SCHEDULING ALGORITHMS FOR BOUNDED DELAY SERVICE IN VIRTUAL NETWORKS

Rahul Garg
rahul@cse.iitd.ernet.in

Huzur Saran
saran@cse.iitd.ernet.in

Department of Computer Science and Engineering
Indian Institute of Technology Delhi, India

Abstract

With their widespread deployment, the need to provide QoS in virtual networks is being realized. We show that traditional scheduling algorithms like WFQ are incapable of providing bounded delay service in a virtual network. The problem arises because traffic of a number of sessions sharing a virtual link in a virtual network is aggregated and tunneled through the physical network, which cannot isolate the traffic of well behaved sessions from that of misbehaving sessions. We show that this problem can be solved by regulating output burstiness on virtual links. Using the theory of latency rate servers, we show that latency rate servers with bounded output burstiness, may be used in a virtual network to provide bounded delay service. This gives a method to design a generic class of scheduling algorithms for virtual networks. Finally, we discuss how to integrate best effort and real-time traffic in a virtual network.

Keywords: Virtual Networks, VPN, QoS, Scheduling, Latency-Rate Servers, Output Burstiness.

1 Introduction

Virtual networking is an important step in the evolution of data networks. It allows quick deployment of new services over legacy networks, eases network operation and management by hiding the unnecessary details and presenting a simplified topology, allows development of experimental protocols in a controlled and safe environment and eases interoperability between networks of different types.

Future corporate virtual private networks (VPN) will carry real time voice and video conference data along with the regular application data. Therefore providing QoS guarantees to real-time applications in virtual networks is very important. Most of the research in virtual networks has been focused on developing new services or managing the virtual networks more efficiently [1, 2, 3]. The research related to QoS in virtual networks has been limited. In [4, 5] the concept of link sharing and packet scheduling algorithms to implement link sharing have been proposed. Using link sharing, a physical link may be partitioned into several logical links, which may further be partitioned to carry traffic of different classes. Application of link sharing to build QoS capable virtual networks, where a virtual link is realized by a sequence of logical links, has not been discussed.

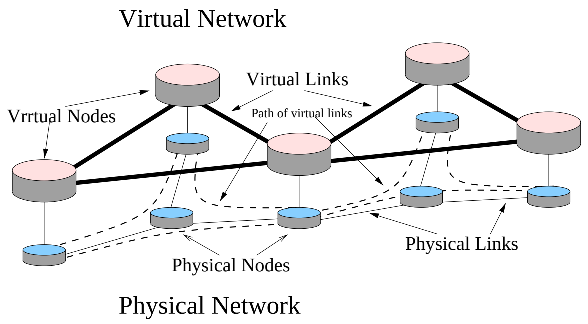

Figure 1: Example of a virtual network

A virtual network has virtual nodes overlaid on physical nodes as shown in Figure 1. Pairs of virtual nodes are connected to each other by virtual links, which are realized by tunnels [6] traversing a path between them. Packets arriving on a virtual node are forwarded to the next virtual node via a virtual link based on the routing table of the virtual network. Before forwarding, packets of virtual network are encapsulated into packets of the physical network. These packets carry the required header to route them in the physical network. At the destination node, the encapsulated packet of the virtual network is reconstructed and given to the virtual node.

While providing QoS in a virtual network, it is desirable to maintain the partitioning introduced by the virtual network. The physical network should keep the state information only for the tunnels, not for individual sessions of virtual network. The entire traffic of a tunnel should be transported as if it was a single session of the physical network. Packet scheduling algorithms are key to providing QoS in any network. Therefore, the scheduling algorithms in the physical network should treat the virtual network tunnels as physical network sessions. We also assume that the packet headers do not contain any QoS specific marking.

With these assumptions we show, in Section 2, why traditional work conserving schedulers cannot be used in virtual networks. When, a work conserving scheduler, because of some misbehaving session, sends excess traffic on a virtual link at a rate larger than its allocated rate, traffic of some well behaved sessions may get excessively delayed. In Section 3, we quantify this excess traffic as output burstiness and argue that worst case session delays are related to the output burstiness. We extend the theory of latency rate servers [7] to show how latency rate servers with bounded output burstiness may be used to provide bounded delays to sessions of virtual network. This gives a flexible method for designing a generic class of scheduling algorithms for virtual networks. We also present an example of scheduler in this class. Because of the output burstiness constraint, the bandwidth sharing among real-time traffic is limited within the same virtual link. In Section 4 we briefly discuss how best effort traffic can utilize the residual bandwidth to achieve more sharing. We summarize our results in Section 5.

2 Drawback of Traditional Schedulers

In a virtual network, the scheduling is done at two different levels: At the physical level, the link level scheduler schedules aggregate traffic of different virtual links, and at the level of virtual nodes, the virtual scheduler decides the order in which packets from different session are encapsulated. This may be viewed as a hierarchical scheduler [5] with two levels of hierarchy. At the top level of the hierarchy, the link level scheduler partitions the link bandwidth into virtual links. At the next level, the virtual scheduler divides the bandwidth of the virtual link among the sessions.

Let a virtual link of rate 1.5Mbps be realized by a two hop path in the physical network. Let the first physical link in the path be of rate of 45Mbps and let the second physical link be of rate 2Mbps as shown in Figure 2. Assume that one voice sessions of rate 64Kbps and a video session of rate 1Mbps are carried by this virtual link. Also assume that the link level scheduler and the virtual scheduler at node 1 are both work conserving. Assume that no other traffic is present in the system. Assume that for some reason, the video session misbehaves and queues up a large amount of data in its output queue at time $t = 0$. Since the virtual scheduler and the link level scheduler at node 1 are both work conserving, the entire traffic of the video session (along with a little traffic of voice session) will be sent from node 1 at a rate of 45Mbps. The rate available to the virtual link at node 2 is limited by the physical link rate which is 2Mbps. Therefore the backlog virtual link traffic at node 2 at time t in this busy period will be $(45 - 2)t$ Mbits. The link level scheduler at node 2 simply serves all the virtual link traffic in first come first serve manner. Therefore any voice packet sent by the link level scheduler of node 1 at time t will experience a queueing delay of at least $43/2t$ at node 2. Thus, an MPEG I-frame burst of 50KB may cause a 191ms queueing delay for voice packets at node 2. In general, if the link level scheduler and virtual scheduler are both work conserving, then well behaving sessions may be penalized because of a misbehaving session in the same virtual link.

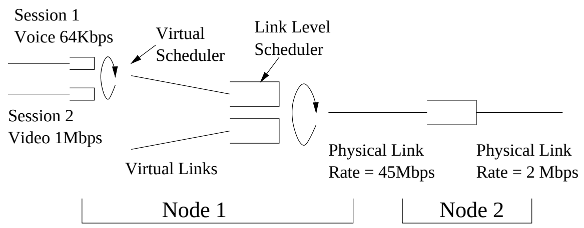

Figure 2: Queues at two different nodes in the path of a virtual link

3 Bounded Delay Service in Virtual Networks

In [7] the theory of latency rate servers was presented. Using this theory, tight end-to-end delay bounds for an arbitrary network of schedulers can be proved. A number of schedulers including weighted fair queueing (WFQ or PGPS) [8, 9], and recursive round robin [10] belong to the class of latency rate servers.

Unfortunately, this theory cannot be used in its present form in a virtual network because of traffic aggregation. We extend this theory by first defining the concept of output burstiness and then showing that if output burstiness of a scheduler is bounded and it satisfies the latency rate property, then a tandem of such schedulers also satisfies the latency rate property even if the downstream schedulers act on aggregate traffic. With this, each virtual link can be modeled as a latency rate server and the latencies of these virtual links can be combined to obtain tight end-to-end delay bounds for individual sessions (as in physical networks) [7].

Definition 3.1 (Output burstiness) *The output burstiness of a virtual link l is the smallest value b_l such that, for any interval of the form $(t_1, t_2]$ the following is true:*

$$W(t_1, t_2) \leq r_l(t_2 - t_1) + b_l$$

where $W(t_1, t_2)$ is the amount of traffic of virtual link scheduled by its virtual scheduler in the interval $(t_1, t_2]$ and r_l is the rate of the virtual link l.

Note that worst case session delays are directly related to the output burstiness. Even if the downstream scheduler is an ideal fluid-flow scheduler serving at a constant rate of r_l, it may take time equal to $\frac{b_l}{r_l}$ to clear its queue. It is therefore desirable to have the output burstiness as small as possible.

Before formally defining a latency rate server, we need the following definitions taken from [7].

Definition 3.2 (Session Busy Period) *A session i busy period is a maximal interval of time $(t_1, t_2]$ such that for any time $t \in (t_1, t_2]$, packets of session i arrive with rate greater than or equal to the allocated rate r_i of the session. If $A_i(t_1, t_2)$ is the amount of session i traffic arriving in the interval $(t_1, t_2]$ then*

$$A_i(t_1, t) \geq r_i(t - t_1), t \in (t_1, t_2) \tag{1}$$

Definition 3.3 (Latency Rate Server) *Let τ be the beginning of a busy period of a session and $W(\tau, t)$ be the amount of session traffic served by the server in the interval $(\tau, t]$. A server is in the class of latency rate servers if and only if for all times t after τ and until all the packets arriving in the busy period beginning at time τ are serviced,*

$$W(\tau, t) \geq r(t - \tau - \theta). \tag{2}$$

The parameter r is called the rate of the server and the parameter θ is called the latency of the server.

The model for a virtual network is shown in Figure 3. The traffic of virtual network is first scheduled by the

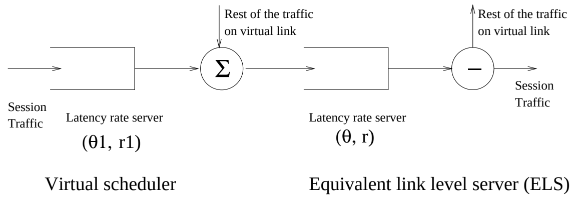

Figure 3: A logical model of a virtual network consisting of schedulers satisfying the latency rate property

virtual scheduler at virtual node. It is then aggregated with rest of the traffic in the virtual link and scheduled by a series of link level schedulers in the physical network until it reaches the next virtual node. We assume that the virtual scheduler is a latency rate servers with parameters (θ_1, r_1). We also assume that the link level schedulers are latency rate servers and model them as a single equivalent latency rate server (ELS) of parameters (θ, r) [7]. Let the rate of the virtual link be $r_l, (r_l \leq r)$, and its output burstiness be b_l.

Let $W_1^s(t_1, t_2)$ and $W_2^s(t_1, t_2)$ respectively denote the amount of session traffic served by the virtual scheduler and the ELS in the interval $(t_1, t_2]$. Similarly let $W_1(t_1, t_2)$ and $W_1(t_1, t_2)$ denote the amount of total traffic of the virtual link served by the virtual scheduler and the ELS in the interval $(t_1, t_2]$. Note that the session traffic served by the combination of the two servers is given by $W_2^s(.)$. Following theorem summarizes the main result.

Theorem 1 *The amount of session traffic served by the combination of servers shown in Figure 3 at any instant t_2 in the session busy period starting at t_1, is bounded by:*

$$W_2^s(t_1, t_2) \geq r_1(t_2 - t_1 - \theta_1 - \theta - \frac{b_l}{r}) \tag{3}$$

Proof: Let a session busy period at the virtual scheduler start at time t_1. Let t_2 be any instant such that the traffic arriving in this busy period is not completely serviced by the tandem of schedulers. In case ELS has no backlog of session traffic at t_2, we have:

$$W_2^s(t_1, t_2) = W_1^s(t_1, t_2) \tag{4}$$

Since virtual scheduler is a latency rate server and t_1 marks the beginning of a session busy period at virtual scheduler, we have:

$$W_1^s(t_1, t_2) \geq r_1(t_2 - t_1 - \theta_1) \tag{5}$$

After substituting the value of $W_1^s(t_1, t_2)$ from Eq. 4, the above equation may be rewritten as Eq. 3.

For the case when ELS has some backlog at time t_2, let t' be the instant before t_2 when the last session busy period at ELS started. Let the last packet served by ELS on or before t_2 arrived ELS at time t''. We claim that:

$$t'' \geq t_2 - \theta - \frac{b_l}{r} \tag{6}$$

We prove this by contradiction. Assume that

$$t'' < t_2 - \theta - \frac{b_l}{r} \tag{7}$$

Since the current session busy period started at time t' we have $t'' \geq t'$. From the latency rate property of ELS we have:

$$W_2(t', t_2) \geq r(t_2 - t' - \theta) \tag{8}$$

Since all of the above traffic arrived in the interval $(t', t'']$ from the virtual scheduler we have:

$$\begin{aligned} W_1(t', t'') &= W_2(t', t_2) \qquad (9)\\ &\geq r(t_2 - t' - \theta) \qquad (10)\end{aligned}$$

Substituting t_2 from Eq. 7 yields:

$$W_1(t', t'') > r(t'' - t') + b_l \tag{11}$$

Since $r_l \leq r$, this implies that the virtual scheduler violates the output burstiness constraint for the virtual link, which is impossible. Therefore our assumption must be wrong and Eq. 6 must be true.

The ELS serves all the traffic on a virtual link in first come first serve manner. So, if it services all its traffic arriving till the time t'', it also services all the session traffic arriving till time t''. Thus we have:

$$\begin{aligned} W_2(t_1, t_2) &\geq W_1(t, t'') \qquad (12)\\ \Rightarrow W_2^s(t_1, t_2) &\geq W_1^s(t_1, t'') \qquad (13)\end{aligned}$$

Eq. 5, 6 and 13 yield Eq. 3.

This shows that the tandem of two latency rate server as shown in Figure 3 is also a latency rate server with rate r and latency $\theta_1 + \theta + \frac{b_l}{r}$.

With the help of this property, the entire theory of latency rate servers may be used in designing bounded delay service in a virtual network. We now show how to design virtual schedulers which have bounded output burstiness and satisfy the latency rate property for individual sessions.

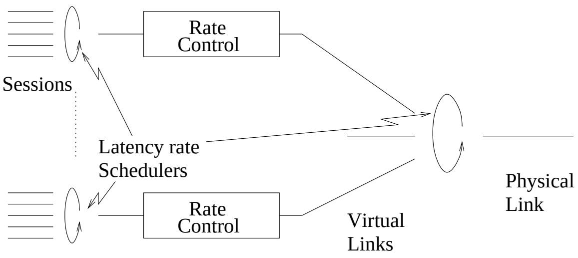

Figure 4: An generic latency rate scheduler with bounded output burstiness

3.1 Example Scheduler for Virtual Networks

Figure 4 shows a generic scheduler with bounded output burstiness. The virtual scheduler for each virtual link is a latency rate server like WFQ [8]. There is a rate control after each virtual scheduler which introduces inter-packet spacing equal to the packet size divided by the rate of the virtual link. Because of this rate control, the virtual link appears just like a physical link to the virtual schedulers. Because of discrete service boundaries, the rate control introduces an output burstiness of $\frac{1}{r_l} S_{max}$ where S_{max} is the size of the maximum packet on the virtual link and r_l is the rate of the virtual link. After the rate control, the packets are queued before the link level scheduler, which is again a latency rate server like WFQ. If WFQ is used then the latency of equivalent scheduler for a session of rate r_1 is $S_{max}(\frac{1}{r_1} + \frac{3}{r} + \frac{1}{l})$, where l is the rate of the physical link.

4 Best Effort Service in Virtual Networks

The output burstiness constraint limits the rate at which traffic may be sent on a virtual link. This reduces the extent of statistical multiplexing which could have been achieved if the entire network was a single layered physical network. The amount of statistical multiplexing is limited among the traffic going through the same virtual link. So, real-time sessions cannot use the excess bandwidth on other virtual links even if these virtual links are carried on the same physical link.

The bandwidth sharing among different virtual links may be achieved by the best effort traffic. Since there are no tight delay constraints on this traffic, excess of this traffic may be sent on a virtual link. The end-to-end congestion control algorithm adapts the sending rate of sessions to take all the excess bandwidth in the path. Because of output burstiness constraint on tunnels carrying real-time traffic, a different tunnel is needed for best effort traffic. At link level schedulers, separate best effort queues are maintained for all the virtual links. The queues containing real-time traffic are given higher priority over this queue. Best effort traffic is scheduled only if real-time traffic cannot be scheduled. Any work conserving fair packet scheduler like weighted fair queueing may be used at link level to ensure fairness to the non real-time traffic from different virtual links. An alternate approach could be, to mark the best effort traffic with a single bit and transport the real-time traffic and best effort traffic of a virtual link using the same tunnel. The link level scheduler can use this bit to separate best effort traffic from real-time traffic and give higher priority to real-time traffic.

5 Conclusions and Future Work

We have shown that traditional work conserving scheduler including the hierarchical schedulers are inappropriate for providing bounded delay service in generic virtual networks. The traffic of multiple sessions of a virtual network is tunneled through the physical network which treats the entire traffic of the tunnel as a single session. As a result, the physical network is not able to protect the well behaved sessions from misbehaving session which share the same virtual link. The problem can be solved by restricting the rate at which traffic may be sent on a virtual link. We introduce the concept of output burstiness which we use to extend the theory of latency rate servers [7]. We show that a latency rate server having bounded output burstiness may be used in a virtual network to provide end-to-end delay bounds for sessions in virtual network. This gives us a method to design a class of schedulers which may be used in virtual networks. We also presented an example scheduler in this class.

The output burstiness constraint limits the inter virtual link bandwidth sharing, some of which may be regained by creating separate tunnels for best-effort and real-time traffic. However, because of traffic aggregation, it may be difficult to carry out congestion management at a granularity of virtual network sessions. This is an interesting problem for future study.

We assume that packets do not carry any QoS specific information in their headers. It would be interesting to see how the packet tagging approaches like the differentiated services [11] extend to virtual networks.

References

[1] D. Shah and H. Holzbaur, "VPNs: security with an uncommon touch," *Data Communications International*, vol. 26, pp. 53–63, Sept. 1997.

[2] J. Vicente, "Genesis: Toward programmable virtual networking," in *Proceedings of OPENSIG'98 - Workshop on Open Signaling for ATM, Internet and Mobile Networks*, (Toronto), Oct. 1998.

[3] D. Florissi, "The NetScript approach to programmable networks," in *Proceedings of OPENSIG'98 - Workshop on Open Signaling for ATM, Internet and Mobile Networks*, (Toronto), Oct. 1998.

[4] S. Floyd and V. Jacobson, "Link sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, August 1995.

[5] H. Zhang and J. C. R. Bennett, "Hierarchical packet fair queueing algorithms," in *Proceedings of ACM SIGCOMM*, pp. 143–156, ACM, September 1997.

[6] K. Hamzeh, "Ascend tunnel management protocol - ATMP," Feb. 1997. RFC 2107.

[7] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," in *Proceedings of INFOCOM*, (San Fransisco, California), pp. 111–119, Mar. 1996.

[8] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *Journal of Internetworking Research and Experience*, pp. 3–26, Oct. 1990. Also in Proc. ACM SIGCOMM'89, pp 3-12.

[9] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control - the single node case," in *Proceedings of INFOCOM*, pp. 915–924, IEEE, May 1992.

[10] R. Garg and X. Chen, "RRR: Recursive round robin scheduler," in *Proceedings of IEEE Global Telecommunications Conference 1998*, (Sydeny), November 1998.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," Oct. 1998. Internet Draft <draft-ietf-diffserv-arch-02.txt>.