ENHANCED PROVISIONING ALGORITHM FOR VIRTUAL PRIVATE NETWORK IN HOSE MODEL WITH QUALITY OF SERVICE SUPPORT USING WAXMAN MODEL

R. Ravi
Department of Computer Science & Engineering, Francis Xavier Engineering College, Tamil Nadu, India
E-mail : fxhodcse@gmail.com

Abstract
As Internet usage grows exponentially, network security issues become increasingly important. Network security measures are needed to protect data during transmission. Various security controls are used to prevent the access of hackers in networks. They are firewall, virtual private networks and encryption algorithms. Out of these, the virtual private network plays a vital role in preventing hackers from accessing the networks. A Virtual Private Network (VPN) provides end users with a way to privately access information on their network over a public network infrastructure such as the internet. Using a technique called “Tunneling”, data packets are transmitted across a public routed network, such as the internet that simulates a point-to-point connection. Virtual private networks provide customers with a secure and low-cost communication environment. The basic structure of the virtual circuit is to create a logical path from the source port to the destination port. This path may incorporate many hops between routers for the formation of the circuit. The final, logical path or virtual circuit acts in the same way as a direct connection between the two ports. The K-Cost Optimized Delay Satisfied Virtual Private Networks Tree Provisioning Algorithm connects VPN nodes using a tree structure and attempts to optimize the total bandwidth reserved on the edges of the VPN tree that satisfies the delay requirement. It also allows sharing the bandwidth on the links to improve the performance. The proposed KCDVT algorithm computes the optimal VPN Tree. The performance analysis of the proposed algorithm is carried out in terms of cost, the number of nodes, and the number of VPN nodes, delay, asymmetric ratio and delay with constraints with Breadth First Search Algorithm. The KCDVT performs better than the Breadth First Search Algorithm.

Keywords:
Provisioning, VPN and Waxman

1. QUALITY OF SERVICE GUARANTEES IN VPN

The general objective of quality of service routing is to improve the efficient utilization of network resources and to provide flexibility in support for various services, with two goals. The goals are, maximizing the overall throughput of the VPN and providing a mechanism that enables VPN customers to allocate the bandwidth according to their own requirements, thus achieving the predicted quality of service performance.

Quality-of-Service is defined as a set of service requirements to be met by the network while transporting a flow. Here, a flow is a packet stream from a source to a destination with an associated Quality of Service (QoS). In other words, QoS is a measurable level of service delivered to network users, which can be characterized by packet loss probability, available bandwidth, and end-to-end delay.

In this work, a new algorithm K-Cost Optimized and Delay Satisfied Virtual Private Network Tree Provisioning algorithm is proposed for finding the optimal VPN tree by considering both cost as well as delay.

1.1. K - COST OPTIMIZED AND DELAY SATISFIED VPN TREE PROVISIONING ALGORITHM (KCDVT)

The previous algorithm considers only the cost parameter for performance analysis, but the KCDVT considers both the cost as well as the delay.

Input: The topology graph, VPN nodes and ingress/egress bandwidth of the VPN nodes.

Output: K-Cost Optimized and Delay Satisfied Virtual Private Network Provisioning Tree.

Step 1: Get the number of nodes and generate random topology using the Waxman Model with our program.

Step 2: Fix 10 to 50% of the boundary region nodes as VPN nodes.

Step 3: Randomly assign the bandwidth and delay for each link of the topology graph within the specified bandwidth and delay constraints.

Step 4: Assign all the bandwidth and delay of each link of the topology to zero. Also assign the ingress/egress bandwidth for each VPN node only.

Step 5: Get the source and destination VPN nodes from the topology constructed using our program.

Step 6: Find all possible shortest paths from the source VPN node to the destination VPN nodes using the K-Optimized traffic lane algorithm.

Step 6.1: Find all the shortest paths from the source to other VPN nodes using the Dijkstra Algorithm and add it to the path set.

Step 6.2: For each path in the path set do the following steps. Take the first path from the path set.

a. Take the least cost link in the selected path and add to the least cost link.

b. Remove the least cost link from the network and find the new shortest path using the Dijkstra Algorithm.

c. If a new path exists then add the path to the path set.

d. Repeat the step b to c for each least cost link.

e. Reinsert the removed link into the network.

Step 6.3: Repeat step 6.2 for each path in the path set until all the paths in the path set have been processed.
Step 7: Construct a BFS Tree by combining any two paths from the shortest path list randomly.

Step 8: Repeat step 7 for different possible combinations of shortest paths.

Step 9: To find the bandwidth requirement for each link of all BFS Trees.

Step 9.1: To find the bandwidth requirement of a particular link, remove it from the BFS tree.

Step 9.2: Now the tree is divided into two components, say T1 and T2.

Step 9.3: Find the sum of the assigned bandwidth of the VPN nodes in the component T1.

Step 9.4: Find the sum of the assigned bandwidth in the component T2.

Step 9.5: Find the minimum of the sum among the two components T1 and T2.

Step 9.6: Assign the chosen minimum cost to the removed link.

Step 9.7: Repeat step 9.1 to step 9.6 for all links in the BFS Tree.

Step 10: Repeat Step 7 to Step 9 until all the BFS trees have been processed.

Step 11: Check whether the constructed BFS Tree satisfies the delay constraints.

Step 12: Separate the BFS Tree which satisfies the delay constraints.

Step 13: Choose the BFS Tree which gives minimum cost; that tree is called as a K-Cost Optimized and Delay Satisfied Virtual Private Network Provisioning Tree.

1.2 TOPOLOGY GENERATION

Fig. 1 is constructed using the following parameters. The size of the plane is set to 1000 and the square of the inner plane of the model is set to 100. The number of nodes is set to 15 (0 to 14). The model selection is Waxman. The $\omega$ and $\beta$ are Waxman specific constant, and its value is set to 0.15 and 0.2 respectively. The node placement strategy is set to random. The growth type is incremental. The bandwidth and delay for each link is set to zero for the calculation of the bandwidth and delay using the KCDVT algorithm. The minimum and maximum bandwidth assigned for each VPN node is randomized from 0 kbps to 999 kbps of symmetric bandwidth. Also, the ingress and egress bandwidth assignment for each of the VPN nodes are symmetric. Due to the random characteristic of the Waxman topology generation model, we randomly generate the following parameters using our program. The other name of VPN node is VPN end point.

According to the hose model concept, we vary randomly the ingress and egress bandwidth to each of the VPN nodes so that it is higher than the randomly generated links bandwidth. In the Fig. 1, we set randomly the symmetric value of the ingress bandwidth of the VPN nodes A, B, C and D as 2/2 Mbps, 1/1 Mbps, 1/1 Mbps and 2/2 Mbps respectively.

Table.1 parameters are constructed randomly according to the Waxman concept which is used in the topology. The Edge identification is set to unique. ‘From’ indicates the node identification of the source. ‘To’ indicates the node identification of the destination. The delay indicated propagation delay of the links. Bandwidth indicates the assigned bandwidth of each link.

<table>
<thead>
<tr>
<th>Edge Id</th>
<th>From</th>
<th>To</th>
<th>Delay (sec)</th>
<th>Bandwidth (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>2</td>
<td>1.21</td>
<td>377.25</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>4</td>
<td>1.66</td>
<td>183.20</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2</td>
<td>1.31</td>
<td>388.64</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>4</td>
<td>1.64</td>
<td>861.35</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>8</td>
<td>1.95</td>
<td>862.49</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td>0.80</td>
<td>428.70</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>1.94</td>
<td>405.97</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>8</td>
<td>1.63</td>
<td>716.00</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>0.12</td>
<td>283.87</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>9</td>
<td>1.57</td>
<td>318.82</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>1</td>
<td>0.94</td>
<td>50.34</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>9</td>
<td>1.87</td>
<td>478.53</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>7</td>
<td>1.56</td>
<td>433.80</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>2</td>
<td>3.02</td>
<td>191.10</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>6</td>
<td>1.94</td>
<td>555.75</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>7</td>
<td>1.35</td>
<td>551.67</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>7</td>
<td>1.54</td>
<td>599.72</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>0</td>
<td>2.86</td>
<td>881.07</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>7</td>
<td>2.23</td>
<td>97.63</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>0</td>
<td>2.32</td>
<td>219.61</td>
</tr>
</tbody>
</table>

1.3 ALGORITHMS USED FOR COMPARISON

Here, we have used the Waxman Model to generate a random topology of the Virtual Private Network for symmetric as well as Asymmetric cases. The number of nodes chosen for
analysis is 100 to 1000. The VPN nodes chosen for analysis are 100 to 500. Also, we have compared our K-Cost Optimized and Delay Satisfied Virtual Private Networks Tree Provisioning Algorithm with the Breadth First Search algorithm (Amit Kumar et al 2002) on the basis of the following parameters: cost, number of nodes, number of VPN nodes and delay. The delay constraint chosen for analysis is 6 sec and 20 sec.

2. SIMULATION RESULTS AND DISCUSSION (WAXMAN MODEL)

2.1. NETWORK SIZE VS COST FOR THE SYMMETRIC VIRTUAL PRIVATE NETWORK USING THE WAXMAN MODEL

Figs. 2, 3, 4, 5 and 6 depict the symmetric provisioning cost of the KCDVT and BFS algorithms for 100, 200, 300, 400 and 500 boundary VPN nodes among the 1000 nodes respectively. The model used for topology creation is the Waxman Model. The bandwidth of each link is assigned randomly by simply specifying the minimum and maximum values in the range of 0 to 2000 Mbps. The VPN nodes are randomly assigned with symmetric bandwidth so that the bandwidth is higher than any randomly assigned bandwidth for the links. The value plotted in the graph was the average of the readings taken from fifteen different runs of the experiment.

![Comparison Of KCDVT Vs BFS Cost](image1)

Fig.2. Effect of the Number of nodes on cost for the Symmetric case of the Waxman Model (100 VPN nodes among 1000 nodes)

![Comparison Of KCDVT Vs BFS Cost](image2)

Fig.3. Effect of the Number of nodes on cost for the Symmetric case of the Waxman Model (200 VPN nodes among 1000 nodes)

![Comparison Of KCDVT Vs BFS Cost](image3)

Fig.4. Effect of the Number of nodes on cost for the Symmetric case of the Waxman Model (300 VPN nodes among 1000 nodes)

![Comparison Of KCDVT Vs BFS Cost](image4)

Fig.5. Effect of the Number of nodes on cost for the Symmetric case of the Waxman Model (400 VPN nodes among 1000 nodes)

![Comparison Of KCDVT Vs BFS Cost](image5)

Fig.6. Effect of the Number of nodes on cost for the Symmetric case of the Waxman Model (500 VPN nodes among 1000 nodes)

Figs. 2, 3, 4, 5 and 6 shows that our KCDVT algorithm is certainly optimal for the symmetric case than the BFS. The cost of the KCDVT is lesser than that of the Breadth First Search algorithm.
2.2 NETWORK SIZE VS DELAY FOR THE SYMMETRIC VPN USING THE WAXMAN MODEL

Figs. 7, 8, 9, 10 and 11 depict the symmetric provisioning delay of the KCDVT and BFS algorithms for 100 boundary VPN nodes with a delay constraint of 6 seconds, 200 boundary VPN nodes with a delay constraint of 7 seconds, 300 boundary VPN nodes with a delay constraint of 11 seconds, 400 boundary VPN nodes with a delay constraint of 12 seconds and 500 boundary VPN nodes with a delay constraint of 13 seconds, among the 1000 nodes respectively. The model used for topology creation is the Waxman Model. The value plotted in the graph was the average of the readings taken from fifteen different runs of the experiment.

Fig. 7. Effect of the Number of nodes on delay for the Symmetric case of the Waxman Model (100 VPN nodes among 1000 nodes with delay constraint 6 seconds)

Fig. 8. Effect of the Number of nodes on delay for the Symmetric case of the Waxman Model (200 VPN nodes among 1000 nodes with delay constraint 7 seconds)

Fig. 9. Effect of the Number of nodes on delay for the Symmetric case of the Waxman Model (300 VPN nodes among 1000 nodes with delay constraint 11 seconds)

Fig. 10. Effect of the Number of nodes on delay for the Symmetric case of the Waxman Model (400 VPN nodes among 1000 nodes with delay constraint 12 seconds)

Fig. 11. Effect of the Number of nodes on delay for the Symmetric case of the Waxman Model (500 VPN nodes among 1000 nodes with delay constraint 13 seconds)
Figs. 7, 8, 9, 10 and 11 shows that our KCDVT algorithm is certainly optimal for the symmetric case than the BFS. The delay of the KCDVT is lesser than that of the Breadth First Search algorithm.

3. SUMMARY OF CONTRIBUTIONS

In the above simulation study we have used both the Waxman Model for topology generation. The number of nodes is varied from 0 to 1000 and the number of VPN nodes is varied from 0 to 500. The delay constraint is varied from 5 to 17 seconds. Both symmetric and Asymmetric bandwidths were studied. The asymmetric ratio is also varied from 0 to 250.

Designed and implemented K-Cost Optimized and Delay satisfied Virtual Private Network Tree Provisioning Algorithm with better performance than the existing Breadth First Search algorithm in the asymmetric case hose model.

Designed and implemented K-Cost Optimized and Delay satisfied Virtual Private Network Tree Provisioning Algorithm with better performance than the existing Breadth First Search algorithm in the Asymmetric case hose model.

REFERENCES