Journal of Computer Science 4 (1): 1-5, 2008
ISSN 1549-3636

Provisioning QoS in Virtual Private Network using Dynamic Scheduling

[1]R. Ravi and [2]S. Radhakrishnan,
[1]Department of Computer Sciences and Engineering,
[2]Department of Computer Sciences and Engineering, A.K. College of Engineering,
Anand Nagar, Krishnankoil, 626190, Virudhunagar District, Tamil Nadu State, India

Abstract: Active and programmable networks change the functionality of routers and switches by using VPN endpoints and active packets. The authors present a new packet scheduling scheme called active scheduling to control and maintain QoS parameters in virtual private networks (VPNs) within the confines of adaptive and programmable networks. In active scheduling a VPN endpoint on the router monitors the accumulated queuing delay for each service. To control and to keep the end-to-end delay within the bounds, the delays of the packet transmitted are adjusted dynamically by VPN endpoints on the routers spanning the VPN. If there is an increase or decrease in delay of packets, VPN endpoints on a source node informs the destination node to adjust the delay of the packets. This keeps the end-to-end delay of services within the specified bounds and offers better QoS compared with VPNs using static priority scheduling. An algorithm for active scheduling is described and simulation results are presented and compared with priority scheduling.

Key words: VPN, Active Scheduling, Priority, Qos

INTRODUCTION

In traditional computer networks the intermediate nodes are vertically closed systems whose functions are rigidly programmed into the embedded software and hardware by the vendors. The VPN also falls in this category. Therefore the development and deployment of new protocols requires a long standardization process. Active networks [1] allow users or operators to inject customized codes into the network to modify the behavior of switches and routers. Scheduling of different classes of traffic within the switches and routers has been identified as one of the most important resources to be reconfigured [2].

This research presents a novel mechanism called, active scheduling for the reconfiguration of the scheduling dynamically to modify the queuing strategy in routers [2-4].

Active scheduling allows the introduction of a procedure by which the packet delay of heavily loaded routers are altered dynamically according to different classes of traffic (voice, video and data), since each class has different QoS requirements. This is achieved by using VPN endpoints on the routers initiating the reconfiguration of delay. This procedure is called active scheduling. One such active scheduling is LST scheduling where the delay of the packet transmitted are adjusted to minimize the packet loss [5].

A survey of packet-scheduling schemes of active and programmable networks shows that projects such as tempest, based on switchlets[6] and spawning networks[7], based on routelets, use static scheduling mechanisms. One of these well-known scheduling schemes is priority. There are certain problems with priority scheduling. First, it allocates a fixed amount of bandwidth over a defined time scale to the sessions over the routers spanning the VPN, that is, the assignment of weights to any queue within a scheduler or across a number of schedulers is essentially static, which leads to bandwidth bottlenecks at the output link of routers for the sessions joining the queues of a router from other sections of the network. Secondly, there is no provision for intelligent load management in this scheme, which results in higher end-to-end delays for the services and wastage of bandwidth. This is because different routers in the network are loaded with different traffic loads. For example, if a router in the core network supporting the VPN is heavily loaded and the upstream routers are lightly loaded, then this can lead to wastage of bandwidth and higher end-to-end delays. This research presents and describes active scheduling and compares its performance with priority scheduling [8] by presenting a simulation example.

Corresponding Author: R. Ravi, Department of Computer Sciences and Engineering, A.K. College of Engineering, Anand Nagar, Krishnankoil, 626190, Virudhunagar District, Tamil Nadu State, India Tel: +91-9442418917

LEAST SLACK TIME ALGORITHM

Each VPN in the active router environment contains two types of agents on the active routers: queue agents and control agents [9]. VPN endpoints control a delay of sessions according to the maximum and minimum delay bounds specified by the clients. Control agents control the end-to-end delay of a VPN by monitoring the delay of a session on a routing path. However the LST scheduling one such active scheduling algorithm, here proposed does not require any control at the VPN endpoints. In Fig. 1 we adjust the delay of the packets routed and is calculated based on the slack time and transmitted to meet the QoS parameters [10].

In Figure 1 the VPN Endpoints with the packets being transmitted effectively with adjusted delay. This delay adjustment is to minimize the packet loss effectively. The Figure 2 shows the LST algorithm base work.

Each node in VPN environment is assigned with the input parameters delay and bandwidth. Based on which the slack time is calculated and the packets are routed in a minimized delay[11]. This scheduling algorithm first selects those packets that have the smallest slack time. Slack time is defined as the temporal difference between the deadline, the ready time and the transmission time. More formally, the slack time for a process is defined as (d-t)-c , where d is the transmission deadline, t is the real time since the cycle start and c is the remaining computation time. The next packet in serious will take this left out slack time for transmission so there is minimized delay. If the delay exceeds the packet drops and is retransmitted at last.

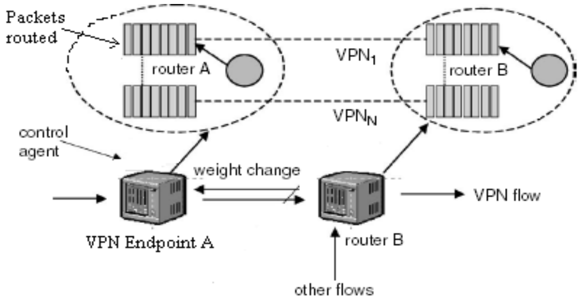

Fig.1: Packet routing based on LST

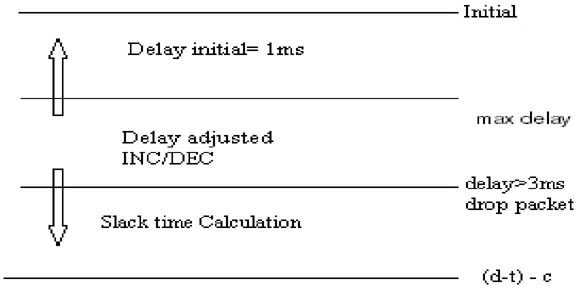

Fig. 2: Basic LST Framework

LST scheduling is functionally compatible with the DiffServ architecture, controlling endpoints can be used to monitor the delay of aggregates. The number of agents on a router depends on the number of aggregates. The delay will be changed whenever there is congestion on the links or a change in the number of sessions. Thus the traffic can be minimized effectively[12].

Proposed Algorithm

Step 1: For each pair of nodes choose the source and destination path compute k-shortest paths.

Step 2: Consider the Max number of packets Pn to be routed via $S_{1....}S_{n+1}$

Step 3: Compute the LST scheduling for finding the packet delay and loss.

Step 4: Then compare the packet routing efficiency based on the algorithm

PERFORMANCE ANALYSIS OF ACTIVE SCHEDULING

Simulation environment: The simulation environment need to be simulated is efficient with packet routing among the routers and switches taken to be the VPN endpoints. The shortest path among the source and destination endpoint is found out and is made as the path for the routing. The minimal cost of the path with the delay adjustment and the bandwidth guaranteed transmission improves the Qos within VPN as in Fig. 3.

Consider the simulated environment in which the number of packets to be routed Pn, is given via S/A based on which the LST is implemented. The packets via S1 seem to be 20 packets then calculate the slack time in routing. The routed are adjusted dynamically to reduce the delay. The slack time is added to the transmission time which could result in improved bandwidth utilization and efficiency. The delay and slack time determines the efficiency of routing. This is shown in calculation part below

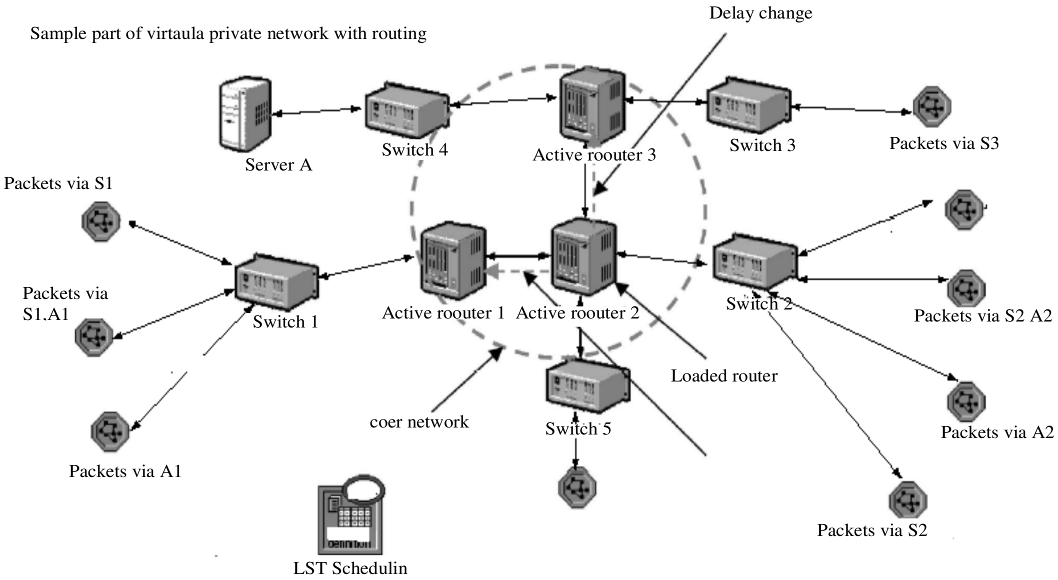

Fig. 3: Environment Need to be simulated

Calculation of the delay is described as: Packets to be routed are P and the transmission time is set based on the link capacity. The delay is altered dynamically so that the efficiency is improved.

Packet transmission Time> Packet Deadline Time

Packet Deadline Time - Packet Time gives the Slack Time

The Slack time left out is added to previous transmission Time

if delay(Packets) >=3ms then drop the packet

Then Count++

When packet is dropped then retransmit the packet.

EXPERIMENTAL STUDIES

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. The (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 ≠ Lji. In the hose model, each VPN specification consists of a set of VPN endpoints P⊆V and the ingress and egress bandwidths of each of the VPN endpoints.

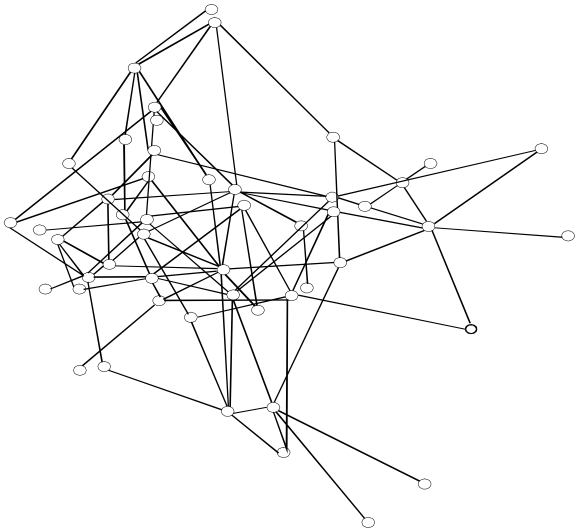

Fig. 4: Waxman Model: Nodes 50

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.

The traffic among the nodes can be scheduled based on LST scheduling algorithm. 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 as shown in Figure 4. For each topology, |P| VPN endpoints are randomly picked up. Based on the chosen end points the shortest path is generated effectively based on which the packets are routing with the LST implemented.

Table 1: Delay bound for packets in active router via S1

Sessions	Maximum Packets	Loss packets packets	Delay in (msec) transmission
Packets via S1/A1	55	28	12
Packets via S2/A1	24	12	6
Packets via S3/A2	21	10	4

Table 2: Delay bound for packets in active router via S2

Sessions	Maximum Packets	Loss packets packets	Delay in (msec) transmission
Packets via S1/A1	50	8	12
Packets via S2/A1	31	11	6
Packets via S3/A2	22	15	4

The routed packets with adjusted delay are efficiently taken into account with minimal bandwidth utilization. The Table 1 and 2 shows the minimal delay in packets routed via nodes S1/A1 or S2/A2 or S3/A3 with calculated packet loss and is overcome by the retransmission.

Based on the Table and the comparative graph, the proposed algorithm Least Slack Time is justified to be efficient than the priority scheduling in terms of cost and delay (Figure 5 and Figure 6).

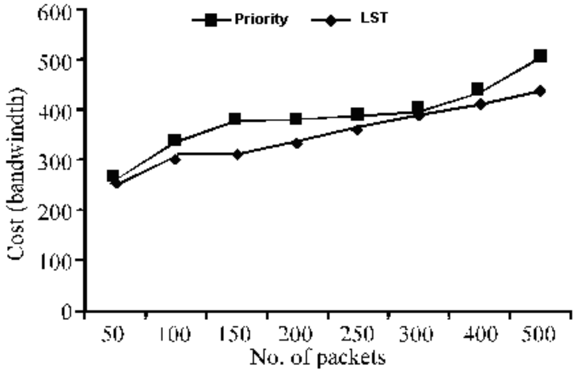

Fig. 5: Graph showing LST Vs Priority (No of packets Vs Cost)

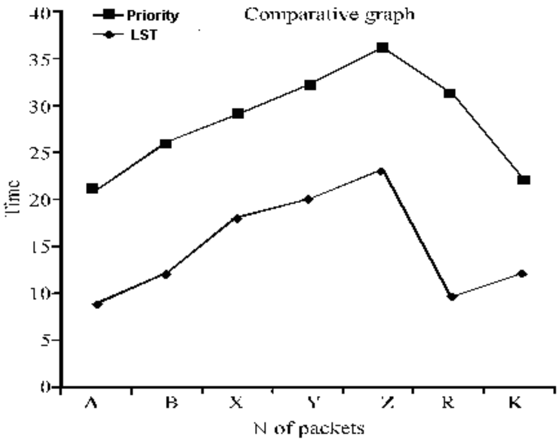

Fig. 6: Graph showing Time delay Vs Packets transmitted

CONCLUSIONS AND FUTURE WORK

The authors have presented a new scheduling scheme called Least Slack Time scheduling. Virtual private networks consisting of active routers have been implemented in NS2. Active routers reconfigure the delays of packet routing in their shortest path. The results have shown that the active routers offer better performance in terms of end-to-end delay and cost.

In future there are still a number of issues relating to VPN routing. For example: (1) The problem of fitting failure of lowest cost path and restoration mechanisms (2) Provisioning for the Asymmetric VPN nodes in active routers.

ACKNOWLEDGEMENT

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

REFERENCES

1. Rocketfuel project, 2006. Computer science and engineering. University of Washington and HIST Algorithm, 12: 123-140.

2. Liang, H., O. Kabranov, D. Makrakis and L. Orozco-Barbosa, 2002. Minimal cost design of virtual private networks. In: IEEE Proceedings of the CCECE ’02. 3: 1610-1615.
3. Erlebach, T. and M. Ruegg, 2004. Optimal bandwidth reservation in hose-model vpns with multi-path routing. INFOCOM., 4: 2275-2282.
4. Duffield, N.G., P. Goyal, 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: Proceeding ACM SIGCOMM, pp: 95-108.
5. Gupta, A., A. Kumar, J. Kleinberg, R. Rastogi and B. Yener, 2001. Provisioning a virtual private network: A network design problem for multicommodity flow. In: Proceeding. ACM STOC., pp: 389-398.
6. Kumar, A., R. Rastogi, A. Silberschatz and B. Yener, 2002. Algorithms for provisioning virtual private networks in the hose model. IEEE/ACM Trans. on Networking, 10 (4): 565-578.
7. Juttner, A., I. Szabo and A. Szentesi, 2003. On bandwidth efficiency of the hose resource management model in virtual private networks. In: Proceeding INFOCOM. 1: 386-395.
8. Matta, I. and L. Guo, 1999. QDMR: An efficient qos dependent multicast routing algorithm. J. Commun. Networks. Real-time Technology and Applications Symposium, pp: 213-222.
9. Mishra, P.P. and H. Saran, 2000. Capacity management and routing policies for voice over ip traffic. IEEE Network, 14 (2): 20-27.
10. Waxman, B.M. 1988. Routing of multipoint connection. IEEE., J. Selected Areas Commun., 6 (9): 1617-1622.
11. Zhanfeng Jia and Pravin Varaiya, 2006. Heuristic methods for delay Constrained least cost routing using k-shortest-paths. IEEE Tran. on Automatic Control, 51 (4): 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, pp: 42-48.