Nonintrusive TCP Connection Admission Control for Bandwidth Management of an Internet Access Link

Anurag Kumar, Malati Hegde, S.V.R. Anand,
B.N. Bindu, Dinesh Thirumurthy and Arzad A. Kherani
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore 560 012, INDIA
email: anurag, malati, anand, bindu, dinesh, alam@ece.iisc.ernet.in

Abstract— **We describe our approach to monitoring and managing the bandwidth of an Internet edge link with a view towards certain quality of service objectives for the services it carries. Such a link could be, for example, a campus's Internet access link, or a small ISP's backbone access link. We use SNMP polls and packet snooping to obtain traffic statistics, and TCP admission control for bandwidth management. Our implementation is completely nonintrusive: we use Ethernet packet capture in the promiscuous mode for traffic analysis, and use IP masquerading for blocking new TCP connections. This approach has been implemented by us in a software system for traffic management. We first justify our approach with a simple analytical model. We give an overview of our software implementation, and discuss some implementation issues. Then we provide measurement results that show the effectiveness of the techniques.**

Keywords: Internet bandwidth management; admission control; quality of service; TCP performance; bandwidth sharing; traffic control

I. Introduction: Objectives and Approach

Figure 1 shows a campus network and its attachment to an ISP. Typical campus backbone speeds are 100Mbps, with more recent installations being 155Mbps, if based on ATM, or 1000Mbps if based on Gigabit Ethernet. Campus endpoints are typically connected to the backbone by departmental LANs of at least 10Mbps. As opposed to these numbers, note that typical Internet access link speeds range from 64Kbps to 2Mbps, the latter being fairly common now in developed countries, but still quite expensive and uncommon in developing countries.

It is immediately clear, from the scenario described above, that the WAN access link can become a bottleneck resource that needs to be monitored and managed.

We have developed a software system for monitoring and managing the WAN access link bandwidth in a scenario such as that depicted in Figure 1.

The work reported in this paper was supported by the ERNET project of the Ministry of Information Technology, Govt. of India.

Our monitoring and control architecture is also shown in Figure 1. All traffic between the WAN access link and the campus network is made to pass through an ethernet segment, or, equivalently, an ethernet hub. On this segment sit two machines: the *monitor* and the *controller*. It is obvious that one machine could serve both functions; the depiction in Figure 1 emphasises the fact that the approach scales easily by adding more machines.

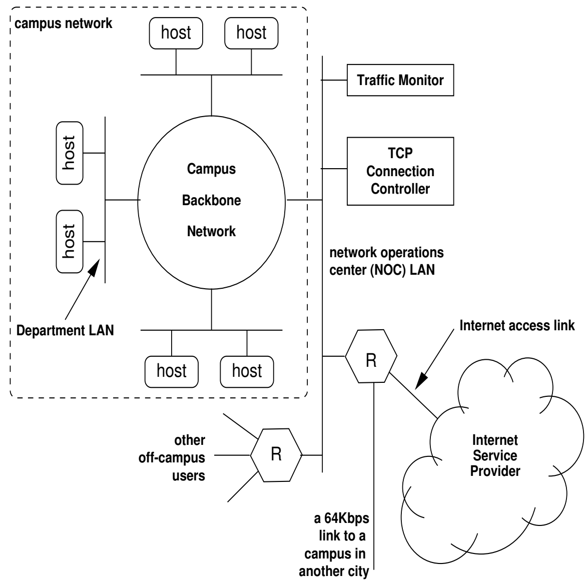

Fig. 1. A campus network with an access link to an Internet Service Provider. Note the off-campus users being served from the same access link, and the link to a campus in another city. Note also the placement of the monitoring and control machines.

The approach for management and control of the WAN access bandwidth is the particularly novel part of our system. The WAN monitoring part of our package is based on the standard SNMP protocol, and utilities for promiscuously capturing IP packets on the ethernet segment. The monitor machine periodically polls the access link router for SNMP data. The raw counts provided by SNMP are used to produce useful network performance measures, such as link utilisation, link up/down percentages, and link bit-error rates. The packet capture data is used to generate service-wise (i.e., email, HTTP, FTP), and source-destination-wise statistics for the packets flowing on the Internet access link.

Based on link utilisation policies (aggregate and service-wise) set up by the network manager, the monitor machine generates control commands that are sent to the controller machine on the ethernet segment (see Figure 1). The controller machine then controls the TCP connections (i.e., we block new connections and slow down existing ones, if necessary; see Section III for details) so that the traffic flowing on the access link conforms to the configured policies. The following are examples of policies that can be set up:

- Do not allow the occupancy of the access link to exceed 90%; this threshold may be obtained, for example, from a model of TCP connections over a wide-area network with this link as the bottleneck link; the objective could be to meet a session throughput objective (see Section II).
- Out of the usable portion (see previous bullet) of the access link rate, in the incoming direction, reserve a part for SMTP (email), and leave the rest for the other services. As the traffic mix usually varies by time of day (owing to differences in time-zones) the bandwidth allocations to the various services could be allowed to automatically vary by time-of-day.
- Out of the usable access link rate, reserve a portion for a particular set of IP addresses on the campus. This feature could be used to provide a "subrate" service to a special group of addresses on campus, who wish to pay for some guaranteed Internet access bandwidth. When unused by this group, this bandwidth could be utilised by the rest of the campus. This may help reduce the total bandwidth that needs to be provisioned for the access link.

There are several advantages of our approach.

1. *Easy installation in the network:* Most WAN access links are attached to campus networks via a router on an Ethernet LAN. Installation of a system based on our approach is then just a matter of installing one or more machines on the same LAN.
2. *Network unaffected by manager hardware failure:* In our approach if the manager machine fails, the controls stop working but the network traffic continues to flow, albeit in the original uncontrolled manner.
3. *Easy scalability to increasing access link speeds:* The simplest installation can just be on one 200Mhz Pentium machine, running both the monitor and manager modules. Additional machines can be added as the link speed, and the complexity of the management policies increase.
4. *Conformity with the RMON philosophy:* The RMON (I and II) philosophy is to create management agents that reside on hardware separate from networking hardware. Our architecture conforms to this approach.

There are now several commercially available products with objectives similar to ours. These are generally referred to as "Bandwidth Managers" (see, for example, [2]). As a rule these are devices that sit physically in the path of the traffic flowing between the WAN access router and the enterprise LAN. Such a device is either a general purpose computer (simply a PC) with the vendor's software running on a standard operating system, or is a proprietary "box" with proprietary software. Since all packets flow through these devices, it is possible to maintain state, and to queue and delay packets. Hence control approaches such as Weighted Fair Queueing (WFQ), TCP acknowledgement pacing, and TCP window adjustments can be used.

Our observation is that none of these devices have a view of the occupancy of the WAN link. Such a view is important in situations where, for example, the WAN access link could be carrying traffic other than that which is passing through the bandwidth manager (e.g., the campus router on which the access link terminates, is also connected by a serial link to a router at another location, say, a different campus; see Figure 1).

In our work we have explored only nonintrusive admission control (along with TCP window quenching) as the control approach. This is among the first attempts to use TCP admission control for bandwidth management. Hence the results are interesting in demonstrating the possibilities in the approach.

We note that our approach cannot control UDP flows (these are minimal in our environment), and nor can it control TCP flows that use IPSEC (such flows are nonexistent in our situation).

In this paper we first use some models to explain our motivation for our approach, we then describe our architecture, and then provide some measurement results that show the efficacy of our system in meeting the desired control objectives.

II. Traffic Engineering and Load Control for TCP Controlled Flows

In this section we discuss a model that will help to motivate the basic bandwidth management approach that we have adopted. Consider again the situation shown in Figure 1. We will assume that the predominant traffic is TCP controlled elastic traffic; e.g., traffic generated by applications such as email, FTP, HTTP, and SMTP. All of these applications essentially involve the transfer of files. Requests for transfer of these files arrive in some way (for example, when a user at a client clicks on a URL, or a mail relay in the Internet opens an SMTP connection to a campus mail server), and then the TCP protocol controls the sharing of the access link bandwidth between the ongoing file transfers. Let us focus on the file transfers from the Internet to the campus clients. Since the campus backbone is typically of much higher speed than the access link, we can assume that the file transfers are bottlenecked at the access link. We will use a model to study this situation.

We assume that the requests for these file transfers arrive in a Poisson process (see [9] for a discussion of this assumption; note that we are *not* assuming that the packet arrival process to the access link is Poisson). We further assume that each such request involves the transfer of a random number of bytes of data, and these random file sizes are independently and identically distributed. It can then be argued that the TCP controlled bandwidth sharing on the access link can be approximately modelled by using the Processor Sharing (PS) model from queueing theory (see [7] for an introduction to the PS model, and [10], [6], [5], [8] for studies involving the use of the PS model for TCP controlled bandwidth sharing). The PS model assumes fair bandwidth sharing; i.e., if there are n sessions active on the access link, then each session obtains exactly $\frac{1}{n}$th of the bandwidth at all times. In practice, for TCP controlled flows this is not true, especially if the file transfers are short, and if they encounter different propagation delays (for some studies of the applicability of such models to TCP, see [6] and [8]).

Let λ denote the rate of arrival of file transfer requests, let X denote the random file size (EX denotes the expected file size), and let b denote the Internet access link bandwidth. For the purpose of analysis we assume a fluid model. For this model we can obtain the time average per-session bandwidth share versus the occupancy (the fraction of time the link is carrying traffic), ρ_b, of the access link. The per-session bandwidth share is a measure of the throughput being received by individual file transfers. We define the normalised offered load ρ by:

$$\rho = \frac{\lambda \times EX}{b}$$

Clearly, for $\rho < 1, \rho_b = \rho$. It can then be shown that the per-session bandwidth share, normalised to b, and as a function of ρ_b is given by

$$\left(\frac{1-\rho_b}{\rho_b}\right) \ \ln \frac{1}{1-\rho_b}$$

It can be seen from this formula that, as expected, the session bandwidth share (normalised to b) decreases from 1 to 0 as ρ_b goes from 0 to 1. There are two consequences of this behaviour:
(i) If $\rho > 1$ (i.e., $\lambda EX > b$) it can be expected that the session throughputs provided by TCP will be very small. Thus there is a need to keep the actual occupancy of the link to some value below 1.
(ii) If the campus network administrator wants to assure the users some minimum quality of service (session throughputs) it is clear that the access link occupancy cannot be allowed to exceed some value, say, ρ_b^* (e.g., it can be shown from the above model that to assure an average bandwidth share of 20% of the access link bandwidth we need $\rho_b \leq 0.93$).

Obviously, the only way of achieving (i) and (ii) is by shedding the excess load if $\rho > \rho_b^*$. In our approach we have used TCP connection admission control to achieve this load limiting (see also [10] for additional arguments for TCP admission control). Note that, from the model, it is clear that if a fraction $\frac{\rho-\rho^*}{\rho}$ of the arriving TCP connections are blocked then the average bandwidth share will be given by the above formula with $\rho_b = \rho^*$. A measurement module polls the access link router for SNMP data; this data provides packet counts that yield link occupancy. This measured link occupancy is compared against the target value ρ_b^* to determine when to block TCP connections. In Section IV we will provide some measurement results (e.g., the throughput distribution with and without admission control) that will demonstrate the efficacy of this approach.

III. The Monitoring and Control Architecture

Figure 1 shows the placement of the monitoring and control devices; the figure is drawn assuming that there are two devices. All traffic between the bottleneck access link and the campus network is made to go through an ethernet segment on which the machines are placed.

A block diagram showing the software modules and their relationships is shown in Figure 2. The software is shown split up over two computers, one computer with the software for monitoring, statistics estimation, determination of the controls to apply, and the other with the software for exercising control.

The control algorithms need statistics for aggregate traffic flow on the WAN access link, and more detailed statistics (e.g., per service, or per source-destination IP address) of traffic flow. The network traffic is monitored using utilities based on SNMP [4], and a TCP packet capture library [3]; see Figure 2. Our implementation of the SNMP based

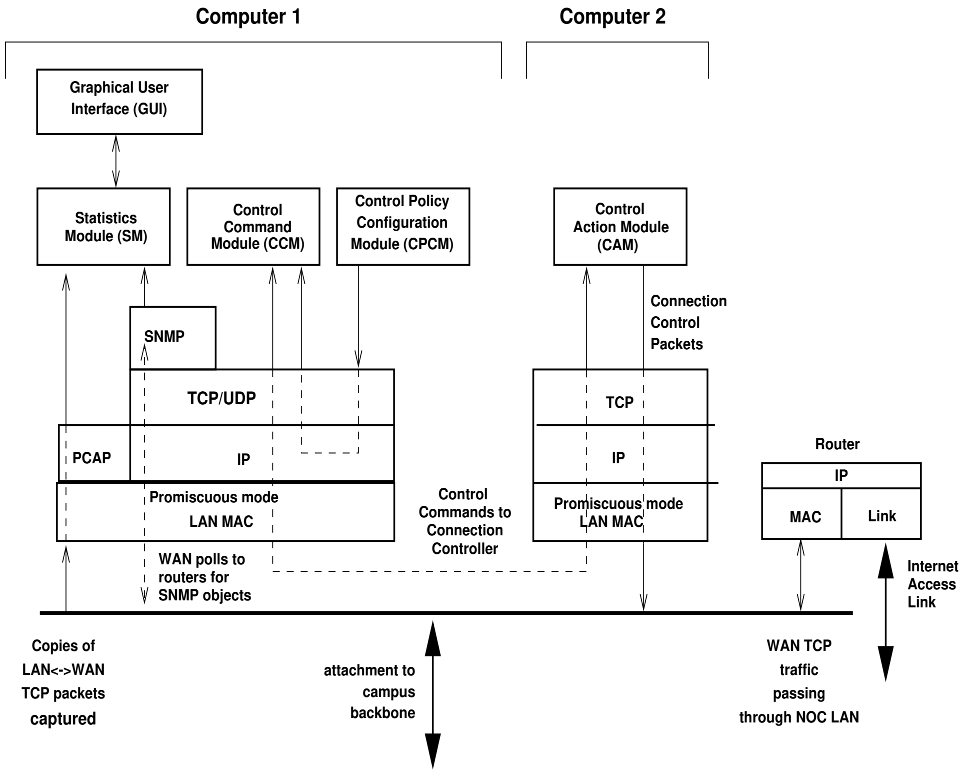

Fig. 2. The software modules, their relationships, and the flow of monitoring and control in the system. A two computer implementation is assumed.

monitoring software is based on the MIT SNMP development kit (see [1])

The detailed breakup of the traffic flow in and out of the campus is obtained by promiscuously snooping the ethernet LAN, picking up all the packets and then parsing them to the extent required, and classifying them in the various categories. Our implementation of this function is based on a packet capture utility available in many UNIX systems [3].

We now describe the various software modules shown in Figure 2, and the flow of monitoring and control information between them.

- Statistics module (SM): Obtains raw measurements from SNMP and TCP packet capture, and uses various algorithms to obtain statistics from these measurements. The SM provides statistics such as the access link occupancy, and the bit rates for various services to and from configured sets of IP addresses on the campus LAN.
- Control Policy Configuration Module (CPCM): We implement a certain control architecture (i.e., how the statistics are used to control the traffic in order to achieve certain objectives). There are several parameters that need to be configured, to set the desired objectives, and to configure the desired dynamics of the control. The CPCM module provides the interface for this purpose. CPCM also defines the network, via link and router interface definitions.
- Control Command Module (CCM): This module uses the statistics, and the configured control policy (see Traffic Control Policies below in this section) to determine whether or not a certain stream of packets, or certain types of connections need to be controlled. Simple algorithms are used to trigger the controls for each type of traffic, so as to enforce the policy settings. These algorithms basically estimate the bandwidth utilisation by each category of traffic, compare it with the objective, and trigger the controls on or off; a hysteresis is built into the controls to avoid excessive oscillations. All *new* connections of a particular class are blocked when a certain occupancy threshold is crossed upwards, and then blocking is turned off when another *smaller* threshold is crossed downwards. Control command packets are generated and sent to the Control Action Module (see below) as UDP packets.
- Control Action Module (CAM): Commands from the CCM cause this module to send TCP connection control packets to the end-points of TCP connections. The CAM snoops for TCP packets on the ethernet LAN, and thus can identify the initiations of new connections (SYN packets), and the activity of ongoing connections. When instructed by the CCM, it can prevent certain types of TCP connections (e.g., all HTTP connections from a certain set of IP

addresses) to be set up, or it can slow down certain other ongoing TCP connections. It does this by IP masquerading, and by sending TCP RST (ReSeT) packets and ICMP source quench packets to one of the hosts involved in a connection. See "Nonintrusive Control of TCP Connections" below in this section for details.

Figure 2 shows the SM, CCM, CPCM, and the GUI on one machine, whereas the CAM is on a different machine. This is basically for load sharing purposes; with a more powerful machine, all the functions could be on one machine. The CCM communicates with the SM, and the CAM. The CCM communicates with SM using TCP socket calls as well as function calls. The CCM communicates with CAM via UDP. Since this communication is over the LAN and very frequent, TCP connection overhead was felt unnecessary. Whenever any information which is available in CAM (e.g., the connection kill rate for a particular set of IP addresses and port numbers) is requested by CCM it is sent via TCP.

Nonintrusive Control of TCP Connections: Connection admission control (CAC) is exercised by issuing TCP RST messages during the TCP connection establishment phase. Since the approach is non-intrusive, the CAM promiscuously captures SYN and SYN-ACK packets, masquerades as one of the connection end-points, and generates a TCP RST message to the other end-point. In the data transfer phase, active connection throughput is controlled by sending ICMP source quench messages to the sender; this brings down the sender's TCP congestion window to one. We have experimented with some alternative implementations of this basic approach.

We tried sending the TCP RST message to the end-point on the campus network with the CAM IP-masquerading as the remote connection end point. Some of the TCP implementations, contrary to the TCP specification (RFC 793), keep resending TCP SYN requests unmindful of the receipt of a TCP RST. As a consequence, the purpose of resetting the connection was not achieved, and the large number of control packets resulted in extra traffic.

Next we experimented by sending a TCP RST to the remote machine (typically a web server). This did not improve the situation much since there were occasions when these packets did not reach the remote host because of network congestion. We also tried using the ICMP reject message with "protocol unkown" type for connection admission control. The problem with the ICMP reject message was that it not only controls the new connection to be admitted, but also terminates all the active TCP connections, if any, between the same source and destination pair. Also most routers have access lists that block ICMP messages. This latter problem also affects ICMP source quenches that we use to control the throughput of active connections.

To overcome the above problems we made sure that we allow FTP and HTTP connections only through a set of identified well-behaved proxies in the campus. These proxies run on Linux or OpenBSD computers, and have implementations that conform to the TCP specification. Connection admission control is achieved by sending a TCP RST to the local end point of a connection, namely a campus proxy.

Traffic Control Policies: We have experimented extensively with control policies that involve: (i) a target WAN access link occupancy, and (ii) service-wise or IP-address-wise aggregate bandwidth allocation.

For example, based on the discussion in Section II we can choose the target access link occupancy, ρ_b, to be 90%. If the access link occupancy tends to exceed its limit, then connections are blocked, as well as existing connections are slowed down using ICMP source quench messages. The choice of connections to be blocked or controlled depends on the service-wise and IP address-wise aggregate rate objectives.

A typical service-wise aggregate bandwidth allocation could be the following. For example, the access link speed is 64Kbps; the target occupancy is 90%, yielding a usable throughput of 57.6Kbps. The inbound direction is the one that limits the flow of traffic (because inbound mail and web traffic is significantly larger). Allot 25Kbps to the set of IP addresses 202.141.x.x/16, and the remainder to the other set of IP addresses 144.16.64.x/19. Out of the bandwidth allotted to 144.16.64.x/19, let 20Kbps be used by SMTP, and the remainder by FTP and HTTP. When any designated flow tends to exceed its allocation, new connections belonging to that class (e.g., new SMTP connections for 144.16.64.x/19) are disallowed. Since all SMTP mail relays attempt to resend emails in case of an attempt failure, connection blocking for SMTP just means that emails get deferred. As for FTP and HTTP connections, these are simply blocked for some time, and users get a message from the web proxy that the network is congested.

IV. Some Measurement Results from a Campus Deployment

Figure 1 shows the campus network, the noncampus nodes connected to the campus Network Operations Centre (NOC)[1], and the Internet access link. A 64Kbps link attaches the campus router to a campus in another city. We call this a "transit" link. The Internet access link was 128Kbps until June 7, 1999, when it became a 2Mbps link.

[1]Our project (namely, ERNET) also provides Internet services to other educational and research institutions

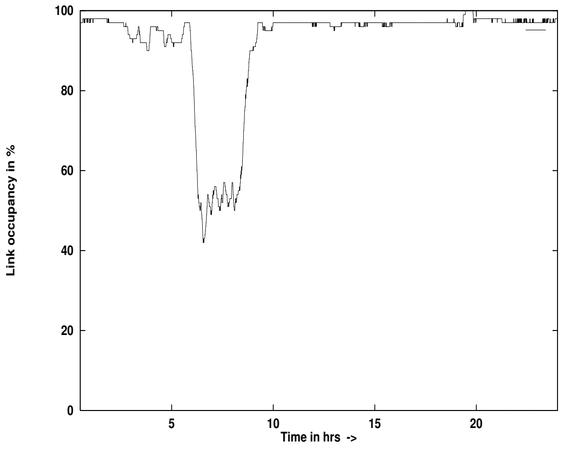

Fig. 3. Inbound occupancy of the access link vs. time of the day, on a day on which there were no bandwidth controls. Link speed 128Kbps.

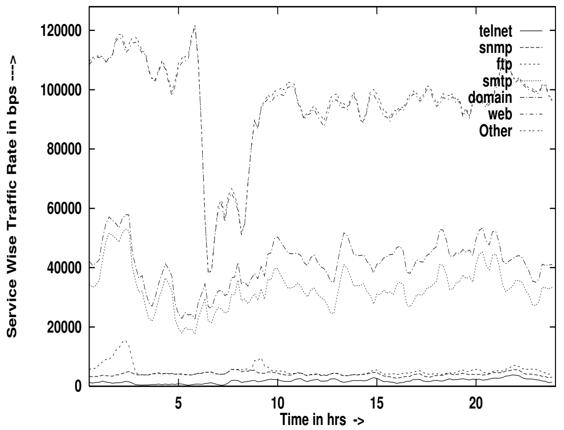

Fig. 4. Service-wise bit rates carried by the access link into the campus vs. time of the day, on a day on which there were no bandwidth controls. Link speed 128Kbps.

The traffic control machines are on the campus Network Operations Center (NOC) LAN. The various departmental Ethernet LANs on the campus are connected over a FDDI backbone, whereas the noncampus nodes are connected to the router by low speed serial lines. In this scenario, the campus network access, unless controlled, can hog all the available bandwidth of the shared link, causing poor service to the noncampus nodes. We present our experience in using the various control strategies available in our bandwidth management system in this scenario.

A. Comparison of Performance With and Without Controls (128Kbps link)

Figures 3 and 4 show measurements for a 24 hour period (midnight to midnight) on a day when no bandwidth management controls were in effect. On that day the access link speed was 128Kbps. Figure 3 shows the inbound occupancy of the access link, plotted every 30 seconds; the occupancy is obtained over a sliding window of 30 minutes. Notice that the link is saturated for most of the day, except for 2 to 3 hours between 6am and 9am, which is consistently a light traffic period on our campus[2]. Note that this is the total occupancy of the link, and includes the traffic into the campus, into the noncampus nodes, and into the 64Kbps transit link (see Figure 1). Figure 4 shows the service-wise break-up of traffic flowing into the campus. The plot shows 7 curves, stacked one on top of the other, in the order (from bottom to top) TELNET, SNMP, FTP, SMTP, DNS, web, and other. The plots are cumulative, i.e., the SMTP curve (fourth curve from the bottom; the light dotted line) shows the total bit rate due to TELNET, SNMP, FTP and SMTP; thus the width of the band between the curves for FTP and SMTP is the bit rate into the campus due to SMTP. Note also that the "other" curve represents the total carried bit rate *into the campus*, including all the services. A sliding window of 30 minutes has been used, and points are plotted every 90 seconds. We observe that, without bandwidth management, the campus utilises at least 100Kbps out of 128Kbps, leaving very little for the noncampus nodes (see the end of this subsection), and for transit traffic. Apart from squeezing out other users, an important consequence of operating the link without any controls is that very poor throughput is seen by all users. We will demonstrate this in Figure 7.

In Figures 5 and 6 we show the inbound access link occupancy and service-wise break-up on a day when controls were applied. The controls attempt to keep the occupancy of the access link between 88% and 90%. Between 9am and 9pm the campus was allowed to use at least 60Kbps of the access link; if spare bandwidth was available this could go up to 80Kbps. During this period SMTP was allowed a maximum rate of 25Kbps. Between 9pm and 9am, the campus could use any spare bandwidth that was available. Notice (see Figure 5) that the controls succeed in keeping the access link occupancy at around 90% during the peak hours. Further, Figure 6 shows that the service-wise controls were also effective. During late nights, when the students are most active on the network, the web util-

[2]The measured occupancy is a couple of percent less than 100% since we are not able to count all the bytes that flow on the link; e.g., HDLC header bytes.

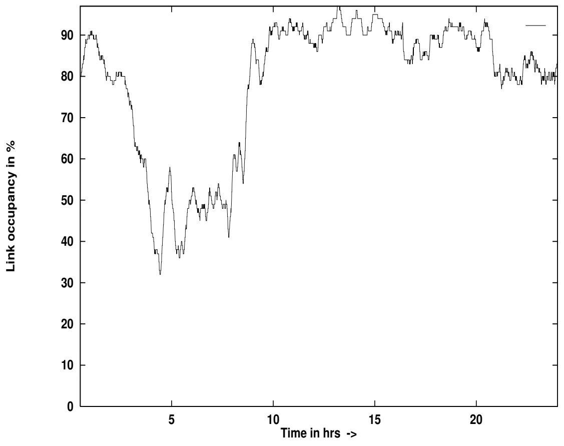

Fig. 5. Inbound occupancy of the access link vs. time of the day, on a day on which bandwidth controls were in place. Link speed 128Kbps.

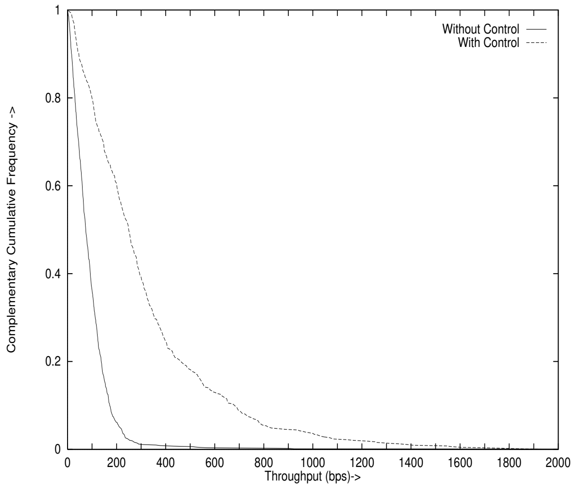

Fig. 7. The complementary frequency distribution of throughputs for the busy hours (10am to 12noon), on a day with no control and a day with control. The throughputs were measured at the campus web proxy. Link speed 128Kbps.

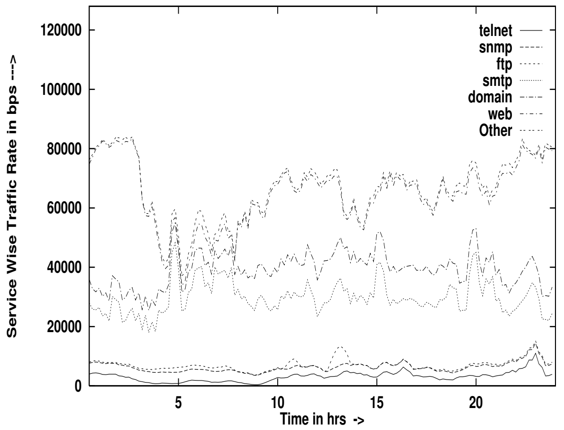

Fig. 6. Service-wise bit rates carried by the access link into the campus vs. time of the day, on a day on which bandwidth controls were in place. Link speed 128Kbps.

isation grows substantially, and since the daytime limits are relaxed, notice that the overall bit rate into the campus grows after 9pm, and typically stays high for much of the night (note that this increase is due to web traffic). Observe also, from Figure 6, that the TELNET traffic (lower most curve) is a little higher with controls in place; this is probably because users obtain better response times, and hence tend to use TELNET more.

Recall that in Section II we argued analytically that as the access link occupancy increases the throughput obtained by sessions will decrease. We now examine how this observation bears out in practice by comparing the session throughputs with and without controls. All the web access in the campus passes through a web proxy. Using the transfer logs on this proxy, and removing partial transfers and local transfers, we obtain the throughputs of file transfers that actually came over the 128Kbps access link. Figure 7 shows the complementary frequency distribution of file transfer throughputs with and without control, during the busy hours of two days, one with control and the other without control. The improvement with control is significant. Without bandwidth management controls, the maximum throughput is about 500Bps (Bytes per second), whereas with control 20% of the transfers obtained throughputs more than 500Bps. The areas under the curves provide the average throughput. Without control the average throughput is about 100Bps, whereas with control this is about 320Bps. As predicted by the analytical models in Section II the throughput does not drop close to 0 since the number of active transfers at any time is bounded, whereas in the model as $\rho_b \to 1$ the number of active sessions becomes unbounded.

We have also observed that when controls are removed much of the additional traffic that is carried comprises *partial transfers*, as users become impatient with the slow response and abort ongoing HTTP transfer requests. Thus with controls the link is utilised more efficiently and the throughputs are also better.

Recall from Figure 1 that the campus access link also carries traffic for some noncampus nodes that are connected to the router via low speed links, and there is a possibility that without control, the higher speed campus access would hog the link. The effect on the noncampus nodes was found to be significant. These nodes are mainly active during the work day. Owing to lack of space we are unable to provide the measurement plots here. We found that without controls the total rate these nodes were able to get was only about 15Kbps, whereas with controls this more than doubled. Under the control policy, the noncampus nodes were allowed a minimum rate of 30Kbps, and a maximum of 40Kbps.

B. Determining How Much Bandwidth the Campus Needs (2Mbps link)

With a 128Kbps access link, and the bandwidth controls in place, the connection blocking probability was as high as 60%, and individual users would get repeatedly blocked for several minutes (since, after crossing the occupancy threshold, it would take long for the occupancy to come down below the threshold at which blocking would be turned off). So although the TCP connection admission control based bandwidth management resulted in good throughputs for successful connections, and efficient utilisation of the link, clearly a higher capacity link was needed. The Internet access link to the campus was upgraded to 2Mbps on June 7,1999. Initially all bandwidth controls were removed. Web traffic then dominated, and the total utilisation of the link reached 500Kbps at times. We wanted to determine the amount of bandwidth that was really needed by the campus subject to a small level of connection blocking (say 10%).

We applied the bandwidth controls at a level of 250Kbps aggregate rate into the campus, and progressively relaxed this constraint over a period of a week until we reached an acceptable operating point. When we limited the total bit-rate into the campus to 300Kbps, large, unacceptable, blocking probabilities of up to 50%, lasting for long time periods, were obtained. With an allowed bit-rate of 350Kbps, the blocking peaked to 10% – 20% during the peak load period, and was close to zero during much of the rest of the work day. Further, the controls would not be in the blocking state for more than a few tens of seconds. To users this kind of performance turned out not to be a serious hinderance. Thus we determined that the campus required about 350Kbps of bandwidth at that time. The rest of the bandwidth of the 2Mbps link could be used to provide off-campus services, perhaps for a fee.

V. CONCLUSION

In the context of the problem of a high speed campus network connected to the Internet by a relatively low-speed WAN access link, we have experimented with a TCP connection admission control based strategy for the control of the bandwidth of this link. Connection admission control appears to be the only way to guarantee some TCP throughput performance on an overloaded Internet access link. In this paper we have described our experiences with a campus deployment of a software system based on our approach, an implementation of the approach and the associated algorithms. We have presented several measurement results that show the efficacy of our approach.

In ongoing work we are exploring better algorithms for invoking connection admission control decisions. We are also studying relationships between admission control and the better known technique of connection queueing with fair service.

REFERENCES

[1] James R. Davin, "SNMP development Kit", Architecture group MIT Laboratory for Computer Science

[2] Robert Mandeville and David Newman, "Traffic Tuners: Striking the Right Note?", *Data Communications Magazine (Asia–Pacific)*, November 21, 1998.

[3] "LIBPCAP 0.4", Lawrence Berkeley Laboratory, FTP://FTP.ee.lbl.gov/libpcap.tar.Z, 1994

[4] J. Case, M. Fedor, M. Schoffstall and J. Davin "A Simple Network Management Protocol (SNMP)", RFC 1157.

[5] A.W. Berger and Yaakov Kogan, "Dimensioning Bandwidth for Elastic Traffic in High-Speed Data Networks," manuscript submitted for publication, 1998.

[6] D.P. Heyman, T.V. Lakshman, A.L. Neidhardt, "A New Method for Analyzing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet," *Performance Evaluation Review*, Vol. 25, No. 1, 1997.

[7] L. Kleinrock, *Queueing Systems: Volume 2*, John Wiley, New York, 1976.

[8] Anurag Kumar, K.V.S. Hari, R. Shobhanjali, and Srikumar Sharma, "Long Range Dependence in the Aggregate Flow of TCP Controlled Elastic Sessions: An Investigation via the Processor Sharing Model," *Proceedings National Conference on Communications, NCC 2000*, New Delhi, January 2000.

[9] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modelling", *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, pp 226-244, 1995.

[10] J.W. Roberts and L. Massoulie, "Bandwidth Sharing and Admission Control for Elastic Traffic," *ITC Specialists Seminar*, Japan, 1998.