22

A new congestion control framework for large bandwidth-delay product networks

Hyogon Kim and David J. Farber
Distributed Systems Laboratory
University of Pennsylvania
200 S. 33rd St., Philadelphia, PA 19104-6389, USA
Phone: (215) 898-5423, Fax: (215) 573-2232
e-mail: hkim@dsl.cis.upenn.edu

Abstract

The performance of adaptive congestion control methods will not scale well with bandwidth increase, due to elongating feedback loop and "wired-in" conservativeness of its rate change algorithms. We prove the point by showing that an experimental approach, with a non-adaptive non-conservative control algorithm, can yield order-of-magnitude larger network power and better bandwidth scalability than TCP Reno for multi-megabit/gigabit networks. This result suggests that generally congestion control algorithms should get more aggressive in high-speed networks, and the additional aggressiveness should be a variable determined by the ratio between transported data size and bandwidth-delay product.

Keywords

Congestion control, bandwidth-delay product, TCP

1 INTRODUCTION

In relatively low-speed networks, we could implicitly assume that data sources have "infinite" data from feedback control's point of view. Congestion control algorithms had to be simply "conservative" enough to keep network stability, maximum network utilization being less of an issue. But under orders-of-magnitudes larger bandwidth-delay product in multi-megabit/gigabit networks, significant fraction of typical transported data can fit within the bandwidth-delay product, and the infinite source assumption of existing congestion control approach is effectively abolished. For feedback control, it is increasingly hard to find the balance between network stability and maximum network utilization. If it gets too aggressive, traffic sources may flood the network before any feedback can choke them, while too cautious control will likely result in under-utilization. It is up to the source rate adaptation algorithms to adjust the degree of conservativeness(or, aggressiveness) and find the balance

R. Puigjaner (ed.), *High Performance Networking*

in large bandwidth-delay product networks. Unfortunately, the degree of conservativeness in existing algorithms is "wired-in", insensitive to the value of bandwidth-delay product. Moreover, the algorithms are too conservative for multi-megabit/gigabit networks. In this paper, we verify the above statements and consider the implications to the design of congestion control methods for high-speed networks.

Many newer congestion control methods have been proposed recently[11, 12, 2, 10] to replace traditional window-based flow/congestion control algorithms such as those used in TCP, but TCP in its current form is expected to continue to be used as the major data transport protocol owing to its enormous installation base, strong adaptability to wide variety of networks owing to its simple assumption on underlying networks, and incremental optimizations that have been proven successful so far. So, we will use TCP's performance to exemplify that of adaptive congestion control algorithms in large bandwidth-delay product networks in this paper. Also, even though the scope of our investigation is limited to window-based control, we conjecture that the implication extends to rate-based protocols which uses similar rate adaptation algorithms[19, 13]. We will further explore this issue in rate-based control in future paper.

In the next section, we perform a set of experiments and observe the effects of various bandwidth-delay product values on the performance of two different congestion control approaches. In the former category is TCP Reno, which we argue is conservative, and the latter is an experimental approach which is very aggressive in exploiting bandwidth. Section 3 concludes the discussion, and considers the implications of the results obtained in the experiments in the design of congestion control methods in high-speed networks. Finally, we discuss our works which are currently under way, as the extension of this investigation.

2 EXPERIMENT

In our experiments, we compare the performance of TCP Reno, and an experimental method, which we will call "Blitz" throughout the paper. Blitz is a non-adaptive and non-conservative algorithm, which sharply contrasts with TCP Reno which uses feedback(adaptive) control and is supposedly conservative. Blitz is non-adaptive in that it does not take any information on the network load to control the traffic source, and it is non-conservative in that it feeds the network with more than sufficient data at all times.

2.1 Blitz

In Blitz, the source can even transmit the entire file in a large burst, unless it is notified of a packet loss (in Fig. 1, packet n gets lost), in which case it starts a new burst starting from the lost packet. Each Blitz stream is prioritized with monotonically decreasing number, as can be seen in Figure 1. When a new burst starts, the priority is also reset to the highest level. As a data stream persists, the priority gradually goes down, and a switch can easily discriminate such streams in favor of shorter, newer, or equivalently, higher priority streams. This scheme distributes bottleneck bandwidth among contending sources in a relatively fair manner and gives priority to short, interactive-type traffic over long bursts. Network switches adopt a pushout scheme similar to [18] to prevent congestion collapse and

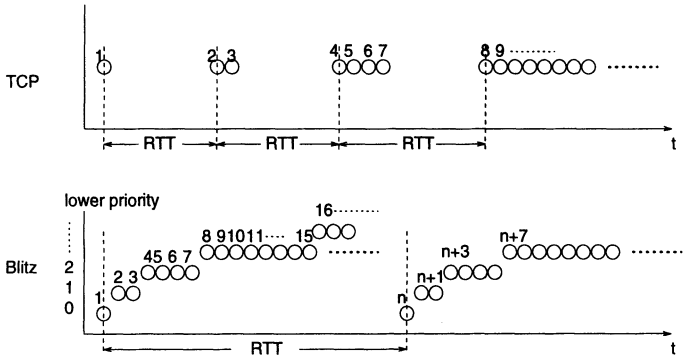

Figure 1: Comparison of TCP(in Slow Start) and Blitz stream structures; numbers are packet sequence numbers.

severe fairness problems. Other than these differences, Blitz adopts the same underlying mechanisms for window-based flow control as TCP (for instance, Blitz uses the same cumulative acknowledgement as TCP). In fact, Blitz is similar to old TCPs before Tahoe version where there was no mechanism to control or avoid congestion was incorporated. Both Blitz and TCP Reno in this paper assume that advertized receiver window is sufficiently large, and only the bottleneck bandwidth in the network matters for performance.

Finally, note that Blitz is a hypothetical scheme, and it is used only to provide a performance mark for TCP Reno to compare with. We will briefly discuss the practical problems of Blitz in being deployed in real networks.

2.2 Simulation setting

The simulation setting is intentionally simplified and generalized to observe the effects of a few important parameters more clearly, which could get diffused in a more complex environment. Our network model shown in Figure 2, is conceptually similar to what is used in [5] for theoretical analysis of delayed feedback in congestion control. The bottleneck link between switches $S1$ and $S2$ is shared by all connections.

There are unspecified number of traffic sources whose aggregate arrival rate to the network is controlled by mean arrival rate parameter λ. The connections arrive with the inter-arrival time having exponential distribution[16], and generate file transfer-like traffic. To get good confidence intervals quickly, we made all connections transport the same sized file. The file size F in different simulations varies from $10KB$ to $100KB$ to $1MB$. This follows the observation[4, 15] on data traffic size, although we will have to use TCP Library to generate more realistic file size distribution. We limit our traffic characteristics to delay-insensitive, file transfer traffic similar to `ftp`, because interactive traffic carries so little data that more or less it is unaffected by flow control[4]. The packet size used is $500B$, which roughly corresponds to the default TCP MTU size $536B$ without path MTU discovery. The switches have at least one (link bandwidth $\times$ link RTT) worth of buffer space, and all connections share the buffer.

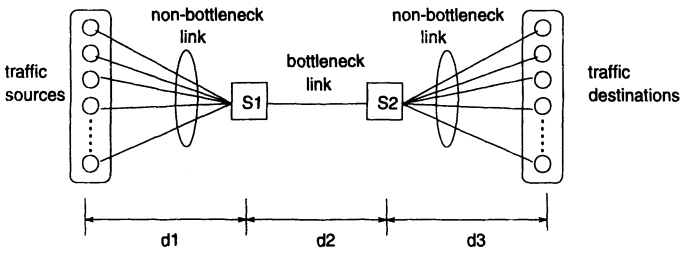

Figure 2: Configuration of the simulated network.

To see the effect of bandwidth-delay product on the performance of congestion control schemes, we vary two parameters, *network bandwidth* BW_N and *propagation delay* D. Specifically, we use $30ms$, $3ms$ and $1.8ms$ as D values. The first and the last roughly correspond to the inter-nodal distances of NSFNET and AURORA testbed[3], respectively. For BW_N, we use $150Mbps$ and $1Gbps$, with the bottleneck bandwidth BW_b ranging from $\frac{1}{5}$ to $\frac{3}{5}$ to full BW_N. Varying BW_N and BW_b, we intend to observe the scalability of the experimented schemes with increasing bandwidth.

We measured the (maximum) network powers[7] of TCP and Blitz. Given the same raw bandwidth, the network power characterizes the efficiency of congestion control algorithms in exploiting the bandwidth. The reason we used network power as the primary performance measure is twofold. First, response time is a measure that easily quantifies performance from the user's viewpoint. But from the viewpoint of the entire network, system throughput is more important. Because throughput and response time are really redundant metrics, they can be combined into a single measure known as the *network power*[6], defined as: $P = \frac{\Theta}{R}$. Since the two measures Θ and R compete with each other, P has a single maximum[9]. A higher power means either a higher throughput or a lower delay in a given system; in either case it is considered better than a lower power. We can uniquely determine the power characteristics for a given system as shown in Figure 3 by only varying the load(X axis). So we can compare the performance of two different systems, one using TCP and the other using Blitz, in terms of the power characteristics by giving the same system configuration parameter values and varying the load. Second, for today's applications, the boundary between interactive traffic and bulk data traffic is not as clear as in the past. New bulk transfer applications such as WWW and gopher constitutes large percentage of today's wide-area network traffic[14], but these services are definitely interactive, recently being classified as *interactive bulk* class[17]. Therefore, it is important to measure both throughput and response time, which are combined into the network power.

2.2 Simulation results

Reference Network: (150Mbps, 30ms, 100KB)

Our reference network configuration is a $150Mbps$ network with $D = 30ms$, where each connection has $100KB$ of data to send. Figure 4(left) shows network power of TCP and Blitz with different bottleneck sizes. In the graph, the initial alphabet represents the congestion

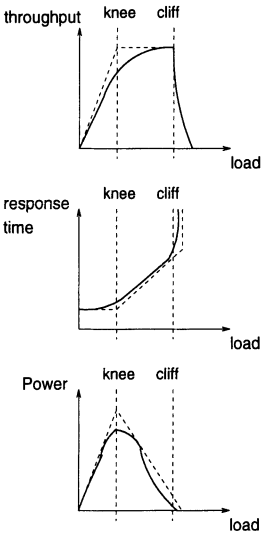

Figure 3: Network performance as a function of the load(from [JR88]).

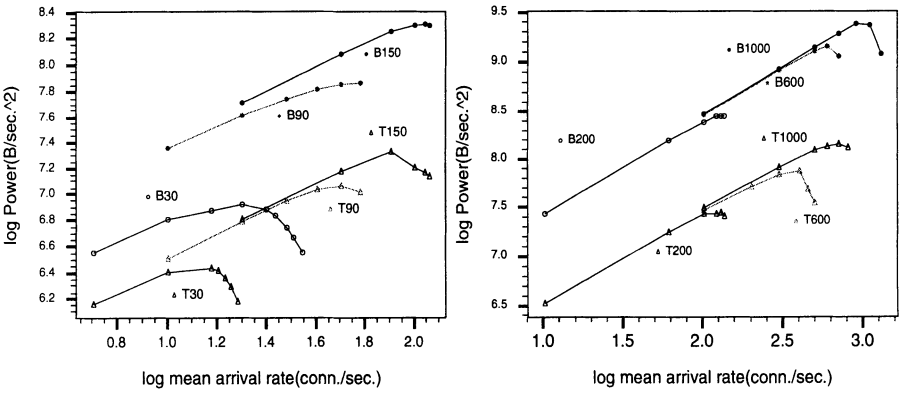

Figure 4: Network powers for various bottleneck sizes in the Reference Network(left) and in the Faster Network(right).

control scheme used("T"=TCP, "B"=Blitz), and the number after the initial denotes the bottleneck size in $Mbps$. The X axis represents λ in log_{10} scale, and the Y axis is P also in log_{10} scale. In Figure 4, the powers of TCP and Blitz both increase with the increase of physical bottleneck bandwidth. But for all BW_b, the power of Blitz is much higher, by a factor of ten at the maximum, when $BW_b = 150Mbps$. Even Blitz power with $BW_b = 90Mbps$ excels that of TCP with $BW_b = 150Mbps$, meaning Blitz performance with $90Mbps$ bottleneck exceeds TCP performance with no bottleneck. If we see the maximum powers, Blitz increasingly outperforms TCP by the factor of 3, 6, and 10(0.5, 0.8 and 1 in *log* scale) as BW_b increases. In the reference network with varying BW_b, Blitz's performance and scalability are better than TCP's. (In [8] we give a detailed analysis of the result, including the bandwidth usage, bottleneck utilization, and fairness, where in all three categories Blitz shows satisfactory performance.)

Faster Network: (1Gbps, 30ms, 100KB)

Now we increase BW_N to examine the effect of larger network bandwidth on performance and scalability of the two schemes. Note that BW_b is automatically scaled with BW_N so that it ranges from $\frac{1}{5}$ to $\frac{3}{5}$ to 1 times of BW_N. Figure 4(right) shows the power difference increases with larger BW_b. But notice that the gap in power increases with the same $\frac{BW_b}{BW_N}$ (*i.e.*, $\Delta P(Blitz) - \Delta P(TCP)$) has increased for all $\frac{BW_b}{BW_N}$. For instance, at $BW_b = 200Mbps$, Blitz maximum power has increased by 1.5 in *log* scale, but it is 1.1 in TCP. Similarly, it is 1.3 versus 0.8 and 1.1 versus 0.7 at $BW_b = 600Mbps$ and $BW_b = 1Gbps$, respectively. This means the maximum power difference between Blitz and TCP has increased by another 2.5 to 3.5 times in network power with the bandwidth increase. So increase in network bandwidth brought larger performance improvement in Blitz than in TCP. This is a strong indication that in larger bandwidth-delay product networks, more aggressive approach is needed than conventional adaptive control schemes.

Smaller Network: (150Mbps, 1.8ms, 100KB)

Now we turn our attention to a smaller network, to examine if our idea results in favorable performance even in smaller bandwidth-delay product networks. But one should note that our main focus is on larger bandwidth-delay product networks, into which networks will eventually evolve. The experiments in smaller bandwidth-delay product networks are for completeness. One way to reduce the bandwidth-delay product is to scale D down. Here we set $D = 1.8ms$, and this amounts to about 170 *miles* in one-way distance and the bandwidth-delay product of about $35KB$. This network roughly corresponds to AURORA network in terms of D. In Figure 5(left), we notice two major changes. First, the network power is overall larger than when D was larger. This is because the response time and throughput both improve with smaller propagation latency. Second, the gap between BLT power and TCP power narrowed. This shows that as bandwidth-delay product decreases, the relative performance gain of BLT gets smaller. With smaller D, the effect of propagation latency on adaptive window size change shrinks in TCP, but since there had been no penalty due to adaptive window change in BLT, the performance gap narrowed down. Still, BLT's power is safely over TCP's, with increasing gap as BW_b increases.

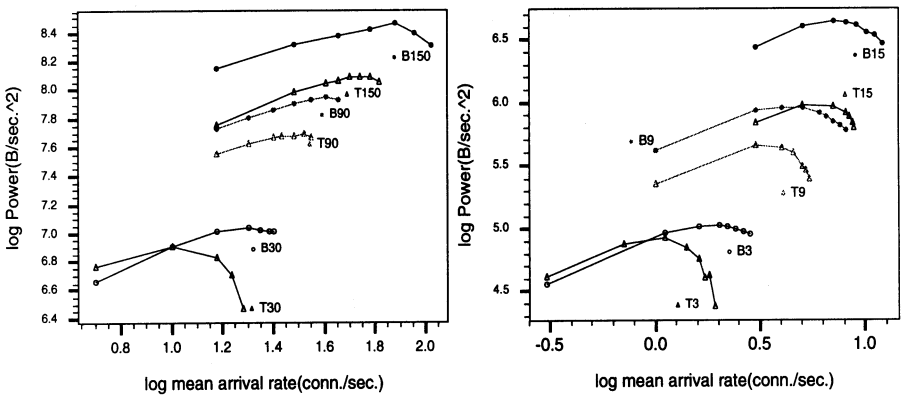

Figure 5: Network powers in an AURORA-size network(left) and Slower network(right), with $BW_N = 150Mbps$, $F = 100KB$.

Slower Network: (15Mbps, 30ms, 100KB)

Another way to reduce bandwidth-delay product is to decrease BW_N. We set $BW_N = 15Mbps$ and $D = 30ms$, where the network size roughly corresponds to NSFNET. Figure 5(right) shows that the power is more than 1 order of magnitude smaller compared with the Reference Network case. This is largely explained by the fact that the raw bandwidth has been decimated. We also note that when $BW_b = 3Mbps$, TCP power is slightly better with low arrival rate. But this soon vanishes with increased arrival rate and/or larger BW_b. In all cases, the maximum power with BLT is greater than with TCP.

Scaling data size: (150Mbps, 30ms, 1MB) and (150Mbps, 30ms, 10KB)

So far we have seen the performance with $F = 100KB$ only. Here, we vary F. We expect as file size grows, the performance difference between Blitz and TCP will narrow down because the relative overhead of adaptive window sizing such as Slow Start will be smaller in long transmissions. Figure 6(left) shows the power characteristics of TCP and Blitz with $F = 1MB$. Overall power decreases compared with Reference Network. In case of Blitz, the maximum powers in all bottleneck sizes fell at least one order of magnitude. In TCP, the decrease is milder, about half order of magnitude. The nominal cause of the overall drop is the increase of response time R. With F ten times larger than in Reference Network, we can expect that the response time will roughly increase tenfold. This amounts to one order decrease in power, and that is precisely what happened to Blitz. The fact that the increase of F has direct impact on Blitz response time(and power) means there are little overhead factors in Blitz that would "buffer" the drop. However, for TCP, as the overhead of adaptive window size maneuvering decreases, the drop was milder. It again confirms how

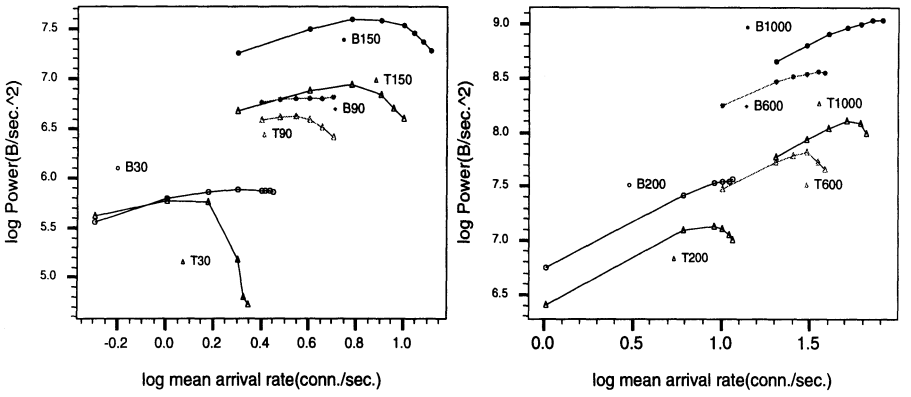

Figure 6: Network powers in transfer of 1MB file in Reference Network(left) and Faster Network(right).

more streamlined Blitz is than TCP.

The usefulness of Blitz, represented by the power difference between Blitz and TCP, decreases when file size increases, especially with small λ that lowers the chance of congestion. But when BW_b increases, the bandwidth-delay product increases, and TCP cannot outperform Blitz even with small λ. Also, with the increase of λ, even when the bandwidth-delay product small, TCP rapidly deteriorates. Also, when we increase BW_N to $1Gbps$, we see that the power difference between Blitz and TCP widens again(Figure 6, right). This is the strong point of Blitz: it scales better with bandwidth.

The disadvantage of conventional schemes like TCP is amplified in small file transfers. Almost all data transfer protocols adopt some form of adaptive rate change algorithm, and this adaptation overhead is critical to small files because for them the time spent in adjustment could be very long. Here, we decreased F to $10KB$ in the reference network. In Figure 7, we also notice that Blitz is more scalable. The scalability here is not about bandwidth, but rather about data size. In other words, as data size decreases, the performance gap between Blitz and TCP widens.

3 CONCLUSION

We observe that the existing rate adaptation algorithms in TCP is too conservative to exploit multi-megabit/gigabit bandwidth. An experimental method was used to show that more aggressive approach can yield much higher performance for large bandwidth-delay product networks. But we could observe by varying bandwidth-delay product and data size that the

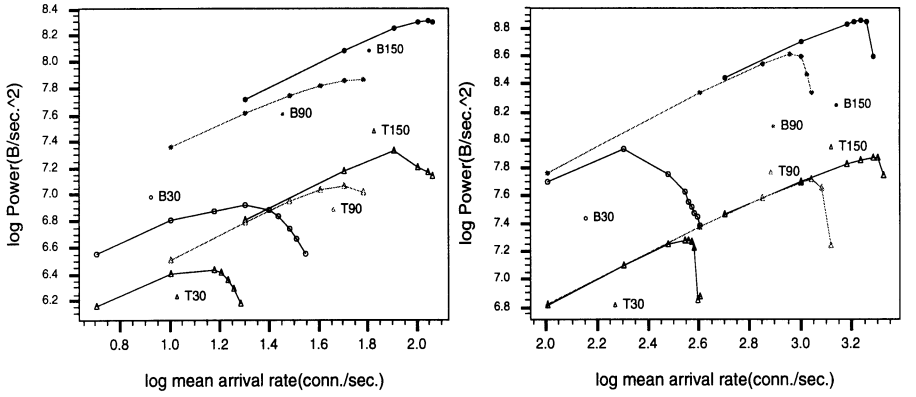

Figure 7: Network powers in transfer of 100KB file(left) and 10KB file(right) in Reference Network.

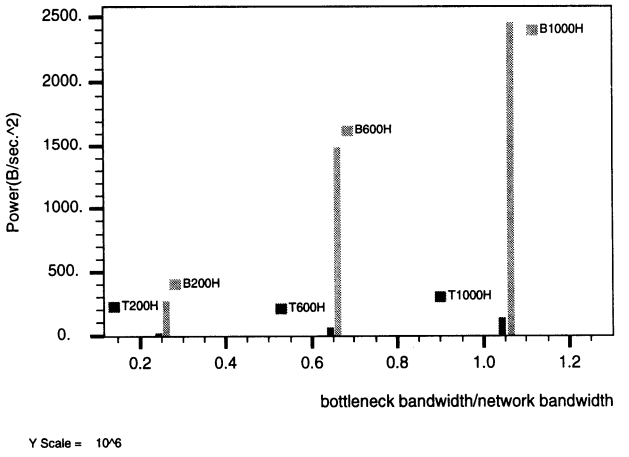

Figure 8: Maximum powers in *normal* scale, $F = 100KB$, $BW_N = 1Gbps$.

performance improvement for the aggressive approach is not just a function of bandwidth-delay product, but also of the data size being transported. In fact, the result suggests that the aggressiveness should be a variable depending on the ratio between bandwidth-delay product and transported data size. Since it is almost impossible and hardly desirable to employ the priority discarding at every multiplexing point within the network architecture all at once, we need a more evolutionary approach than Blitz. So it might be desirable to retain the sophisticated adaptive algorithms to change source rates, but to make the algorithms more sensitive to the ratio between bandwidth-delay product and transported data size. (On the other hand in existing algorithms, the rate changing algorithms are constant functions of the ratio.) But by making the rate changing algorithms like Slow Start and Congestion Avoidance in TCP possibly more aggressive, the network stability can be threatened. It has been proved that additive increase and multiplicative decrease is a sufficient condition for network stability, but it has not been shown if it is a necessary condition. We conjecture there are a set of more aggressive algorithms than linear-increase/exponential-decrease that do not hurt network stability, and this issue is currently under investigation.

Rate-based congestion control for ATM Available Bit Rate(ABR) services also uses similar rate adaptation techniques as TCP algorithms or DECbit. Unfortunately, the ongoing effort to set the standard for ABR traffic control also lacks the concern on bandwidth-delay product and transported data size. However, since the details on how to set parameters like Initial Cell Rate(ICR), Multiplicative Decrease Rate(MDR), Additive Increase Rate(AIR), *etc.* are up to the traffic source-destination pair[1], we believe this is where the ratio between bandwidth-delay product and transported data size can be considered, to maximally exploit network bandwidth without hurting the network stability.

References

[1] F. Bonomi and K.W. Fendick. The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service. *IEEE Network Magazine*, 9(2):25–39, March/April 1995.

[2] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proceedings of ACM SIGCOMM*, August 1989.

[3] D.D. Clark *et al.* The aurora gigabit testbed. *Computer Networks and ISDN Systems*, 25(6), January 1993.

[4] R. Caceres *et al.* Characteristics of Wide-Area TCP/IP Conversations. In *Proceedings of ACM SIGCOMM*, pages 101–112, 1991.

[5] K. Fendick, M. Rodrigues, and A. Weiss. Analysis of a Rate-Based Control Strategy with Delayed Feedback. In *Proceedings of ACM SIGCOMM*, pages 136–148, 1992.

[6] A. Giessler, J. Hanle, A Konig, and E. Pade. Free Buffer Allocation – An Investigation by Simulation. *Computer Networks*, 1(3):191–204, July 1978.

[7] R. Jain and K.K. Ramakrishnan. Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology. In *Proceedings of the Computer Networking Symposium*, pages 134–143, 1988.

[8] H. Kim and D.J. Farber. Congestion Control by Bandwidth-delay Tradeoff in Very High-Speed Networks: The Case of Window-based Control. Technical report, University of Pennsylvania, November 1994. MS-CIS-94-55/DSL-80.

[9] L. Kleinrock. On Flow Control in Computer Networks. In *Proceedings of ICC*, volume 2, pages 27.2.1–27.2.5, 1978.

[10] L.Zhang. Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proceedings of ACM SIGCOMM*, August 1990.

[11] P. Mishra and H. Kanakia. A Hop by Hop Rate-based Congestion Control Scheme. In *Proceedings of ACM SIGCOMM*, pages 112–123, 1992.

[12] A. Mukherjee, L. Landweber, and T. Faber. Dynamic Time Windows and Generalized Virtual Clock:Combined Closed-Loop/Open-Loop Congestion Control. In *Proceedings of IEEE INFOCOM*, 1992.

[13] P. Newman. Traffic Management for ATM Local Area Networks. *IEEE Communications Magazine*, pages 44–50, August 1994.

[14] V. Paxson. Growth Trends in Wide-Area TCP Connections. *IEEE Network Magazine*, pages 8–17, July/August 1994.

[15] V. Paxson. Private communication, November 1994.

[16] V. Paxson and S. Floyd. Wide-Area Traffic:The Failure of Poisson Modeling. In *Proceedings of ACM SIGCOMM*, August 1994.

[17] S. Shenker, D.D. Clark, and L.Zhang. A Service Model for an Integrated Services Internet, October 1993. draft-shenker-realtime-model-00.txt.

[18] L. Tassiulas, Y. Hung, and S.S. Panwar. Optimal buffer control during congestion in an ATM network node. In *Proceedings of IEEE GLOBECOM*, 1993.

[19] N. Yin and M. Hluchyj. On closed-loop rate control for atm cell relay networks. In *Proceedings of IEEE INFOCOM*, 1994.