v:=Prune_Tree(T_v, vr_i);
  Compute_ dia_RS(PT_v, vr_i);
  if (Cost(PT_v)<Cost(VT_MC) )
      VT_MC:= PT_v;
}
if (Cost(VT_MC) = ∞)
{
      Reject(vr_i);
      Return Ø;
}
else{
      For each link l_x∈VT_MC
        {
          B(l_x):= B(l_x)-RS(l_x);
        }
      Accept(vr_i);
      Return(VT_MC);
      }
```

When processing a request, the MBDCTA tries to find a VPN tree that minimizes the cost subroutine defined in Eq. (2). It is clear that the additional cost of using a link, l_x, in building a VPN tree is proportional to the value of $RS(l_x)$ and is reciprocal to the value of $B(l_x)$ and $D(l_x)$. Therefore, the MBDCTA tries to find a VPN tree with links that have abundant residual bandwidth, larger delay value and lower overall bandwidth allocation.

Given a network graph, G, consisting of n nodes, to process a VPN setup request, vr_i, the MBDCTA iterates a total of n times, once for each $v \in N$. In each iteration, the MBDCTA first finds a candidate VPN tree, PT_v, rooted at v for vr_i, and then computes the bandwidth allocation that is needed for 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 is no PT_v ($v \in N$) that satisfies the delay requirement, d ($dia(PTv) \leqq d$), or if there is no PT_v ($v \in N$) on which all links have enough residual bandwidth for allocation, the MBDCTA will reject vr_i. When vr_i is accepted, the MBDCTA will return the VPN tree that has the minimum cost value among all PT_v ($v \in N$) for vr_i, which is denoted by VT_{MC}. In addition, the MBDCTA then allocates bandwidth to each link, l_x, of VT_{MC} as follows: $B(l_x) := B(l_x) - RS(l_x)$.

To find a candidate VPN tree, PT_v, rooted at v, the MBDCTA first finds a breadth first search tree [21], T_v, rooted at v, by calling subroutine *BFS_Tree*. T_v contains all nodes in G and, in addition, T_v may contain leaf nodes that are not VPN access routers. Therefore, the MBDCTA prunes T_v and obtains a candidate VPN tree, PT_v, on which all leaf nodes are VPN access routers, by calling subroutine *Prune_Tree*. The MBDCTA computes the diameter of PT_v, $dia(PT_v)$, and the amount of bandwidth needed, $RS(l_x)$, on each link, l_x, of PT_v, by calling subroutine Compute_ *dia_RS*.

The time complexity of each iteration in the MBDCTA is $O(m)$, which is determined by the subroutine *BFS_Tree*. To process a request, a total of n iterations are required. So, it is clear that the time complexity of the MBDCTA in processing a request is $O(mn)$.

5. Performance Evaluation

In this paper, we compare the performance of MBDCTA, bandwidth-optimization VPN tree algorithm (BOVTA) [5], and shared tree algorithm that combined the diameter constrained heuristic with the least-delay tree algorithm (DC-LDTA) [12]. These were implemented in Java programming language. Note that, all of the three algorithms are tree based, that is, given the four inputs as in Table 2, the output of these algorithms is a VPN tree. We have also conducted extensive simulations to compare the rejection ratio (defined in Eq. (1)) of the three algorithms on the OEHMVEP, introduced in section 3. Since much of the literature on network traffic engineering [17-19] adopts the KL topology as the backbone network, G, we also adopt it in our simulations. Note that the KL topology is a network graph comprised of 15 routers and 28 links, as shown in Figure 2. The routers numbered 1, 2, 4, 5, 9, 13 and 15 are selected as VPN access routers (nodes in the set AR), and are labeled as *ar1* - *ar7*, respectively.

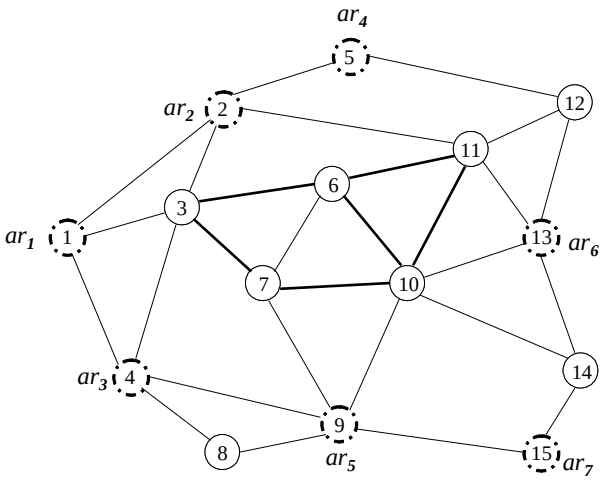

Figure 2. The KL topology.

In the simulations reported, we consider 3 cases. In the first case, we assume that the delay values on links of G are so small that delay requirements of all enhanced hose model VPNs are satisfied, and the sole reason for rejecting VPN setup requests is that there being some links on VPN tree have insufficient bandwidth for allocation. In the second case, we assume that the residual bandwidths on links in G are so abundant that bandwidth requirements of all enhanced hose model VPNs are satisfied, and the sole reason of rejecting VPN setup requests is that the VPN trees output by traffic engineering algorithms cannot satisfy the delay requirement. In the latest case, we consider general parameter configurations.

In each case, we conduct 10 simulation runs, the parameter configurations are listed in Table 3. Let

$B(l)$ and $D(l)$ denote the residual bandwidth and delay value on link l of G, and let d, q and Max_br denote the delay requirement, the maximum number of endpoints contained in a VPN setup request and maximum bandwidth requirement on VPN endpoints, respectively. In each simulation run, 100 VPN setup requests are randomly generated, as is the number of endpoints contained in a VPN setup request, being assigned a value between 2 and q. The bandwidth requirement of each endpoint is also randomly generated, between 1 and Max_br. For simulations of case 1, the delay requirement, d, in each VPN setup request, is fixed to 10, and for simulations of case 2 and 3, it is generated randomly from 10 to 30 and 10 to 20, respectively.

Table 3. The parameter configurations of simulations for case 1 to 3.

	$B(l)$	$D(l)$	Max_br	q	d
Case 1	1,700	1	75	7	10
Case 2	25,000	5	75	7	10~30
Case 3	2,000	4	75	7	10~20

The simulation results of cases 1 to 3 are shown in Figure 3 to Figure 5, respectively. In all simulations of the three cases, the MBDCTA achieved the lowest rejection ratios among the three algorithms. In Figure 3, the rejection ratios achieved by the MBDCTA, BOVTA and DC-LDTA ranged from 0% to 5%, 24% to 36% and 32% to 46%, respectively. Clearly the BOVTA performed better than the DC-LDTA in most of the simulations, because the bottleneck resources in case 1 are the bandwidths that are available for allocation. The DC-LDTA tries to find a least-delay VPN tree, vt, regardless of on the residual bandwidth and bandwidth allocation needed on links of vt. In Figure 4, the rejection ratios achieved by the MBDCTA, BOVTA and DC-LDTA, ranged from 3% to 10%, 32% to 44% and 8% to 18%, respectively. The DC-LDTA performed better than the BOVTA in all the simulations of case 2, because the BOVTA tries to find a bandwidth-optimization VPN tree, vt , regardless of the delay value on links of vt. Hence the BOVTA tends to fail in meeting the delay requirement. In Figure 5, the rejection ratios achieved by the MBDCTA, BOVTA, in case 3, ranged from 0% to 3%, 22% to 35% and 21% to 33%, respectively.

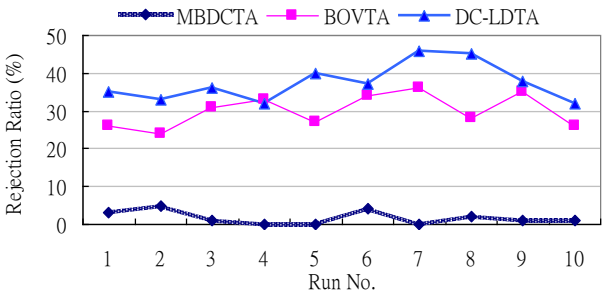

Figure 3. Simulation results of case 1.

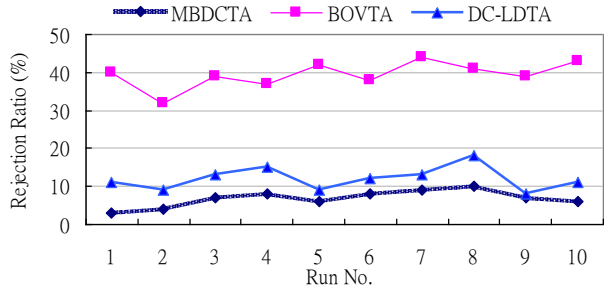

Figure 4. Simulation results of case 2.

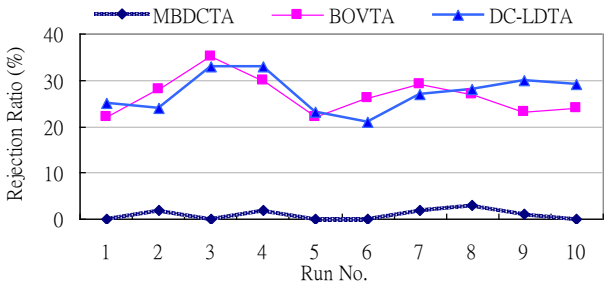

Figure 5. Simulation results of case 3.

6. Conclusions

In this paper, we investigated a VPN traffic engineering problem called the OEHMVEP, in which multiple enhanced hose model VPNs are to be established. We first reviewed the QoS specification of the enhanced hose model, and then proposed a new algorithm, called the minimum bandwidth-delay cost tree algorithm (MBDCTA), to address issues associated with OEHMVEP. The optimization goal of VPN traffic engineering algorithms considered in our work is to minimize rejection ratio. According to experimental simulations conducted in our paper, the MBDCTA can indeed achieve lower rejection ratios compared to previous algorithms. For the future research, it is worthwhile to extend VPN traffic engineering algorithms on asymmetric network backbone, and considers another optimization goal, such as, maximizing revenue of VPN service provisioning.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments that greatly helped improve the quality of this paper. This work was supported partially by National Science Council NSC 99-2221-E-156-005.

References

[1] R. Venkateswaran, Virtual Private Networks, *IEEE Potentials*, Vol. 20, Issue 1, (2001), 11-15.

[2] B.S. Davie and Y. Rekhter, MPLS technology and applications, Morgan Kaufmann, San Francisco, 2000.

[3] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan and J.E.V.D. Merwe, Resource Management with Hoses: Point-to-Cloud Services for Virtual Private Networks, *IEEE/ACM Transactions on Networking*, Vol. 10, No. 3, (2002), 679–692.

[4] S. Raghunath and K. K. Ramakrishnan, Resource Management for Virtual Private Networks, *IEEE Communications Magazine*, Vol. 45, Issue 4, (2007), 38-44.

[5] 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, (2002), 565-578.

[6] G.-S. Poo and H. Wang, Multi-path Routing versus Tree Routing for VPN Bandwidth Provisioning in the Hose Model, *Computer Networks*, Vol.51, Issue 6, (2007), 1725-1743.

[7] Juttner, I. Szabo and ´A Szentesi, On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks, in Proceedings of the 2003 IEEE Conference of the Computer and Communication (INFOCOM), 2003.

[8] T. Erlebach and M. Rürich, Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing, in Proceedings of the 2004 IEEE Conference of the Computer and Communication (INFOCOM), 2004.

[9] E. Oki and A. Iwaki, Load-Balanced IP Routing Scheme Based on Shortest Paths in Hose Model, *IEEE Transactions on Communications*, Vol. 58, Issue 7, 2088-2096, (2010).

[10] P. P. M and H. S, Capacity Management and Routing Policies for Voice over IP Traffic, *IEEE Network*, Vol. 14, No. 2, 20-27, (2000).

[11] S. Firestone, T. Ramalingam and S. Fry, Voice and Video Conferencing Fundamentals, Cisco Press, Indianapolis, 2007.

[12] L. Zhang, J. Muppala and S. Chanson, Provisioning Virtual Private Networks in the Hose Model with Delay Requirements. In Proceedings of the 2005 International Conference on Parallel Processing (ICPP), 2005.

[13] C. T. Chou, Traffic Engineering for MPLS-based Virtual Private Networks, *Computer Networks* Vol. 44, Issue 3, (2004), 319-333.

[14] M. Naraghi-Pour and V. Desai, Loop-free traffic engineering with path protection in MPLS VPNs, *Computer Networks*, Vol. 52, Issue 12, (2008), 2360-2372.

[15] J. Chu and C.-T. Lea, New Architecture and Algorithms for Fast Construction of Hose-Model VPNs, *IEEE/ACM Transactions on Networking*, Vol. 16, Issue 3, (2008), 670-679.

[16] J. Chu and C.-T. Lea, Optimal Link Weights for IP-Based Networks Supporting Hose-Model VPNs, *IEEE/ACM Transactions on Networking*, Vol. 17, Issue 3, (2009), 778-788.

[17] Y. Yang, L. Zhang, J. K. Muppala and S. T Chanson, Bandwidth-Delay Constrained Routing Algorithms, *Computer Networks*, Vol. 42, Issue 4, (2003), 503-520.

[18] K. Kar, M.Kodialam and T.V. Lakshman, Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications, *IEEE J. Selected Areas in Communications*, Vol. 18, No. 12, (2000), 2566-2579.

[19] S. Suri, M. Waldvogel and P. R. Warkhede, *Computer Communications*, Vol. 26, Issue 4, (2003), 351-365.

[20] G. Apostolopoulos, R. Gu´erin, S. Kamat and S. K. Tripathi, Server-Based QoS Routing, in Proceedings of the 1999 IEEE Global Telecommunications Conference (GLOBCOM), 1999.

[21] E. Horowitz, S. Sahni, D. Mehta, Fundamentals of data structures in C++, Computer Science Press, New York, 1995.

Yu-Liang Liu received B.S. and M.S. degrees from National Taiwan University of Science and Technology, Taiwan, in 1997 and 1999, respectively, and the Ph.D. degree from National Taiwan University in 2006, all major in information management. He joined Aletheia University as an assistant professor in the Department of Computer Science and Information Engineering in August 2007. Dr. Liu's research interests include QoS provisioning, resource management and traffic engineering on computer or telecommunication networks.

Der-Jiunn Deng received the Ph.D. degree in electrical engineering from National Taiwan University in 2005. He joined National Changhua University of Education as an assistant professor in the Department of Computer Science and Information Engineering in August 2005 and then became an associate professor in February 2009. His research interests include multimedia communication, quality-of-service, and wireless networks. In 2010, he received the Top Research Award of National Changhua University of Education. Dr. Deng served or is serving as an editor and guest editor for several technical journals. He also served or is serving on several symposium chairs and technical program committees for IEEE and other international conferences. Dr. Deng is a member of the IEEE.