This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Resource Optimization to Provision a Virtual Private Network Using the Hose Model

Monia Ghobadi, Sudhakar Ganti, Gholamali C. Shoja
University of Victoria, Victoria BC, Canada V8W 3P6
e-mail: *{monia, sganti, gshoja}@cs.uvic.ca*

***Abstract*— Virtual Private Networks (VPN) provide a secure and reliable communication between customer sites over a shared network. With increase in number and size of VPNs, providers need efficient provisioning techniques that adapt to customer demands. The recently proposed hose model for VPN alleviates the scalability problem of the pipe model by reserving for its aggregate ingress and egress bandwidth instead of between every pair of VPN endpoints. Existing studies on quality of service guarantees in the hose model either deal only with bandwidth requirements or regard the delay requirement as the main objective ignoring the bandwidth cost. In this work we propose a new approach to enhance the hose model to guarantee delay requirements between endpoints while optimizing the provisioning bandwidth cost. We connect VPN endpoints using a tree structure and our algorithm attempts to optimize the total bandwidth reserved on edges of the VPN tree. Our proposed approach takes into account the user preferences in meeting the delay requirements and provisioning cost to find the optimal solution of resource allocation problem. Our experimental results indicate that the VPN trees constructed by our proposed algorithm meet minimum delay requirements while reducing the bandwidth requirements as compared to previously proposed algorithms.**

I. INTRODUCTION

A Virtual Private Network (VPN) is a group of computer systems connected as a private network but communicates over a public network. The aim is to provide the VPN endpoints with a service comparable to a dedicated private network established with *leased lines*. Thus, providers of VPN services need to address the QoS and security issues while deploying a VPN over a shared IP network. In recent years, substantial progress in the IP security technologies have enabled existing VPN service offerings to provide customers with a level of privacy comparable to that offered by a dedicated line [5]. The emergence of IP technologies such as MPLS and RSVP-TE [1] have made it possible to realize IP-based VPNs that can provide the end customers with QoS guarantees. In the context of QoS guarantee in VPNs, only delay and aggregate bandwidth guarantee has been considered and we will consider the same in this work. In this paper, we address the problem of resource allocation in VPN hose model with QoS guarantees while optimizing total provisioning bandwidth cost.

Two popular models have been proposed for providing QoS in the context of VPNs: the "pipe" model [1] and the "hose" model [2]. As depicted in Fig. 1, in the pipe model, a VPN customer buys a set of customer-pipes, i.e., allocations of specific bandwidth on paths between every source-destination pair of the VPN endpoints. However, in the hose model, as illustrated in Fig. 2, each VPN endpoint connects to the network by a hose, which is specified by its aggregate ingress and egress bandwidth requirements. The hose model has desirable characteristics such as ease of specification, flexibility, and multiplexing gain [2]. A number of provisioning algorithms for VPNs in the hose model have been proposed [2, 3, 5, 6, 10]. In [2], a hose is realized with a source-based tree and a factor of two to three in capacity savings over the pipe model is achieved. The authors suggested that using a Steiner tree connecting VPN endpoints would optimize the total bandwidth provisioning cost. Further, in [5], it has been shown that optimal bandwidth allocation problem in VPN hose model is NP-complete. In their work, hoses are implemented with a single shared tree and the proposed algorithms attempt to optimize the total bandwidth reserved on the edges of the VPN tree. The bandwidth efficiency of the hose model is studied in [3] where the over-provisioning factor of the model is evaluated in networks with various sizes and node densities. In [7], a multi-path routing provisioning approach is proposed for the hose model.

A VPN network is modeled 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 (u, v) is associated with two QoS metrics: maximum bandwidth capacity over the link and delay between link endpoints. The delay value of a path is defined as the sum of the delay values of all the links along it. The VPN specification in the hose model includes: A subset of endpoints $P \subseteq V$ corresponding to the VPN endpoints; and for each VPN endpoint $i \in P$, the associated ingress and egress bandwidths B_i^{in} and B_i^{out}, respectively [5].

The nature of the Service Level Agreement (SLA) between a customer and a service provider is driven by the traffic characteristics and QoS requirements of the customer applications that make use of the VPN. For example, a voice-over-IP VPN service might require tight bounds on the packet loss rate, delay, and possibly jitter. On the other hand, a data-only VPN service might have relatively less stringent or no delay requirements. To ensure that appropriate requirements can be met, it is essential for the provider to know the traffic

1-4244-0353-7/07/$25.00 ©2007 IEEE

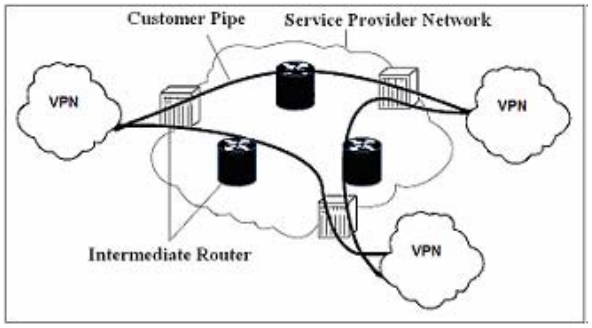

Fig. 1: VPN pipe model

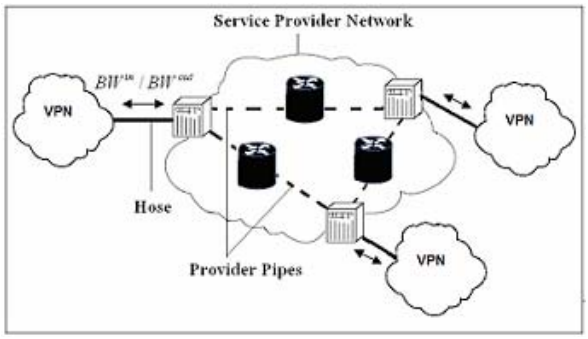

Fig. 2: VPN hose model

characteristics.

Although the hose model provides customers with simpler and more flexible SLAs, the model presents the provider with a more challenging problem of resource management. It is also difficult to provide QoS guarantee in the hose model since VPN customers specify QoS requirements per VPN endpoint and not for every pair of endpoints. The main problem of interest in this work is the problem of constructing a shared tree connecting all the VPN endpoints with the objective of satisfying the end-to-end delay requirements and guaranteeing the aggregate bandwidth while minimizing the provisioning bandwidth cost.

Compared to the pipe and source-based tree approaches, the shared tree approach makes the best use of statistical multiplexing to reduce the provisioning cost. Thus, we consider tree structures to connect the VPN endpoints in P since trees are scalable and simplify routing and restoration. Furthermore, trees allow the bandwidth reserved on a link to be shared by the traffic between the two sets of VPN endpoints connected by the link [5].

The rest of the paper is organized as follows. A short review on the previous works that this algorithm is based on is presented in section II. The Optimal Bandwidth Delay Shared Tree (OBDST) algorithm that meets the delay requirement while minimizing the provisioning cost is described in section III. Section IV presents the discussion on time complexity. Simulation results comparing the performance of the proposed algorithm are presented in section V. Finally, section VI concludes the paper.

II. Motivation and Related Work

The primal-dual algorithm for computing VPN tree was developed by Kumar et al. [5]. Their approach finds the near optimal bandwidth provisioning tree in which a 10-approximation is obtained by solving the linear program relaxation and rounding the fractional solution. This approach can find a tree with cost smaller than a Steiner tree suggested in [2]. Following the notation of Kumar's paper, for a link (u, v), let $P_u^{(u,v)}$/ $P_v^{(u,v)}$ denote the set of VPN endpoints in the connected component of tree T connecting nodes u/v when link (u,v) is deleted from T. In [5] it has been shown that the bandwidth to be reserved on link (u,v) of T is

$$C_T^{(u,v)} = \min\{\sum_{l \in P_u^{(u,v)}} B_l^{out}, \sum_{l \in P_v^{(u,v)}} B_l^{in}\} + \min\{\sum_{l \in P_v^{(u,v)}} B_l^{out}, \sum_{l \in P_u^{(u,v)}} B_l^{in}\}. \quad (1)$$

The total bandwidth reserved for tree T is given by $C_T = \sum_{(u,v) \in T} C_T^{(u,v)}$. For the purpose of this paper, this allocation of total bandwidth (C_T) is used as bandwidth cost or provisioning cost. Since we are interested in minimizing the reserved bandwidth for tree T, the problem of computing the optimal VPN tree becomes the following:

Optimal Bandwidth-constrained Shared Tree Problem (OBSTP): Given a set of *VPN* endpoints P and their ingress B_p^{in} and egress B_p^{out} bandwidths, compute a shared *VPN* tree T, connecting these nodes for which C_T is minimum. In [5] it is proved that the OBSTP is NP-hard.

Zhang et al. [8] suggested classifying applications that use VPN in order to solve the *Optimal Delay-constrained Shared Tree Problem (ODSTP).* Each class, j, is characterized by its end-to-end delay requirement, *maximum_allowable_delay(j)* that must hold between every pair of endpoints. ODSTP addresses the problem of finding a shared tree T connecting VPN endpoints that satisfies the delay requirement. In practice, the delay classes are obtained by measuring characteristics of typical applications over the VPN. The ODSTP Problem has also been proved to be NP-hard [8].

Zhang et al. [8], find near optimal delay constrained shared tree by formulating a Minimum Diameter Steiner Tree (MDStT) problem to make the shared tree support the delay requirements. The diameter of a Steiner tree connecting the VPN endpoints is defined as the maximum delay between any two endpoints. The minimum delay requirement that can be supported by the shared tree can be obtained by finding the minimum diameter Steiner tree.

The MDStT supports the lowest delay requirement using a tree structure. However to build a low provisioning cost tree, Zhang et al. [8] suggested a Least-Cost-Least-Delay (LCLD) approach which tries to reduce the provisioning bandwidth while maintaining the delay requirement. We found no previous proposed way to construct the shared tree connecting VPN endpoints ensuring that both provisioning cost and delay requirement would be minimized. LCLD algorithm [8] satisfies the delay requirement, but there is no guarantee that the provisioning cost is optimal.

In this paper, we use a combination of both MDStT [8] and primal-dual [5] algorithms to find the trees with minimum

provisioning cost that also satisfy the delay requirement.

For clarity of presentation, in the following sections, only the provisioning of one specific delay class with its given delay requirement and the associated ingress and egress bandwidths for each VPN endpoint is discussed. Note that the delay requirement is identical for all the VPN endpoints for a given class. Thus provisioning a VPN network can be viewed as provisioning each of the delay classes separately [8].

III. OUR CONTRIBUTIONS

This paper's main objective is to find a near optimal shared tree considering the delay requirement while trying to minimize the total provisioning bandwidth cost.

The problem can be formulated as follows: *Optimal Bandwidth and Delay-constrained Shared Tree Problem (OBDSTP)*: Given a set of VPN endpoints P with their associated ingress and egress bandwidths and the *maximum_allowable_delay*, compute a shared tree T connecting all the VPN endpoints satisfying the delay requirement in which the total provisioning bandwidth cost is minimized.

In this work, we develop algorithms to solve the OBDSTP ignoring link capacity constraints. Our proposed approach can be extended to handle link capacity constraints.

Our methodology is to find two sets of shared trees: one optimizing the delay requirement and the other optimizing the bandwidth cost. From these sets, we select the best solution regarding both delay and bandwidth requirements using user specified preference parameters. Our approach consists of four phases.

In phase 1, we use a modified version of MDStT algorithm [8] to construct shared tree(s) connecting all VPN endpoints with the objective of satisfying the given maximum allowable delay. In this step, similar to LCLD approach [8], we develop our MDStT algorithm based on absolute center problem [11]. In our approach, if more than one local 1-center points satisfy the delay requirement, in contrast to LCLD approach [8], all of them would be set as one of the candidates for center of the graph. For each center candidate, a Steiner tree connecting it to all VPN endpoints using the minimum delay path would be constructed. The set of constructed trees is called ODST set. This set of trees consists of all the shared trees which satisfy the delay requirement but might not minimize the provisioning bandwidth.

In phase 2, the total provisioning bandwidth of each tree in ODST set, using (1), is computed. The maximum bandwidth cost for trees in ODST set is chosen as the bandwidth threshold to be used in the next phase. This threshold is the maximum amount of provisioning bandwidth that we are interested in. In phase 3, we use a modified version of primal-dual algorithm [5], to construct shared tree(s) connecting all the VPN endpoints with the cost less than the bandwidth threshold. The set of constructed trees in this phase is called OBST set. This means that all the shared trees with total provisioning bandwidth less than the bandwidth threshold would be added to OBST set. Our approach to modify primal-dual [5] algorithm is that we save all the trees satisfying the bandwidth threshold requirement while primal-dual [5] algorithm only returns one tree with the smallest cost. In phase 4, a ranking scheme is used to rank the trees in ODST and OBST sets. The trees with smallest rank would be the best candidate for OBDSTP.

Ranking the trees is done according to user specified bandwidth/delay preference and maximum allowable delay of the particular service class. These parameters depend upon class-of-service of the application and user preference in meeting bandwidth/delay requirements.

We formulate the following scheme for choosing from the above sets of trees the ones that will be closest to satisfy user's bandwidth/delay requirements. For each shared tree T, the following value will be calculated:

$$\forall T \in ODST \vee OBST, Rank(T) = delay\_preference \times \frac{delay\_diameter_T}{maximum\_allowable\_delay} + bandwidth\_preference \times \frac{bandwidth\_cost_T}{bandwidth\_threshold}. \quad (2)$$

In the above formula, the *maximum_allowable_delay* is defined as the maximum allowable end-to-end delay dependent on the class-of-service. The $delay\_diameter_T$ is defined as maximum end-to-end delay between VPN endpoints of each tree T. The $bandwidth\_cost_T$ is defined as sum of provisioning bandwidth over links of T. As explained earlier, the *bandwidth_threshold* is the maximum bandwidth cost for trees in ODST set. The parameters *delay_preference* and *bandwidth_preference* are set by the user. The lowest ranked trees are candidates for providing near optimal solutions.

The justification to use this approach is that it provides needed flexibility with respect to user's preferences in choosing delay versus bandwidth requirement. For example, VoIP applications may use larger delay preference while a guaranteed data service application may use larger bandwidth preference.

As an example consider the graph depicted in Fig. 3. It contains a network with six nodes. Nodes 0, 1, 2, 3 are VPN endpoints with ingress/egress bandwidth equals to 5/5, 6/6, 7/7, 8/8 Mbps respectively. The numbers on each edge indicate the link's delay in milliseconds. Assume that the maximum allowable delay for a particular application is 58 ms.

The result of phase 1 is illustrated in Fig. 4 in which the two shared trees A and B belong to the ODST set and satisfy the maximum allowable delay requirement. Then in phase 2, we compute the *bandwidth_cost* of each tree and the *bandwidth_threshold* is set as the maximum *bandwidth_cost* over the trees in ODST. In this case the maximum *bandwidth_cost* is the cost of the tree labeled as A which is 74 Mbps. Now, in phase 3, we execute our modified primal-dual algorithm on the original network to find all the shared trees with total *bandwidth_cost* less than the *bandwidth_threshold* computed in phase 2. The modified primal-dual algorithm results in a set of trees labeled as C, D

and E in Fig. 5. Note that for the sake of clarity we have not shown the homogeneous trees in Fig. 4 and Fig. 5. In phase 4, the ranking is performed for each tree in ODST and OBST sets using (2). Assume that the *bandwidth_preference* and *delay_preference* are set equal to 1. The ranking value for each tree is as following:

Rank of tree A = 74/74 + 50/58 = 1.86
Rank of tree B = 52/74 + 58/58 = 1.70
Rank of tree C = 52/74 + 58/58 = 1.70
Rank of tree D = 62/74 + 59/58 = 1.85
Rank of tree E = 74/74 + 89/58 = 2.53

For this example, both trees B and C have the same minimum rank. Therefore, one can choose either one of them. However, executing the LCLD algorithm [8] on this network would result in selection of tree A which has a larger *bandwidth_cost*.

IV. COMPLEXITY ANALYSIS

As long as our algorithm uses both MDStT [8] and primal-dual [5] algorithms to find the best near optimal trees regarding both delay and bandwidth, its time complexity is a function of time complexities of MDStT and primal-dual algorithms. The former has time complexity equal to $O(mp+nplogp)$ and the latter has time complexity equal to $O(n(mp^2+mnp+n^2\log n))$ where m is number of edges, n is number of nodes and p is number of VPN endpoints in the network.

In our OBDST algorithm, during phase 1, all the shared trees constructed by setting each edge as the root might have delays less than the delay threshold in the worst case. This will not change the time complexity of finding the global center in the MDStT algorithm [8] but the complexity of tree construction will be affected. Thus the worst case time complexity will be $O(m^2p+nmplogp)$ for phase 1. However, proper selection of maximum allowable delay would eliminate this scenario. In our extensive study regarding the selection of maximum allowable delay, we found that setting this value to be a factor selected in interval [1, 1.5] times the minimum supportable delay of the network (the value of the global center) would be the best selection that reduces the time complexity to a constant factor of MDStT algorithm [8] regardless of number of edges. Thus the time complexity of phase 2 is $O(c)$. Execution of primal-dual algorithm [5] is independent of the bandwidth threshold since all the trees will be constructed in primal-dual approach. The only difference in our approach is that we keep the trees with bandwidth cost less than the bandwidth threshold in a linked list for further ranking. In the worst case, the maximum size of ODST set is the number of edges in the network (m) since each edge can be a candidate center point. The maximum size of OBST set is the number of vertices of the network (n), since the algorithm iterates over all nodes in the network. Thus, the time complexity of ranking phase is $O(m+n)$.

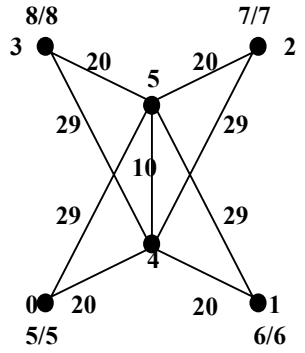

Fig. 3: A sample network with 6 nodes

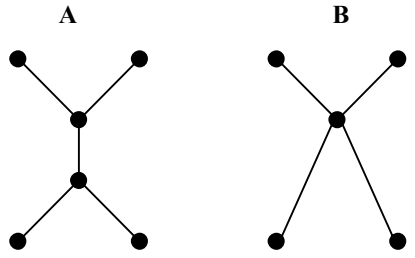

Fig. 4: ODST set, the result of performing phase 1

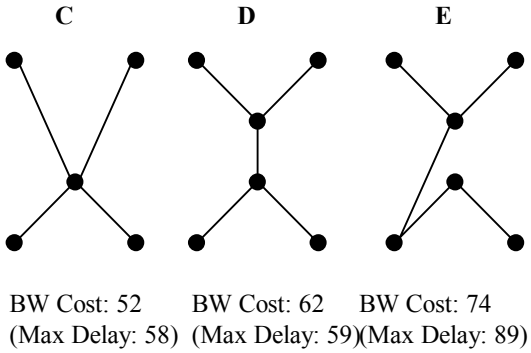

Fig. 5: OBST set, the result of performing phase 3

V. SIMULATION STUDY

Two sets of topologies were used in our simulations. The first set is taken from the Rocketfuel project [9]. Among all the topologies, we selected two dominant tier-1 ISP topologies as listed in Table I. The link delays of these two topologies were computed based on their geographical distances.

The second set was randomly generated using the Waxman Model [4]. Since we can easily control the size of the topologies, we use them to study the effect of the network size. In this model, the nodes are placed on a 3000×2400KM^2 plane, roughly the size of the USA. The probability for two nodes to be connected by a link decreases exponentially with the Euclidean distance between them according to the following probability function: $P_e^{(u,v)} = \alpha \exp(-l(u,v)/L\beta)$ where L is the maximum distance between any two nodes in the network and $l(u, v)$ is the distance between u and v. The parameter β controls the ratio of short links to long links, while the parameter α controls the average node degree of

the network. Large value of β increases the number of long links, and a large value of α results in a large average node degree. In the simulations, α and β were set at 2.2 and 0.15 respectively. These values were selected to be the same as used in the MDStT algorithm [8] implementations. Like the Rocketfuel topologies, the link delay values of the random networks were calculated according to their geographical distances.

For both sets of topologies, the VPN endpoints were randomly selected from the network nodes. The number of VPN endpoints was set to 10% of the total number of network nodes in the network unless explicitly specified. The bandwidth requirement of each VPN endpoint was uniformly chosen between 2 and 100 Mbps. An asymmetry parameter *r* is associated with each endpoint, representing the ratio between the ingress and egress bandwidth requirements. This ratio varies from 1 to 256 in our simulations. Each simulation result given below is the average of 5 rounds of simulation run. The 95% confidence intervals are calculated as $\hat{\theta} - t_{\alpha/2,f}\hat{\sigma}(\hat{\theta}) \le \theta \le \hat{\theta} + t_{\alpha/2,f}\hat{\sigma}(\hat{\theta})$ where $\hat{\theta}$ is the average value of simulations runs, $\hat{\sigma}^2(\hat{\theta})$ is the standard deviation and $t_{0.025,4}$ is 2.78 according to Table A.5 in [12].

The results show that the confidence intervals for provisioning costs are less than 0.5 Gbps and confidence intervals for minimum supportable delay are less than 5 ms.

We define three different scenarios for our OBDST algorithm: Scenario 1) bandwidth preference is equal to delay preference; Scenario 2) bandwidth preference is greater than delay preference; and Scenario 3) bandwidth preference is smaller than delay preference.

The performance of different scenarios of OBDST algorithm and LCLD algorithm [8] are compared in Figures 6 to 11. Fig. 6 compares the diameter of constructed shared trees connecting VPN endpoints. This value can be interpreted as the minimum delay requirement '*supported'* by each tree. The results show that using OBDST algorithm with delay preference greater than bandwidth preference would result in smaller minimum supported delay than LCLD algorithm [8]. The reason is that in LCLD algorithm only one center point will be considered to construct the shared tree while in our algorithm all the candidate centers are considered. Moreover, the results in Fig. 7 show that our algorithm will always result in smaller provisioning bandwidth cost. Note that since the Sprint network is a more linear network than the random networks generated by Waxman model, there is an increase in the minimum supportable delay for Sprint network in Fig. 6.

Fig. 8 illustrates the effect of increasing number of VPN endpoints on provisioning cost for a 100 network node. As stated earlier, the results show that our OBDST algorithm always results in smaller provisioning cost compared to LCLD algorithm [8]. Fig. 9 shows the effect of increasing the number of VPN endpoints on the execution time of algorithms. Fig. 10 shows the effect of increasing the network size on the execution time. As the results shows, running OBDST algorithm on a typical tier 1 network, which has around 100 nodes, does not require more than 1 minute.

TABLE I
ROCKETFUEL ISP TOPOLOGIES USED IN THE SIMULATIONS

AS Number	Name	Tier	No.of Edges	No.of Nodes
1239	Sprint(US)	1	168	52
7018	ATT(US)	1	296	115

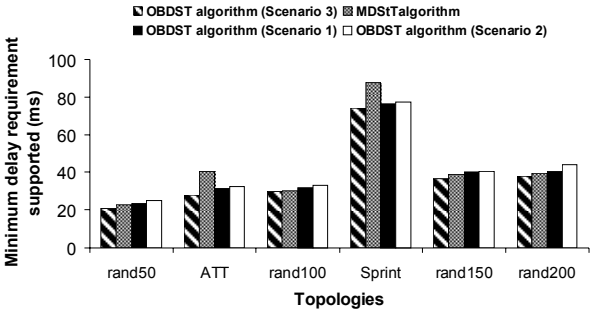

Fig. 6: Minimum delay requirement supported

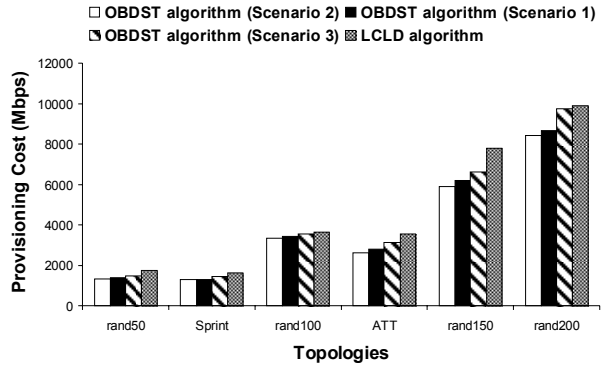

Fig. 7: The provisioning cost comparison

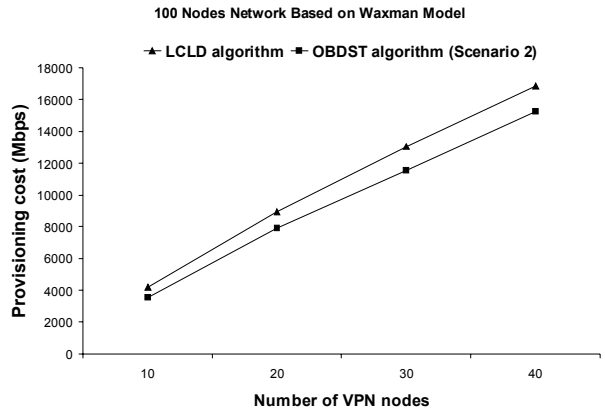

Fig. 8: Effect of number of VPN nodes on provisioning cost

Since the resource allocation is not likely to be performed more often than every one minute, this does not pose a problem. In Fig. 11 the effect of changing the bandwidth asymmetry ratio on provisioning cost is studied. The ratio between ingress and egress bandwidth of VPN endpoints has

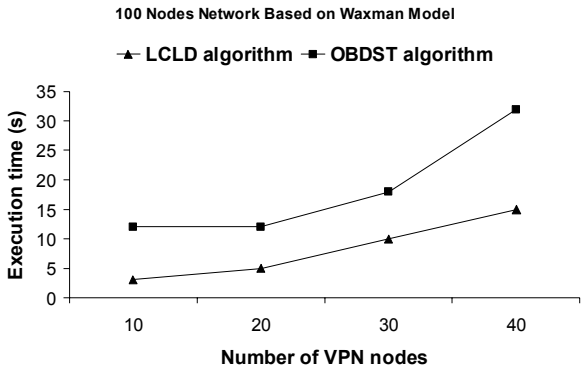

Fig. 9: Effect of number of VPN nodes on execution time

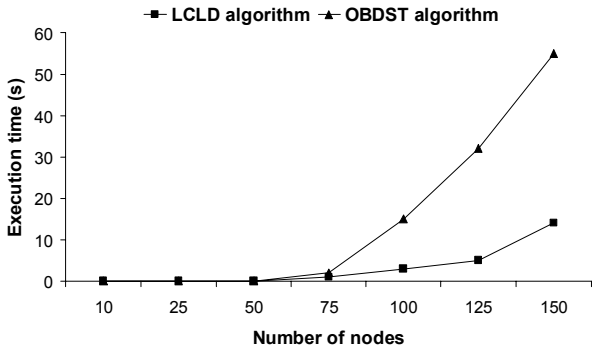

Fig. 10: Effect of number of network nodes on execution time

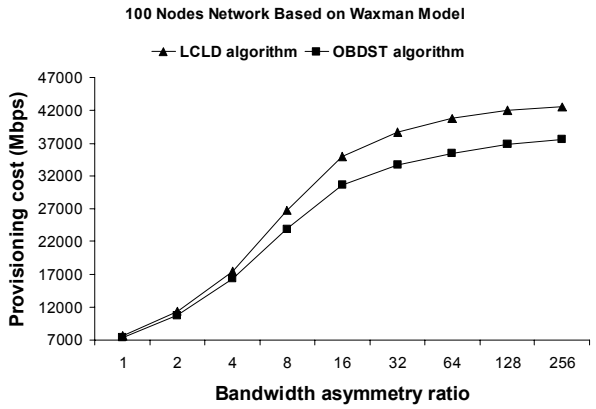

Fig. 11: Effect of asymmetry ratio

been increased from 1 to 256. The results show that our OBDST algorithm would still perform better than the LCLD algorithm.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we introduced a new ranking scheme based on user preferences to optimize the provisioning bandwidth cost while supporting the minimum delay requirement in VPN hose model. This scheme will result in lower bandwidth provisioning costs than the previous work introduced by Zhang et al. Our algorithm modifies and combines primal-dual algorithm [5] and Least-Cost-Least-Delay algorithm [8] to find the MDStT tree that satisfies the maximum allowable delay and provides optimum provisioning cost.

For future work, the ranking scheme will be extended to include link capacity constraints and any link capacity enhancement features (e.g. increase the path or link bandwidth) into our ranking scheme. This can be done by introducing two more parameters such as the enhancement preference and enhancement cost in our ranking scheme. The latter can be interpreted as the preference of network administrator to increase the bandwidth capacity of a particular link while the former is the monetary cost to perform such operation.

REFERENCES

[1] B. Davie, Y. Rekhter. "MPLS technology and applications", San Mateo, CA: Morgan Kaufmann, 2000.

[2] N. G. Duffield, P. Goyal,A. Greenberg, P. Mishra, K. K Ramakrishnan, J. E. van der Merwe, "A flexible model for resource management in Virtual Private Networks", In Proc. ACM SIGCOMM, vol 29(4), 1999, pp. 95-108.

[3] A. Juttner, I. Szabo, A. Szentesi, "On bandwidth efficiency of the Hose resource management model in Virtual Private Networks", In Proc. INFOCOM, vol. 1, 2003, pp. 386-395.

[4] B. M. Waxman, "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, vol. 6(9), 1988, pp. 1617-1622.

[5] A. Kumar, R. Rastogi, A. Silberschatz, B. Yener, "Algorithms for provisioning Virtual Private Networks in the hose model", IEEE/ACM Transaction on Networking, vol. 10(4), 2002, pp. 565-578.

[6] G. de Veciana, S. Park, A. Sang, S. Weber. "Routing and provisioning VPNs based on hose traffic models and/or constraints". In Proc. 40th Annual Allerton Conference on Communication Control and Computing, 2002, pp. 77-86.

[7] T. Erlebach, M. Ruegg, "Optimal bandwidth reservation in hose model VPNs with multi-path routing", In Proc. INFOCOM , vol. 4, 2004, pp. 2275-2282.

[8] L. Zhang; J. Muppala, S. Chanson, "Provisioning virtual private networks in the hose model with delay requirements." Hong Kong University, International Conference on Parallel Processing, 2005, pp.211 – 218.

[9] Rocketfuel project, Computer Science and Engineering, Univ. of Washington. http://www.cs.washington.edu/research/networking/rocketfuel

[10] A. Gupta, A. Kumar, Kleinberg, R. Rastogi, B.Yener, "Provisioning a Virtual Private Network: A network design problem for multicommodity flow", In Proc. ACM STOC, 2001, pp. 389-398.

[11] S. Hakimi. "Optimal locations of switching centers and medians of A graph", Operations Research, vol. 12, 1964, pp. 450-459.

[12] Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol, "Discrete-Event System Simulation", Fourth Edition, Prentice Hall, 2005.