CONGESTION CONTROL BY ADAPTIVE ADMISSION

Zygmunt Haas and Jack H. Winters
AT&T Bell Laboratories
Holmdel, NJ 07733

Abstract

In this paper, we discuss some issues in congestion control and describe a novel congestion control scheme for high-speed networks. The scheme is based on periodic transmission of sample time-stamped packets through the network. Upon reception, the packet delays are calculated, averaged, and used to determine the state of the network. The information on the state of the network is then used to drive the network admission control. The major advantage of the proposed scheme over conventional congestion control techniques is that it copes with traffic surges that are shorter than the network round-trip delay. This is achieved by controlling traffic admission with continuous estimate of the network state. The scheme is targeted towards networks that carry aggregated traffic, and can be applied to ATM-based networks.

1 What is Network Congestion?

There seems to be some confusion in the technical literature of what network congestion and congestion control are. According to our definition, network congestion is: A state of a network, in which some network resource is oversubscribed/overdemanded, **and in which the availability of the resource decreases because of the oversubscription/overdemand.** In other words, network congestion results in a real loss of the overutilized resource. In most of the cases, the resource utilization measure is assumed to be the *goodput*[1].

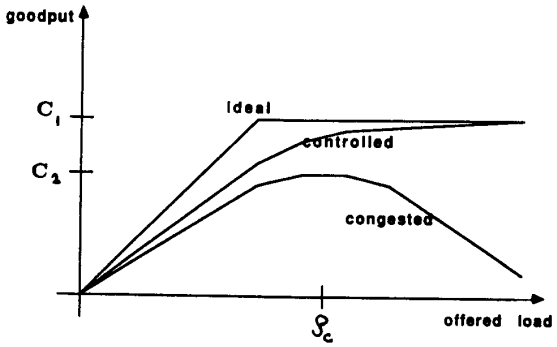

Figure 1: What is congestion?

Network congestion is nearly always an *end-to-end* issue. In other words, we distinguish between congestion and *contention*. Contention is a state in which some network resource is oversubscribed/overdemanded. However, in contrast to congestion, contention does not necessarily and **directly** result in loss of the resource. Thus contention within a switch is not (necessarily) network congestion. Contention can, however, lead to congestion, especially if it lasts long enough.

Figure 1 demonstrates the phenomenon of loss of resources due to congestion (the "congested" curve) and the ideal behavior of a network (the "ideal curve"). In practice, one is satisfied with relatively minor loss of the resource due to congestion[2], as shown by the "controlled" curve. The idea behind congestion control is to get as close as possible to

[1] The *goodput* is the rate of useful data delivered by the network, whereas *throughput* is the total rate of data delivered by the network and includes duplicated, erroneous, and misdelivered packets.

[2] In fact, some loss of the resources is inevitable in networks relying on detection of loss of resource to control congestion; i.e., the *reactive* flow-control methods.

0560 CH2979-3/91/0000-0560 $1.00 © 1991 IEEE

the ideal curve **given the knowledge or ignorance of the traffic pattern that the network is expected to experience.** This is rather a crucial observation. If the traffic pattern is a very predictable one, the congestion control problem is simple to solve. Unfortunately, in most cases the traffic is either completely unknown or insufficiently specified. Thus the congestion control scheme needs to be robust enough to perform relatively well in "nearly all" cases. This is why some researchers believe that a single scheme may not be adequate to solve the congestion control problem and that a combination of methods is needed, each emphasizing different aspect of congestion control.

Probably the most familiar example of congestion is the *positive feedback retransmission* phenomenon. When, for some reason, the network starts excessively delaying packets (maybe because of contention), the end-to-end protocols time-out and ask for data retransmission. This creates multiple copies of the same packet within the network, and since the capacity of the network is finite, the actual amount of useful data delivered to the destination (the goodput) decreases. This decrease in goodput results in more and more retransmission requests, creating more and more replicates in the network and more and more congestion[3]. If the retransmission facilities were "smart" enough to detect that the time-outs and retransmission requests are not due to packet loss (or erroneous packets) but due to congestion, the positive feedback could be eliminated. Unfortunately, mainly due to the layering of communication protocols, the end-to-end retransmission facilities (transport protocol) cannot receive the information from lower layers that is required to decide whether a time-out is caused by actual loss of a packet or by congestion. (See also [1].)

There is also confusion between flow- and congestion-control. Flow-control can be exercised at any layer; for example, at the data-link layer to avoid adjacent buffer overflow, at the network layer to avoid buffer overflow in network interfaces, at the transport layer to avoid overflow of transport buffers, etc. Congestion control is usually resolved in the network and/or transport layer, and can be handled by flow-control methods. In particular, fully implemented flow-control at the data-link layer can eliminate congestion. The question is whether such flow-control is practical in high-speed networking due to large buffering requirements. Specifically, modifications of flow-control schemes for high-speed networks usually result in schemes that are not fully congestion-proof. (For example, virtual circuits with buffer sharing may not be an adequate solution for network congestion).

2 How can congestion be controlled?

The proliferation of congestion control schemes comes in the era of much research on very high speed, hundreds-of-Mbps networks. This correlation is not a mere coincidence. It is believed that, as in the case of flow-control, congestion-control in high-speed networks is much more difficult than in low-speed networks due to lower coupling between the transmitting and the receiving ends. In other words, it is more difficult to control a system with long propagation delay[4].

Refer back to Figure 1. The curve "controlled" in this Figure is the basis for a large family of congestion-control scheme; the *reactive* methods. The reactive methods monitor the network performance (either directly, by monitoring the state of the resource, or indirectly, by monitoring some other correlated parameter), and upon detection of developing congestion, take some action that drives the network out of this state. Reactive methods rely heavily on the feedback from the network; thus these methods can successfully cope with congestion in the network only when the congestion develops with a time constant on the order of at least the round trip delay. Among

[3]The curve in Figure 1 is a static one. Due to the positive feedback phenomenon, the offered traffic increases rapidly, leading to uncontrolled rapid decrease in goodput.

[4]Propagation delay is measured in units of packet transmission time. Thus, even though the propagation delay in seconds remains constant, the propagation delay in packet transmission time increases linearly with transmission rate for fixed packet length.

the reactive methods are dynamic windowing ([2,3]) and schemes that drop excessive (possibly prioritized) traffic ([4]).

Pro-active methods, as opposed to the reactive schemes, constantly activate some mechanism that reduces the possibility of the network getting into the congested state[5]. Some commonly known pro-active methods are bandwidth reservation, traffic shaping/smoothing ([5]), and admission control (leaky bucket [6], for example).

The main disadvantage of reactive methods is the relatively slow reaction time to building congestion, since these methods rely on some indication from the network of increasing congestion. This indication is usually provided only after the actual exchange of traffic takes place. Pro-active methods, in general, do not suffer from this disadvantage, since these schemes are continuously applied. However, most pro-active methods that rely on an open-loop control scheme are not robust enough and cannot, therefore, guard the network against all possible traffic patterns.

In the next section we present a novel approach to congestion control. This scheme relies on continuous feedback from the network to drive the network admission control. Thus the scheme combines the advantage of closed-loop control (usually used in reactive methods) with the continuous activation feature (usually present in pro-active methods).

3 Adaptive Admission for Congestion Control

First, let us introduce our model of the network, which is shown in Figure 2. We concentrate on high-speed communication networks with a large number of users. Each user may have access to some small amount of the total link capacity. In other words, we anticipate that the link capacity far exceeds the sourcing/sinking capability of an individual user. Consequently, a single user is not able to **significantly** change the (congestion) status of

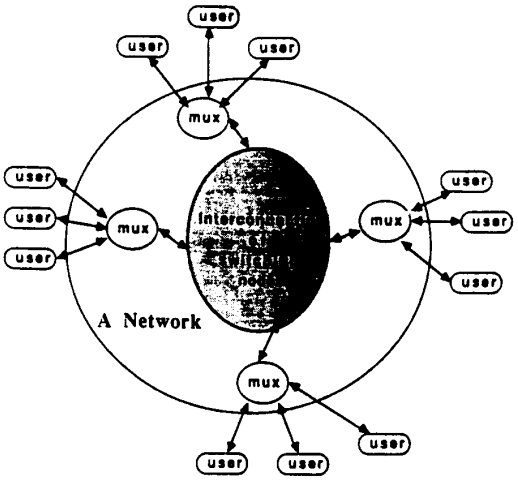

Figure 2: The network model

the network. Also, because of the large number of users, the resulting aggregated link traffic is rather **slowly** varying[6]. Furthermore, we assume that the users are connected to the network by a network interface. What we actually mean by "network interface" includes a variety of devices: multiplexers, routers, or gateways. Moreover, we assume that the congestion control mechanism is implemented in network interfaces that are considered part of "the network." Consequently, a user has no access to the congestion control scheme and cannot gain advantage by disobeying the scheme's rules.

Our model assumes some connection-oriented services. However, the scheme can be modified to accommodate connectionless environment.

The basic idea behind the proposed scheme is to maintain a single parameter per each connection/path within the network that estimates the level of congestion of this connection/path. The parameter is continuously updated by periodic sampling packets sent between the source and the destination[7]. The scheme uses the value of this parameter to adjust the transmission rate from the source to

[5]One can draw an analogy from the medical field: reactive methods compared to pro-active schemes are like curative vs. preventive medicine.

[6]Especially for links "deep" in the network.

[7]Periodic exchange of states was proposed in [7] to improve the performance of transport protocols. Here we use the idea of periodic sampling packets, in which the packets themselves sample the network to provide estimation of the network status. Both the ideas can, however, be combined, so that only one kind of periodic exchange is implemented.

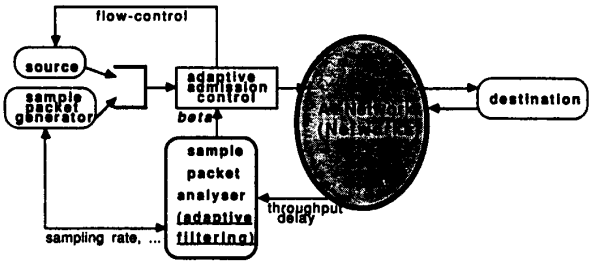

Figure 3: The adaptive admission control scheme

the destination, or to encourage/defer initiation of new data transfers on the path.

In our scheme, shown in Figure 3, the transmitter and the receiver ends maintain constant synchronization for the purpose of congestion control. This synchronization is achieved by the periodic exchange of sample packets. These sample packets carry time stamps that indicate the time a packet was inserted into the network. The sample packets are inserted into the same queues and follow the same path as regular packets. Upon arrival of sample packets at their destinations, the packets' delays are calculated by extracting the time stamp from the packet format[8]. The destination averages the packet delay and forwards this information to the source on the reverse-directed traffic. This information, along with the average sample packets arrival rate, is used to estimate the level of network congestion[9]. (An example of an algorithm is presented below.) The estimate of the network congestion is expressed by a single parameter, β, with $\beta \approx 1$ indicating low congestion and $\beta \approx 0$ high congestion. This congestion estimation can be used in several ways. In the basic arrangement, β is the dynamic token allocation parameter in the *leaky bucket*-like[10] scheme ([6]).

The proposed scheme has several advantages, as opposed to the conventional congestion control schemes. First, because of the periodic exchange of sample packets, the estimation of congestion is performed continuously and not just at the time of an actual transmission[11]. This feature eliminates the problem that occurs in transmission of data that is shorter than the round trip delay[12] and with initialization of new data flows[13] (referred to as the "cold start" phenomenon). Second, the periodic exchange of sample packets decouples the congestion control mechanism from the actual data transfer, reducing the effect of positive feedback on the congestion control mechanism. This is accomplished by the fact that when congestion occurs, the sample packets are also subject to excessive delay detectable at the destination. This delay serves as a direct indication of congestion. In contrast, in most feedback-based schemes, when congestion occurs the information about the congestion is also delayed, increasing the congestion even more[14]. (This decoupling can be further increased by raising the priority for the sampling packets in the event of congestion.) Third, by using the delay of the path within the network as direct indication of congestion, the actual congestion event is detected and treated. Some schemes use other indirect indications, leading to improper or delayed congestion detection. Fourth, the admission control reduces the dependency of the scheme's performance from the actual traffic arrival process. Fifth, the proposed scheme can be combined with other protocols that rely on periodic exchange of information ([7]), thus reducing the bandwidth and processing over-

[8] In practice, changes in the network delay are computed. This eliminates the need for clock synchronization.

[9] The average sample packet arrival is used to model the network by another adaptive technique, as described later.

[10] The leaky bucket technique is an implementation of admission control, in which some number of tokens is allocated during each update interval. Each transmitted packet removes a token from the pool. When no more tokens are available, no more packets are inserted into the network. In our work, we employ the idea of the leaky bucket. However, the number of the allocated tokens is dynamically adjusted based on the congestion status of the network.

[11] Some schemes, like adaptive windowing, for example, stop operating when the channel idles.

[12] Most of the congestion control methods perform well in an environment where the changes of congestion are with time constant on the order of round trip delay. When, however, a short duration traffic is presented to the network, the congestion control schemes have no means to control the source when the congestion is detected after the transmission is actually completed.

[13] Virtual circuits, for example.

[14] The reason for the difference is the means by which the congestion information is acquired: sampling packets, in our case, or acknowledgements/negative acknowledgements, in other schemes.

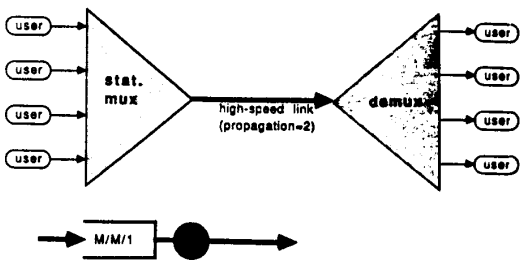

Figure 4: An example of a single M/M/1 queue network

head associated with the periodic exchange of samples. Moreover, the rate at which the sample packets are generated should be such that the total overhead of the scheme is minor[15]. Finally, the scheme is easy to implement and can be integrated with other congestion control means, leading to overall superior performance.

4 The control algorithm

The purpose of the averaging algorithm is to determine the value of congestion parameter β from the arriving samples. The algorithm described here serves as an example, and is by no means the only or the optimal way to implement such an algorithm. The idea behind our algorithm is to use a distributed scheme. Each one of the network interfaces continuously sends sample packets to all its active destinations. Each sample packet contains a time stamp indicating the time the packet was admitted to the network[16]. At every destination, the algorithm accumulates the arriving samples during some sample interval[17], t, and calculates the average delay, D, and the average throughput, T. This is done in each network interface and per "connection." The values of D and T that are returned to the source interface[18] are used to determine the level of congestion within the network and to adjust the value of admission control parameter β. Consequently, the value of β tracks the congestion. One possible approach is to use β as a binary variable. Thus for $\beta = 1$ the network is in normal operation, while $\beta = 0$ indicates a congested network. Let us demonstrate this for the case of a single M/M/1 queue illustrated in Figure 4. The delay D, the throughput T, and the power P, of this simple network are[19]:

$$D = \frac{1}{1-\rho},\ T = \rho,\ P \triangleq \frac{T}{D} = \rho\cdot(1-\rho) \quad (1)$$

and are displayed in Figure 5. (ρ represents the network load.)

For small ρ, as the total traffic into the network increases, the power increases due to the increase in throughput. However, beyond some point, the increase in delay is so significant that the power begins to drop. We define the congestion/no congestion threshold as the delay corresponding to operation at maximum power. Since $P_{max} = P(\rho = 0.5)$, then $D_{threshold} = 2.0$. [20] [21] Thus, as shown in Figure 6, $\beta = 1$ for $D < 2.0$ and $\beta = 0$ for $D \geq 2.0$. β is then used to either accept or inhibit traffic to the network. In a general case, the value of β is used to determine the amount of blockage the traffic will experience at the network interface. In the case of the leaky bucket as admission control, the value of β determines the amount of credits per update interval.

The choice of $\beta(D,T)$ in the general case remains an open question at this time. Obviously, the choice depends on the network topology. Possibly, the approach of neural networks[22] can be used to determine and im-

[15] The scheme does, however, use bandwidth, which is expected to be an excessive resource in future networks, to solve a network control problem.

[16] Actually, the instance the packet was admitted to the network interface queue.

[17] The choice of t is still an open question; it should be large enough to reduce noise, and small in comparison with the path propagation delay, so that β can follow the congestion state changes. A good strategy is to allow t to be dynamically adjusted.

[18] Possibly piggy-backed on sample packets sent in the reverse direction.

[19] We assumed an average packet transmission time of one unit and a propagation delay of 2 units.

[20] This is a somewhat arbitrary definition of $D_{threshold}$. The rationale is to stabilize the network operation at its maximum power.

[21] Delay is the waiting (queueing) and transmission (service) delay, and does not include propagation delay.

[22] Use of neural networks to solve the congestion control problem was already proposed in the literature ([8], for example). Our work, however, uses the neural

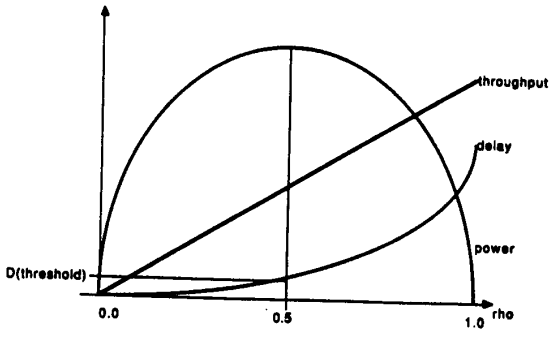

Figure 5: Delay, throughput, and power for the single M/M/1 queue network

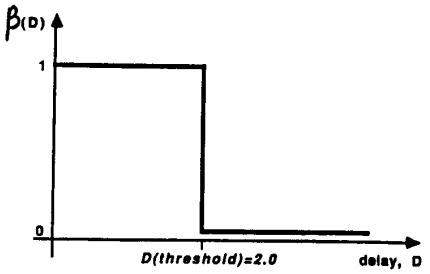

Figure 6: Binary β function

plement the β function, as shown in Figure 7 (ρ_0 in the Figure is defined in Appendix). Likewise, the mechanism of estimation/prediction of the network structure from the D and T parameters needs to be further researched. Also in this case, an adaptive nonlinear approach such as neural networks may ultimately be found to be an excellent choice.

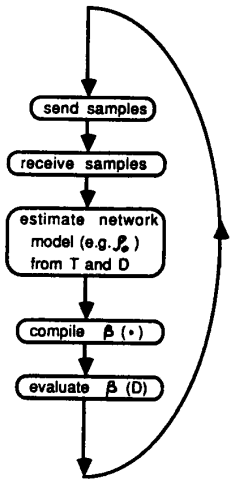

Figure 7: Full adaptive admission congestion control algorithm

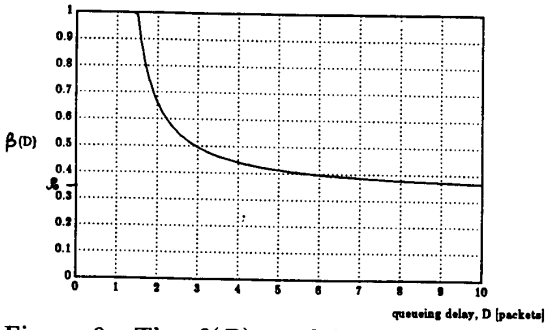

Figure 8: The $\beta(D)$ used in the simulation example

5 Some preliminary simulation results

We have simulated the above control algorithm on the network in Figure 4 with a more sophisticated choice of the β function. The model of the offered traffic is a fixed packet size[23], Poisson arrival source with arrival rate λ. The arrival rate is slowly changing; i.e., every 200 packet units λ is either increased by 5% with probability p, or decreased by 5% with probability[24] $1-p$. The arrival rate represents the total arrival traffic including blocked traffic and retransmissions. The samples are sent every tenth packet[25]. The admission control is based on the leaky bucket principle, allowing $\beta \cdot t$ packets into the network during each t update interval. The $\beta(D)$ function that was used is shown in Figure 8, and is derived in the Appendix. The results of the simulation are shown in Figures 9, 10, and 11.

Each one of the simulation traces includes three graphs, the input offered traffic, the sampling queueing delay, and the throughput. The

network approach in a different way. In our scheme, the neural network models the communication network and drives the per-packet admission control at the network entrances.

[23] Packet transmission time was fixed to one unit.

[24] The parameter p is used to create different arrival patterns. In the results presented here, $p \approx 0.5$.

[25] Thus the total arrival rate is increased by 10% at most. In general, the increase is much smaller, since some of the sampling packets are piggy-backed on regular packets.

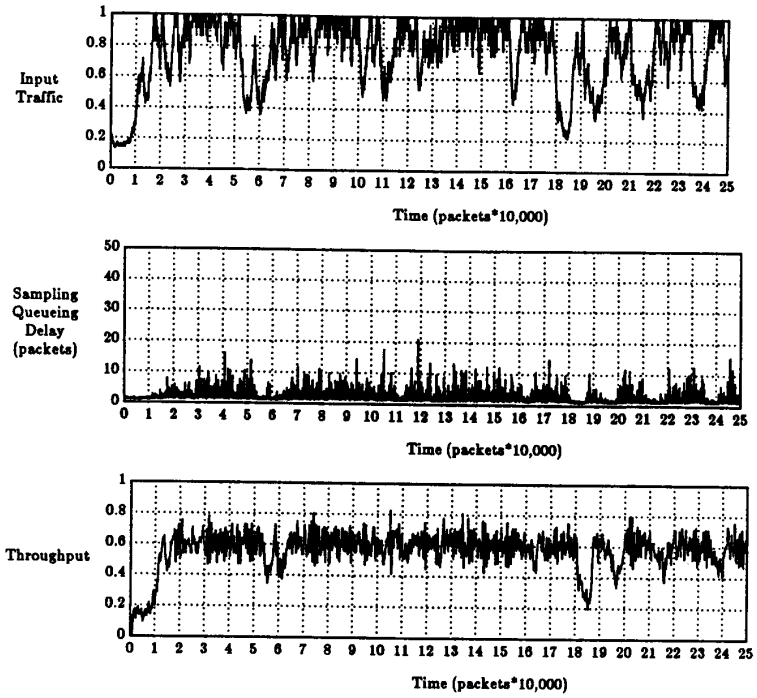

Figure 9: Simulation trace: $t = 50$ units.

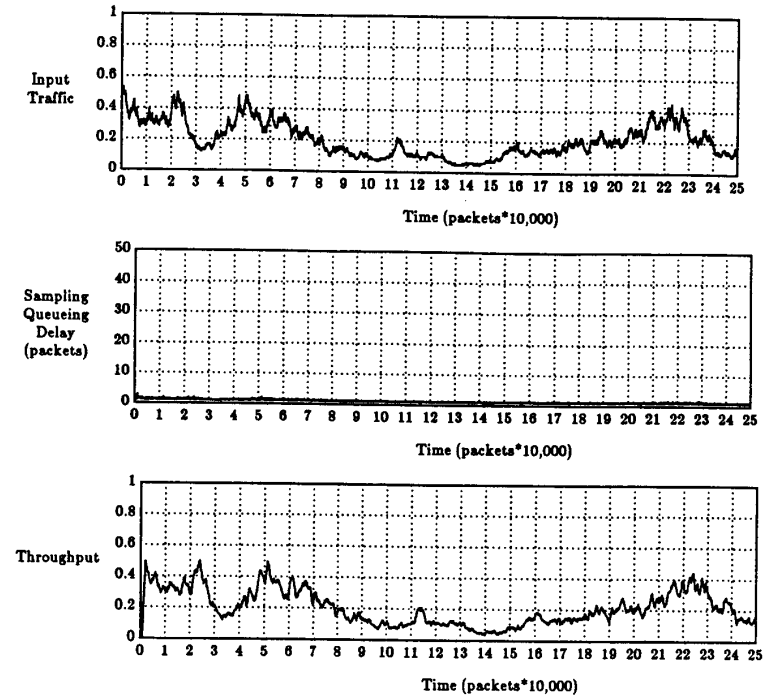

Figure 10: Simulation trace: $t = 100$ units.

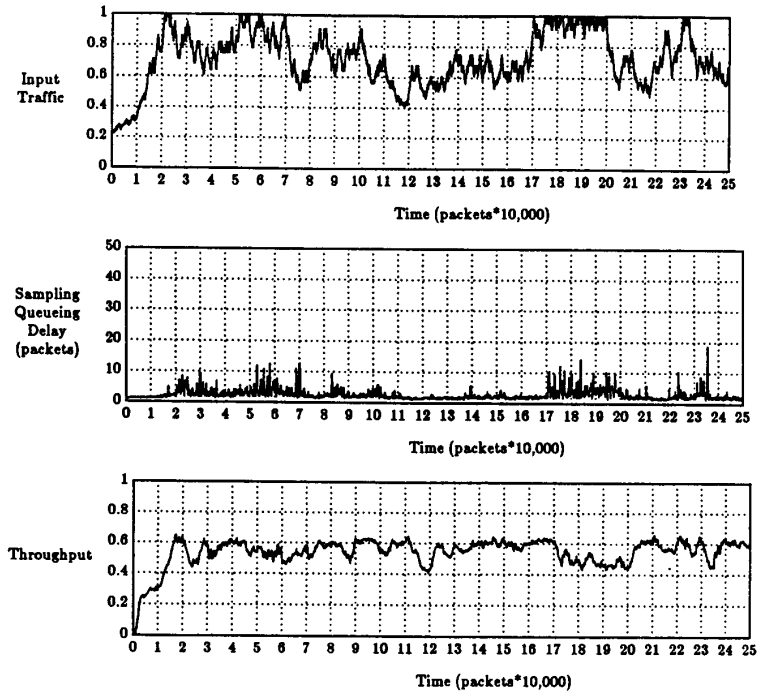

Figure 11: Simulation trace: $t = 200$ units.

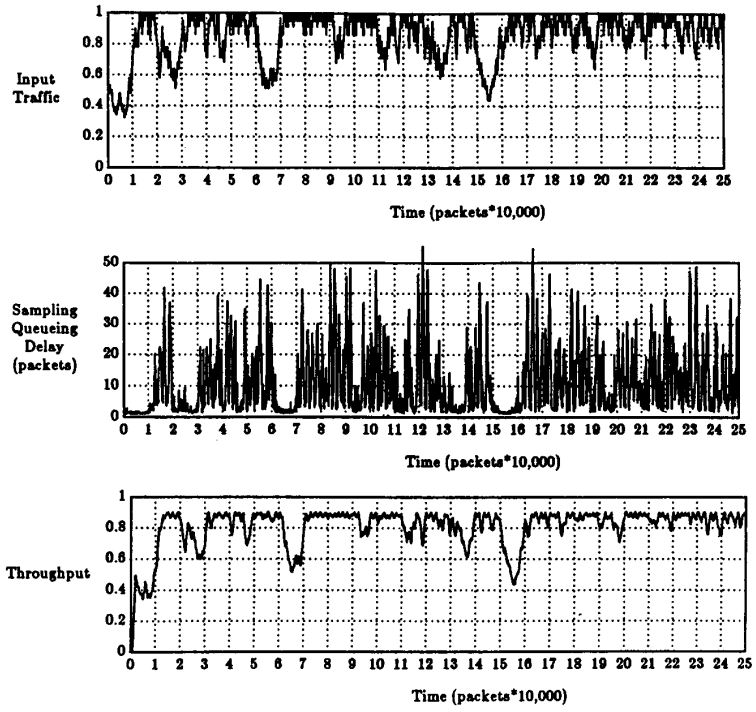

Figure 12: Simulation trace: $t = 100$ units, $\rho_0 = 0.81$.

input offered traffic is generated by a Poisson arrival process with continuously changing λ, as explained before. The sampling queueing delay is the delay that the sampling packets experience at a specific time, as a response to the network state. The throughput is the controlled rate at which packets are received (or admitted, since the network does not drop packets) into the network. The throughput is calculated by averaging over 10 last update intervals. The propagation delay was kept constant, and is equal to 400 units. The update interval, t, was given values of 50, 100, and 200 units. Since in all cases the samples rate is constant (every tenth packet), the number of samples that are averaged in each update interval are 5, 10, and 20, respectively.

The simulation results show the performance of the scheme. Under low-load condition (no congestion) the network does not admission-control the offered traffic, and the throughput approximately equals the offered traffic. As the offered traffic increases significantly above the value of $\sqrt{\rho_0}$ (see Appendix), the admission control limits the amount of offered traffic into the network, stabilizing the network performance. The scheme essentially cuts off the traffic above $\sqrt{\rho_0}$, stabilizing the throughput at this value. It can be seen that the scheme performs very well even in the case of $t = 50$ when only 5 samples are used to determine the network state. Moreover, the scheme performs quite well in this case, taking into account that the round trip delay is 8 times longer than the update interval, leading to a maximum possible increase of 40% in input traffic during one round-trip delay. The trace in Figure 12 shows the effect of ρ_0. In this trace, ρ_0=0.81. Consequently, the admission control mechanism goes into operation when the input offered traffic rate approximately equals 0.9. Moreover, 0.9 is also the average throughput for the network while in the congested state.

We intend to further evaluate the scheme, both analytically and by simulation, and to compare its performance with other congestion control techniques.

6 Summary and concluding remarks

In this work, we have presented a new approach to congestion control. The proposed scheme relies on periodic exchange of sampling packets to provide a means for continuous feedback from the network to the network interfaces that enforces admission control over the input offered traffic. Since the scheme proposed here monitors the network continuously, the phenomenon of "cold start" that some congestion control schemes experience is avoided. Moreover, because of the same reason, the scheme can successfully cope with traffic flows of duration much shorter than the round-trip delays. The scheme is distributed in its nature and incorporates the major advantages of both reactive and pro-active schemes, and can provide a framework for congestion control in future high-speed networks which are expected to be based on very large capacity links, thus reducing the influence of a single source by the aggregation process. Preliminary evaluations of the scheme were very encouraging; however, more research on the various aspects is needed to evaluate fully the performance of the proposed scheme.

7 Appendix

In the Appendix, we derive the $\beta(D)$ function that was used in the example simulation in section 5.

We start with the basic equation that on the average the modulated offered traffic rate (i.e., the admission-controlled input rate) should be equal to the network rate, ρ. Thus, if ρ_i is the offered traffic rate at the input,

$$\beta \cdot \rho_i = \rho. \tag{2}$$

We assume that the offered and the network rates are related as follows (pictured in Figure 13)[26]:

$$\rho^2 = \rho_i \cdot \rho_0, \tag{3}$$

[26] The desired function is linearly increasing (with slope 1) till $\sqrt{\rho_0}$ and constant thereafter (see the "ideal" curve in Figure 1). The function used is an approximation to the desired function.

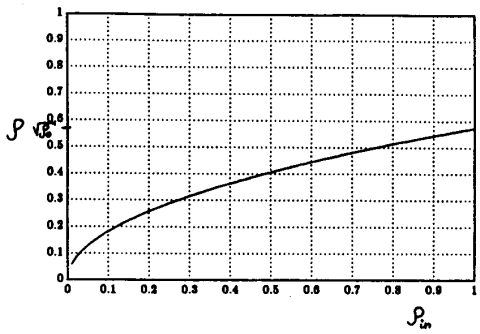

Figure 13: The suggested relation between the offered and the network traffic

where ρ_0 is a constant. The significance of ρ_0 is explained below.

Substituting Equation 3 into Equation 2 results in:

$$\beta \cdot \rho = \rho_0. \tag{4}$$

The delay D, as measured by the samples, is used to estimate the actual network rate ρ. Assuming M/M/1 network topology[27], we solve for $\rho(D)$:

$$\rho(D) = \frac{D-1}{D}. \tag{5}$$

Substituting Equation 4 into Equation 5 gives:

$$\beta(D) = \left[\frac{\rho_0 \cdot D}{D-1}\right]_0^1, \tag{6}$$

where $[x]_0^1 \triangleq \min(\max(x, 0), 1)$.

The parameter ρ_0 controls the average utilization of the network under congestion. In other words, when the network experiences congestion, the throughput tends to $\sqrt{\rho_0}$. Roughly speaking, ρ_0 is the point (value of D) at which the admission control mechanism starts its operation. According to the discussion in Section 4, a possible approach is to set ρ_0 to be equal $\rho^2_{threshold}$, where $\rho_{threshold}$ is the value that maximizes some cost function, the power for example. Furthermore, the parameter ρ_0 controls the steady-state operating point; i.e., how far the "normal" network operation is from the congestion state. The optimal value for ρ_0 depends on the current network topology and traffic pattern, and we view that it should be adaptively estimated from the values of parameters D and T, as referred to in Figure 7. In our simulation examples, $\rho_0 = 0.33$, which leads to about 0.57 utilization under congestion.

[27] The network is actually M/D/1, however we use the M/M/1 formulas because of their analytical simplicity. This is justified, since the function $\beta(D)$ is an approximation to begin with.

References

[1] Zygmunt Haas, "A Communication Architecture for High-speed Networking," *INFOCOM'90.*

[2] Raj Jain, "A Timeout-based Congestion Control Scheme for Window Flow-controlled Networks," SAC-4(7), October 1986.

[3] V. Jacobson, "Congestion Avoidance and Control," in Proceedings of *Sigcomm'88.*

[4] A. E. Eckberg, Jr., D. T. Luan, and D. M. Lucantoni, "Meeting the Challenge: Congestion and Flow Control Strategies for Broadband Information Transport," *Globecom'90.*

[5] M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks," JSAC, vol.8, no.3, April 1990.

[6] Moshe Sidi, Wen-Zu Liu, Israel Cidon, and Inder Gopal, "Congestion Control through Input Rate Regulation," *Globecom'90.*

[7] K. Sabnani, M. H. Nguyen, and C. D. Tsao, "High Speed Network Protocols," 6^{th} *IEEE Int. Workshop on Microelectronics and Photonics in Communications,* New Seabury, MA, June 6-9, 1989.

[8] Atsushi Hiramatsu, "ATM Communication Network Control by Neural Networks," *Transactions on Neural Networks,* vol.1, no.1, March 1990.