

Providing QoS for Real-time Applications

Teresa Tung, Jean Walrand
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
{teresat, wlr}@eecs.berkeley.edu

Abstract—This paper describes a framework for packet-switched networks that balances the needs of real-time and data applications. Routers use virtual queues to mark packets reflecting a measure of congestion. These marks are then used to control the admission of time-sensitive traffic and to regulate other traffic.

We present a theoretical analysis of this framework. We evaluate the performance characteristics of this approach in terms of overall link utilization and quality of service of the time-sensitive traffic.

We conclude with a discussion of simulation results and of an implementation on a test network.

Index Terms—Quality of service, Congestion Control.

I. INTRODUCTION

Internet accommodates many types of traffic. These types can be categorized as real-time and elastic. Elastic traffic refers to that of applications where the transmitted information is not time sensitive, but requires eventual correct delivery. Examples of applications that generate elastic traffic are email, web-browsing, file transfers (FTP), Telnet, and any application that works without timely delivery. Internet accommodates elastic traffic very well. Protocols like TCP control the transmission rate of elastic traffic and allow for reliable transmission.

Real-time traffic refers to that of applications where the transmitted information is only useful if it is received within a small delay. Such traffic often does not benefit from retransmissions of lost packets since they would arrive too late. Examples of applications that generate real-time traffic are voice over IP (VoIP), video conferences, and generally any application that requires small end-to-end delay. Internet does not cater well to the time constraints needed by such real-time applications. There is significant interest in developing an Internet that can accommodate real-time applications.

This paper studies a scheme to guarantee low end-to-end delay and loss rate for latency-sensitive traffic without reducing the Internet's ability to transport elastic traffic efficiently. The main goals are the following:

- Minimize the queuing delay of time-sensitive flows.

- Preserve the ability to serve elastic traffic.
- Provide fair resource allocation in the sense that neither real-time traffic nor elastic traffic monopolizes network resources.

The scheme has the following additional objectives:

- Minimize packet loss to make retransmissions less frequent and improve the throughput of the network.
- The scheme should admit a simple implementation that does not require complex signaling nor a centralized manager.

This paper is organized as follows. Section 2 describes the scheme. The scheme combines previously proposed ideas in a new way that enables the estimate of its performance. Section 3 provides a theoretical analysis of the performance of time-sensitive applications. Section 4 presents simulation results. Section 5 describes an implementation on a test network. Section 6 concludes with related work and deployment issues.

II. PROVIDING QOS TO EF TRAFFIC

This section describes a networking scheme that balances the needs of applications with real-time and elastic traffic.

Consider VoIP as a representative time-sensitive application. If the network cannot guarantee a small latency for a VoIP call, it should block that call. The network can block the call with a busy signal as in the public switched telephone network. This observation suggests that the network should control real-time traffic by some admission control policy based on the latency.

In contrast with real-time traffic, the network can adjust the transmission of elastic traffic with a congestion control policy and does not need to subject such traffic to admission control.

We study the feasibility of a scheme that uses packet marking based on virtual queues for feedback congestion control and admission control. When it gets congested, a router marks a packet by turning on a bit in the header of the packet. The Random Early Detection (RED) scheme is an example of a scheme used to determine the onset of

congestion and to mark packets [8]. Using virtual queues with reduced transmission rates to track the operations of the router with is another method of predicting congestion and marking packets. We study a virtual queue scheme which we describe later.

Our scheme subjects elastic traffic to a congestion control scheme that reacts to packet marking. TCP with ECN (Explicit Congestion Notification) is a congestion control scheme ideal for controlling elastic traffic [7]. TCP with ECN reacts to marked packets as if they were dropped packets, by decreasing of the transmission rate of the flow. Using virtual queues, TCP with ECN can decrease a flow’s transmission rate before the router queue builds up.

An admission control policy based on packet marking governs the time-sensitive flows. Gibbens and Kelly propose an admission control scheme for time-sensitive traffic where decisions are made in a distributed manner by the source and the destination [10]. We study the same admission control policy as Gibbens and Kelly: the systems we study respond to marked packets in the same manner. However our work differs in how routers mark packets. Gibbens and Kelly study a generalized packet marking scheme. We study a specific scheme using virtual queues for which we can provide more detailed analysis.

We investigate the admission control scheme using packet marking based on a specific configuration of virtual queues for which we offer a unique theoretical interpretation, simulation results, and experimental results.

These previously proposed schemes have been studied separately using a packet marking system. We study the combination of these congestion control and admission control schemes using a packet marking system based on a pair of virtual queues. Next, we describe the suggested behavior of the router, the source, and the destination.

A. Admission Control of Real-Time Flows

Real-time applications such as VoIP generate fixed size packets at fixed time intervals. We assume that real-time flows generate a fairly constant rate traffic.

The first T seconds of a real-time flow are a trial period. The source uses this trial period to determine how adding this new flow perturbs the existing flows in the network. If congested, routers mark these trial packets. The destination sends ACKs to the source to acknowledge packets received without marks. Thus the source knows how many packets are successfully transmitted without marks. The source has a fixed threshold D . If fewer than D test packets are dropped or marked, then the new flow is admitted for its entire duration. Otherwise the flow is not admitted and the source aborts it. The trial period is short;

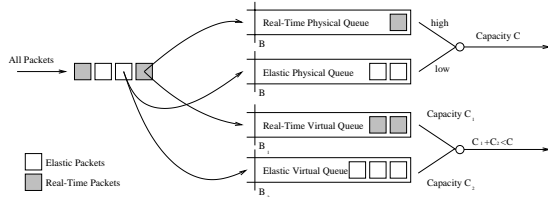


Fig. 1. Our Configuration: This configuration has two physical queues and two virtual queues.

consequently, trial flows will not perturb accepted real-time flows. We discuss below how using virtual queues can ensure that the network keeps some safety margin on its available resources to make room for these short-lived trial flows.

The trial period measures if the path has enough spare bandwidth to accept a new real-time flow. Assume that each real-time flow requires at most a small fraction, say 5%, of each link’s bandwidth. Assume also that at most 5 real-time calls occur in T seconds. If the virtual queues start marking packets when the link utilization exceeds 75%, we can expect this scheme to work. Our analysis confirms this basic intuition for more realistic assumptions.

B. Congestion Control of Elastic Flows

Our scheme uses TCP with ECN to transmit elastic flows and to control congestion. With TCP with ECN, sources slow their transmission rates when a packet is marked. A router marks packets when the utilization exceeds some level, triggering the sources to slow their transmission rates before packet drops occur. In this way, the scheme limits the aggregate rate of elastic traffic.

C. Virtual Queues in the Router

In our scheme, a router marks packets at the onset of congestion, which is determined by the use of virtual queues. Specifically, the router maintains two packet queues: one for real-time traffic, the other for elastic traffic. Moreover, the router maintains one virtual queue for each traffic type. When a packet arrives, it enters the appropriate physical packet queue based on its type (if there is sufficient space in that queue). At this time, an imaginary packet is enqueued in the virtual queue of the same type regardless of the length of the virtual queue. If the new arrival overflows the virtual buffer, its corresponding real packet in the physical queue is marked. The router serves (with rate C) the real-time packet queue with non-preemptive priority over the elastic packet queue. The virtual real-time queue is served with rate C_1 whenever nonempty. The elastic virtual queue is served with rate

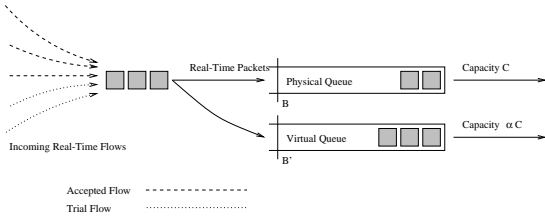


Fig. 2. The Real-Time System from our configuration.

$C_1 + C_2$ whenever the real-time virtual queue is empty and with rate C_2 when it is not. The rates C_1 and C_2 are chosen so that $C_1 + C_2 = \alpha C$ for some $\alpha < 1$. Figure 1 illustrates this arrangement.

D. Meeting the Goals

With its packet marking scheme, the routers reduce the chance that the physical queue fills up, thus reducing the likelihood of large queuing delays. The routers further minimize the queuing delay of real-time packets by serving them with strict priority. Also, the scheme limits the bandwidth that the real-time traffic can use while guaranteeing an available bandwidth for it.

The scheme is decentralized, with decisions made at the edges of the network on a per flow basis. The network does not need to perform any accounting of accepted flows or available resources. The source and destination perform admission decisions and congestion control. Signalling uses a simple scheme of packet marking. The virtual queue pair can be implemented as counters as in [19].

III. THEORETICAL JUSTIFICATION

This section provides theoretical analysis of the performance of time-sensitive applications. The main issue is to bound the end-to-end delay of real-time packets.

A. One Router

We assume that a flow is admitted only if none of its test packets are marked. We first compute the bound for one router.

The router serves real-time packets with strict priority over classic packets. So for the purposes of analyzing the delay of real-time packets, we can ignore the elastic traffic in Figure 2. Accordingly, in the discussion below we focus exclusively on real-time flows.

The virtual queue remains empty so long as the total incoming rate of the traffic is less than αC . Otherwise, the virtual queue fills up and marking occurs. Let X and Y be the rates of admitted and trial flows, respectively. Marking occurs if $X + Y > \alpha C$. Hence, as soon as $X = \alpha C$, the

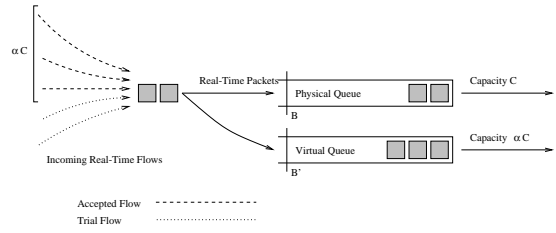


Fig. 3. Real-time System: The accepted flows are transmitting at αC .

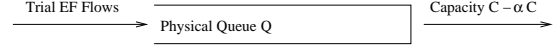


Fig. 4. This is an equivalent representation of the real-time physical queue when the accepted flows are transmitting at αC .

system—shown in Figure 3—no longer accepts new trial flows.

To compute a bound for the delay, we assume that the accepted flows are always transmitting at the maximum sending rate, $X = \alpha C$. Thus the trial flows experience capacity $c = C - \alpha C$ and computing the bound for one router reduces to computing the bound for a router with capacity c fed only by trial flows shown in Figure 4.

We use a large deviation and effective bandwidth argument that is used in [18] to analyze this system. Consider the following model. Fix N to be the number of possible trial flows. Model each trial flow as an on-off Markov fluid source with the following transition rates: $Q_{off\ to\ on} = \frac{\lambda}{N}$ and $Q_{on\ to\ off} = \frac{1}{T}$. Each traffic source sends at a constant rate R when on and at rate 0 when off.

Let Q be the amount of traffic in the physical queue found by the typical arriving packet. From [18],

$$P(Q > X) \leq e^{-X\delta} \quad (1)$$

holds when the sum of effective bandwidths of the arrivals is less than the capacity of the link c . In this case, the effective bandwidth for each source is $\frac{h(\delta)}{\delta}$ where

$$h(\delta) = \frac{1}{2}(-a(\delta) + \sqrt{a(\delta)^2 - 4b(\delta)}) \quad (2)$$

$$a(\delta) = Q_{off\ to\ on} + Q_{on\ to\ off} - \delta R \quad (3)$$

$$b(\delta) = -\delta Q_{off\ to\ on} R \quad (4)$$

The sum of the effective bandwidths is less than c when

$$\delta \leq \frac{N(\lambda R - c(\lambda/N + 1/T))}{c(c - NR)}. \quad (5)$$

As N goes to infinity, this system of on-off Markov sources approaches an $M/M/\infty$ system and

$$\delta = \frac{1}{TR} - \frac{\lambda}{c}. \quad (6)$$

Hence we have that the probability that the queueing length at a router exceeds X is less than or equal to $e^{-X\delta}$ where $\delta = \frac{1}{TR} - \frac{\lambda}{C-\alpha C}$. The following example shows the use of this bound.

B. Example

Consider an example of a router under high demand of VoIP flows. Each VoIP source generates packets of 80 bytes every 20ms. Suppose there are 10000 users, 15% of users are active at a time, and the average duration of a call is 200 seconds. Let the capacity of the link be 10Mbps and the average duration of a trial period be T seconds.

If all the calls are accepted, then the cumulative sending rate is $15 \times 200 \times 32000 = 96Mbps \gg 10Mbps$. Congestion definitely occurs resulting in poor QoS for every flow. $15 \times T \times 32000$ is the average number of test packets in the system at any given time. In order to be sure that the packets from the trial calls do not disturb the system, we must fix the capacity of the virtual queue to be

$$c \leq 10Mbps - (15 \times T \times 32000).$$

Let Q denote the queue length of the router. Table I shows various values of $P(Q > X)$ for different values of t , for $c = 3$ Mbps, and for $T = 1$. These bounds verify that large delays are unlikely even under high demand.

TABLE I
PROBABILITY OF DELAY THROUGH ONE ROUTER.

X (bits)	Delay (seconds)	Upper Bound for $P(Q > X)$
1e4	0.001	0.7691
5e4	0.005	0.2691
1e5	0.01	0.0724
5e5	0.05	1.9947e-6
1e6	0.1	3.9790e-12
5e6	0.5	9.9735e-58

We simulate this scenario using Network Simulator version 2 (ns-2) to verify the computed bound [25]. Our scenario spans 600 seconds. Initially the system is empty and the initial calls are all be accepted. In fact, all the calls within the first 5 seconds are accepted. We skip the first 100 seconds before collecting the following data:

- The largest queueing delay experienced by any packet of an accepted flow is 0.266 ms.
- The largest average queueing delay of an accepted flow (computed over all the packets of an accepted flow) is 0.178 ms.
- The average number of accepted flows maintained (after the initial period) is 81 flows; so the accepted

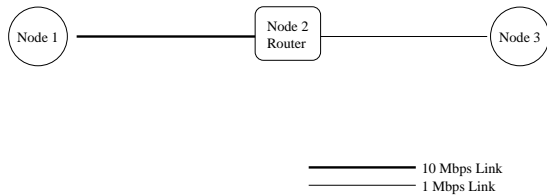


Fig. 5. Single Router Network Topology

calls generates traffic at a rate of 2.592 Mbps. The rate from the test packets should be 0.48 Mbps. The total VoIP traffic entering the router is limited so that the delay through the router is very small even under high demand.

These results verify that the bound computed is indeed an upper bound and that the admitted VoIP flows receive the necessary quality of service even when demand is high. Specifically we verify that the test packets do not overwhelm the router causing large delay to the packets of admitted flows.

C. Network of Routers

Now we extend this result to a network of routers for which traffic is fed only in one direction so that there are no loops. At router i , the real-time physical queue is served with capacity C_i and the virtual queue is served with capacity $\alpha_i C_i$. Let $c_i = C_i - \alpha_i C_i$, and let Q_i be the amount of traffic in the physical queue found by the typical arriving packet.

From [24], if for each router i the sum of the effective bandwidths of the original trial flows is less than the available capacity c_i , then at each router have that

$$P(Q_i > X) \leq e^{-X\delta}. \quad (7)$$

For each router, one only needs to consider the original arrival process of the incoming trial flows rather than the arrival process seen at the router within the network.

IV. SIMULATIONS

In this section we present ns-2 simulation results to verify that the proposed scheme is implementable. We consider simulations involving only one router to study the role of the virtual queues as control mechanisms.

We show how the pair of virtual queues control the amount of elastic and real-time traffic allowed through a router. We specifically verify that the delay through the router is small for real-time traffic even with the presence of elastic traffic. Consider the network topology shown in Figure 5. Bandwidth is as indicated being 10 Mbps on the link between nodes 1 and 2 and 1 Mbps on the link

Flow ID	Start Time
ftp0(TCP)	0.0
ftp1(TCP)	1.0
ftp2(TCP)	46.0
voice1(64kbps)	5.0
voice2(64kbps)	8.1
voice3(64kbps)	39.0

TABLE II
TRAFFIC SENT OVER SINGLE ROUTER NETWORK.

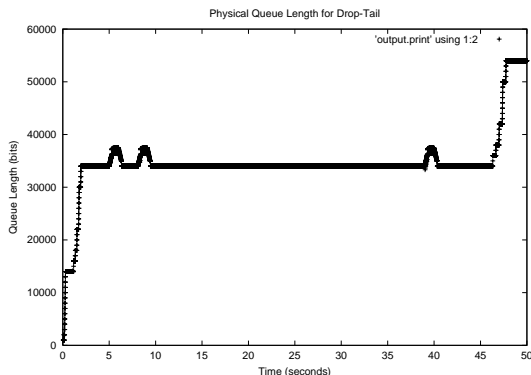


Fig. 6. Physical queue length for Drop-tail router without virtual queues.

between nodes 2 and 3. Propagation delay is 10 ms on all the links. Suppose that the traffic listed in Table II is sent over this network in the following simulations.

We first examine this scenario with a traditional drop-tail router in node 2. Figure 6 shows the resulting physical queue length of the drop-tail router. Even the mode value of 34kb results in a queuing delay of 34 ms. The maximum value 54kb results in a queuing delay of 54ms. The bandwidth used by the TCP flows grows until a packet is dropped at which time the queue length is quite long. Thus voice packets experience a large queuing delay. Additionally, this setup accommodates every voice flow so that it is possible for the voice flows to use all the available bandwidth to the detriment of the TCP flows. This simulation shows that without special provisions we cannot guarantee the timely delivery of voice packets or the fair allocation of bandwidth between data flows and voice flows.

With our scheme, we can use the virtual queue pair to limit the bandwidth used by admitted voice flows and to limit the bandwidth used by TCP. Suppose we wish to limit the maximum number of voice flows passing through the router to n . Set the capacity C_1 of the real-time virtual queue to be greater than the rate of n calls but less than the

rate of $n + 1$ calls. Then set the buffer size such that the buffer fills up within the trial period if the flows through the router exceed n . Choose the capacity C_2 of the elastic virtual queue such that the sum of $C_1 + C_2$ is approximately the desired amount of total traffic one wishes to accommodate.

For example, suppose we wish to limit the number of admitted voice flows passing through the router to one flow. Let the capacity of the real-time virtual queue $C_1 = 100$ kbps and the buffer length $B_1 = 25$ kb. Let the capacity of the elastic virtual queue $C_2 = 800$ kbps and the buffer length $B_2 = 300$ kb. The length of the real-time virtual queue is shown in Figure 7. Flow voice1 is admitted for the duration of the simulation. The two spikes at 8.1 seconds and 39 seconds correspond to when voice2 and voice3 begin their trial periods respectively. Both voice2 and voice3 are not admitted and cease transmission so that only one voice flow is admitted through the router.

The length of the elastic virtual queue for data packets is shown in Figure 9. The virtual queue length hovers near B_2 suggesting that the presence of TCP flows ensures that the link utilization is near the total capacity of the virtual queues.

Figure 8 shows the length of the real-time physical queue over time. Notice that the length of the physical queue for voice packets is small. Figure 10 shows the length of the elastic physical queue over time. The elastic queue does not grow without bound so elastic packets are served despite the fact that real-time packets are served with strict priority. In fact, the virtual queue pair limits the proportion of transmission of each type of traffic through the router.

V. EXPERIMENTAL NETWORK

This section describes an implementation of our scheme on a test-bed of notebook computers. This test-bed serves as a “real-world” proof of concept to verify the correctness of the scheme in a real-world setting.

We use IBM Thinkpad laptops and wired Ethernet cards to implement our scheme. These laptops run a modified version of the Linux kernel. This section shows an example of how to deploy our routers in a small network.

We study a situation where the edge routers implement our scheme but the internal routers are ordinary drop-tail. Additionally, consider the case when the internal network links at the drop-tail routers are slower. Consider the small network of routers shown in Figure 11. Node C and node D are routers that implement our scheme with 100Mbps links, and node E and node G are ordinary drop-tail routers with 10Mbps links. Table III shows the traffic sent.

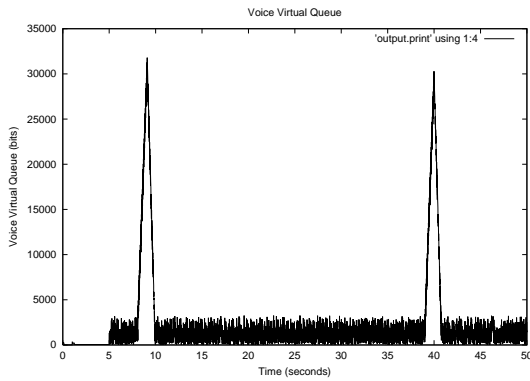


Fig. 7. Real-Time Virtual Queue Length when $C_1 = 100\text{kbps}$

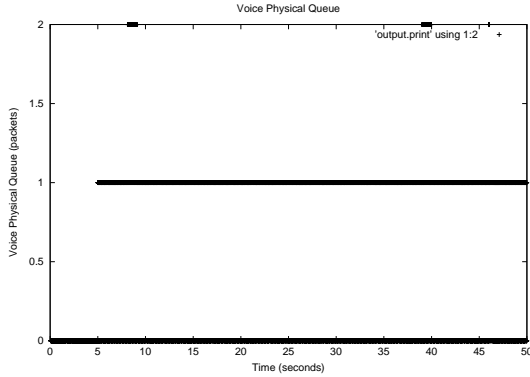


Fig. 8. Real-Time Physical Queue of our configuration

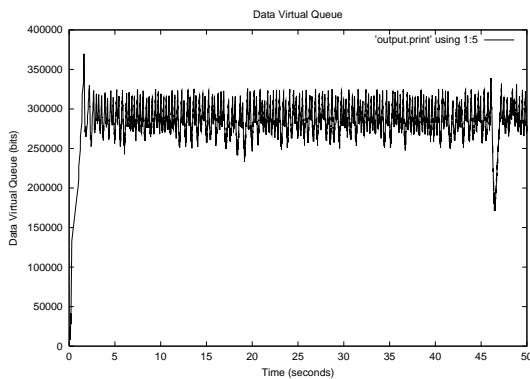


Fig. 9. Elastic Virtual Queue Length when $C_2 = 800\text{kbps}$

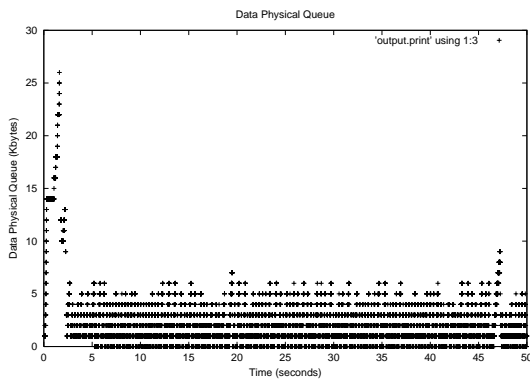


Fig. 10. Elastic Physical queue of our configuration

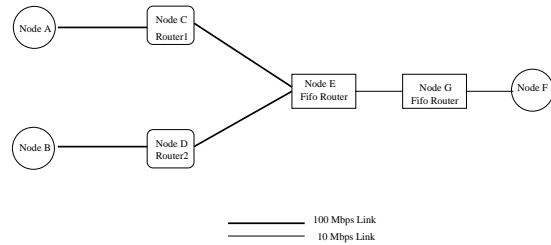


Fig. 11. Laptop Topology for Small Network Experiments

TABLE III

THE FTP FLOWS (FTP0, FTP1, FTP2) ARE DATA FLOWS TRANSMITTED USING TCP WITH ECN. FLOW VIDEO-HIGH IS A HIGH-QUALITY VIDEO FLOW THAT TRANSMITS AT 512KBPS, FLOW VIDEO-LOW IS A LOW QUALITY VIDEO FLOW THAT TRANSMITS AT 256KBPS, AND FLOW VOICE IS A VOIP FLOW THAT TRANSMITS AT 64KBPS. ALL FLOWS LAST FOR THE DURATION OF THE SIMULATION EXCEPT FOR REAL-TIME FLOWS THAT ARE NOT ADMITTED. THE DESTINATION OF ALL FLOWS IS NODE F.

Flow ID	Flow Type	Start Time (seconds)	Source
ftp0	TCP	0.0	A
ftp1	TCP	1.0	A
ftp2	TCP	46.0	A
video-high	real-time(512 Kbps)	5.0	A
video-low	real-time(256 Kbps)	8.1	A
voice	real-time(64 Kbps)	39.0	A
ftp3	TCP	0.0	B
ftp4	TCP	0.0	B
video-low1	real-time(256 Kbps)	0.0	B
voicel	real-time(64 Kbps)	0.0	B

Since the natural bottlenecks occur at the drop-tail routers, without special provisions those routers will be overwhelmed resulting in packet drops and long-queueing delays. The special routers at nodes C and D must conspire to limit the total traffic through the drop-tail routers. Let the virtual queue pair at node C be set so that the capacity of the real-time virtual queue is 0.6Mbps and the capacity of the elastic virtual queue is 4Mbps. Let the virtual queue pair at node D be set so that the capacity of the real-time virtual queue is 0.6Mbps and the capacity of the elastic virtual queue is 3Mbps.

Under such a setup, the admitted real-time traffic through nodes C and D never exceeds the capacity allocated by the virtual queue. Indeed, node C admits flows video-high and voice, but not flow video-low. Node D admits both flows video-low1 and voicel. Furthermore, the cumulative TCP bandwidth allowed approximately equals the capacity available at the virtual queue. Thus in the ab-

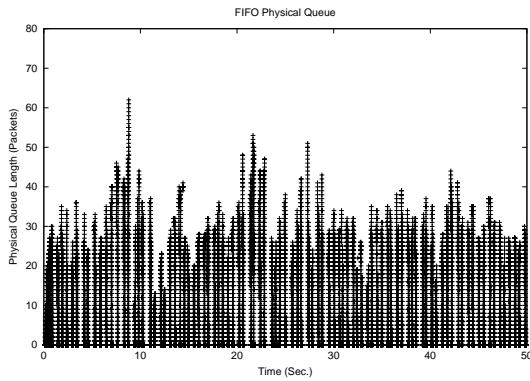


Fig. 12. FIFO Queue Length at Node E

sence of real-time traffic, node C admits approximately 4.6Mbps of TCP traffic and node D admits approximately 3.6Mbps. The total bandwidth allowed for all traffic is 8.2Mbps which is less than the capacity at the drop-tail routers.

Under such conditions, the resulting physical queues of nodes C and D are near empty because the virtual queue parameters are provisioned to limit the incoming traffic to be much less than their actual capacities. However, the introduction of the artificial bottlenecks is necessary to keep the queueing delay low at the internal drop-tail routers. Figure 12 shows the length of the physical queue at the drop-tail router at node E. The reported jitter of the accepted real-time flows is always less than 28ms as shown in Figures 13 and 14. The end-to-end delay experienced by real-time flows is limited to be less than 100ms resulting in acceptable QoS for all accepted flows.

We limit the queueing delay of the real-time packets by limiting the total bandwidth entering the network. However we want to provide QoS while maintaining good link utilization at the bottleneck link. We plot the link utilizations from only the TCP traffic. As expected, the usage at nodes C and D is low compared to the total link capacity of 100 Mbps as shown in Figures 15 and 16. However, at node E's drop-tail router, in Figure 17, the link utilization hovers near 7Mbps.

Furthermore, the drop-tail routers also accommodate the real-time flows. Node C admits flows video-high and voice, and node D admits flows video-low1 and voice1. With the admission these flows, there is a cumulative real-time transmission rate of 1152 Kbps. The total bandwidth used at this drop-tail router is 8.12Mbps which is close to the 8.2Mbps allocated by the routers at nodes C and D. The drop-tail router has a line speed of 10 Mbps so link utilization is approximately 80%.

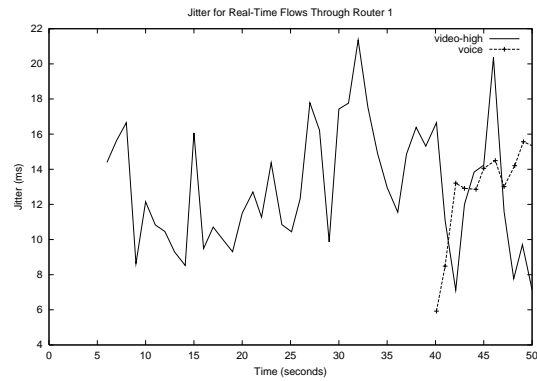


Fig. 13. Jitter of Accepted Real-Time Flows Through Node C

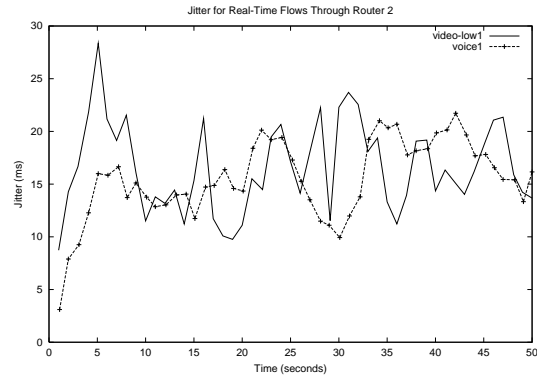


Fig. 14. Jitter of Accepted Real-Time Flows Through Node D

VI. DISCUSSION

A. Deployment Issues

Our assumption is that routers with our scheme will be placed at strategic locations within the network. For instance, one can put them at bottleneck links, at the edges of a network, or at the ends of a MPLS pipe.

Additionally, we cannot assume that applications will be prepared to react to packet marking as defined. This report leaves an unanswered question about policing, how to enforce that flows respond to packet marking. Not all applications may use the transport protocols necessary to react to packet marking. Future work involves investigating how policing and deployment should be done.

In the experiments, parameters for the virtual queues are chosen to accommodate a desired load of traffic. We do not address how to choose the appropriate load. There are many effects to explore. For example, round-trip time determines the reaction time of our scheme. In choosing the desired load of traffic, one needs to account for the delay in the reaction time and provision to accommodate the traffic that arrives in that delay.

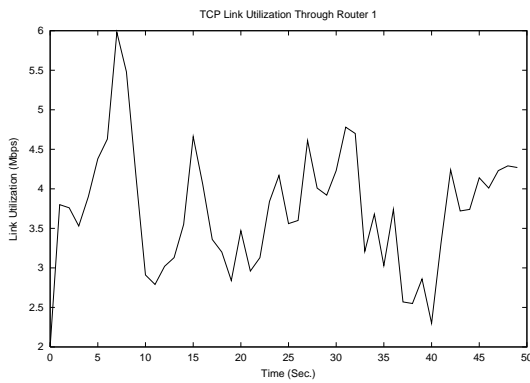


Fig. 15. TCP Link Utilization Through Router 1

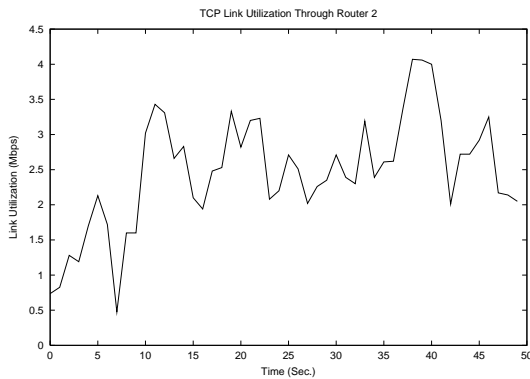


Fig. 16. TCP Link Utilization Through Router 2

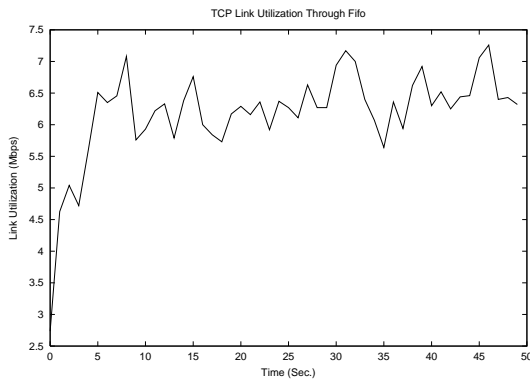


Fig. 17. TCP Link Utilization Through FIFO Router at Router 3

B. Related Work

There exists much work using virtual queues and packet marking to control networking traffic. Packet marking with one physical queue and one virtual queue is the most commonly analyzed. [17] and [16] show why a packet marking scheme is desirable especially in conjunction with TCP with ECN [15]. Alternative theoretical analysis for similar schemes is provided in [9], [10], and [12]. The packet based admission control used in our scheme was first proposed in [10]. Some variations we did not

include are as follows: [19] uses adaptive parameters to adapt the virtual queue to better respond to the varying of traffic loads over time offering better control and link utilization. [2] and [11] analyze the effects of pricing according to packet marks for multi-class QoS.

C. Conclusions

This paper shows how the use of virtual queues can control the amount of time-sensitive and data traffic that is allowed through a router and hence through a network. The suggested scheme is simple in its implementation aspects as well as its decentralized structure. By controlling the amount of traffic allowed, the routers allocate enough throughput to ensure low queuing delay providing QoS to time-sensitive applications thereby making transmission of time-sensitive applications over packet-switched networks a feasible option.

REFERENCES

- [1] <http://dast.nlanr.net/Projects/lperf/>
- [2] J. Alvarez and B. Hajek, *On Using Marks for Pricing in Multiclass Packet Networks to Provide Multidimensional QoS* <http://www.comm.csl.uiuc.edu/hajek>
- [3] Mark Allman and Aaron Falk, *On the Effective Evaluation of TCP* ACM Computer Communication Review, 29(5), October 1999.
- [4] C. Chuah and R. Katz, *Network Provisioning and Resource Management for IP Telephony* Report No. UCB/CSD-99-1061, August 1999.
- [5] Amire Dembo and Ofer Zeitouni, *Large Deviations Techniques and Applications* Jones and Bartlett Publishers, 1993.
- [6] Rick Durrett, *Essentials of Stochastic Processes* Springer-Verlag New York, Inc. 1999.
- [7] S. Floyd, *TCP and Explicit Congestion Notification* ACM Computer Communication Review 24, p 10-23.
- [8] S. Floyd and V. Jacobson, *Random Early Detection Gateways for Congestion Avoidance* IEEE/ACM Trans. Networking, 1:397-413, 1993. <ftp://ftp.ee.lbl.gov/papers/early.pdf>
- [9] R. J. Gibbens and F. P. Kelly, *A note on packet marking at priority queues* IEEE Transactions on Automatic Control.
- [10] R. J. Gibbens and F. P. Kelly, *Distributed connection acceptance control for a connectionless network* <http://www.statslab.cam.ac.uk/frank/dcac.html>
- [11] R. J. Gibbens and F. P. Kelly, *Resource Pricing and the Evolution of Congestion Control* <http://www.statslab.cam.ac.uk/frank/evol.html>
- [12] R. J. Gibbens, P. B. Key, and S. R. E. Turner, *Properties of the Virtual Queue Marking Algorithm* <http://www.statslab.cam.ac.uk/Reports>
- [13] Donald Gross and Carl Harris,

Queuing Theory

John Wiley & Sons, Inc. 1998.

- [14] Paul Hurley, Jean-Yves Le Boudec, and Patrick Thiran,
A note on the Fairness of Additive Increase and Multiplicative Decrease
- [15] V. Jacobsen,
Congestion Avoidance and Control
Proc. ACM SIGCOMM '88, p. 314-329.
- [16] Tom Kelly,
The Case for a New IP Congestion Control Framework
- [17] Tom Kelly,
An ECN Probe-Based Acceptance Control
- [18] George Kesidis, Jean Walrand, and Cheng-Shang Chang,
Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources
IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August 1993.
- [19] Srisankar Kunniyur and R. Srikant,
Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management
SIGCOMM'01, August 27-31, 2001.
- [20] Srisankar Kunniyur and R. Srikant,
End-to-End Congestion Control Schemes: Utility Functions, Random Losses, and ECN Marks
INFOCOM 2000.
- [21] Larry L. Peterson and Bruce S. Davie,
Computer Networks, Second Edition
Academic Press, 2000.
- [22] V. Paxson and S. Floyd,
Difficulties in Simulating the Internet
IEEE/ACM Transactions on Networking, February, 2001.
- [23] Adam Shwartz and Alan Weiss
Large deviations for performance analysis: queues, communication, and computing
Chapman and Hall, 1995.
- [24] Jean Walrand and Pravin Varaiya,
High-Performance Communication Networks, Second Edition
Morgan Kaufmann Publishers, 2000.
- [25] www.isi.edu/nsnam/ns/