A Rate Based Back-pressure Flow Control for the Internet

Carlos M. Pazos and Mario Gerla

Computer Science Department
University of California, Los Angeles
405 Hilgard Ave., Los Angeles, CA 90024
{pazos,gerla}@cs.ucla.edu

ABSTRACT

The Internet has traditionally relied on end-to-end congestion control performed at the transport layer by TCP. In this paper, we discuss the limitations of this approach to address the large number of flows and the large delay-bandwidth product scenarios typical of next generation Internets. As a result, we propose a link layer back-pressure flow control which can be applied to Internet backbones over ATM. More precisely, we use the ABR service and flow control, where routers become virtual sources (VSs) and virtual destinations (VDs) for the ABR control loop. We introduce a VS/VD "behavior" that implements a rate based back-pressure flow control and that addresses max-min fairness. Finally, for the case of edge router connectivity to the ATM network (as opposed to ATM all the way to the host), we discuss approaches that can be used to convey ATM rate control indications to TCP sources connected to legacy LANs.

1 Introduction

Following the rapid evolution of Internet services in recent years, the Best Effort (BE) service is no longer adequate to address the requirements of new services, some of which have real time (RT) constraints and thus require resource commitment from the network in order to implement Quality of Service (QoS) guarantees. The IEFT Internet Integrated Services (ISS) working group is investigating new services and features to allow the Internet to transport multimedia traffic [1].

As a result, we are bound to experience an ever increasing demand for transmission resources, already a reality given the exponential growth of current Internet traffic. The conventional approach to address this problem has consisted on adding faster links to Internet backbones, upgrading the available pool of resources. While this course of action satisfies the Internet cravings for bandwidth, it also makes it more difficult to control traffic and prevent congestion. With faster transmission rates, we are unavoidably faced with problems due to large delay-bandwidth products on backbones links (a huge number of packets is in transit on these links).

The RT traffic can reserve resources and it does receive priority service on routers in the backbones. For this traffic, appropriate resource allocation and efficient admission control are probably enough to avoid the effects of congestion. On the other hand, the bulk of BE traffic is transported over the backbones under TCP window flow control applied to individual sessions on the edges of the network. Namely, TCP sources adjust their transmission rates in reaction to congestion in the network, which is detected through packet loss. Hence, actions to remedy congestion are only taken when congestion sets in. The associated reaction time is in the order of end-to-end round trip times (RTTs) (excluding timer granularity issues).

In addition, the majority of TCP sessions actually transmit only a few kilobytes of data [2]. For these short sessions, TCP never achieves steady state and thus cannot exercise proper congestion control. Recall that TCP probes for congestion by increasingly sending more packets on every RTT (i.e., increasing the tranmission window) until packets eventually get dropped and the transmission window is reduced. With these adaptive window adjustments TCP tries to achieve fair sharing of bandwidth for each session at steady state. However, for short-lived sessions, this steady state condition is never reached.

The dynamics of TCP with many flows have been recently analyzed in [3], where it is argued that TCP alone cannot fairly and effectively control a very large number of sessions, an increasingly frequent scenario in the Internet. Two main reasons are cited: the scaling limitations of window flow control approaches [4], and the aggressive TCP window increase policy during slow-start. Increasing the bandwidth available on backbone links and/or increasing buffering space on routers can help accommodate the traffic of a very large number of sessions [3], but this is not a costeffective solution.

In this paper we describe a rate based back-pressure flow control that acts on the **aggregate** traffic between each pair of routers. Each router continuously notifies its neighbors of the rate it can accept from them. This back-pressure control propagates all the way to the ingress edge routers. Hence, the response time is in the order of half the RTT between the edge router and the point of congestion. Furthermore, the back-pressure is applied to the aggregate traffic. Thus, it is actually irrelevant how many flows traverse the links or how long lived these flows are.

Our flow control approach builds on the work in [5], where the use of the ATM ABR service was suggested for Internet backbones. Namely, the links interconnecting IP routers are ABR VPs, rather than the more conventional CBR VPs. With ABR, the backbone VPs are no longer restricted to the CBR peak rate allocation. Namely, the ABR flow control applied to backbone VPs allows the Internet traffic carried over them to recover the bandwidth leftover by other ATM connections and the bandwidth unassigned on the ATM links traversed by the VPs. Hence, we improve ATM resource utilization and Internet throughput. In this paper, we acknowledge this advantage but we take a step further implementing a rate based congestion control scheme.. One considerable advantage of this approach is that it is almost entirely implemented at the ATM layer on routers and hence it requires no modifications to either TCP or IP protocols.

The balance of the paper is organized as follows. In section 2 we describe the congestion problem we address in this paper and in section 3 we present the network scenario we consider for the remainder of the paper. In section 4 we review current approaches for congestion control in Internet backbones and we introduce our back-pressure flow control scheme; in section 5 we present simulation results. In section 6 we describe the transfer of rate indications to TCP sources not directly connected to the ATM backbone. Finally, in section 7, we make some concluding remarks.

2 Congestion in the Backbone

As Figure 1 illustrates, congestion occurs in Internet backbones when the incoming traffic ($\lambda_{ad} + \lambda_{bd}$) feeding a router VP ($VP_d$) exceeds the VP capacity (C_d). As a result, packets may be dropped, impacting effective network utilization and compromising performance. In order to avoid such losses, we need to selectively slow down the traffic flows λ_{ad} and λ_{bd} at the respective sources. This is clearly a hard problem to solve unless an end-to-end rate based (or credit based) feedback mechanism such as the ABR flow control [6] is employed. However, the direct application of ABR on individual sessions is made difficult by the large delay-bandwidth product associated with backbone links, the large Round Trip Times (RTTs) to which such feedback congestion indications may be subjected (response time), and the extremely large number of sessions, each with a very bursty behavior. The Explicit Congestion Notification (ECN) approach proposed for IPv6 in [7] is another attempt to provide feedback control from the network layer (IP) to TCP.

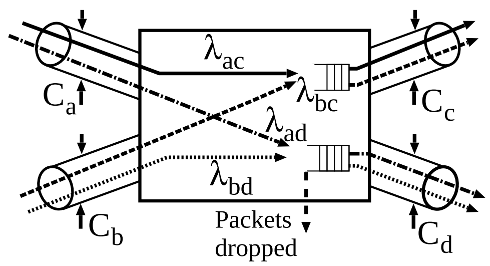


Figure 1: A congested router.

The current approach for congestion control in backbones relies on the TCP window flow control, a reactive mechanism. A very nice property of this approach is that only those sessions suffering packet losses (experiencing congestion) reduce their transmission rate (window). In Figure 1, for instance, the sessions using VP_c are unaffected by the congestion on VP_d and the TCP flow control does not act on them. However, if the flow λ_{ad} alone is responsible for congesting VP_d, the flow λ_{bd} is also penalized in the process and the TCP mechanism cannot selectively slow down the λ_{ad} flow. Namely, TCP alone cannot ensure fairness [3]. Furthermore, for satisfactory TCP operation the routers should have buffering proportional to the average end-to-end delay-bandwidth product or proportional to the number of active flows [3]. Thus, queueing delay becomes an issue.

Hence, maybe the best we can do to alleviate the congestion on the router of Figure 2 and to avoid packet losses is to selectively back-pressure the flow λ_{ad}. Such preventive action is more appropriate to address the large delay-bandwidth product issue if we make the back-pressure propagate from the congestion point all the way to the sources. However, this is very difficult to implement since routers do not have end-to-end information. In our approach, we back-pressure the whole input port traffic and we discuss when this is actually alright and discuss the performance penalties we pay when it is not.

3 The Network Scenario

In this paper we address the congestion control problem in next generation Internets and we describe a solution for the IP over ATM backbone scenario. This is a realistic assumption, since it is common practice today. Nonetheless, we are not actually assuming ATM connectivity end-to-end, rather, we assume that the sources and sinks of Internet traffic reside on legacy LANs. An edge router with ATM connectivity functions as ingress router for the traffic crossing from the LAN to the backbone (and vice versa), see Figure 2. Hence, in our model, the edge routers are the effective sources and sinks of Internet traffic. In section 6 we survey some approaches to convey the back-pressure rate notifications to the actual sources.

As Figure 2 also illustrates, we consider the Internet Integrated Services (IIS) scenario; namely, the backbone transports Best Effort (BE) and Real Time (RT) traffic. In our simulation studies, we consider the BE traffic being TCP traffic and the RT traffic being video traffic sent over UDP. We placed RT and BE sources and sinks in different legacy LANs in Figure 2 for simplicity only since we are not concerned, for the moment, with the individual traffic flows. Our interest is on the aggregate streams for each class because we exercise back-pressure on the aggregate traffic sent between routers. The RT traffic is given priority over the BE traffic on edge and backbone routers by a Class Based Queueing (CBQ) [8] scheduler. We also assume that RSVP and some form of flow admission control are employed to allocate the RT traffic flows appropriate bandwidth on VPs such that RT constraints can be guaranteed. Finally, we use the ABR service on the VPs interconnecting the routers in Figure 2 where a Minimum Cell Rate (MCR) is allocated to implement bandwidth guarantees to the VP traffic. This approach, as described in [5], improves ATM network utilization and throughput for the Internet traffic as compared to the more common approach of using the CBR service.

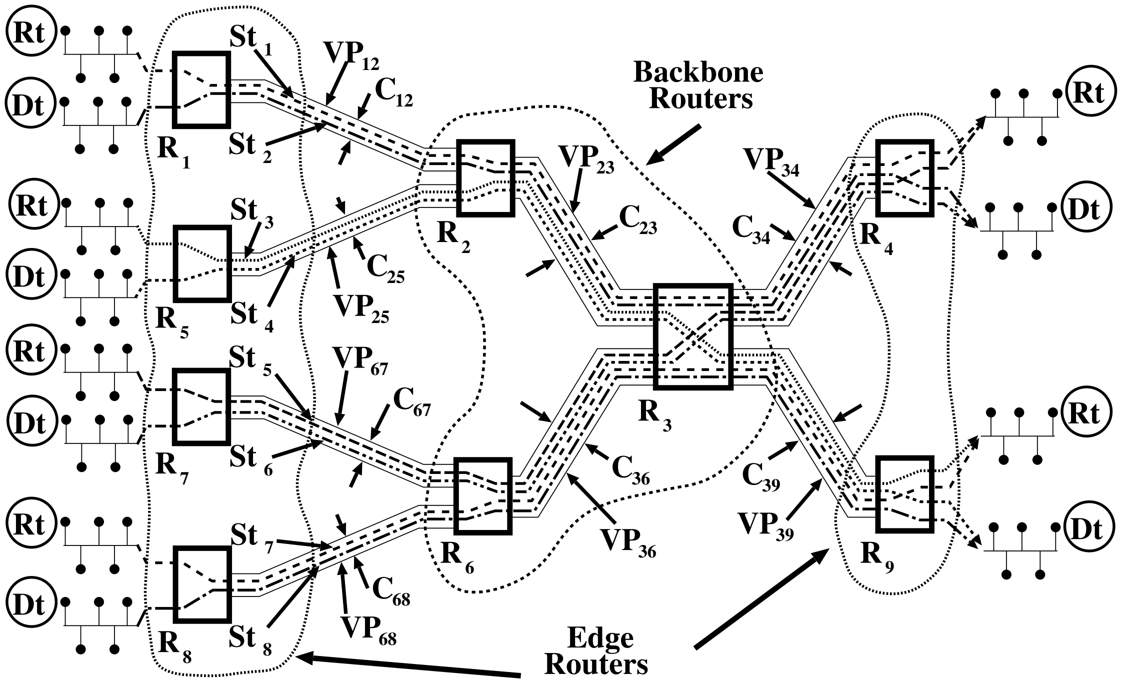


Figure 2: The network scenario.

4 Flow Control Approaches

Let us now elaborate a little further on the congestion issue we are addressing in this paper. First of all, RT sessions are likely to last for more than a few minutes, they reserve resources in the network and they are given priority over the BE traffic by the CBQ scheduler. Hence, in this scenario if the RT traffic is well behaved or policed, it is unaffected by congestion in the network. On the other hand, TCP sessions are likely to be responsible for the majority of the BE traffic but these sessions are usually short-lived (some 90% of them transfer less than 20kB [2]). In addition, current TCP implementations test for congestion in the network by increasingly sending more packets. Packets must then be dropped for the transmission window to be reduced and the source rate to be controlled. However, the dropping of packets is very undesirable because it can lead to global synchronization and throughput collapse, and it compromises the performance of delay sensitive traffic (e.g., telnet) [7, 3]. Besides, depending on when congestion hits, the drop of packets may come too late if the source has already sent out all the data it intended to (i.e., short lived sessions).

Approaches such as Random Early Detection (RED) [9] and Explicit Congestion Notification (ECN) [7] try to make the packet dropping more "effective and fair" and to enhance the congestion feedback, respectively. RED uses queue length information to randomly decide when to drop packets. ECN improves on RED by marking the packets RED would normally drop as a means to indicate congestion to the sources.

In any case, the source's reaction to the congestion signal is felt at the congested router only after a RTT. Even then, the reduction in the source rate may not have been sufficient, in which case packets are still dropped and a new RTT has to go by before the rate is reduced again.

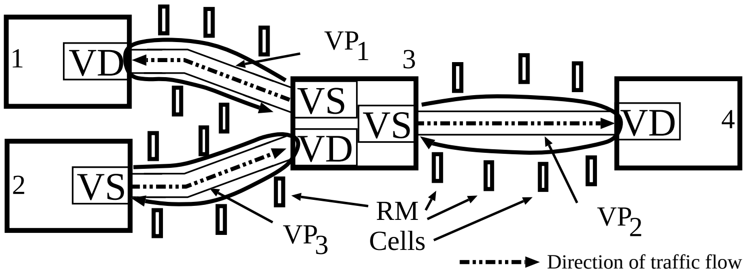
The equivalent of ECN has also been proposed in ATM under the name of FECN (Forward ECN) for the flow control of ABR connections [10]. In this case, the sources implement a multiplicative decrease of their sending rates upon congestion indication, and an additive increase when there is no congestion in the network. Such binary feedback may reduce implementation complexity, but it is well known to lead to unfairness, oscillation, and slow response to congestion [11]. Hence, many ABR flow control schemes employ explicit rate indication [6, 12] or relative rate indications [13], as a means to solve or at least alleviate these problems.

Challenges also arise in the transport of TCP traffic over ATM using the UBR service. The UBR traffic is not subjected to flow control at the ATM layer and cells are dropped when a congestion threshold is exceeded. In this scenario, the ATM switches usually employ Early Packet Discard (EPD) mechanisms [14, 15] to alleviated the source synchronization problem caused by tail dropping. Fairness problems also arise and selective dropping approaches such as the "Virtual Queueing" [15] address this issue. Since RED and ECN are also binary indication mechanisms which feedback congestion indications based on the state of a common FIFO queue, they are bound to experience similar fairness problems.

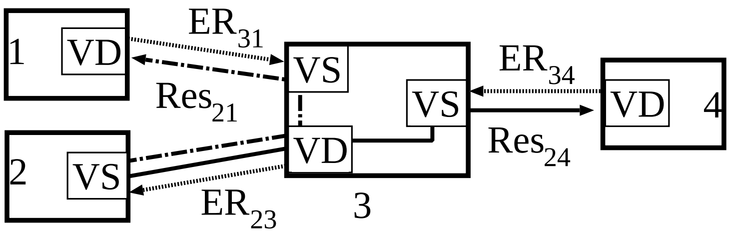
To sum up, the TCP flow control is reactive in nature with actions taken only after congestion sets in. Congestion notification based on packet drop is inadequate to preserve network throughput, to maintain fairness, or to deliver a quality service to delay sensitive traffic. Schemes are available to alleviate the consequences of dropping packets, but the response time is still in the order of RTTs. With high transmission rates becoming increasingly more common and with the number of simultaneous sessions rapidly increasing, such traditional approach may not react fast enough to alleviate congestion. In the next section we introduce a back-pressure approach that allows a router the ability to notify upstream neighbors of the amount of traffic they can send. This back-pressure propagates all the way to the edge router and from it to the TCP source via other means (see section 6). Hence, the response time is on the order of half the RTT between the source and the point of congestion. Since this approach is based on the ABR flow control using Explicit Rate (ER) indications, we are capable of addressing max-min fairness issues adequately.

4.1 A Back-pressure Flow Control

Our goal here is to address the congestion problem of Figure 1 as discussed in section 2. However, from now on we will consider the more general scenario of Figure 2 in which router R_3 can experience the congestion illustrated in Figure 1. The rationale on our approach is that since the VPs interconnecting the routers use the ABR service, the Internet traffic can make a better utilization of the available ATM resources and still allow the routers to enforce a back-pressure flow control. In addition, routers at the end points of a VP function as virtual sources (VSs) and virtual destinations (VDs) for the aggregate traffic flowing between them through the VP. Figure 3(a) illustrates the VSs and VDs roles played by each router port as seen by the traffic flowing in the direction indicated.



(a)



(b)

Figure 3: The ABR flow control applied to VPs.

Figure 3(a) illustrates how the ABR rate-based flow control is applied to each VP. A VS is continuously informed of the bandwidth available along its associated VP path through Explicit Rate (ER) indications on a stream of Resource Management (RM) cells the VS injects along with the data stream to the corresponding VD, which then returns the RM cells back to the VS [6]. While ABR protects the VP path from congestion, it does not prevent congestion at the terminating router acting as VD. For instance, if all VSs transmit at the available rate on the respective VPs, some routers, say router 3 in Figure 3(a), may become congested and drop packets. In this case, a back-pressure flow control can be implement by having VDs reduce the ER indications on the RM cell streams, before sending them back to the VSs.

It is then critical to determine how to dynamically adjust the ER indications and to assess the resulting impact on throughput, cell loss and fairness. Let us illustrate the issues involved by considering Figure 3(b). In this figure router 3 is an intermediary hop for the bidirectional traffic flowing between any two of the other routers shown. Let us further focus on the traffic (BE + RT) from router 2 to routers 1 and 4. The RT traffic directed to routers 1 and 4 has made resource reservations Res_{21} and Res_{24} on the VPs traversed by the respective RT streams, as illustrated in Figure 3(b). These reservations are guaranteed by the MCR allocated to the VPs traversed by the respective RT flows and they are enforced by the CBQ scheduler on each output port. The BE traffic is entitled to a fair share of the unused MCR bandwidth, the unassigned bandwidth and the bandwidth leftover by other ATM connection along the paths of the respective VPs. For the sake of argument, we further assume that ER_{31} and ER_{34} are the rates advertised by routers 1 and 4 and by the corresponding VPs. The issue is then to determine ER_{23} (i.e., the rate router 3 should advertise to router 2) such that the traffic from router 2 does not congest router 3.

In our approach, such back-pressure is actually provided as part of the VS and VD behaviors to be implemented by the ATM cards in the router ports. In [16], a VS and VD behavior was proposed for a simple tandem topology. Here we consider the more general topology of Figure 2 and we use the measured bandwidth utilization and the available bandwidth to slow down sending routers when their offered traffic would make downstream routers congested. Ideally, this hob-by-hop back-pressure would propagate to the sources, slowing them down. In the following, we first disregard, for simplicity, the notion of selective back-pressure (e.g., selectively slowing down the components of traffic from router 2 to 3 which are forwarded to routers 1 and 4, respectively) . Then, at the end of this section we discuss the resulting implications on throughput and fairness.

We compute ER_{23} by taking into account the bandwidth reservation for the RT traffic and the fair shares of ER_{31} and ER_{34} to be used by the BE traffic out of router 2. Since the reserved resources are accounted for by MCR_{23} (the MCR allocated to VP_3, Figure 3(a)) and ER_{23} cannot be smaller than MCR_{23}, the estimate for ER_{23} is determined by the Fair Shares (FSs) FS_{21} and FS_{24} of ER_{31} and ER_{34} used by the BE traffic out of router 2. As a result, the ER_{23} supported by router 3 of Figure 3(b) is bounded by:

$$\mathrm{MCR}_{23} \leq \mathrm{ER}_{23} \leq \mathrm{Res}_{21} + \mathrm{Res}_{24} + \mathrm{FS}_{21} + \mathrm{FS}_{24} \tag{1}$$

For max-min fairness [4], we measure the BE traffic contribution from each input port and we allocate fair shares to input ports based on whether they have traffic to use the fair shares. In our studies, we consider a variation of the approach in [4], which involves measuring used and available bandwidth. This does not necessarily lead to more complexity in our scenario since our CBQ scheduler[1] knows how much bandwidth has been assigned to each class and it measures how much bandwidth each class is actually using in order to enforce the link sharing policy. A minor modification can then be made to allow the scheduler to keep track of bandwidth usage by individual input ports. Note, however, that the CBQ queues contain

[1]Our CBQ was developed based on code publicly available from http://ftp.ee.lbl.gov/floyd/cbq.html.

packets and determining the input port a packet comes from may not be possible given that it would involve the same amount of processing overhead as that needed to make packet routing decisions. In our studies, though, we make the input ports stamp their own local identification on packets, on a form of encapsulation header, before sending them to the output port which then strips this extra header off before sending the packet onto the outgoing link (such approach is not uncommon on the architecture of some ATM switches).

The only real issue is that this bandwidth bookkeeping has to be done at the IP layer (we implement it as part of the CBQ scheduler code) and periodically communicated to the ATM layer, as described below. In our implementation, the CBQ scheduler on each output port keeps track of the traffic offered (in bytes) by each class and each input port i (Dt_i and Res_i) and it keeps track of the traffic actually sent (also in bytes) through the output port (Tf_{out}) during a fixed sampling period. Tf_{out} actually reflects the ER indication (the available bandwidth) for the outgoing VP. Hence, the Virtual Source behavior is split between the ATM layer and the CBQ scheduler in the following way:

1. At the beginning of a new sampling period:
   (a) The CBQ scheduler computes the fair shares as:
   $Av_{Dt} = Tf_{out} - \sum_i^N Res_i, \qquad FS = \frac{Av_{Dt}}{N},$
   Av_{Dt} is the available resources (in bytes) for the BE traffic and N is the number of input ports contributing traffic to the output port.
   (b) If an input port offered traffic Dt_i is such that $Dt_i <$ FS, the traffic from input port i is constrained somewhere else and the respective input port is assigned an actual share:
   $$AS_i = \frac{Dt_i + Res_i}{Av_{Dt}}.$$
   (c) The remaining input ports j are constrained on the router and they are assigned an actual share:
   $$AS_j = \frac{\dfrac{Av_{Dt} - \sum_i^{N_{un}} Dt_i}{N - N_{un}} + Res_j}{Av_{Dt}}$$
   N_{un} is the number of input ports not constrained at the output port.
   (d) Less obvious is the possibility that $Dt_i >$ FS. This means that input port i is actually sending too much traffic to the output port and this can happen for a number of reasons. Let $Ex_i = Dt_i$ - FS be the excess traffic sent from input port i. Since this excess traffic on the previous sampling time was caused by reasons outside the scope of the output port, we expect at least another Ex_i in the next sampling period. Hence, we make:
   $AS_i = \frac{FS - 2Ex_i + Res_i}{Av_{Dt}}.$

(e) If an input port is constrained somewhere else, assigning it an $AS_i = 1/\mathrm{N}$ does not congest the router while the input port is constrained. However, if the input port is no longer constrained somewhere else before getting to the router, having $AS_i = 1/\mathrm{N}$ allows the corresponding traffic to reclaim its fair share. Hence, all unconstrained input ports are assigned $AS_i = 1/\mathrm{N}$.

(f) Finally, the CBQ scheduler sends the vector AS to the ATM layer through a control primitive.

2. At the ATM layer, a flow of backward[2] RM cells is continuously being returned as part of the ABR flow control. The ER indication on these cells advertise the bandwidth actually available along the VP and supported by the router on the other end. Hence, upon the arrival of a new forward RM cell, the ATM layer uses the current vector AS and sends to the ATM interface on each input port i in the router a control message with AS_i*RM.ER as the bandwidth available for the input port i to send traffic to the output port.

The Virtual Destination behavior is then entirely carried out by the ATM layer on input ports as:

1. A vector Bw_i keeps track of the bandwidth available for the input port traffic on every other output port i, and an entry Bw_i is updated every time a new control message arrives from the output port i.

2. Whenever a forward RM cell arrives, we test if RM.ER is smaller than the sum of all Bw_i. If it is, the ER indication is left intact, otherwise it is replaced by the sum of all Bw_i.

3. The RM cell is sent back to the VS as a backward RM cell.

In our simulation results we demonstrate the effectiveness of this approach to implement back-pressure and to achieve max-min fairness. Now, let us elaborate on the implications to the congestion control problem of Figure 1. If we apply the VS and VD behaviors above to the scenario of Figure 1, we would get for instance an ER_b, the bandwidth available to the router feeding link b. However this ER indication does not reflect AS_{bc} and AS_{bd}. Rather it reflects $\mathrm{AS}_{bc} + \mathrm{AS}_{bd}$ and the sending router has no way of enforcing AS_{bc} and AS_{bd} selectively. If λ_{bc} and λ_{bd} are smaller than or equal to AS_{bc} and AS_{bd}, respectively, we have no problems. However, if both flows λ_{bc} and λ_{bd} have enough traffic load, they would be assigned half of $\mathrm{AS}_{bc} + \mathrm{AS}_{bd}$, which is clearly a problem if $\mathrm{AS}_{bc} \neq \mathrm{AS}_{bd}$.

In this case, it is easy to see that one link will get congested while the other will be under utilized. However, with appropriate buffering, we can accommodate the excess traffic and, from the VS behavior action 1(d), we try to alleviate this effect by reducing the actual share for the input port creating congestion. This is the only type of scenario in which we cannot enforce a better max-min fairness without sacrificing our commitment to alleviate congestion, as we illustrate on our simulation results.

[2] An RM cell can make part of a forward or backward loop and a bit Dir in the RM cell payload defines the appropriate direction [6].

5 Simulation Results

In this section we present simulation results to illustrate the effectiveness of our back-pressure approach. For our study, we used the topology illustrated in Figure 2, in which all links are 150Mbps, the propagation delay on the indicated VPs is 400μs, and the propagation delay from a host to an edge router is 10μs. Each Rt stream is composed by seven individual real-time H.261 video [17] flows transported over UDP at a target rate of 1.5Mbps and with 15 consecutive H.261 frames sent on every IP packet. Each Dt stream (the even St_is in Figure 2) is made up of ten individual ftp session transferring a large file (persistent sources), while each Rt stream (the odd St_is in Figure 2) is the aggregation of seven video sessions. The routers have 100-packet and 30-packet queues for the aggregate BE and RT streams, respectively. Our simulator code implements the TCP Tahoe version and we consider a maximum segment size of 1024 bytes. Finally, each VP in Figure 2 traverses two ATM switches (not shown on that figure to avoid overloading it) and one of the links traversed by each VP is shared by other CBR traffic. Hence, we chose the load for the background CBR traffic such that the C_{ij}s in Figure 2 and the MCR_{ij}s for the VP_{ij}s are:

C_{12}	C_{25}	C_{67}	C_{68}	C_{23}	C_{36}	C_{34}	C_{39}
70Mbps	70Mbps	70Mbps	70Mbps	95Mbps	95Mbps	70Mbps	70Mbps

MCR_{12}	MCR_{25}	MCR_{67}	MCR_{68}	MCR_{23}	MCR_{36}	MCR_{34}	MCR_{39}
20Mbps	20Mbps	20Mbps	20Mbps	40Mbps	40Mbps	40Mbps	40Mbps

As these numbers show, the resources (the MCR_{ij}s) committed by the networks to the VPs are less than what it is actually available. We use this fact to modify the CBR load to test different congestion scenarios. Furthermore, the MCR_{ij} for a VP is taken by the CBQ scheduler to be the link bandwidth for the purpose of imposing the link sharing policy of 70% of MCR dedicated to the RT video traffic, and 30% dedicated to the BE data traffic. Nevertheless, as our simulation results show, the BE traffic uses all of the actual bandwidth available on the links. This is a reminder of the benefit of using the ABR service.

The way we study the effectiveness of our scheme is by looking at the traces in Figures 4 and 5. These traces show the load offered to the VP connected to the indicated routers by the traffic streams traversing them. Each graph shows the traces for the BE stream, for the RT stream and for the aggregate stream. Figure 2 indicates the streams St_is traversing each router. Figure 4 plots the traces for the backbone routers indicated, while Figure 5 plots the traces as seen by the edge routers indicated. All connections start transmitting at different times, but the actual starting time is uniformly distributed over an initial time interval. As these figures show the TCP traffic reaches a steady state, i.e., their offered rate matches what the network can take.

We study different load scenarios that present themselves on the different time instants t_i shown in Figures 5 and 6. So, before t_1, we have all connections (RT and BE) active, and the initial available

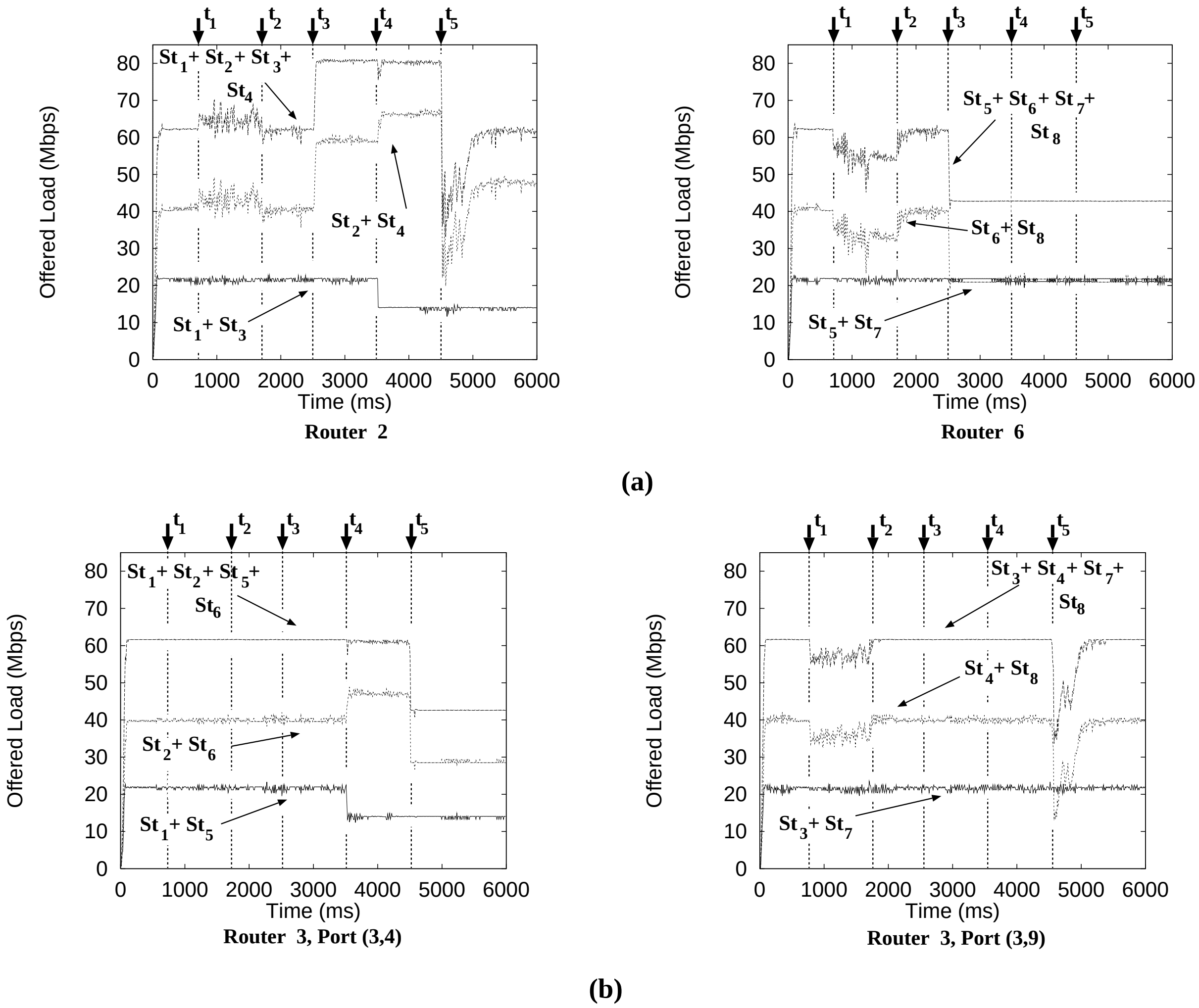


Figure 4: The load offered by backbone routers.

bandwidth on the links are the ones indicated on the table above. However, as the traces in Figure 4(b) indicate, the actual rate available on the respective links is roughly 60Mbps, not the 70Mbps presented in the table. This is so because we plot the effective rate, discounting all the different protocol (UDP, TCP, IP and ATM) overheads. Note that 60Mbps is also the actual rate at which R_2 and R_6 transmit to R_3, even though the VPs connecting them to R_3 have 95Mbps of available bandwidth (see table above). This means that the VD entities on R_3 are appropriately reducing the ER indication they receive on forward RM cells, before returning them to the VS source entities in the sending routers. Hence, R_3 is effectively back-pressuring R_2 and R_6.

If we now look at the traces in Figure 5, we see that each router is roughly transmitting at a maximum aggregate rate of 30Mbps, when the table above indicates that the bandwidth on the VPs connecting the

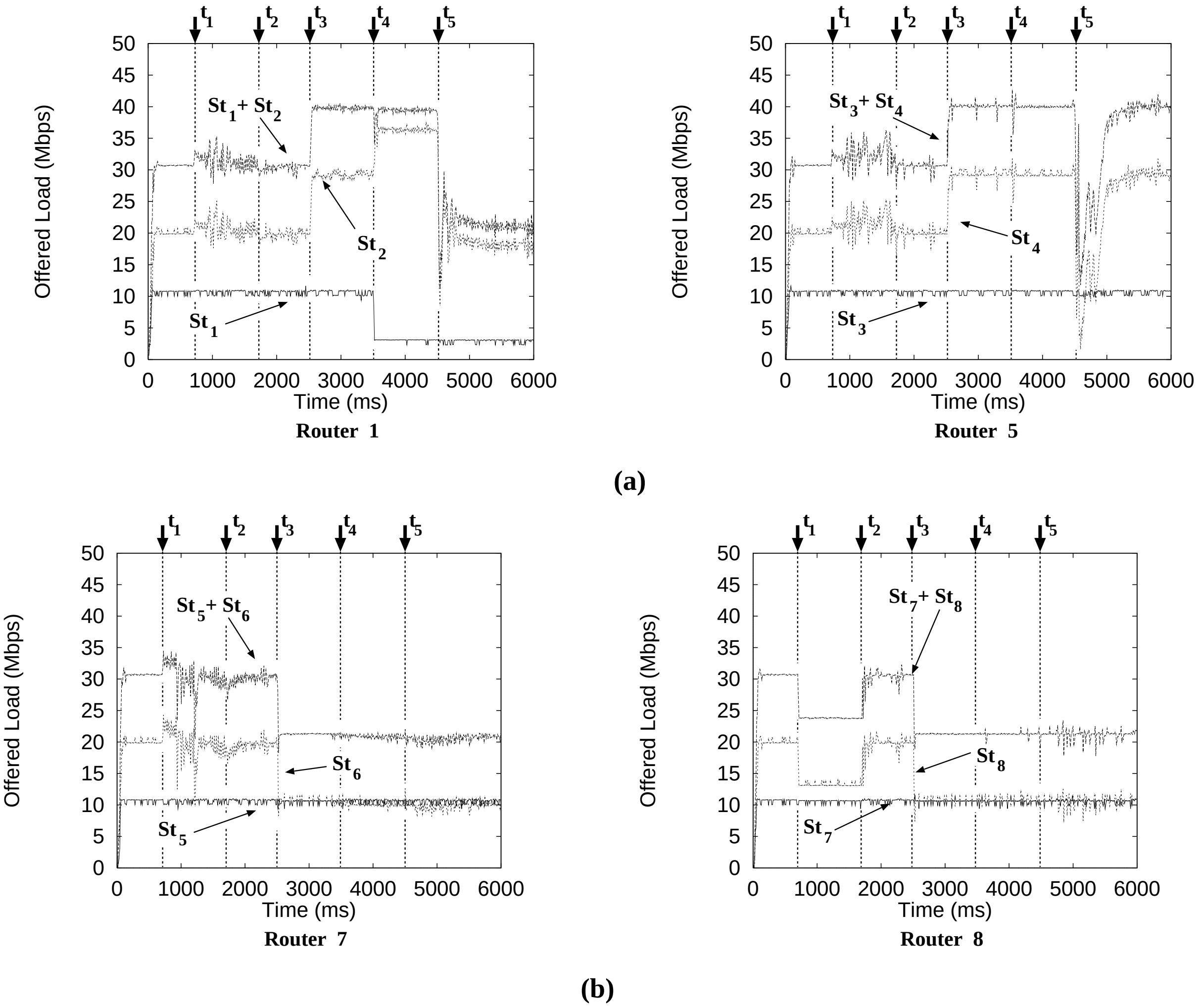


Figure 5: The load offered by edge routers.

routers in Figure 5 to R_2 and R_6 is actually 70Mbps. The reason is that the traffic from R_1 and R_5 and from R_7 and R_8, share VPs with 60Mbps of "supported" bandwidth from R_2 and R_6 to R_3, respectively. Again, we observe that R_2 and R_6 effectively back-pressure routers R_1 and R_5 and routers R_7 and R_8, respectively. Hence, the back-pressure from R_3 is effectively propagated to R_1, R_5, R_7 and R_8!

At time t_1, when we simulate the scenario in which the aggregate stream from R_8 in Figure 2 is reduced. This would be the case of sessions terminating or sources going into idle periods. For simplicity, we actually increase the CBR traffic that reduces C_{68} to roughly 30Mbps. The result is that the BE traffic (St_8) is reduced accordingly since it was recovering the bandwidth that was available on the VP path until t_1. Note from Figure 4(b) that since we have the CBQ scheduler, the RT traffic (St_7) is unaffected.

Most importantly, though, is the fact that for R_6, the input port fed by R_8 becomes constrained somewhere else, and the VS on R_6 increases the Actual Share (AS) of the VP_{36} bandwidth for the input port fed by R_7. This is an attempt to deliver max-min fairness. However, the bandwidth left over by the BE traffic from R_8 cannot be claimed by the BE from R_7 because, as Figure 2 indicates, these streams only share VP_{36}. The BE from R_7 is still constrained by the bandwidth available on VP_{34}. Hence, some of the excess packets are unavoidably lost on R_3.

Then, after the reduction on the traffic load from R_8 is felt on R_3, the VS acting on VP_{39} deems VP_{36} constrained somewhere else and it increases the Actual Share for R_2, again in an attempt to implement max-min fairness. However, the newly available bandwidth is split in R_2 between the traffic from R_1 and R_5. The traffic from R_5 could recover the bandwidth released by R_8, but we cannot enforce this in R_2 and the result are the oscillation we see between t_1 and t_2. The excess traffic sent from R_2 goes through unharmed while the excess traffic from R_1 collides with the excess traffic from R_7 and loses cannot be prevented if the scenario persists for a long time.

At time t_2, the CBR load is reduced and C_{68} returns to its original value and the BE traffic from R_8 recovers the bandwidth that is available along the path to R_9. This is due to the fact that the R_3 output port (3,9) regarded the input port (3,6) constrained but it still allocated this input port its fair share (see VS behavior 1(e)). So, when the R_8 traffic started using its fair share, the AS for input port (2,3) is also brought back to the fair share and the oscillations between t_1 and t_2 are terminated.

At time t_3 another change on the CBR load reduces this time the bandwidth C_{36} to 50Mbps. As Figure 4(a) illustrates, the BE traffic from R_6 is reduced accordingly. With the VP_{36} constrained by the excess CBR traffic, the BE traffic from R_7 and R_8 are also reduced in response to the back-pressure from the VS in R_6. Since, in this scenario, the reduction equally affects the load on VP_{34} and VP_{39}, the respective output ports regard the input port (3,6) constrained and they increase the AS for input port (2,3). This extra bandwidth is then notified by the back-pressure mechanism to R_1 and R_5 which can use the newly available bandwidth. Hence the load offered by these and by R_2 is adjusted to the new network scenario as illustrated on Figures 4(a) and 5(a).

At time t_4 we illustrated another type of traffic load variation: the aggregate RT traffic load offered through R_1 is reduced. This simulates the case in which some of the RT flows are terminated, but this does not reduce the reservations that are made on CBQ schedulers for the RT flows from that router. As a result, the BE traffic from same R_1 is the one benefiting from the bandwidth released by the RT sessions.

Finally, at a time t_5 another increase in CBR load reduces the bandwidth C_{34} to 50Mbps which reduces the offered traffic from R_3, see Figure 4(b). Also from Figure 5(a) we see that the offered traffic from R_1 is reduced since there is much less bandwidth on the path to R_4. What it is not intuitive though is the effect on the R_5 load. The reason here is that the reduction in C_{34} is back-pressured by R_3 to R_2 and the new available rate is split in R_2 between R_1 and R_5, again we cannot impose a selective control on the R_1 flow alone. As a result, packets from R_1 are unavoidably lost in R_3 which triggers the slow-start on corresponding sessions. This reduces the offered rate of R_1 and R_2 increases the share of R_5 for a rate not supported by R_3 and packets from R_5 are also lost in R_3. The effect of VS rule 1(d) does not have enough time to be propagated to R_2 to avoid these loses. As we said before, propagation delays become an issue in such high speed scenarios. Eventually, when the effects of slow-star subsides, the offered traffic converges to the appropriate available rates.

The simulation results presented so far were designed to illustrate some of the features and limitations of our back-pressure control. In addition, we also felt that it would be nice to compare the results without back-pressure. Hence, we presented the simulation results for the same scenario considered so far and for the edge routers only in Figure 6. First of all, the TCP sources become active with a slow-start and the offered traffic increases exponentially (doubling every RTT). Since we no longer employ the back-pressure, this traffic is always admitted into the network. As a result, very many packets are dropped and all sessions are synchronized. Hence, before t_1 we have a lot of wasted resources by packets dropped and eventually retransmitted. Comparing these with the results in Figure 5, it is clear that the back-pressure is effective in avoiding these packet losses.

Then, at t_1, the available rate on VP_{68} is reduced which affects only the traffic from R_8, which accommodates the traffic mismatch it is offered. Notice, though, that there is much oscillation on the traffic offered by R_1 and R_7 because the R_6 backlog is cleared, quickly draining the traffic from R_7 stored in R_6. Hence, there is no way to enforce max-min fairness here. Furthermore, the traffic from R_5 cannot claim the bandwidth leftover from the traffic from St_8 since this involves a few RTTs to increase the window. Then, at t_2 the available rate on VP_{68} is increased again and the backlog in R_8 is dumped into R_6 and then R_3 overflowing the buffers in this latter router. Here again, in the back-pressure scenario of Figure 5, the new offered load reclaims its fair share while the load from R_5 gets back-pressured and these losses are avoided. The remaining scenarios do not illustrate meaningful congestion issues so obviously as these first two do.

To further explore the benefits of our back-pressure, we plot in Figure 7 traces for the same sequence of events described above, but we replace the persistant TCP (ftp) sources by sources transmitting only 32KBytes. Upon transmitting these, the source goes idle for a period of time uniformely distributed in [0,50ms] and then it begins sending another block of 32KBytes from slow-start. With this we simulate the persistant arrival of new TCP sessions which are short-lived. This is a typical scenario observed for instance during web browsing, except for the idle period distribution. We use such distribution, though, as an approximation for the scenario with many, almost simultaneous, active TCP sessions.

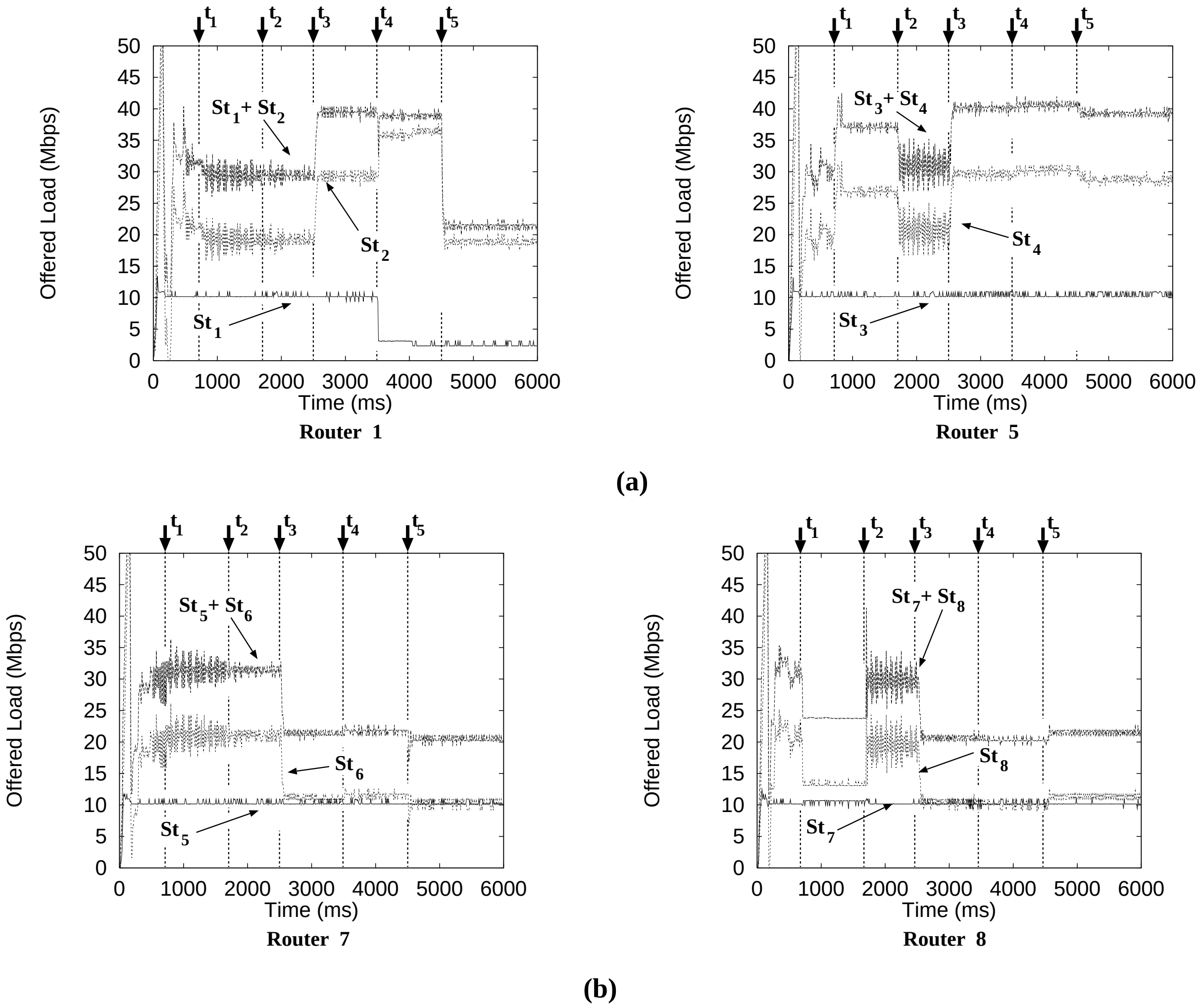


Figure 6: The load offered by edge routers under no back-pressure.

The rationale here, as Figure 7(a) illustrates, is that under no back-pressure the frequent slow-starts keep the routers in a permanent transient state, we do not see the nice convergence to steady state demonstrated in the results of Figure 6. The consequence is a high packet loss rate, even though the output links in R_3 are kept at full utilization, indicating that resources are wasted with retransmitted packets. With back-pressure, Figure 7(b), the frequent slow-starts are smothed out and packet losses only occur when an effective max-min fairness cannot be enforced, as discussed above.

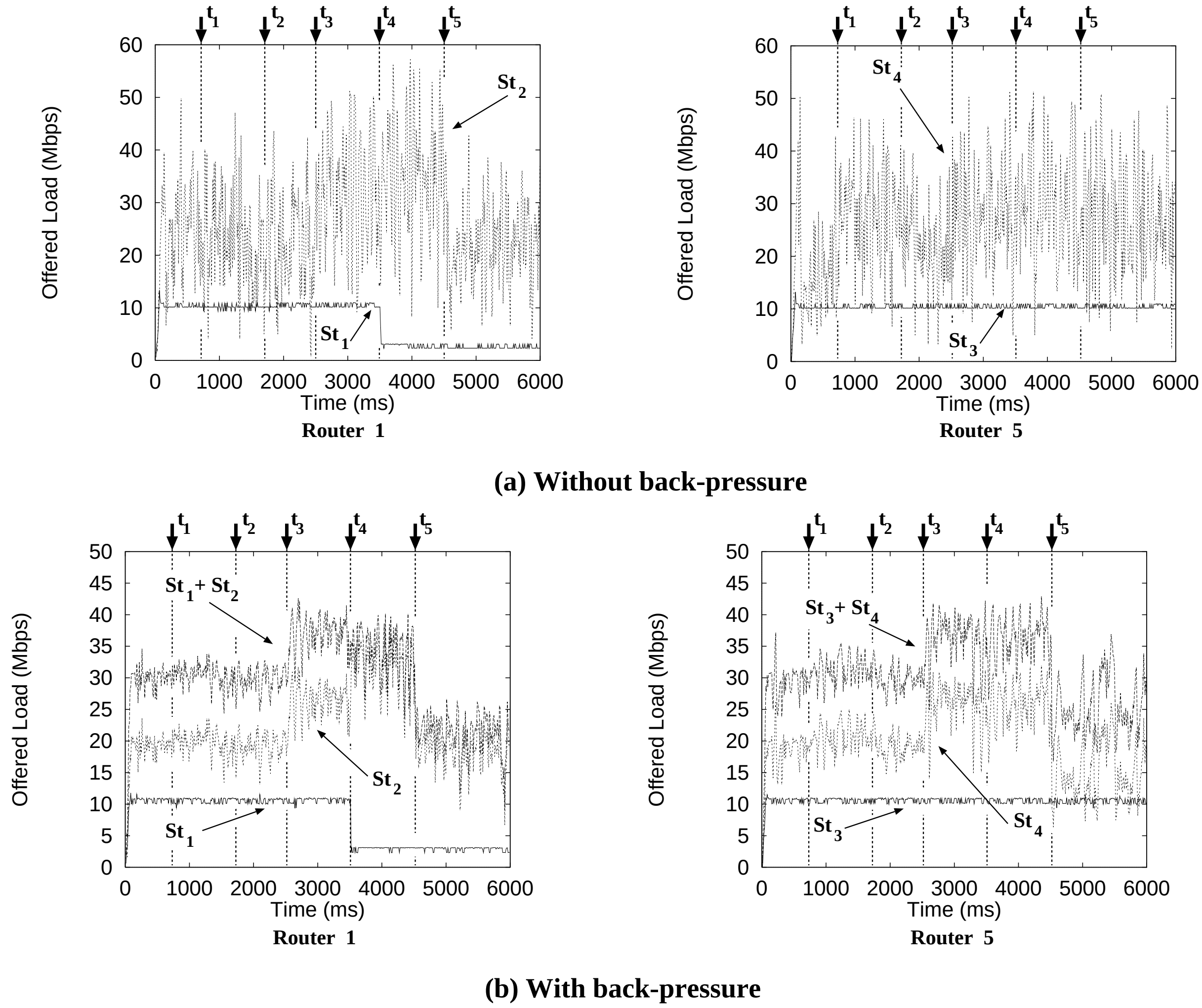


Figure 7: The effect of many short-lived connections.

6 Explicit Rate and the Last Hop

As mentioned previously, the Explicit Rate indications that are provided through the ABR service to edge routers cannot be communicated to the actual TCP sources as part of the VS/VD behavior if the sources do not have ATM connectivity. In fact, the sources are most likely connected to legacy LANs. The issue is then the usefulness of the ER indications if the actual sources do not receive this information as already addressed in [18, 19]. Basically, with the arrival of acknowledgments, the TCP sources can double their transmission window and send more packets. Since the sources are connected to a Legacy LAN and they are not subjected to any flow control mechanism other than TCP, the traffic reaches the edge router at a rate higher than that supported by the ATM network.

If the number of active TCP sessions is high, it is easy to see that the edge router may run out of buffers and it may start dropping packets on the edge of the ATM network. The advantage of this approach is that no network resource is consumed with packets that would be eventually dropped inside the network. The disadvantage is that time sensitive session could experience time-out delays. An alternative approach proposed in [18] consists on equipping edge devices with enough buffers to hold many TCP-receiver-window worth of packets. The rationale consists in letting the TCP sources transmit as if there was no congestion in the path to the destination. Eventually, the transmission window would equal the receiver's window and rate increases would stop. The problem with this approach is the large buffer requirement and a possible complex management thereof.

A more interesting idea was presented in [19] as a means of "effectively extending the ABR flow control all the way to the TCP source". In their approach the edge router uses an acknowledgment bucket that translates the ER indications into a sequence of TCP acknowledgments to prevent the TCP source from sending at rates higher than the ER indications. A potential difficulty with this approach is that the TCP acknowledgments may in fact be returned piggy-backed on the packets from the reverse data flow and implementing this approach effectively may require altering the acknowledgment field in the TCP header, a clear layer violation.

Probably the best approach is to use the ECNs in [7]. Again, with the arrival of acknowledgments the source would increase their transmission windows and the buffers on the edge routers would fill up. Using the ECNs in the forward direction has the same problems as discussed in section 4. However, we can stamp the ECN bit on acknowledgment packets coming back through the edge device. Actually, this could be provide a very selective feedback if the edge device provides per session queueing where the complexity involved may not be too large since and edge router deals with a much limited number of sessions. With such feedback, we could control the sources' transmission rates to match the ER indications. This is left as direction for future studies.

7 Conclusions and Future Research

In this paper we address the effectiveness of the TCP flow control to deal with the large delay-bandwidth product which, along with the TCP inability to handle very many short-lived connections [3], indicate that new solutions are needed to address the congestion problem in next generation Internet backbones. Since ATM has been extensively used as a link layer for such backbones, and this trend is likely to persist for a number of years, in this paper we address this congestion control problem in this common IP over ATM scenario. Our approach follows the suggestions in [5] to use the ABR service in the backbones and we present Virtual Source and Virtual Destination behaviors for the IP routers terminating the ABR control loop. Such approach effectively implements a back-pressure mechanism that addresses max-min fairness. In addition, it is also more appropriate to handle the large RTTs and the large number of TCP sessions because we back-pressure from the point of congestion and because we act on the aggregate traffic between routers, respectively.

As for future research, we are looking into the use of Label Swapping Routers (LSR) over ATM to implement this back-pressure mechanism. The use of LSR should lead to a more elegant mechanism because all queueing, scheduling and bandwidth bookkeeping can be done entirely at the ATM layer. This removes the need for passing control messages across layer boundaries as described in section 4.1 as part of the Virtual Source behavior.

References

[1] R. Braden, D. Clark, and S. Shenker. Integrated services in the Internet architecture: an overview. *Request for Comments 1633*, June 1994.

[2] V. Paxson. Empirically-derived analytic models of wide-area TCP connections: Extended report. Technical report, Lawrence Berkeley Laboratory, July 1993.

[3] R. Morris. TCP behavior with many flows. In *Proc. of ICNP '97*, October 97.

[4] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[5] C. M. Pazos and M. Gerla. ATM virtual private networks for the Internet data traffic. In *Proc. of MMNS '97*, Montréal, Canada, July 1997.

[6] ATM Forum Technical Committee. Traffic management specifications, version 4.0. April 1996.

[7] S. Floyd. TCP and explicit congestion notification. *ACM Computer Communication Review*, 24(5), October 1995.

[8] S. Floyd and V. Jacobsen. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4), August 1995.

[9] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), August 1993.

[10] Peter Newman. Traffic management for ATM local area networks. *IEEE Communications Magazine*, 32(8), August 1994.

[11] L. An, N. Ansari, and A. Arulambalam. TCP/IP traffic over ATM networks with ABR flow and congestion control. In *Proc. of GLOBECOM '97*, November 1997.

[12] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan. ERICA switch algorith: A complete description. In *ATM Forum Contribution 96-1172*, August 1996.

[13] F. Chiussi, Y. Xia, and V. Kumar. Virtual queueing techniques for ABR service: Improving ABR/VBR interaction. In *Proc. INFOCOM '97*, April 1997.

[14] A. Romanow and S. Floyd. Dynamic of TCP traffic over ATM networks. In *ACM SIGCOMM '94*, August 1994.

[15] Y. Wu, K-Y Siu, and W. Ren. Improved virtual queueing and dynamic EPD techniques for TCP over ATM. In *Proc. of ICNP '97*, October 97.

[16] C. M. Pazos and M. Gerla. Bandwidth efficiency in Internet AVPNs. In *Proc. of GLOBECOM '97*, November 1997.

[17] T. Turletti and C. Huitema. RTP payload format for H.261 video streams. *Request for Comments 2032*, October 1996.

[18] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and J. Jiang. Performance of TCP over ABR on ATM backbone and with various VBR traffic patters. In *ATM Forum Contribution 96-1294*, October 1996.

[19] P. Narcáez and K-Y. Siu. An acknowledgment bucket scheme for regulating TCP flows over ATM. In *Proc. of GLOBECOM '97*, November 1997.