

# LAS scheduling to avoid bandwidth hogging in heterogeneous TCP networks

Idris A. Rai, Guillaume Urvoy-Keller, Ernst W. Biersack

Institut Eurecom  
2229, route des Crêtes  
06904 Sophia-Antipolis, France  
{rai, urvoy, erbi}@eurecom.fr

**Abstract.** We propose using *least attained service* (LAS) scheduling in network routers to prevent some connections against utilizing all or a large fraction of network bandwidth. This phenomenon, which is also known as *bandwidth hogging*, occurs in heterogeneous networks such as networks with connections that have varying propagation delays, networks with competing TCP and UDP applications, and networks with multiple congested routers. LAS scheduling in routers avoids bandwidth hogging by giving service priority to connections that have sent the least data. In other words, a connection under LAS scheduler does not receive service if there is another active connection with less attained service. Simulation results in this paper show that this scheduling approach is more efficient than FIFO scheduling, and offers close to fair sharing of network bandwidth among competing connections in congested heterogeneous networks.

## 1 Introduction

TCP is the most widely used transport protocol in the Internet. The studies [4, 13] indicate that TCP controls about 80-90% of traffic sent over the Internet. TCP uses a closed loop mechanism between source and destination that attempts to fairly allocate bandwidth to competing users. The tasks of TCP include controlling the transmission rate of the source and ensuring reliable delivery of data from source to destination. For this purpose, TCP must constantly adapt to network conditions in terms of the available bandwidth and congestion. Despite its popularity, TCP poses performance problems in some network environments such as in *heterogeneous networks*. This paper considers three types of heterogeneous networks, namely networks with varying propagation delays, networks with applications using either TCP and UDP protocols, and networks with multiple congested routers. These networks are known to allow some connections to *unfairly* occupy a large fraction of bandwidth, which is also called *bandwidth hogging*. The terms unfairness and bandwidth hogging will be used interchangeably in this paper.

Network measurements have shown that RTTs experienced by TCP connections are widely varying (Figure 1 in [1]). TCP inherently causes each TCP flow to receive a bandwidth that is inversely proportional to its *round trip time* (RTT) [14]. Hence, TCP connections with low RTT may unfairly receive a large allocation of network bandwidth compared to other TCP connections in the network with a high RTT. This also explains

the problem of TCP in networks with multiple congested routers. In these networks, a connection that traverses more links also has longer RTT than connections that cross fewer links. It is shown in [6] that TCP networks with multiple congested routers bias against a connection with long RTT to the extent of attaining unacceptable throughput. UDP-based applications are oblivious to congestion and make little or no attempt to reduce transmission rates when packets are lost by the network. Instead, they attempt to use as much bandwidth as their source rates. Thus, UDP-based streaming applications are likely to cause congestion and unfair bandwidth allocations against TCP-based applications. Most of the proposed solutions to avoid bandwidth hogging suggest modifying the TCP protocol itself, e.g., [3] and a few of the previous work propose solutions inside the network. The work in this paper is more related to network-based solutions that propose to use different buffer management and scheduling algorithms in network routers.

Random early detection (RED) [7] is a buffer management scheme that distributes packet loss rates to a small number of connections to prevent oscillation problem that occurs in FIFO routers with drop tail queues when multiple connections repeatedly experience packet losses. It is well known that connections get the same throughput if they have the same per-connection queue length. RED does not try to control per-connection buffer occupancy; therefore, it is shown that it doesn't always avoid unfairness [12]. Bandwidth hogging problem in RED networks still exists, since it is possible for other connections to occupy a large fraction of network bandwidth.

Flow RED [12], on the other hand, improves the fairness by sharing the buffer occupancy fairly among active TCP connections and also when TCP connections are mixed with UDP connections. Deficit Round Robin (DRR) [17] assigns a sub-queue to each flow in the router buffer, and each sub-queue to a deficit counter. DRR uses deficit counter to make sure that each connection utilizes no more than the pre-defined service rate in a round manner. Through this, DRR provides reasonable fair service among connections. Using simulations, [9] shows that DRR is fair for heterogeneous networks where connections have different propagation delays and capacities. When service rates of each DRR queue are set proportional to each connections input link bandwidth, an almost complete fairness is observed.

This paper proposes to use a priority based scheduling known as *least-attained service* (LAS) to prevent bandwidth hogging in heterogeneous networks. The simulation results presented in this paper show that, unlike FIFO scheduling, LAS scheduling in routers prevents any connection, regardless of its propagation delay or its transport protocol, to utilize all network resources throughout the active duration of the connection.

## 2 LAS Scheduling in Packet Networks

Least attained Service (LAS) scheduling is a well studied policy in operating systems and is also known in the literature as *foreground-background* (FB) [11] or *shortest elapsed time* (SET) first [5] scheduling. Recently, LAS has also been considered as a possible scheduling policy in packet switched networks that can replace FIFO [15]. In packet scheduling, LAS is defined as a scheduling policy that gives service to a connection in the system who has received the least service. In the event of ties, the set

of connections having received the least service shares the processor in a Round-Robin fashion [8]. A newly arriving connection always preempts the connection currently in service and retains the processor until it terminates, or until the next arrival appears, or until it has obtained an amount of service equal to that received by the connection preempted on arrival, whichever occurs first. LAS is therefore a priority based scheduling that gives service to the highest priority connection, which is the one with the least attained service of all.

We simulate LAS-based routers by using the network simulator ns-2 [10]. LAS-based routers maintain a single priority queue and insert each incoming packet at its appropriate position in that queue. The less service a connection has received so far, the closer to the head of the queue its arriving packets will be inserted. When a packet arrives and the queue is full, LAS first “inserts” the arriving packet at its appropriate position in the queue and then drops the packet that is at the end of the queue. A fast and efficient hardware architecture of highest-priority-first schedulers (like LAS) is implemented in [2]. In particular, the work [2] shows that the implementation can support high speed connections with up to 10Gb/s rates and over 4 billion priority levels.

LAS should avoid bandwidth hogging since it inserts packets of a connection that has received the most service at the tail of the queue. Thus, this connection does not receive service until other connections have received equal amount of service or these connections are idle. Also, packets at the tail of queue (belonging to connections that have received the most service under LAS) are dropped in case the queue being full. Recall that dropping these packets in TCP networks, makes corresponding sources to reduce their rates.

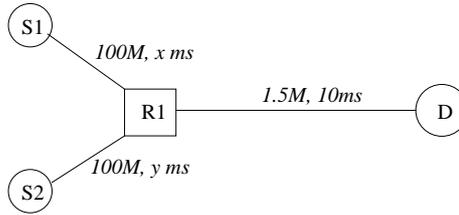
In our previous works [15, 16], we looked at the interaction between LAS and TCP and showed that LAS scheduling in packet networks reduces the transfer times of short TCP connections without starving the largest ones. The improvement seen by short TCP flows under LAS is mainly due to the way LAS interacts with TCP algorithm in the slow start (SS) phase, which results in shorter round-trip times and very low packet loss rates for short flows. This paper shows that LAS prevents unfair bandwidth allocations among competing connections in all heterogeneous networks considered.

### **3 LAS in Heterogeneous Networks with Single Congested Router**

In this section, we analyze the performance of LAS in simple heterogeneous networks with a single congested router. We consider networks with varying propagation delays and networks with TCP-based applications competing against UDP applications. We simulate a simple network topology shown in Figure 1, where two sources S1 and S2, send data to the same destination D through a bottleneck link R1-D, where LAS or FIFO scheduler is deployed. We vary propagation delay values  $x$  and  $y$  and we set source types to TCP or UDP.

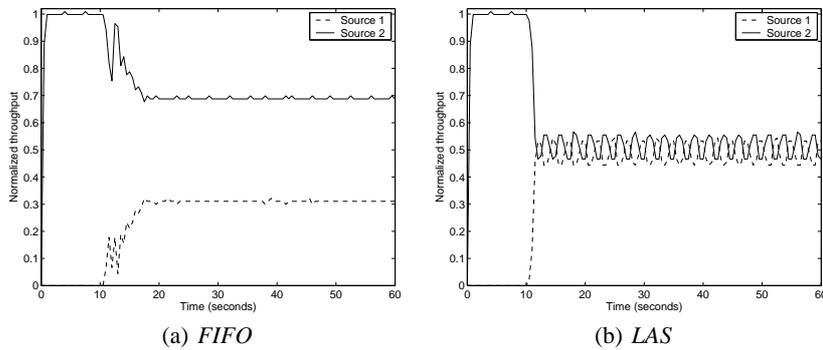
#### **3.1 TCP Sources with varying RTTs**

Internet traffic is a mixture of traffic traversing different paths, with links of different capacities and propagation delays e.g., asymmetric links such as those that use both



**Fig. 1.** A network with a single shared link

terrestrial and satellite links, ADSL or cable modem. In this section we study the impact of LAS to the performances of TCP connections with varying RTTs. The fact that the offered throughput under TCP is inversely proportional to connection's RTT, is widely known to cause ineffective bandwidth utilization among competing flows. That is, TCP gives a higher share of bandwidth to connections with low RTTs than to connections with high RTTs. FIFO schedulers, on the other hand, service packets in accordance of their arrivals: it takes no action to avoid a flow monopolizing all the service. In this section, we present simulation results that show that LAS scheduling is a suitable policy that avoids this problem. LAS schedules connections taking into account the received service of connections, and schedules first the packets that belong to a connection that has received the least service. Thus, LAS either drops or buffers packets of a connection that has attained the most service. In doing so, LAS increases the queuing delay of the connection, which stretches its RTT and so reduces its throughput.



**Fig. 2.** Throughput obtained by TCP flows with different RTTs

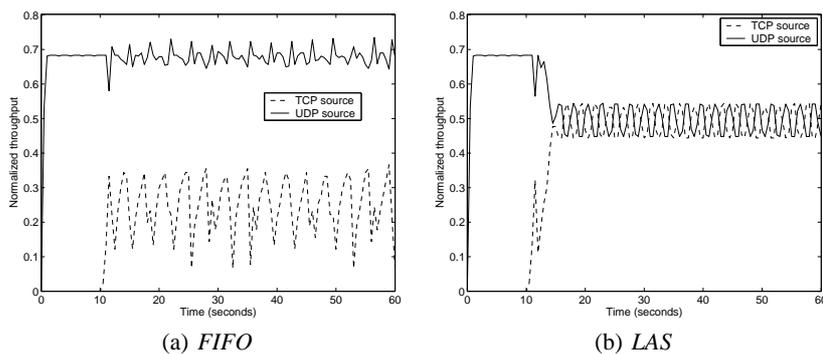
We study the network shown in Figure 1, where both sources are TCP-based and propagation delays for links S1-R1 and S2-R2 are  $x = 100ms$  and  $y = 1ms$  respectively. Source S2 starts transmitting data at time 0 and source S1 starts 10 seconds later. Figure 2(a) shows simulation results when the queuing discipline at link R1-D is FIFO with droptail mechanism and Figure 2(b) shows the results for LAS. We clearly note the benefits of LAS over FIFO; source S2 attains equal share of bandwidth as source S1

under LAS throughout a simulation duration. For the case of FIFO, however, the source transmitting over low RTT links (S2), receives a significantly larger share of bandwidth than does source S1.

### 3.2 Competing UDP and TCP Sources

Supporting UDP-based applications in the Internet is known to be difficult mainly because they can not respond to network congestion like TCP-based applications. As a result, when transmitted over the same link with TCP-based applications, UDP applications tend to take as much bandwidth as their source rates. In this section, we present simulation results that first illustrate this problem in the current Internet architecture (with FIFO schedulers) and then show that, under the same network, LAS can fairly allocate bandwidth among competing UDP and TCP sources.

We again use the network shown in Figure 1, where S1 is a TCP source and S2 is a UDP source transmitting constant bit rate (CBR) application at a rate of 1024Kbps. The propagation delays  $x$  and  $y$  are both set to 100ms.



**Fig. 3.** Throughput obtained by UDP and TCP connections

Figure 3(a) shows the simulation results in terms of the throughput obtained by connections under FIFO and Figure 3(b) shows the result under LAS. We note the unfairness effect for the case of FIFO, where UDP applications occupies the same amount of bandwidth in the presence of a TCP connection as when it is alone in the network. We also observe for the case of FIFO that the throughput of the TCP source shows oscillations with sharp peaks. This is due to frequent packet losses of the TCP connection. In contrast, the TCP connection acquires the same throughput as the UDP connection in the case of LAS. This is achieved by dropping some packets of the UDP connection. These results demonstrate that LAS fairly allocates bandwidth among flows with different protocols.

## 4 LAS in Networks with Multiple Congested Routers

The performance of TCP in a network with multiple congested routers was first studied by Floyd [6]. The work in [6] reveals that a network with multiple routers and FIFO schedulers biases against connections that traverses multiple routers (also have long RTT values) to the extent that they may receive very low throughput performance. In this section we consider the effect of multiple congested routers on the throughput of connections when LAS scheduling is implemented in the routers. We compare the results obtained for the case of LAS routers to FIFO routers at bottleneck links.

### 4.1 All ftp connections

We first study LAS under the topology used in [6], also shown on Figure 4. All buffer sizes are limited to 60 packets, and the maximum window size is 100 packets. From Figure 4, connections 1-5 are ftp connections that traverse a single congested router and have a low propagation delay of 70 ms each, whereas connection 0 is an ftp connection that traverses multiple congested routers and its propagation delay is 470 ms. Thus, connections 1-5 have short RTTs and connection 0 has a long RTT. We analyze the throughput performance of both types of connections when schedulers at bottleneck links, i.e., links 1a-1b, 2a-2b, ..., and, 5a-5b, are either all FIFO or all LAS. All connections send packets during the whole simulation duration.

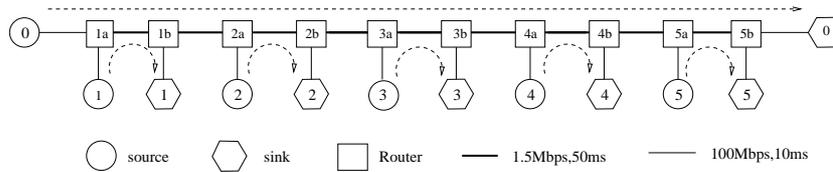


Fig. 4. Simulated network topology

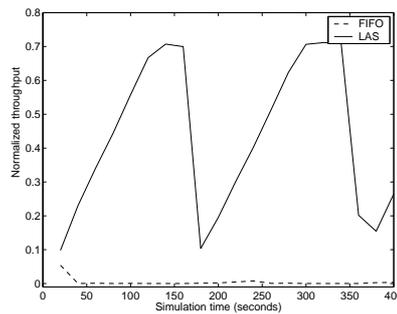
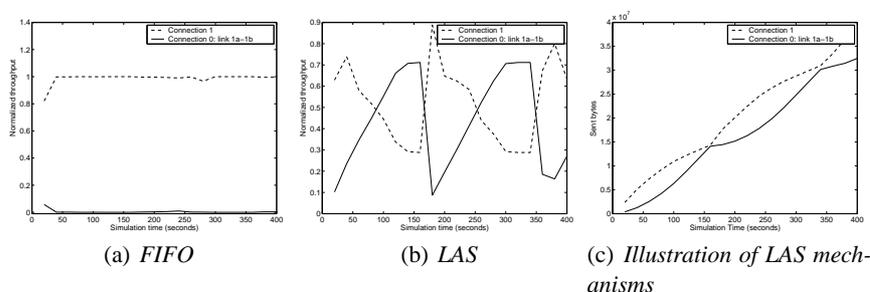


Fig. 5. Throughput of connection 0

We first analyze the overall throughput obtained by the connection with high propagation delay, (connection 0) under LAS and under FIFO. Figure 5 shows the throughput of connection 0 for the network of congested routers with FIFO and LAS schedulers. The performance of the connection under FIFO is bad, as expected; its throughput stays at zero during almost the whole simulation, and all the network bandwidth is taken by connections with short RTTs. However, we observe that the throughput of the connection under LAS schedulers is high and the network does not bias against the long connection.

Figure 6 shows the throughput of connection 1 and connection 0 as observed at link 1a-1b. This figure illustrates how the two connections share the bandwidth at bottleneck link 1a-1b. We observe that for the case of LAS 6(b), two connections at the bottleneck almost evenly share the link bandwidth as opposed to the case of FIFO 6(a), where connection 1 occupies all the bandwidth and completely starves connection 0. The results at other congested links were observed to be the same as the results at link 1a-1b.



**Fig. 6.** Throughput at link 1a-1b

The throughput of both connections under LAS (see Figure 6(b)) is observed to oscillate in time. This is primarily caused by the way LAS scheduling works, but also depends on the network parameters used (see next section). We consider connections at link 1a-1b to illustrate the effect of LAS scheduling to oscillations in throughput. At time time 0, both connections have the same priority. Connection 1 initially has a higher source rate due to its short RTT and so rapidly occupies the available bandwidth. As connection 1 sends more packets, the priority of its packets decreases, and packets of connection 0 attain a higher service priority in the router. However, the rate of connection 0 increases very slowly due to its long RTT. These are the epochs during which we observe the slowly increase and slowly decrease in the throughput of connection 0 and connection 1 respectively. This continues until both connections have sent the equal amount of data, at the simulation time slightly later than 150 sec (also slightly before 350 sec, see Figure 6(c)). After this point, we observe a sharp increase in the throughput of connection 1 and a sharp decrease in the throughput of connection 0. These are the times when packets of connection 1 have a higher priority again than those of connection 0. Here, connection 1 rapidly occupies the bandwidth due to its high source rate. Observe in Figure 6(c) that connection 1 tends to send a larger amount of data than connection

0 in almost all times. This is again the impact of varying RTTs of the connections, which determine their source rates. Finally, Table 1 shows the number of lost packets

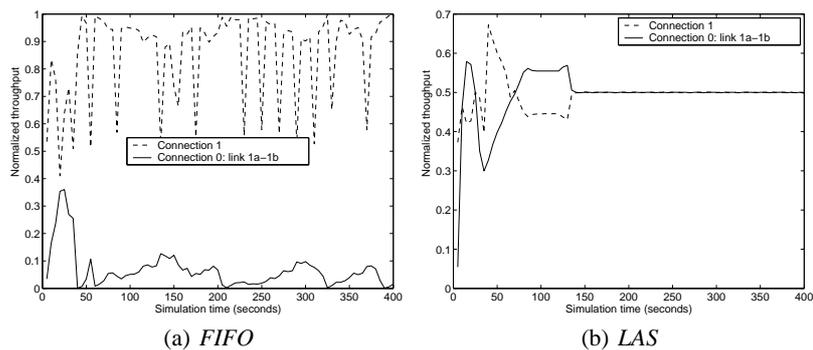
	All	Connections 1-5	Connection 0
LAS	1013	926	87
FIFO	649	569	80

**Table 1.** Number of lost packets of connections

for networks with congested routers with FIFO and LAS schedulers. The table shows that LAS loses more packets from short connection than does FIFO. The reason for a smaller number of all lost packets under FIFO than under LAS is that the source of connection 0 under FIFO schedulers completely backs off and does not send packets to the network for a long duration (has zero throughput). Despite giving acceptable throughput to connection 0, LAS also maintains approximately equal number of lost packets as FIFO for this connection. These results also show that LAS avoids the network bias towards connections with short RTT at the expense of only a slight increase in packets loss rate for connections with long RTT.

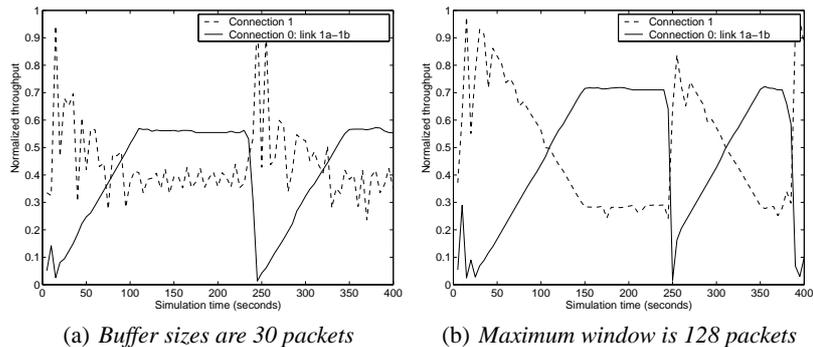
#### 4.2 Sensitivity of LAS to network parameters

We simulated LAS scheduling for the network topology shown in Figure 4 using slightly different parameters to investigate the sensitivity of the LAS scheduling to network parameters. We consider changing either link capacities, the maximum advertised window size, or the buffer size from the parameter set used in Section 4.1. Each of these parameters has an impact on the throughput of connections. For example, TCP source rate increases when increasing the maximum window size and decreasing only the buffer sizes increases the packet loss rate ( and thus the TCP source rate decreases).



**Fig. 7.** Throughput at link 1a-1b for network access link speeds of 10Mbps

Figure 7 shows the throughput results of connection 1 and connection 0 obtained at link 1a-1b when the network access links speeds are changed from 100Mbps to 10Mbps while keeping all other parameters the same as in Section 4.1. Observe the fairness in terms of throughput of the connections under LAS after a short time interval. The performance under FIFO remains almost the same as before where connection 1 occupies almost all the link bandwidth.

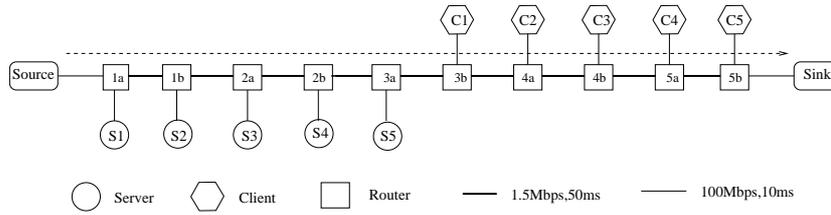


**Fig. 8.** Throughput under LAS scheduling at link 1a-1b

Figures 8 show the throughput of connections at link 1a-1b when buffer sizes are changed from 60 to 30 packets (Figure 8(a)), and when the maximum advertised window size is changed from 100 to 128 packets (Figure 8(b)), all other parameters being the same. We observe that the performance of both connections under LAS scheduling are similar to the results obtained in Section 4.1. Hence, we conclude that the performance of LAS is sensitive to network parameters. However, the results shown in this section indicate that the sensitivity of parameters only changes the transient throughput of connections and confirm that the LAS scheduler tends to fairly distribute available bandwidth among active connections.

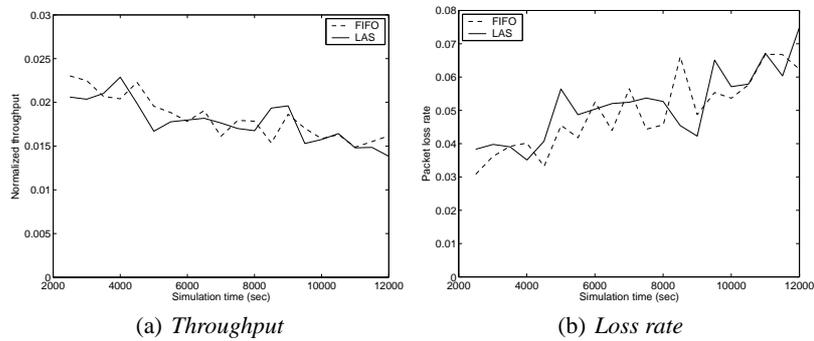
## 5 Network with Multiple Routers and Web Connections

In this section, the connections with short RTTs are Web file transfers with sizes distributed according to heavy-tail distribution. The heavy tail distributed file sizes constitute of many short connections and a few very large connections. This flow size distribution agrees with realistic traffic distribution in Internet today. We consider a Web model with a pool of Web clients that request files from a pool of servers. We simulate the topology (Figure 9) with network parameters as shown in the figure. C1-C5 denotes a pool of five clients and S1-S5 denotes a server pool of five servers. Thus, Web files can traverse at least one router (have low RTTs), whereas the ftp connection traverses all routers (has long RTT). The ftp connection starts sending packets after a warm-up period of 2000 seconds of simulation. All buffer sizes are limited to 60 packets, and the maximum window size is 64 packets.



**Fig. 9.** Simulated network topology

The topology in Figure 9 emulates a network scenario where clients from different autonomous systems (AS) access files stored in Web servers located in the same AS. In this case, the access link of the AS where servers are located is likely to be the bottleneck. We will consider the case when LAS is implemented only at this bottleneck access link (link 3a-3b). The generated Web traffic is expected to show a different impact to the performance of the ftp connection with long RTT under LAS compared to when connections with short RTTs are ftp file transfers. Since each time a client requests a file, it receives it using a new connection. This new connection has the highest priority under LAS since it has sent no data to the network. Thus, all arriving connections of Web files in the system are likely to maintain higher priorities than the ftp connection until they complete. The goal is to examine the performance of an ftp connection with long RTT that must traverse a number of routers populated with Web connections with short RTTs.



**Fig. 10.** Performance of an ftp connection under LAS and FIFO

Figure 10(a) shows the throughput of the connection with long RTT under LAS and FIFO when the load due to Web traffic is about 0.7. We observe that the throughput of the ftp connection under LAS and FIFO is closely the same at all times, and the ftp connection under both policies has low throughput. This shows that the throughput of a connection in this topology is more limited by the low data rate of the source, which is a result of its long RTT. Similarly, Figure 10(b) shows that packet loss rates of the ftp

connection under LAS and FIFO are also the same. The loss rates were computed in non-overlapping time windows of 500 sec. These results show that the fact that the ftp connection has a lower priority than Web connections during their transfer times under LAS has no negative impact on the throughput of the ftp connection for moderate load of Web traffic (i.e., load  $\rho < 0.9$ ). Instead, the high RTT value of the ftp connection is a dominant factor affecting its throughput. The results shown in this section also support the results in [15] that LAS in packet networks favors short flows while only negligibly penalizing large flows. We also simulated the topology of Figure 9 when LAS is implemented in all routers, the results is observed to be similar to results shown in this section when LAS is deployed only at the bottleneck link.

## 6 Conclusion

This paper shows that the bandwidth hogging problem, which is commonly experienced by some connections in heterogeneous TCP networks, can be alleviated when a router schedules packets according to the *least attained service* (LAS) scheduling. LAS schedules connections by giving service to a connection that has attained the least amount of service. As a result, LAS prevents any connection from occupying all or a large fraction of network bandwidth regardless of the difference or variation in propagation delays or varying transport protocols of competing connections as is the case for FIFO scheduling.

The simulation results presented in this paper show that for a network with a single bottleneck link, LAS maintains the same throughput between connections with long and short RTTs and between competing connections that use UDP and TCP transport protocols in the same network. When LAS is simulated in networks with multiple bottleneck links, the results indicate that connections with short RTTs do not starve connections with long RTTs. The performance of a connection with long RTT in a network with multiple bottlenecks is observed to be the same under LAS and under FIFO when Web transfers with short RTTs and with a realistic flow size distribution are used. This shows that while LAS is well known to favor short connections, it does not penalize long flows and the main factor that limits the throughput of the long flow in the topology used is its long RTT.

The results in multiple bottlenecks network with Web transfers also show that deploying LAS schedulers in all routers does not lead to performance improvement to ftp connection with long RTT compared to the results when LAS is implemented only at the bottleneck link. Thus, it is necessary to deploy LAS only at bottleneck links to benefit from its advantages. Fortunately, most bottleneck links in the Internet are access links where LAS implementation is scalable due to a moderate number of active connections available there.

## References

1. J. Aikat et al., "Variability in TCP Round-trip Times", In *Internet Measurement Conference 2003*, October 2003.

2. R. Bhagwan and B. Lin, "Fast and Scalable Priority Queue Architecture for High-Speed Network Switches", In *INFOCOM 2000*, pp. 538–547, 2000.
3. L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", In *Proceedings of the ACM SIGCOMM Conference*, London, England, 1994.
4. M. G. Claffy, K. and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone", In *Proceedings of INET '98, July 1998*, July 1998.
5. E. G. Coffman and P. J. Denning, *Operating Systems Theory*, Prentice-Hall Inc., 1973.
6. S. Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks", *ACM Computer Communication Review*, 21(5):30–47, Oct. 1991.
7. S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
8. E. L. Hahne, "Round-Robin Scheduling for Max-Min Fairness in Data Networks", *IEEE Journal of Selected Areas in Communications*, 9(7):1024–1039, 1991.
9. G. Hasegawa and M. Murata, "Survey of Fairness Issues in TCP Control Mechanisms", *IEICE Trans. on Communications*, E84-B(6):1461–1472, June 2001.
10. <http://www.isi.edu/nsnam/ns/>, "The Network Simulator ns2",
11. L. Kleinrock, *Queuing Systems, Volume II: Computer Applications*, Wiley, New York, 1976.
12. D. Lin and R. Morris, "Dynamics of Random early Detection", In *Proc. of the ACM/SIGCOMM'97*, pp. 127–137, October 1997.
13. B. Nandy et al., "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks", In *Proc. IFIP High Performance Networking, HPN 2000*, Paris, June 2000.
14. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A simple Model and its Empirical Validation", In *Proceedings of the ACM SIGCOMM Conference*, Vancouver, British Columbia, Canada, September 1998.
15. I. A. Rai, E. W. Biersack, and G. Urvoy-Keller, "Analyzing the Performance of TCP Flows in Packet Networks with LAS Schedulers", RR-03.075, April 2003.
16. I. A. Rai, G. Urvoy-Keller, and E. W. Biersack, "Analysis of LAS Scheduling for Job Size Distributions with High Variance", In *ACM Sigmetrics 2003*, pp. 218–228, June 2003.
17. M. Shreedhar and G. Varghese, "Efficient Fair Queueing using Deficit Round Robin", *ACM Computer Communication Review*, 25(4):231–242, October 1995.