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,
Indian Institute of Science

ABSTRACT

We describe our approach to monitoring and managing the bandwidth of an Internet edge link with a view toward 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 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.

INTRODUCTION: OBJECTIVES AND APPROACH

Figure 1 shows a campus network and its attachment to an Internet service provider (ISP). Typical campus backbone speeds are 100 Mb/s, with more recent installations at 155 Mb/s if based on asynchronous transfer mode (ATM) or 1000 Mb/s if based on Gigabit Ethernet. Campus endpoints are typically connected to the backbone by departmental LANs of at least 10 Mb/s. As opposed to these numbers, note that typical Internet access link speeds range from 64 kb/s to 2 Mb/s, 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 Fig. 1.

Our monitoring and control architecture is also shown in Fig. 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 Fig. 1 emphasizes the fact that the approach scales easily by adding more machines.

The approach to 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 utilization, link up/down percentages, and link bit error rates. The packet capture data is used to generate service (i.e., e-mail, HTTP, FTP) and source–destination statistics for the packets flowing on the Internet access link.

Based on link utilization policies (aggregate and per service) set up by the network manager, the monitor machine generates control commands that are sent to the controller machine on the Ethernet segment (Fig. 1). The controller machine then controls the TCP connections (i.e., we block new connections, and slow down existing ones, if necessary) so that the traffic flowing on the access link conforms to the configured policies. Some examples of policies that can be set up are:

- Do not allow the occupancy of the access link to exceed 90 percent; this threshold may be obtained, for example, from a model of TCP connections over a WAN with this link as the bottleneck link; the objective could be to meet a session throughput objective.
- Out of the usable portion (see previous item) of the access link rate in the incoming direction, reserve a part for Simple Mail Transfer Protocol (SMTP — e-mail), and

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

0163-6804/00/$10.00 © 2000 IEEE

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 utilized by the rest of the campus. This may help reduce the total bandwidth that must be provisioned for the access link.

There are several advantages to our approach:

- *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.
- *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.
- *Easy scalability to increasing access link speeds*: The simplest installation can just be on one 200 MHz Pentium machine, running both the monitor and manager modules. Additional machines can be added as the link speed and the complexity of management policies increase.
- *Conformity with the RMON philosophy*: The Remote Monitoring specification, 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* (e.g., [1]). 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 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 Queuing (WFQ), TCP acknowledgment 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 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; Fig. 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 of the approach.

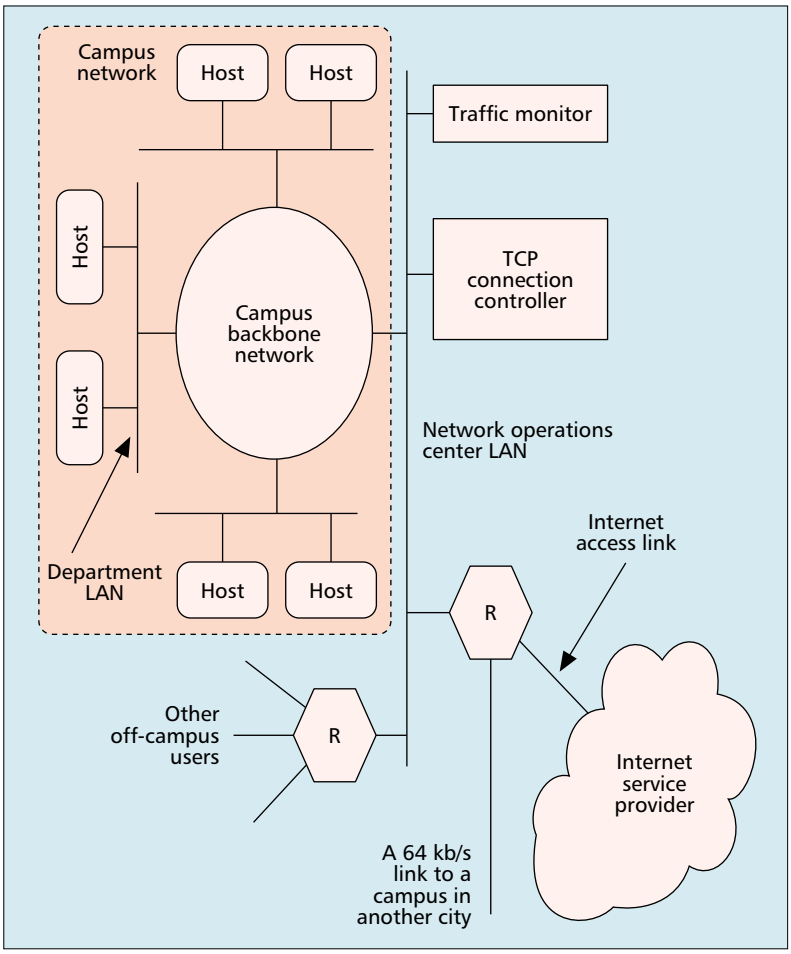

Figure 1. *A campus network with an access link to an Internet service provider. Note the off-campus users being served from the same access link, and the link to a campus in another city. Also note the placement of the monitoring and control machines.*

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

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

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 we have adopted. Consider again the situation shown in Fig. 1. We will assume that the predominant traffic is TCP-controlled elastic traffic (e.g., traffic generated by applications such as e-mail, FTP, HTTP, and SMTP). All of these applications essentially involve the transfer

The control algorithms need statistics for aggregate traffic flow on the WAN access link, and more detailed statistics of traffic flow. The network traffic is monitored using utilities based on SNMP and a TCP packet capture library.

of files. Requests for transfer of these files arrive in some way (e.g., 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 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 [2] 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 modeled by using the Processor Sharing (PS) model from queuing theory (see [3] for an introduction to the PS model, and [4–7] for studies involving the use of the PS model for TCP-controlled bandwidth sharing). The PS model assumes fair bandwidth sharing; that is, if there are n sessions active on the access link, then each session obtains exactly $1/n$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 encounter different propagation delays (for some studies of the applicability of such models to TCP, see [5, 7]).

Let λ denote the rate of arrival of file transfer requests, X the random file size (EX denotes the expected file size), and b 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 vs. 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 normalized 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, normalized 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 (normalized to b) decreases from 1 to 0 as ρ_b goes from 0 to 1. There are two consequences of this behavior:

- 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.
- If the campus network administrator wants to assure the users of 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 ensure an average bandwidth share of 20 percent of the access link bandwidth we need $\rho_b \le 0.93$).

Obviously, the only way of achieving these two objectives is to shed the excess load if $\rho > \rho_b^*$. In our approach we have used TCP connection admission control to achieve this load limiting (see also [4] for additional arguments for TCP admission control). Note that, from the model, it is clear that if a fraction $(\rho - \rho^*/\rho)$ of the arriving TCP connections are blocked, 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. Later we will provide some measurement results (e.g., the throughput distribution with and without admission control) that will demonstrate the efficacy of this approach.

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 Fig. 2. The software is shown split up over two computers: one with the software for monitoring, statistics estimation, and 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 [8] and a TCP packet capture library [9] (Fig. 2). Our implementation of the SNMP-based monitoring software is based on the MIT SNMP development kit [10].

The detailed breakup of the traffic flow on and off 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 [9].

We now describe the various software modules shown in Fig. 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 access link occupancy, and bit rates for various services to and from configured sets of IP addresses on the campus LAN.
- Control policy configuration module

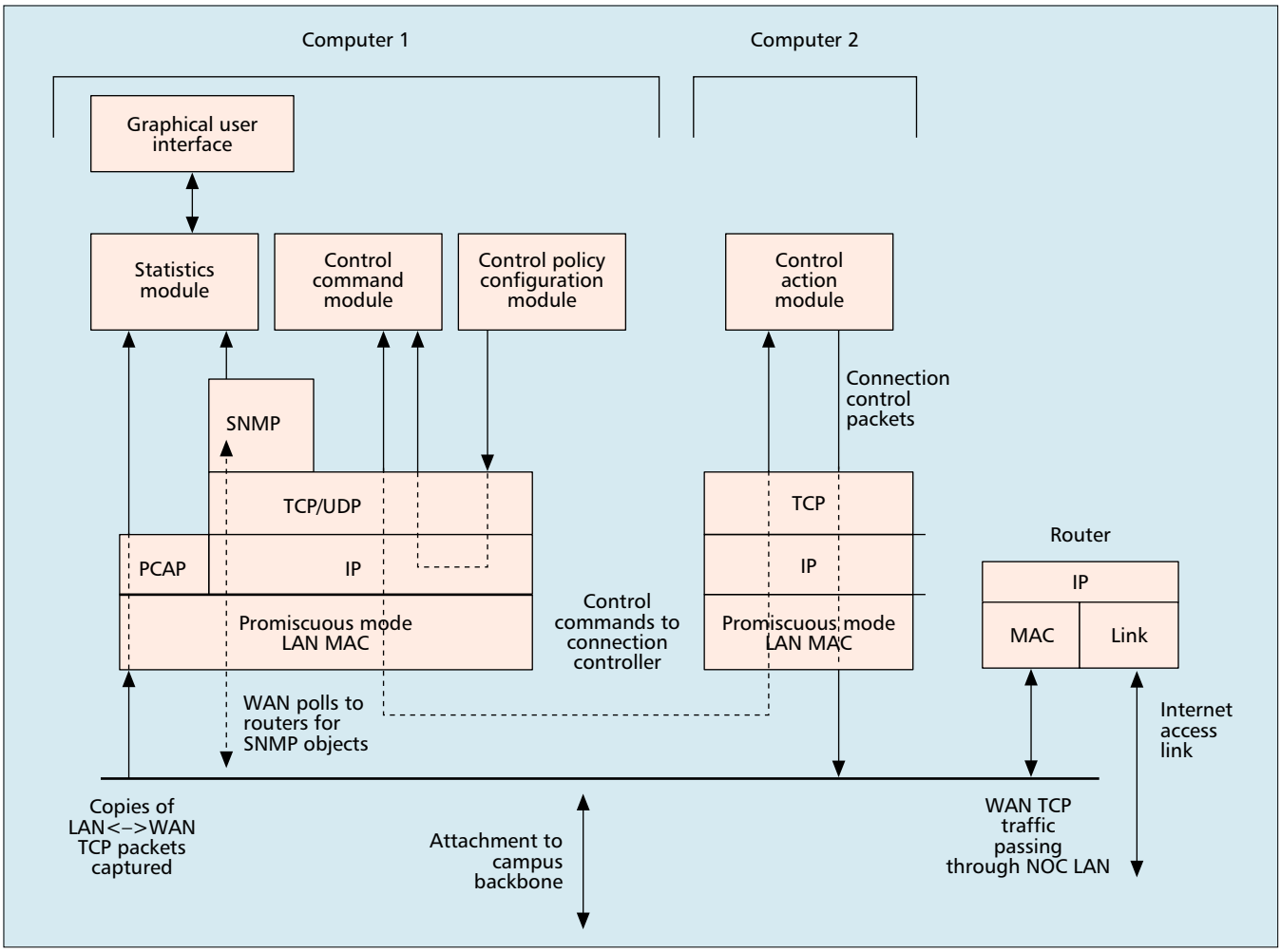

Figure 2. *The software modules, their relationships, and the flow of monitoring and control in the system. A two-computer implementation is assumed.*

(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 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 to determine whether or not a certain stream of packets or type of connections needs to be controlled. Simple algorithms are used to trigger the controls for each type of traffic, to enforce the policy settings. These algorithms basically estimate the bandwidth utilization 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 upward; blocking is turned off when another *smaller* threshold is crossed downward. Control command packets are generated and sent to the control action module (below) as UDP packets.
- Control action module (CAM): Commands from the CCM cause this module to send TCP connection control packets to the endpoints 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 slow down certain other ongoing TCP connections. It does this by IP masquerading, and sending TCP reset (RST) packets and ICMP source quench packets to one of the hosts involved in a connection. See "Nonintrusive Control of TCP Connections" below for details.

Figure 2 shows the SM, CCM, CPCM, and graphic user interface (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 CAM. The CCM communicates with the SM using TCP socket calls as well

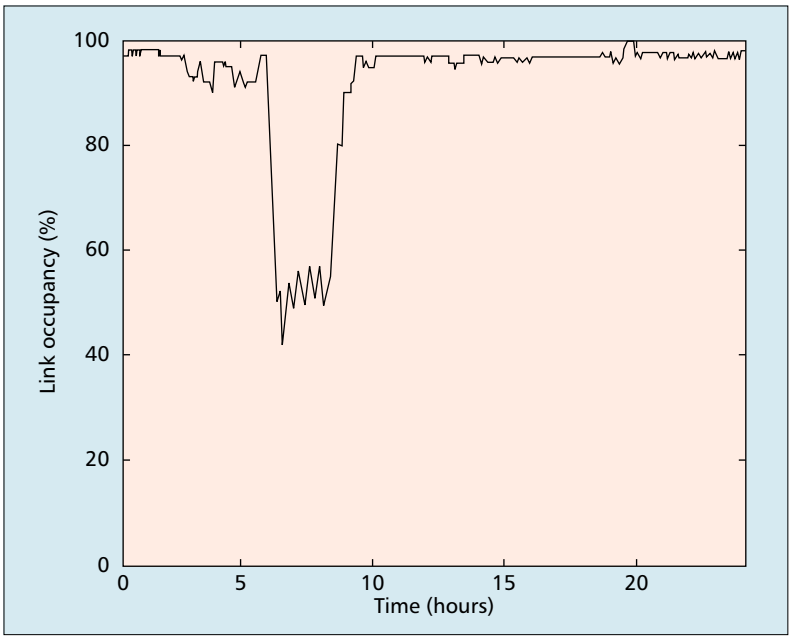

Figure 3. *Inbound occupancy of the access link vs. time of the day, on a day on which there were no bandwidth controls; link speed 128 kb/s.*

as function calls. The CCM communicates with the CAM via UDP. Since this communication is over the LAN and very frequent, TCP connection overhead was felt unnecessary. Whenever any information available in the CAM (e.g., the connection kill rate for a particular set of IP addresses and port numbers) is requested by the 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 nonintrusive, the CAM promiscuously captures SYN and SYN-ACK packets, masquerades as one of the connection endpoints, and generates a TCP RST message to the other endpoint. 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 endpoint on the campus network with the CAM IP-masquerading as the remote connection endpoint. 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 unknown" type for CAC. 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–destination pair. Also, most routers have access lists that block ICMP messages. This latter problem also affects the ICMP source quenches 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. CAC is achieved by sending a TCP RST to the local endpoint of a connection, a campus proxy.

Traffic Control Policies — We have experimented extensively with control policies that involve:

- A target WAN access link occupancy
- Per-service or per-IP-address aggregate bandwidth allocation

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

A typical service aggregate bandwidth allocation could be the following. For example, the access link speed is 64 kb/s; the target occupancy is 90 percent, yielding a usable throughput of 57.6 kb/s. The inbound direction is that which limits the flow of traffic (because inbound mail and Web traffic is significantly larger). Allot 25 kb/s 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 20 kb/s be used by SMTP, and the remainder by FTP and HTTP.

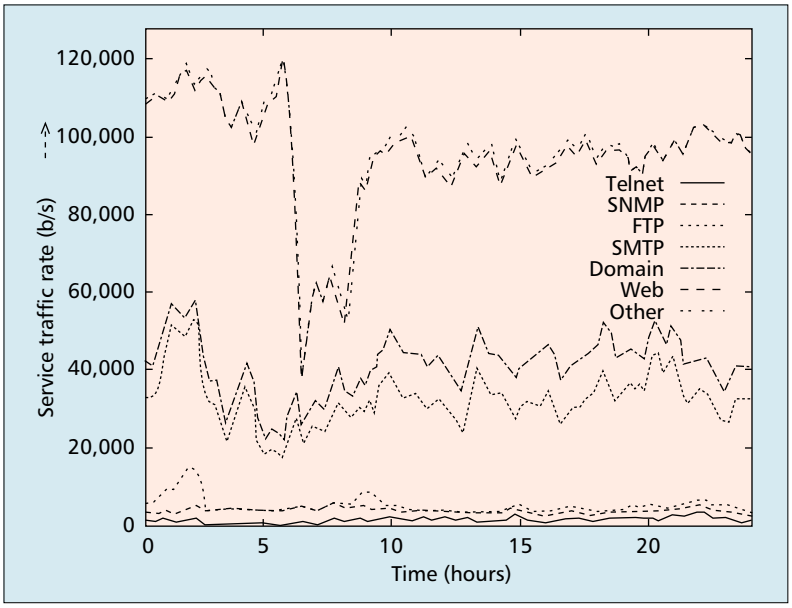

Figure 4. *Service bit rates carried by the access link into the campus vs. time of the day, on a day on which there were no bandwidth controls; link speed 128 kb/s.*

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 e-mails in case of an attempt failure, connection blocking for SMTP just means that e-mails 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.

Some Measurement Results from a Campus Deployment

Figure 1 shows the campus network, the noncampus nodes connected to the campus network operations center (NOC),[1] and the Internet access link. A 64 kb/s link attaches the campus router to a campus in another city. We call this a *transit* link. The Internet access link was 128 kb/s until June 7, 1999, when it became a 2 Mb/s link. The traffic control machines are on the campus NOC LAN. The various departmental Ethernet LANs on the campus are connected over a fiber-distributed data interface (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.

Comparison of Performance with and without Controls (128 kb/s Link)

Figures 3 and 4 show measurements for a 24-hr period (midnight to midnight) on a day when no bandwidth management controls were in effect. On that day the access link speed was 128 kb/s. Figure 3 shows the inbound occupancy of the access link, plotted every 30 s; the occupancy is obtained over a sliding window of 30 min. Notice that the link is saturated for most of the day, except for two to three hours between 6 a.m. and 9 a.m., 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 64 kb/s transit link (Fig. 1). Figure 4 shows the service breakup of traffic flowing into the campus. The plot shows seven 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; for example, the SMTP curve (fourth from the bottom) 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 min has been used, and points are plotted every 90 s. We observe that, without bandwidth management, the campus utilizes at least 100 out of 128 kb/s, leaving very little for the noncampus nodes and 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 Fig. 7.

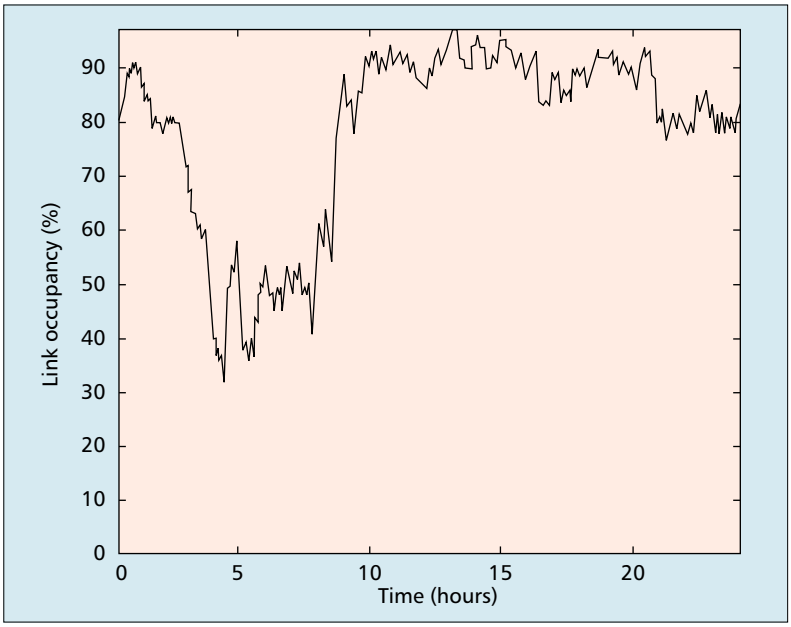

Figure 5. *Inbound occupancy of the access link vs. time of the day, on a day on which bandwidth controls were in place; link speed 128 kb/s.*

In Figs. 5 and 6 we show the inbound access link occupancy and service breakup on a day when controls were applied. The controls attempt to keep the occupancy of the access link between 88 and 90 percent. Between 9 a.m. and 9 p.m. the campus was allowed to use at least 60 kb/s of the access link; if spare bandwidth was available this could go up to 80 kb/s. During this period SMTP was allowed a maximum rate of 25 kb/s. Between 9 p.m. and 9 a.m., the campus could use any spare bandwidth available. Notice (Fig. 5) that the controls succeed in keeping the access link occupancy at around 90 percent during peak hours. Furthermore, Fig. 6 shows that the per-service controls were also effective. During late nights, when the students are most active on the network, Web utilization grows substantially, and since the daytime limits are relaxed, notice that the overall bit rate into the campus grows after 9 p.m. and typically stays high for much of the night (note that this increase is due to Web traffic). Observe also, from Fig. 6, that the TELNET traffic (lowermost 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 earlier 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 Web access in the campus passes through a Web proxy. Using the transfer logs on this proxy, and removing partial and local transfers, we obtain the throughputs of file transfers that actually came over the 128 kb/s

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

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

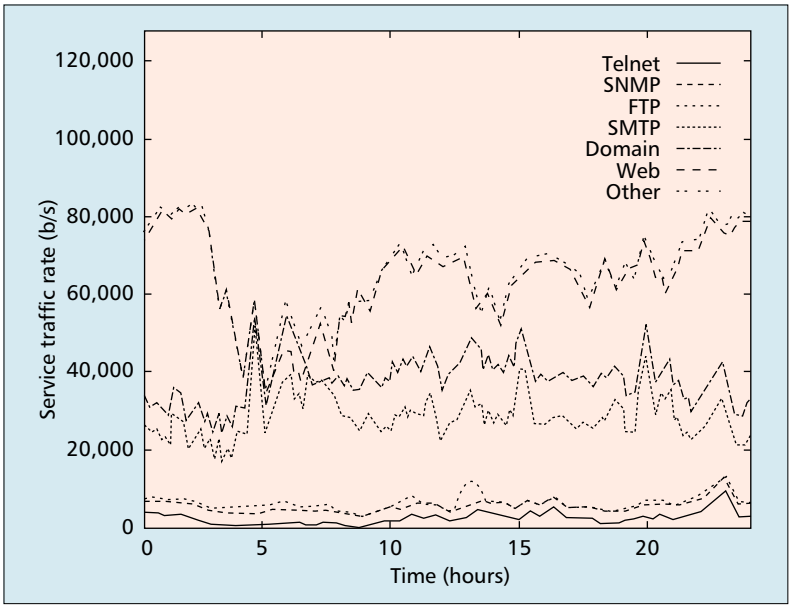

Figure 6. *Service bit rates carried by the access link into the campus vs. time of the day, on a day on which bandwidth controls were in place; link speed 128 kb/s.*

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 and the other without control. The improvement with control is significant. Without bandwidth management controls, the maximum throughput is about 500 bytes/s, whereas with control 20 percent of the transfers obtained throughputs more than 500 bytes/s. The areas under the curves provide the average throughput. Without control the average throughput is about 100 bytes/s, whereas with control this is about 320 bytes/s. As predicted by the analytical models earlier, the throughput does not drop close to 0 since the number of active transfers at any time is bounded, whereas in the model as $\rho_b \rightarrow 1$, the number of active sessions becomes unbounded.

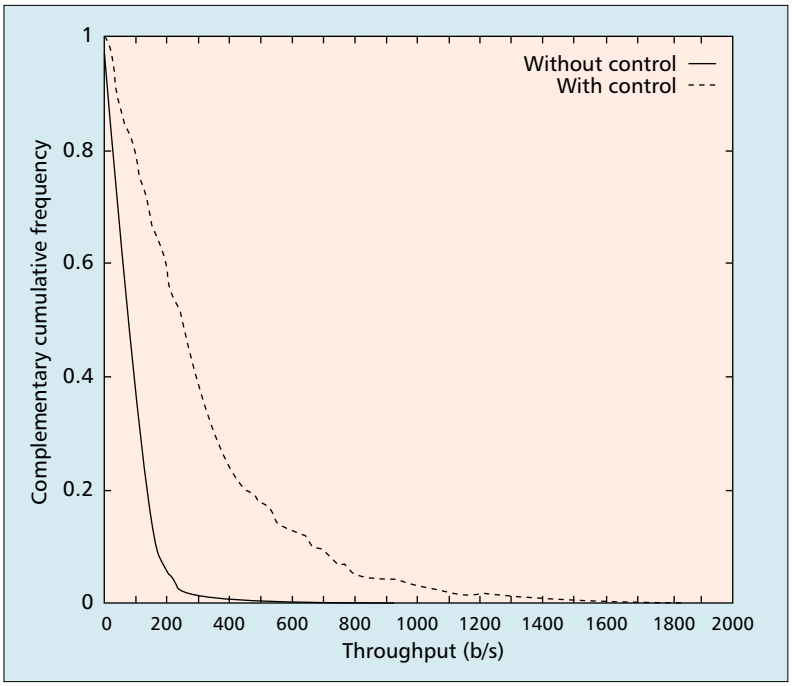

Figure 7. *The complementary frequency distribution of throughputs for the busy hours (10 a.m. to 12 noon), on a day with no control and a day with control. The throughputs were measured at the campus Web proxy; link speed 128 kb/s.*

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

Recall from Fig. 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 workday. Due 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 15 kb/s, whereas with controls this more than doubled. Under the control policy, the noncampus nodes were allowed a minimum rate of 30 kb/s and a maximum of 40 kb/s.

Determining How Much Bandwidth the Campus Needs (2 Mb/s link)

With a 128 kb/s access link and the bandwidth controls in place, the connection blocking probability was as high as 60 percent, and individual users would get repeatedly blocked for several minutes (since, after crossing the occupancy threshold, it would take a long time for the occupancy to come down below the threshold at which blocking would be turned off). So, although the TCP CAC-based bandwidth management resulted in good throughputs for successful connections and efficient utilization of the link, clearly a higher-capacity link was needed. The Internet access link to the campus was upgraded to 2 Mb/s on June 7,1999. Initially all bandwidth controls were removed. Web traffic then dominated, and the total utilization of the link reached 500 kb/s 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 percent).

We applied the bandwidth controls at a level of 250 kb/s 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 300 kb/s, large, unacceptable, blocking probabilities of up to 50 percent, lasting for long time periods, were obtained. With an allowed bit rate of 350 kb/s, the blocking peaked to 10–20 percent during the peak load period, and was close to zero during much of the rest of the workday. Furthermore, 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 hindrance. Thus we determined that the

campus required about 350 kb/s of bandwidth at that time. The rest of the bandwidth of the 2 Mb/s link could be used to provide off-campus services, perhaps for a fee.

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 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 article we describe our experiences with a campus deployment of a software system based on our approach, an implementation of the approach and the associated algorithms. We present 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 queuing with fair service.

REFERENCES

[1] R. Mandeville and D. Newman, "Traffic Tuners: Striking the Right Note?," *Data Commun. (Asia-Pacific)*, Nov. 21, 1998.

[2] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modelling," *IEEE/ACM Trans. Net.*, vol. 3, no. 3, 1995, pp. 226–44.

[3] L. Kleinrock, *Queuing Systems: Volume 2*, New York: Wiley, 1976.

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

[5] D. P. Heyman, T. V. Lakshman, and A. L. Neidhardt, "A New Method for Analyzing Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet," *Perf. Eval. Rev.*, vol. 25, no. 1, 1997.

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

[7] A. Kumar *et al.*, "Long Range Dependence in the Aggregate Flow of TCP Controlled Elastic Sessions: An Investigation via the Processor Sharing Model," *Proc. Nat'l. Conf. Commun.*, New Delhi, India, Jan. 2000.

[8] J. Case *et al.*, "A Simple Network Management Protocol (SNMP)," RFC 1157.

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

[10] J. R. Davin, "SNMP development Kit," Architecture Group, MIT Lab. for Comp. Sci..

BIOGRAPHIES

ANURAG KUMAR (anurag@ece.iisc.ernet.in) has a B.Tech. in electrical engineering from the Indian Institute of Technology, Kanpur, and a Ph.D. from Cornell University. He was with AT&T Bell Labs, Holmdel, New Jersey, for over six years. He is now a professor in the Department of Electrical Communication Engineering, Indian Institute of Science (IISc), Bangalore. His research interests are in the area of modeling, analysis, control, and optimization problems arising in communication networks and distributed systems. He is a Fellow of the Indian National Academy of Engineering.

MALATI HEGDE (malati@ece.iisc.ernet.in) received her Ph.D. from the Indian Institute of Technology, Kanpur, in graph theory. Currently she is working as a senior scientific officer in Education and Research in Computer Networking (ERNET), IISc, Bangalore. Her areas of interest are network management and protocol conformance testing.

S. V. R. ANAND (anand@ece.iisc.ernet.in) has a Bachelor's degree in electrical communications engineering from the IISc, Bangalore. He is a network software consultant to ERNET, IISc, and a consultant to the HFCL-IISc Research Programme. His interests are network applications development and network management.

B. N. BINDU (bindu@ece.iisc.ernet.in) has a Bachelor's degree in communications engineering from the University of Mysore, India. She was a project assistant at ERNET, IISc, Bangalore, and is currently at Oracle Software Development Centre, Bangalore.

DINESH THIRUMURTHY (dinesh@ece.iisc.ernet.in) is a project associate in the ERNET Project at IISc. He received an M.Sc.(Eng.) degree from the Department of Computer Science and Automation, IISc, in 1997, and a B.Tech. degree in Computer Science from the Indian Institute of Technology, Madras, in 1994. His research interests are computer networks, and network spplications that bring people closer to each other.

ARZAD A. KHERANI (alam@ece.iisc.ernet.in) received a B.E. (honors) in electronics from Government Engineering College, Raipur, in 1997, and an M.E. in telecommunications from IIS in 1999. Currently he is a Ph.D. student in the Electrical Communication Engineering Department of IISc. His research interests include performance analysis of communication networks.

We have observed that when controls are removed, much of the additional traffic that is carried consists of partial transfers, since users become impatient with the slow response and abort ongoing HTTP transfer requests.