Chapter 4

Admission and Access Control in Broadband Networks

In the previous chapter we focused on models for characterizing traffic in broadband networks. As noted, this is critical in the design of such networks. It is expected that these networks, when deployed, will be carrying high-resolution images, higher-resolution video, multimedia traffic, and other types of "bandwidth-hungry" traffic, in addition to interactive data, computer traffic, and voice, that existing networks currently handle. This projected mix of traffic is much more difficult to characterize than current traffic, particularly if differing qualities of service (QoS) are required for each. It is clear that the common Poisson model for traffic may not always provide accurate design results. This point was made at the beginning of the last chapter. This fact motivated our discussion of other traffic models.

In this chapter we begin to apply some of these models to a fundamental question that arises with the design and deployment of any network: How much traffic can it handle if a prescribed quality of service for each traffic class is to be maintained while the network utilization, that is, traffic throughput (hence revenue), is to meet some minimum goal? Alternatively put, what resource requirements, in terms of link and buffer capacities, number of links, number of nodal switching points, etc, are imposed on a network design if it is to handle a specified broadband integrated traffic load? We clearly cannot hope to answer this question definitively here. What we will do in this chapter is to focus first on admission control in the context of B-ISDN: Given virtual paths set up in a network, how many virtual connections (that is, calls with specified QoS) can it handle? Given this number, does one admit a new call when a request to set one up arrives? Even here, we shall have to restrict our discussion sharply because of the complexity of the issue and the lack of specific results at this point: We shall discuss admission policies based solely on traffic occupancy at one point in the network, generally the access

node. We shall not attempt to study admission control based on bandwidth available along an entire virtual path.

Given a decision to admit a call, the traffic generated by this call must be monitored to ensure it does not start congesting the network. We have already noted in the previous chapter that traffic can only be described statistically. This has always been the case, whether in the voice-based, circuit-switched telephone networks, or the more recent data networks. As a result, congestion may develop despite a good admission policy. (Malicious misuse of the network by some users must be guarded against as well). To prevent congestion from occurring, control at the access point (the user-network interface of UNI), as well as within the network, must be exerted. In the context of B-ISDN, this is sometimes referred to as a "policing function".

ATM documents refer to this policing function as *Usage Parameter Control* (UPC). Usage Parameter Control is used to ensure that users do not violate their traffic contracts, negotiated during admission control. These contracts include various conditions of operation such as peak and average rate of cell transmission, and maximum cell burst size, among others. It is the possible violation of these contracts that can result in network congestion.

A literal torrent of studies on both admission and access control has been carried out, worldwide, in the context of B-ISDN. Because of the complexity of the issues involved and the difficulty of both analysis and simulation, there is no universal agreement on methods of control. There probably never will be, because of the enormous number of networks expected to be deployed, as well as the variety of wideband applications expected for them. This chapter thus serves only as an introduction to this field. In the first section, we shall describe and analyze quantitatively some simple procedures proposed for handling call admission at the access to a network. The analysis will rely on the traffic model described in the last chapter; that of multiplexed on-off sources. Recall that this model, with appropriate adjustment of parameters, can be used to describe voice, compressed (VBR) video, and image traffic, as well as other bursty traffic. We will also apply some of our fluid-flow analysis to the study of this problem.

In section 4.2 we introduce the concept of access control by focusing on one particular strategy, the "leaky bucket" algorithm, that has received a great deal of attention in the literature. We evaluate the performance of this technique in a number of ways, concluding in section 4.3 with a fluid-flow analysis for the case of a single on-off (bursty) source.

4.1 ■ ADMISSION CONTROL

The objective here is clearly very simple: given a call arriving, requiring a virtual connection with specified QoS (bandwidth, loss probability, delay, etc.), should it be admitted? The implementation of the control can be quite com-

plex. In a real network control, messages would have to be sent along the complete virtual path that would have to provide this connection to ascertain whether the QoS objectives could be met without adversely affecting other calls already in progress. In general, this would imply checking not only the specific VP within which this VC would be established, but all other VP's sharing a part or all of the route to be used in the network, end-to-end.

A simple solution to this problem would be to allocate a specific bandwidth to each VP along its complete path, end-to-end, and then allocate that bandwidth to each VC using that VP on the basis of its *peak-rate* requirement. The path chosen and buffer allocations at each nodal switch along the path could then be used to guarantee the desired QoS. Such a solution to the problem has in fact been proposed [CCITT 1992d]. The control is indeed quite simple. Monitoring peak rate is also relatively simple: This rate is, by definition, the reciprocal of the minimum cell spacing for a given connection. The only problem is that peak-rate allocation precludes the use of statistical multiplexing, requiring possibly much wider bandwidth usage than might otherwise be the case.

Other, more complex, control algorithms can presumably provide statistical multiplexing, thus enabling network resources to be used more efficiently. But there is clearly a tradeoff: The greater the complexity of the algorithm the more costly and difficult its implementation becomes. The objective is thus to come up with reasonably simple control algorithms, capable of being implemented at B-ISDN rates, that provide the desired QoS for each connection, while at the same time allowing efficient use of the network. This is, in general, a difficult problem to solve. In this section we provide some proposed, simple, solutions, focusing exclusively at a single access point, rather than along the full VP.

As stated earlier, we will focus in this section on on-off models of traffic, since we have shown in Chapter 3 that these may be used to model voice, bursty sources, including images, and video, among other sources. Recall that an on-off traffic source requires at least three parameters to represent it: the peak rate R_p of cell transmission, in units of cells/sec (or bps, as the case may be), the average length of a burst $1/\beta$, in sec, and the average off-time $1/\alpha$. These are indicated in Figure 4–1a. We use the same notation adopted in the latter part of Chapter 3. Recall that for voice $\alpha/\beta \doteq 0.4$, and $R_p \doteq 170$ cells/sec, for 64-kbps signals and 48-byte cells. The peak rate parameter R_p is equivalent to the parameter V used in modeling voice in Chapter 3. Bursty traffic would have $1/\beta << 1/\alpha$, or $\alpha/\beta << 1$. A 2-Mbyte image transmitted from a source at a 50-Mbps rate would require 0.4 sec to be transmitted, ignoring ATM-cell overhead. In this case R_p is 50 Mbps, or roughly 120,000 cells/sec. The on-, or

[CCITT 1992d] "Traffic and Congestion Control in B-ISDN, I. 371," Recommendation, CCITT Study Group XVIII; Contained in Temporary Document 62 (XVIII), CCITT, Geneva, June 1992.

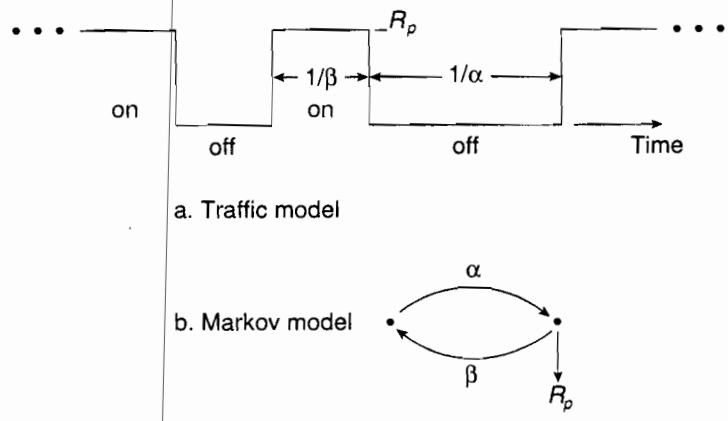


FIGURE 4–1 ■ Basic on-off traffic source.

burst-time $1/\beta$ is 0.4 sec. If the average time $1/\alpha$ between images is 10 sec, $\alpha/\beta = 0.04$. For the video minisource model discussed in section 3.6, $R_p = AK$, the minisource parameter used there, with K in units of pixels/sec.

These three parameters are examples of what are called "traffic descriptors," the parameters a user must provide the network, in addition to QoS requirements, to be used for both admission and access control. Peak rate control requires, of course, only one parameter, R_p. We say three is the *minimum* number of parameters required for this model because the statistics of the burst ("on") interval and the "off" interval might be of interest as well. In this section we implicitly assume the statistics of both intervals are exponential. Each source is thus represented by the two-state, continuous-time Markov chain of Figure 4–1b. Heterogeneous sources would each have different values for the three parameters. For homogeneous sources they would be the same. We are also implicitly assuming that during the burst period, cells are either transmitted continuously or periodically, as in the case of voice, at the peak rate R_p.

Using this basic on-off source model, we are now in a position to begin our discussion of admission control. Say that there are k traffic classes, each with its own traffic descriptors, to be multiplexed onto one network access link, of capacity C_L cells/sec. Initially, let each traffic class access its own buffer, as shown in Figure 4–2. It is clear from the pictorial representation of Figure 4–2 that the network throughput and traffic class quality of service are closely related to the way in which each buffer is served by the access link. Improved performance, in terms of increased throughput and/or better QoS parameters, should be obtained by dynamically scheduling classes for transmission, depending on class traffic statistics, buffer occupancies, and desired QoS. The type of scheduler used, in turn, impacts on the admission policy. The design of "optimum" and "near-optimum" schedulers, in the sense noted above, has been discussed in the literature and is an active area of re-

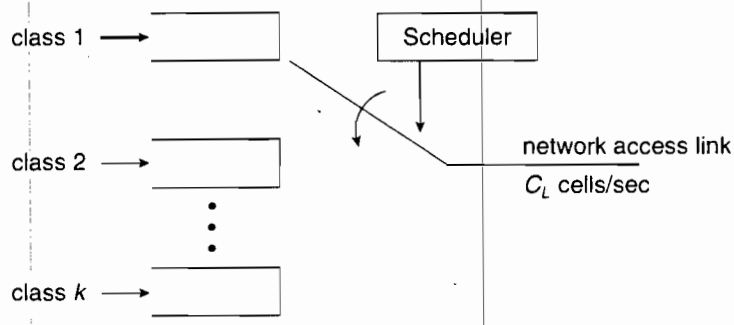


FIGURE 4–2 ■ Network access node with access scheduler.

search [HYMA 1992]. In our discussion in this section, we focus exclusively on the simplest scheduling strategy—FIFO. In this case all source buffers are combined to form one buffer holding cells from all the sources in FIFO order.

Whether more optimum scheduling or the simple FIFO strategy are used, it is clear that there must be a maximum number of connections or calls of each class for the given strategy that can be handled by the access link. This maximum number depends on the scheduling strategy, the access capacity C_L, the number of classes, and the QoS requirements for each. These maximum numbers, in turn, represent vertices in a k-dimensional region representing an admissible region for this system. The "optimum" strategy would then provide the "largest" region; the FIFO strategy the "smallest" one. Given the admissible region, in principle a call is admitted if the system, with the call present, still operates within the admissible region. A call is blocked if its acceptance would take the system outside of the admissible region.

Hypothetical two-dimensional ($k = 2$) regions appear in Figure 4–3. Some work has been done on joint scheduling—admission control for three classes of traffic [HYMA 1991], [HYMA 1992]. Simulation and analysis have been used to obtain the admissible regions for a variety of two-class admission control strategies, with or without scheduling. Selected examples appear in [FERR 1990], [DZIO 1990], [DECI 1990a]. The strategies that do not involve

---

[HYMA 1992] Hyman, J., et al., "Joint Scheduling and Admission Control for ATS-Based Switching Nodes," *Proc. ACM SIGCOMM 92,* Baltimore, MD, Aug. 1992: 223–234.

[HYMA 1991] Hyman, J., et al., "Real-Time Scheduling with Quality of Service Constraints," *IEEE JSAC, 9* (Sept. 1991): 1052–1063.

[FERR 1990] Ferrari, D., and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE JSAC, SAC-8* (April 1990): 368–379.

[DZIO 1990] Dziong, Z., et al., "Admission Control and Routing in ATM Networks," *Computer Networks and ISDN Systems, 20* (1990): 189–196.

[DECI 1990a] Decina, M., et al., "Bandwidth Assigment and Virtual Call Blocking in ATM Networks," *Proc. IEEE Infocom 90,* San Francisco, June 1990: 881–888.

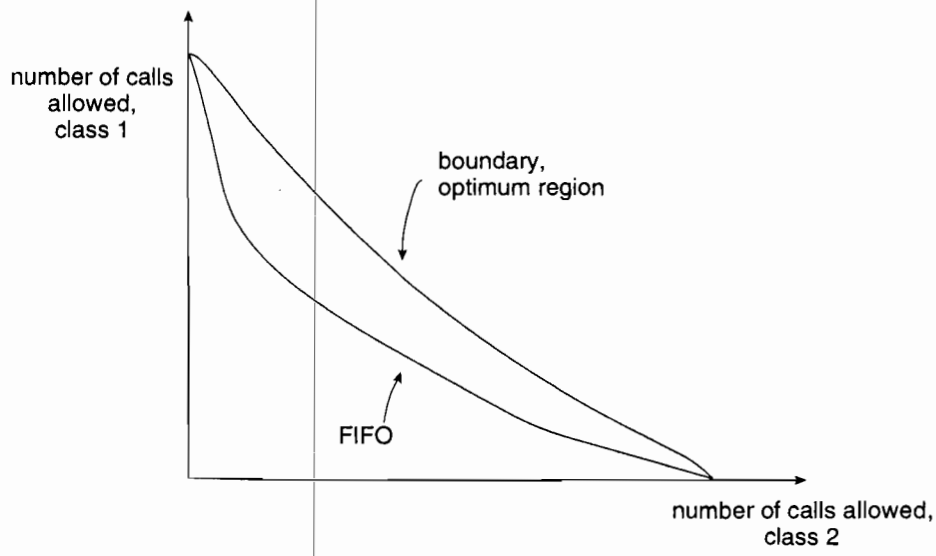


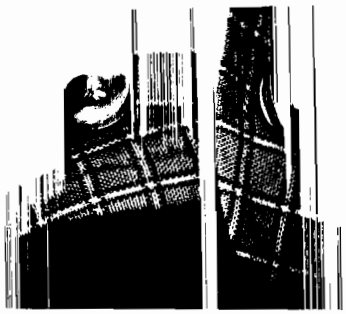
FIGURE 4–3 ■ Two-class admissible regions.

scheduling implicitly assume FIFO transmission, but with some bandwidth assignment rule defined for each class.

This work on admissible regions for ATM admission control is motivated, in part, by earlier work on call admission strategies in the circuit-switched environment. In this earlier work, the objective was to develop admission-control policies for classes of traffic, each with possibly different bandwidth requirements, and each with different blocking probability and delay objectives [KRAI 1984], [GOPA 1983]. In the work described in this chapter, we focus on single-class admission control only. This means each user connection, with possibly different traffic descriptors, must receive the same grade of service. In ATM terms, this means we focus on a particular VP that is assumed to have already been set up that is designed to provide the given QoS. This VP is assigned the capacity C_L. The question now is to determine how many calls or virtual connections of each type can be handled. For homogeneous user connections, the question is to determine the total number of VCs that may be handled. Alternatively, for a given number of connections to be handled, one would like to know the capacity C_L required for a given performance objective.

[KRAI 1984] Kraimeche, B., and M. Schwartz, "Circuit Access Control Strategies in Integrated Digital Networks," *Proc. IEEE Infocom 1984*, San Francisco, April 1984: 230–235.

[GOPA 1983] Gopal, I. S., and T. E. Stern, "Optimal Call Blocking Policies in an Integrated Services Environment," *Proc. 17th Conf. on Info. Sciences and Systems*, Johns Hopkins University, Baltimore, MD, 1983.



One obvious way of admitting calls is to determine the maximum number whose aggregate *average* rate of transmission (bandwidth) does not exceed the capacity C_L. A little thought will indicate that this provides the maximum statistical multiplexing advantage or gain. In particular, for many very bursty sources ($\alpha/\beta << 1$) one would expect very high gain. This procedure clearly allows the absolute maximum number of calls to be admitted if FIFO transmission (scheduling) is used.

The problem with this technique is that it doesn't take statistical fluctuations into account, and may, in fact, result in a high cell loss probability. Designing on the basis of average bandwidth assignment is equivalent to a link utilization of 1. This is easily demonstrated for the on-off traffic source model of Figure 4–1. Consider the single FIFO access buffer feeding a link with capacity C_L, shown in Figure 4–4. For simplicity's sake, assume homogeneous sources. (The heterogeneous case provides the same results with details left to the reader). Figure 4–4 is thus precisely the statistical multiplexor model described in Chapter 3 (see Figure 3–21 for example). The difference now is that we are interested in finding the maximum number of sources N that may be multiplexed. Conversely, given N, what is the capacity C_L required?

For the source model of Figure 4–1 we have precisely the same model and same statistics described in Chapter 3. The probability a source is in the "on," or burst, state is $p = \alpha/(\alpha + \beta)$. Its average rate of transmission is thus pR_p. For N such sources, the average rate of transmission is NpR_p and the utilization is provided by our familiar expression.

$$\rho = NpR_p / C_L \tag{4–1}$$

Setting the average rate to the link capacity C_L, to determine the maximum number of sources that can be multiplexed, is obviously equivalent to having $\rho = 1$. It is left to the reader to demonstrate that a similar result is obtained when k classes, each with different traffic descriptors, are maximally multiplexed on the basis of *average* bandwidth occupancy. In that case, however, tradeoffs are possible among the maximum number of calls allowed per user class. Specifically, the maximum utilization of 1 is again attained when

$$\sum_{i=1}^{k} n_i p_i R_{pi} = C_L \tag{4–2}$$

FIGURE 4–4 ■ Statistical multiplexing of homogeneous sources.

Different choices of n_i are possible in this case.

We focus on the homogeneous case for the time being. The "best" multiplexing strategy in terms of utilization, or multiplexing gain, is the average bandwidth assignment method just described. This provides the maximum number of calls N allowed for C_L given, but it may be unacceptable in terms of cell loss. It thus provides an upper bound on the number of calls that can be accepted or, conversely, provides a lower bound on the capacity C_L required to handle a given number of calls. The peak assignment strategy, which guarantees no cells lost (that is, the best cell loss performance) provides the lower bound on the number of calls that can be accepted, given a link capacity C_L. In this case $NR_p = C_L$. As $p \to 1$ (that is, the sources become less and less bursty) all admission control schemes merge. A source that is always "on" must be given its full bandwidth requirement R_p at all times. This is essentially circuit switching, and statistical multiplexing provides no gain.

A graphical representation of these comments is provided in Figure 4–5. It shows the $C_L - N$ tradeoff as well as the one-dimensional admissible region with C_L given and the QoS specified. (We shall focus on cell loss probability or related parameters). As the QoS parameters become more strict, the control assignment curve approaches the peak assignment curve; as they are relaxed, the assignment curve approaches the average assignment line. The admissible region widens correspondingly.

The objective is to find the admissible region: Given a set of sources with traffic descriptors specified, what is the maximum number of calls N that can

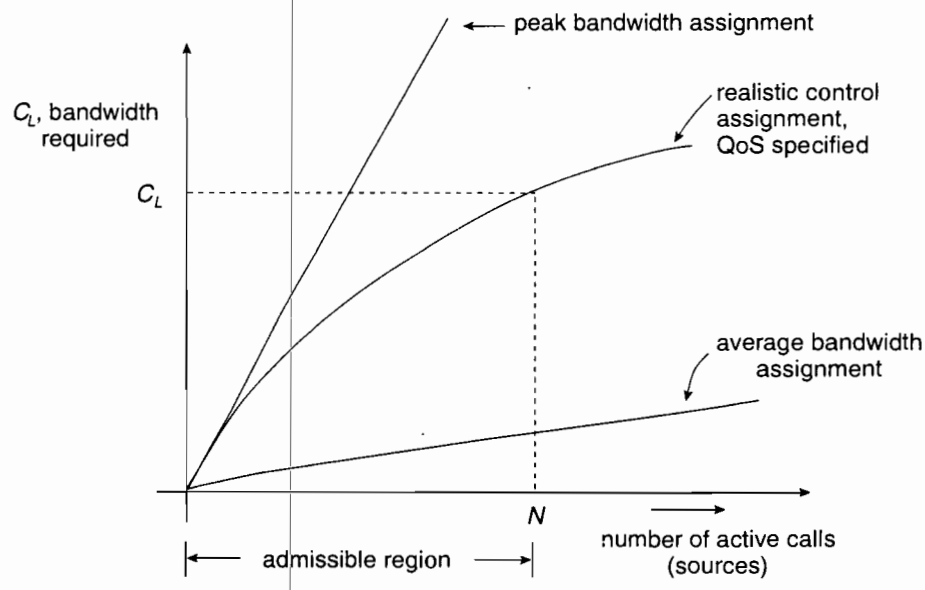


FIGURE 4–5 ■ Admission control, homogeneous on-off sources.

be admitted for a given QoS? If the number connected is less than N, any new call can be admitted without affecting the quality of service adversely. We shall approach this problem in the equivalent, alternative, manner already indicated. For a given QoS and given traffic descriptors, we find the capacity C_L required to handle N calls. This is then inverted to find N, given C_L. We proceed intuitively and heuristically at first, then more systematically later. To simplify the notation, define $m \equiv pN$ as the mean (average) number of sources "on." The mean bit rate inputted by these m connections is, as already noted, mR_p. From the discussion above, as well as from consideration of Figure 4–5, it is apparent that $C_L \geq mR_p$. A simple first guess at an acceptable value for the capacity C_L is to have it differ from the mean value mR_p by some multiple of the standard deviation σR_p. We thus write, as our first possible value of C_L,

$$C_L = (m + K\sigma)R_p \tag{4–3}$$

with K a constant to be determined, which varies with the quality of service specified. This constant should *increase* as the QoS is more strictly defined, and should *decrease,* approaching 0, as the QoS is made more lax. We now propose two simple procedures for finding K. They are related, as we shall see, and give results which are nearly the same.

Note first, however, that the capacity allocation specified by equation (4–3) is not restricted to multiplexed on-off sources. This could represent any multiplexed VBR sources as well, with mR_p and σR_p the mean and standard deviation, respectively, of the multiplexed bit rates. But focusing here on the on-off burst model of Figure 4–1, we again write $m = Np$ and note that $\sigma^2 = Np(1 - p) = m(1 - p)$. To simplify the analysis and notation again, we rewrite equation (4–3) in the equivalent normalized form,

$$C = m + K\sigma = Np + K\sqrt{Np(1-p)} \tag{4–3a}$$

with $C = C_L/R_p$ just the capacity parameter $C \leq N$ introduced in Chapter 3 (see Figure 3–4, for example).

Recall from our discussion of multiplexed on-off sources in Chapter 3 that they may be represented by an (N + 1)-state birth-death process, with the familiar state diagram reproduced in Figure 4–6. We focus particularly on the overload state J_o, the underload state $J_u = J_o - 1$, and the capacity parameter C set between them.

Recall from Chapter 3 that $J_o = \lceil C \rceil$ and $J_u = \lfloor C \rfloor$. All states $i \geq J_o$ represent the overload region in which the buffer of Figure 4–4 tends to fill; $i \leq J_u$ corresponds to the underload region, in which the buffer tends to empty. The average number of sources "on," m, lies somewhere in the underload region, either from equation (4–3a), or from noting that $\rho = NpR_p/C_L = m/C < 1$.

We now propose the following heuristic method for obtaining the constant K in equation (4–3a) and, from this, one admission control procedure for on-off sources. Let the desired QoS be the cell loss probability P_L. This is the

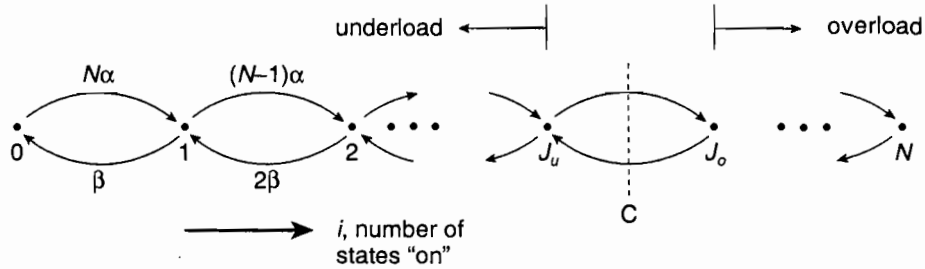


FIGURE 4–6 ■ State diagram, N on-off sources.

QoS measure we shall use consistently in our discussion of admission control. (Maximum buffer delay will be introduced as an additional measure later). A conservative estimate of loss probability proposed in the literature is the ratio of the cell input rate when the $(N + 1)$-state system of Figure 4–6 is in the overload region to the average input rate [COST 1991]. From Figure 4–6, this is the number

$$P_L = \sum_{i=J_o}^{N} (i - C)\pi_i / m \tag{4–4}$$

with π_i the probability the system is in state i (Chapter 3). This is a conservative estimate since it assumes that any input rate above the capacity will result in cell loss. It ignores the buffer completely. Given P_L, and the one traffic descriptor p that a source is "on" in this case, we can find the desired relation between the capacity C and the number of calls N connected.

A related measure of loss suggested in the literature is simply the probability of being in the overload region, that is, the probability the input rate exceeds the capacity C [GUER 1991]. Calling this measure ε, we have

$$\varepsilon = \sum_{i=J_o}^{N} \pi_i \tag{4–5}$$

We shall see that for very small P_L and ε, as would be desired in practice, either measure provides very nearly the same $C - N$ relation; that is, the same value of K.

To attain small P_L and/or ε, requires a relatively large number of sources to be multiplexed. We thus assume $N >> 1$. We also take $p << 1$; that is, we as-

---

[COST 1991] *Performance Evaluation and Design of Multiservice Networks*, J. W. Roberts, ed., Final report, Cost 224 project. Luxembourg: Commission of the European Communities, Oct. 1991.

[GUER 1991] Guerin, R. et al., "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks," *IEEE JSAC, 9,* 7 (Sept. 1991): 968–981.

sume the sources to be quite bursty. These assumptions enable us to find the constant K quite readily. Once we do this we can simply conjecture, as a heuristic, that the admissible region, or the equivalent capacity C, with N given, obeys equation (4–3a). With N large and p small, it is well known that the binomial distribution solution for π_i,

$$\pi_i = \binom{N}{i} p^i (1-p)^{N-i} \tag{4–6}$$

is approximated quite closely by the normal distribution with the same mean value $m = Np$ and variance $\sigma^2 = Np(1-p)$. This approximate distribution is sketched in Figure 4–7. The overload region corresponds to the tail of the distribution, as indicated. The sums of equations (4–4) and (4–5) must then be approximated by integrals of the Gaussian distribution over the same region. This is, in fact, another reason for requiring $N >> 1$. It should be apparent that the tail of the Gaussian distribution should be a poorer approximation to the tail of the binomial distribution than the main body of the distribution centered about the mean m, necessitating N large.

Converting sums to integrals implies converting the integer random variable i to the equivalent continuous r.v. x, as shown in Figure 4–7. The two

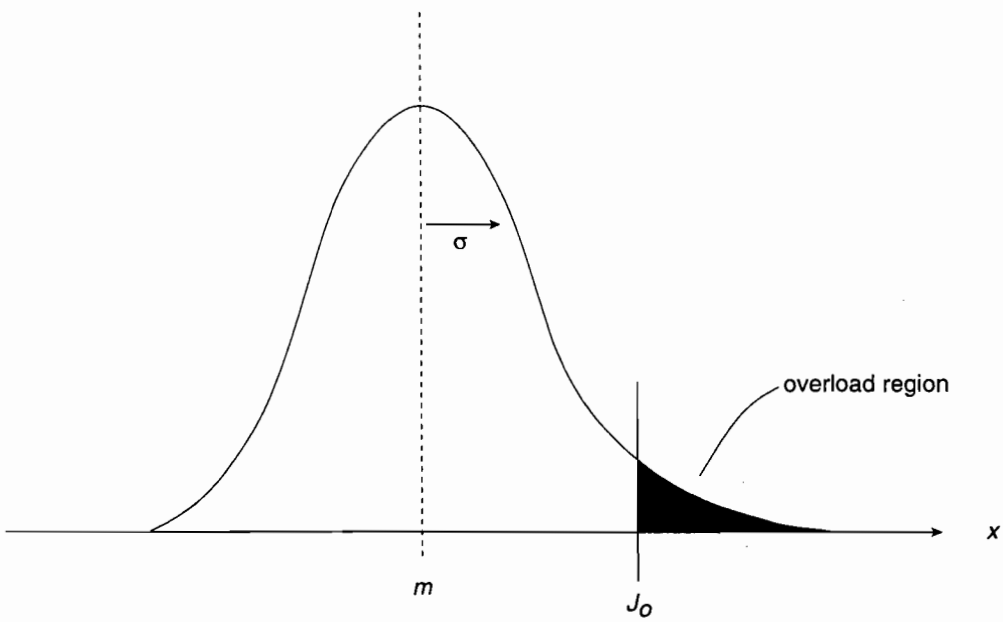


FIGURE 4–7 ■ Gaussian approximation to binomial distribution: $m = Np$; $\sigma^2 = m(1-p)$.

probabilities P_L and ε, given by equations (4–4) and (4–5), are then approximated respectively by

$$P_L \doteq \frac{1}{m}\int_{J_o}^{\infty} \frac{e^{-(x-m)^2/2\sigma^2}(x-C)}{\sqrt{2\pi\sigma^2}}\,dx \tag{4–4a}$$

and

$$\varepsilon \doteq \int_{J_o}^{\infty} \frac{e^{-(x-m)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}\,dx \tag{4–7}$$

Note that because of the continuous distribution involved, we can just take $J_o = C$, and shall do so in the calculations following. Note also that although we have obtained equations (4–4a) and (4–7) by approximating the binomial distribution by the Gaussian distribution, the distribution of multiplexed variable bit rate (VBR) traffic sources, such as multiplexed video sources, tends to the Gaussian directly (see Chapter 3). The analysis to follow applies to these sources directly, with m and σ^2 appropriately defined.

Equation (4–4a) may be rewritten in an equivalent form by writing $(x - C) = (x - m) - (C - m)$, and replacing equation (4–4a) by the sum of two integrals. This then gives us

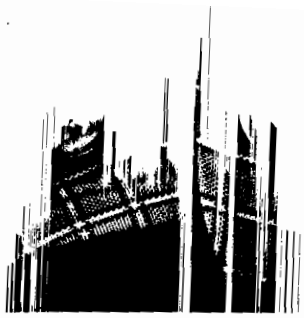
$$\begin{aligned} P_L &\doteq \frac{1}{m}\left[\int_C^{\infty} \frac{(x-m)e^{-(x-m)^2/2\sigma^2}\,dx}{\sqrt{2\pi\sigma^2}} - (C-m)\varepsilon\right] \\ &= \frac{\sigma}{m}\,\frac{e^{-(C-m)^2/2\sigma^2}}{\sqrt{2\pi}} - \frac{(C-m)\varepsilon}{m} \end{aligned} \tag{4–8}$$

after replacing J_o by C, integrating the first integral directly, and noting that the second integral is just the definition of ε.

We now have to evaluate ε, as defined by the integral of equation (4–7). This is a standard problem. Two equivalent solutions are possible. One, which we use, replaces the integral by an asymptotic series. The second invokes a Chernoff bound [HUI 1988]. The asymptotic series, appropriate for small tail probabilities, is found in the following way. Consider the integral on the left-hand side of equation (4–9), following, which is equivalent to the expression for ε. Multiply numerator and denominator by y. (This can only be done for $x > 0$. We shall, in fact, require x to be greater than 3).

$$\int_x^{\infty} e^{-y^2}\,dy = \int_x^{\infty} \frac{ye^{-y^2}\,dy}{y} \tag{4–9}$$

[HUI 1988] Hui, J. Y., "Resource Allocation for Broadband Networks," *IEEE JSAC, SAC-6* (Dec. 1988): 1598–1608.



Integrating the right-hand side of equation (4–9) by parts, one obtains

$$\int_x^\infty e^{-y^2} dy = \frac{e^{-x^2}}{2x} - \int_x^\infty \frac{e^{-y^2} dy}{2y^2} \tag{4–9a}$$

Integrating the second integral by parts in the same manner, (this can be done as often as desired), one obtains

$$\int_x^\infty e^{-y^2} dy = \frac{e^{-x^2}}{2x} - \frac{e^{-x^2}}{4x^3} + \frac{3}{4} \int_x^\infty \frac{e^{-y^2}}{y^4} dy \tag{4–9b}$$

This suffices for our purposes. Continuing in the same manner results in an asymptotic series in the sense that convergence is guaranteed only for large x. Comparing the second term with the first, note that it is small compared to the first term (less than 5% of the first term say) if $x > 3$. Extending this approach to more terms shows that the successive terms get progressively smaller and negligible.

It is a simple exercise to now show that the first term in equation (4–9b) is sufficient to provide a good approximation for ε if $(C - m) > 3\sqrt{2}\,\sigma$, with ε then given by

$$\varepsilon \doteq \frac{\sigma\, e^{-(C-m)^2/2\sigma^2}}{\sqrt{2\pi}\,(C-m)} \qquad \frac{(C-m)}{\sqrt{2}\,\sigma} > 3 \tag{4–10}$$

Similarly, the loss probability P_L, defined by equation (4–8), is readily shown, after some algebra, to be given by

$$\begin{aligned} P_L &\doteq \frac{\sigma^3 e^{-(C-m)^2/2\sigma^2}}{\sqrt{2\pi}\, m(C-m)^2} \\ &= \frac{(1-p)\sigma\, e^{-(C-m)^2/2\sigma^2}}{\sqrt{2\pi}\,(C-m)^2} = \frac{(1-p)}{(C-m)}\,\varepsilon < \varepsilon \quad \frac{(C-m)}{\sqrt{2}\,\sigma} > 3 \end{aligned} \tag{4–11}$$

if one replaces σ^2 by $m(1 - p)$. Note that P_L and ε differ by the factor $(C - m)/(1 - p)$. For P_L and ε very small, however, as is true for $(C - m) > 3\sqrt{2}\,\sigma$, the exponent in equations (4–10) and (4–11) dominates, and these differences do not affect the results very much. (Generally speaking, however, the ε approach is more conservative than the P_L approach resulting in a larger capacity requirement if used.)

Specifically, consider equation (4–10) for ε. Taking natural logs, one finds

$$\ln(\sqrt{2\pi}\,\varepsilon) = \ln\left(\frac{\sigma}{C-m}\right) - \frac{(C-m)^2}{2\sigma^2}$$

Neglecting the first term on the right-hand side in comparison with the second term, (this corresponds to focusing principally on the dominant exponential in equation (4–10)), one can solve easily for the capacity C to obtain a desired value of ε. This is given simply by

$$C \doteq m + \sigma\sqrt{-\ln(2\pi) - 2\ln\varepsilon} \tag{4–12}$$

Note that this is precisely in the form of equation (4–3a), just what we had started out to show! We thus have

$$C = m + K\sigma \tag{4–3a}$$

with the constant K now given by

$$K = \sqrt{-\ln(2\pi) - 2\ln\varepsilon} \tag{4–13}$$

As an example, if we take $\varepsilon = 10^{-5}$, we find

$$C = m + 4.6\,\sigma \qquad \varepsilon = 10^{-5} \tag{4–14}$$

Inserting the value for m for the N multiplexed on-off sources, we get

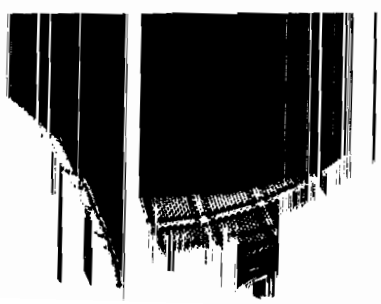
$$C = Np + 4.6\sqrt{Np(1-p)} \qquad \varepsilon = 10^{-5} \tag{4–14a}$$

For other values of ε, the constant K can be adjusted accordingly. Note also that equation (4–11) for P_L, which is dominated by the same exponent, would give about the same result.

Consider an example now. Let $p = 0.02$. (This represents a very bursty source which is "off" 98% of the time, "on" only 2% of the time.) Then, from equation (4–14a), we have

$$C = 0.02N + 0.64\sqrt{N} \qquad \varepsilon = 10^{-5}, \quad p = 0.02 \tag{4–14b}$$

This is shown plotted, versus the number of sources N, in Figure 4–8. Note the curve has precisely the shape expected, as sketched previously (in unnormalized form) in Figure 4–5. For larger values of ε (or P_L), the constant K in equation (4–13) would decrease, moving the equation for C closer to the average bandwidth assignment curve shown in Figure 4–5. For smaller ε (or P_L) the curve would move closer to the peak assignment curve of Figure 4–5, just as expected. More interestingly, Decina and Toniatti have reported on simulation results to find the equivalent bandwidth of N bursty sources required to attain a cell loss probability of 10^{-5} [DECI 1990b]. Their Figure 3 plots the results for $p = 0.02$, a cell burst size (our $1/\beta$) of 100 cells, and a buffer size of 50 cells, as the number of sources N increases. Their simulation curve is tracked almost exactly by equation (4–14b)! Figure 4–8 shows their simulation points superimposed on the analytical curve obtained using equation (4–14b). Our simple expression for the capacity C (or $C_L = CR_p$), originally motivated by in-



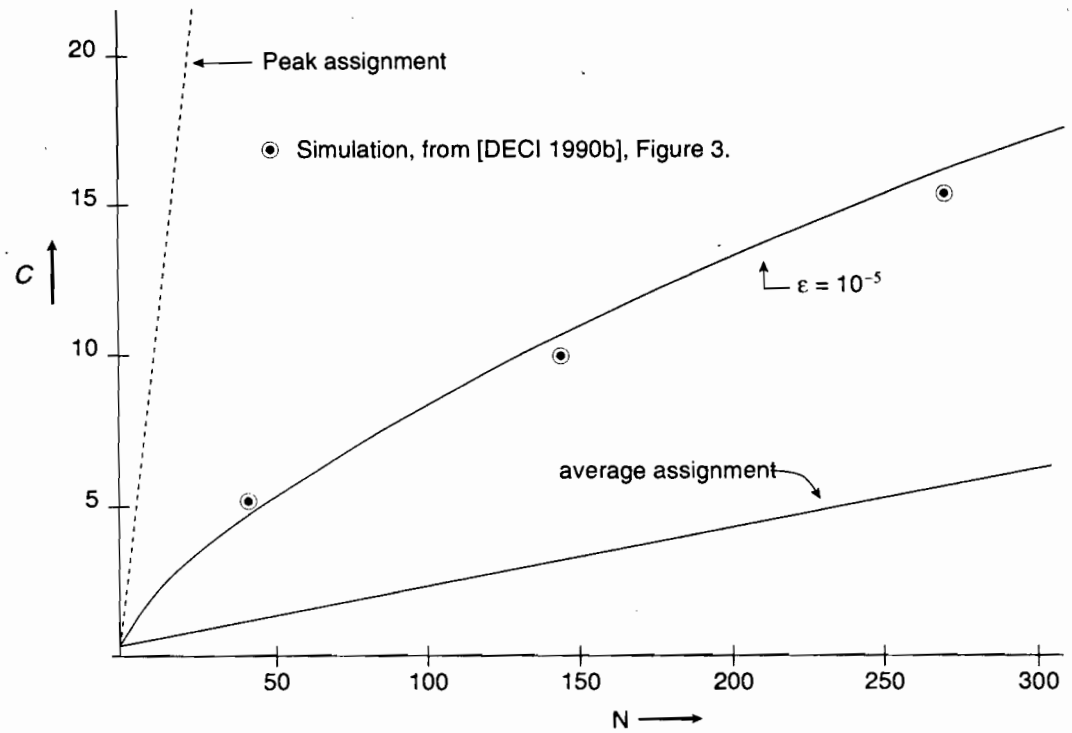


FIGURE 4–8 ■ Admission control, N homogeneous sources, $p = 0.02$.

tuition [equation (4–3)], and then found analytically by use of the Gaussian approximation to the distribution of N on-off sources (Figure 4–7), is thus verified by simulation. We now show that this expression agrees closely with other results appearing in the literature.

The simple normalized capacity expression of equation (4–12) defines the capacity required at an access buffer to handle a traffic source whose average normalized bit rate is m, with a standard deviation about m of σ. (Recall that normalization is with respect to the peak bit rate R_p, so that the actual link rate is $C_L = CR_p$). This is often called the "equivalent" or "effective" capacity of the source. For N multiplexed homogeneous on-off sources, we have $m = Np$ and $\sigma^2 = Np\,(1-p)$, with $p = \alpha/(\alpha + \beta)$ the probability a source is in the "on" state. We can invert this expression easily in this case to obtain a specific value for N, the number of such on-off connections allowed, given the actual link capacity C_L (or its normalized form $C = C_L/R_p$) and the QoS para-

[DECI 1990b] Decina, M., and T. Toniatti, "On Bandwidth Allocation to Bursty Virtual Connections in ATM Networks," *Proc. IEEE ICC 90,* Atlanta, GA, April 1990.

meter ε. Specifically, introducing the constant K defined by equation (4–13), we have,

$$C = Np + K\sqrt{Np(1-p)} \tag{4–15}$$

This is solved quite easily for N, resulting in the equation

$$N = \frac{C}{p} - \frac{1}{p}\left[\sqrt{4\alpha(C+\alpha)} - 2\alpha\right] \tag{4–16}$$

where the parameter α is defined as

$$\alpha \doteq K^2(1-p)/4$$

to simplify the appearance of the expression. Equation (4–16) defines the admissible region in this case: It is the specific expression for N shown generally in Figure 4–5. It represents, for example, the value of N in the curve of Figure 4–8 with C specified and $\varepsilon = 10^{-5}$. An admission control using this relation would thus admit a new call if fewer than N calls were connected.

In addition to providing a specific expression for the admissible region in this case of multiplexed homogeneous on-off sources, equation (4–16) allows us to focus specifically on the advantage to be obtained by statistically multiplexing these sources. Refer back again to Figure 4–4. This shows schematically that the N sources, once admission has been allowed, are combined statistically at the access buffer, with cells from each transmitted in FIFO order over the outgoing access link. One can use the analysis leading to equation (4–16) to define a statistical multiplexing gain to be accrued by combining and smoothing the source cell arrivals at the buffer.

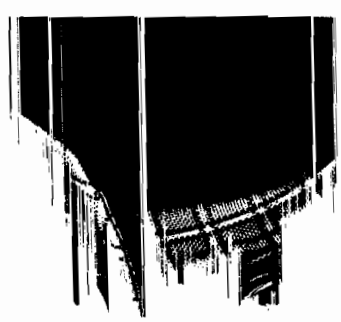
This gain G_ε, which in our case is specified by the allowable probability QoS parameter ε and more generally by the tolerable probability of loss P_L, is the increase in connections allowed over the case in which peak bit rate allocation was to be used. If peak bit rate admission control were used, the number of calls allowed into the system would be

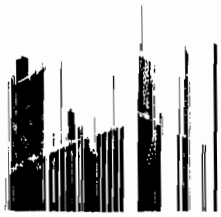
$$C_L/R_p = C$$

This would of course result in no cell loss; that is, $P_L(\text{or } \varepsilon) = 0$. In terms of our multiplexing model of Figure 4–6, this says that the number of connections allowed is such as to keep the system always in the underload region. Multiplexing gain is obtained by allowing the number N to increase beyond C, thereby incurring a cell loss penalty. The multiplexing gain G_ε is thus defined as

$$G_\varepsilon \equiv N/C \tag{4–17}$$

with the QoS parameter implicitly contained in the value of C.





The maximum possible value of gain is easily found. This is obtained when admission control is based on *average* bandwidth assignment, that is, when the number of calls allowed in is determined from $mR_p = C_L$, or $m = C$. Since $m = Np$ for the on-off source model, the maximum multiplexing gain G is given by

$$G = N/C = 1/p \tag{4–18}$$

This has a nice intuitive feel: for highly-bursty traffic, with $p << 1$, $G = 1/p$ can be quite large. As the probability p that a source is in the "on" (transmitting) state increases, the gain possible by multiplexing such sources decreases. Sources that are "on" all the time, with $p = 1$, cannot be statistically multiplexed, peak rate control must be used, and the gain $G = 1$.

The problem is that the maximum gain cannot be attained. This corresponds to average rate admission control, which we have already seen is not acceptable. This is apparent as well from equation (4–18), which shows that to attain this gain the system must be operated at a utilization of one:

$$\rho \equiv NpR_p/C_L = Np/C = 1 \tag{4–19}$$

The actual gain G_ε possible, for a given normalized capacity C, the burstiness parameter p, and an allowable loss probability ε, is obtained directly from equation (4–16). Writing the parameter α out to show the specific dependence on p, and dividing N in equation (4–16) by C to obtain the gain as defined by equation (4–17), we obtain

$$G_\varepsilon = \frac{1}{p} - \frac{1}{p}\left\{\sqrt{\frac{K^2(1-p)}{C}\left[1+\frac{K^2(1-p)}{4C}\right]} - \frac{K^2(1-p)}{2C}\right\} \tag{4–20}$$

The parameter K depends on the loss probability ε, as given by equation (4–13). Note, as expected, that as either the capacity C increases or K decreases, a greater gain is possible.

Figure 4–9 is a plot of equation (4–20) for $\varepsilon = 10^{-5}$ (in this case $K = 4.6$, as already noted), for $C = 10$ and 100. The curves plotted agree with those appearing in [COST 1991], Figure 3–2–2, p. 38. Also shown is the "ideal" gain curve $G = 1/p$.

These curves have been plotted on a log-log scale to show that over most of the range of p they are very nearly linear. This implies that $G_\varepsilon p$ = constant. But note from the definition from equation (4–17) of the multiplexing gain that $G_\varepsilon p = Np/C$ is the utilization p. We thus have

$$G_\varepsilon p = \rho = \text{constant} \qquad \rho < 1 \tag{4–21}$$

This observation has been previously made in [COST 1991] and, in fact, is shown by curves plotted there of ρ versus p for a loss probability of 10^{-9}. For the curves of Figure 4–9, with $\varepsilon = 10^{-5}$, we find that $G_\varepsilon p = \rho \doteq 0.64$ for $C = 100$

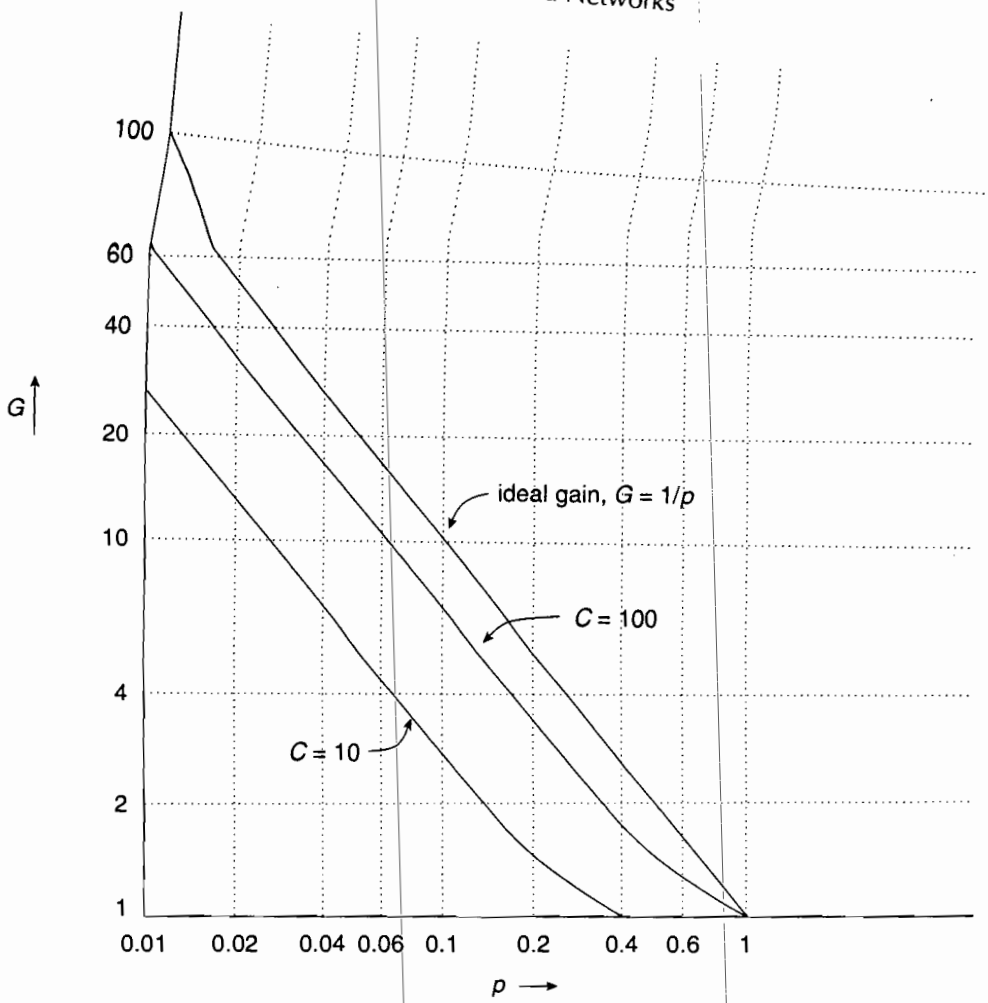
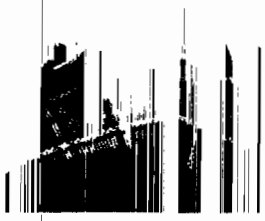


FIGURE 4–9 ■ Multiplexing gain, on-off sources, $\varepsilon = 10^{-5}$.

and $\rho \doteq 0.27$ for $C = 10$. A higher gain is possible if the allowable capacity C is increased, with correspondingly more sources allowed to be connected, and the utilization thus increasing.

In the material preceding, we have focused on a simple formulation of the admission control problem, relying both on an intuitive, heuristic approach as exemplified by equation (4–3), and a more rigorous analytical approach appropriate to many multiplexed bursty on-off sources, as well as "Gaussian-type" VBR sources, as exemplified by equations (4–10) and (4–11). The problem with this approach is that it ignores the multiplexing buffer completely, and relies on conservative bounds on the cell loss probability, captured, in the case of multiplexed on-off sources, by the probability the compos-



ite traffic stream will exhibit overload behavior. To include the effect of the access buffer and resultant cell loss probability on admission control, as well as the buffer delay (another significant QoS parameter), is an inherently difficult problem. It turns out, however, that the fluid-flow analysis of the previous chapter can be used for this purpose, in quite a straightforward, relatively simple manner. We summarize one method proposed, following the work of [GUER 1991]. This again assumes multiple on-off sources are multiplexed, with heterogeneous sources included quite naturally. To simplify the analysis, one has to again adopt conservative estimates of cell loss probability, leading to conservative estimates of the "equivalent," or "effective", capacity required. To redress this problem somewhat, Guérin et al. [1991] propose choosing, as the equivalent capacity (hence implicitly adopting an admission control strategy), the smaller of either of two capacities: the capacity obtained by using the fluid-flow analysis and the capacity calculated from equation (4–15).

The equivalent capacity using fluid-flow analysis follows directly from section 3.6. We focus on homogeneous sources first. Refer, in particular, to Figure 3–21, parts (b) and (c), which shows M on-off minisources statistically multiplexed at an access buffer. If we compare these parts of Figure 3–21 with Figures 4–1 and 4–4 in this chapter, we note that they represent the same multiplexing model if M, there, is replaced by N, here, KA bits/sec ($K = 7.5 \times 10^6$ pixels/sec) is replaced by R_p, the peak on-off source rate, and KC bits/sec is replaced by the link capacity C_L. But recall from section 3.6 that the survivor function $G(x)$ for the M-minisource model, the probability the buffer occupancy exceeds x bits, was given by equation (3–111). Introducing the changes in notation indicated above, we immediately have as the probability the buffer occupancy in Figure 4–4 exceeds x bits, the asymptotic approximation

$$G(x) \sim A_N \rho^N e^{-\beta r\, x/R_p} \tag{4–22}$$

with the parameter r given by

$$r = (1-\rho)\left(1+\frac{\alpha}{\beta}\right)/(1-C_L/NR_p) \tag{4–23}$$

and

$$\rho = NpR_p/C_L < 1 \tag{4–24}$$

Note that the capacity C_L appears explicitly in the expression for $G(x)$ in the exponent r and in the term ρ^N, as well as implicitly in the parameter A_N. Letting $G(x)$ be an approximation to the probability of loss P_L (the approximation mentioned first in Chapter 3), and letting all the parameters in equation (4–22) be specified, except for the capacity C_L, one can, in principle, solve equation (4–22) for C_L. This then results in the link (actually virtual path) ca-

pacity required to obtain a specified quality of service P_L. Note that the buffer size x must be specified as well. One can choose the maximum buffer size on the basis of some hardware or processing constraint or on the basis of maximum delay. This latter constraint requires some iteration, since the delay depends both on the buffer size and the capacity C_L. Given x, one can find C_L, check to see that the maximum delay constraint is not violated, and if it is, repeat the process with a smaller value of x. Clearly the larger the value of x, the smaller the loss probability or, with P_L fixed, the smaller the capacity required.

The solution for C_L with loss probability $G(x)$, x, and the traffic parameters specified, requires iteration of equation (4–22). A simple, closed-form expression for the capacity C_L required can, however, be obtained from equation (4–22) if one focuses on the dominant exponent only. Specifically, set the function $A_N\rho^N = 1$. This is expected to be a relatively good approximation when the utilization ρ is relatively close to 1, in which case $A_N > 1$ and $\rho^N < 1$ compensate for one another to some extent. When ρ becomes smaller, however, ρ^N decreases rapidly for large N. This means that the capacity found by using only the exponent in equation (4–22) in this case is larger than it should be; that is, the capacity obtained is a conservative estimate. These observations are borne out by calculations summarized in [GUER 1991], as will be noted later.

Using the exponent in equation (4–22) only, one then has as the proposed approximation for the loss probability

$$P_L \doteq e^{-\beta r\, x/R_p} \tag{4–25}$$

From this expression one finds immediately that

$$\beta\, rx/R_p = -\ln P_L \tag{4–26}$$

Using equations (4–23) and (4–24) in equation (4–26), one obtains a quadratic equation for C_L. The solution is easily shown to be given by

$$C_L/R_pN = \frac{1-k}{2} + \sqrt{\left(\frac{1-k}{2}\right)^2 + kp} \tag{4–27}$$

with the parameter k defined as

$$k \equiv \beta x/R_p(1-p)\ln(1/P_L) \tag{4–28}$$

As a check, let $p \to 1$ with β fixed. Then $k \to \infty$. This implies the "off" interval $1/\alpha$ goes to zero. From equation (4–27), one readily shows that $C_L/R_pN \to 1$. This is as to be expected: With all N sources "on" all the time, the capacity required to handle them is NR_p, that is, the peak assignment, independent of loss probability P_L.

Consider an example [GUER 1991]. Let $P_L = 10^{-5}$, take as the buffer size $x = 3$ Mbits, $R_p = 4$ Mbps, and $1/\beta = 100$ msec. (The average number of bits generated by each source during an average burst interval is $R_p/\beta = 400$ kbits. Note that this is much less than the buffer size x.) For these numbers, we have $k = 0.65/(1 - p)$. Equation (4–27) now predicts that the equivalent capacity per source C_L/N is independent of the number of homogeneous sources N. Its specific value depends on the probability p a source is in the "on" state.

How valid is this result? Guérin et al have tested this conclusion out for this specific example, among others. Their results for this example appear reproduced in Figures 4–10 and 4–11. Figure 4–10 is for the case of $N = 1$ on-off source only. For this case $A_N\rho^N = \rho = pR_p/C_L$ (see equation (3–38) in Chapter 3). One would expect the effect of ignoring the term $A_N\rho^N$ in equation (4–22) to obtain the exponential approximation of equation (4–25) to P_L to be negligible for $P_L << 1$, as is the case here. This is borne out by Figure 4–10, which compares the "flow approximation" capacity obtained using equation (4–27) (derived from equation (4–25)) to the "exact value" that is calculated using the full fluid-flow

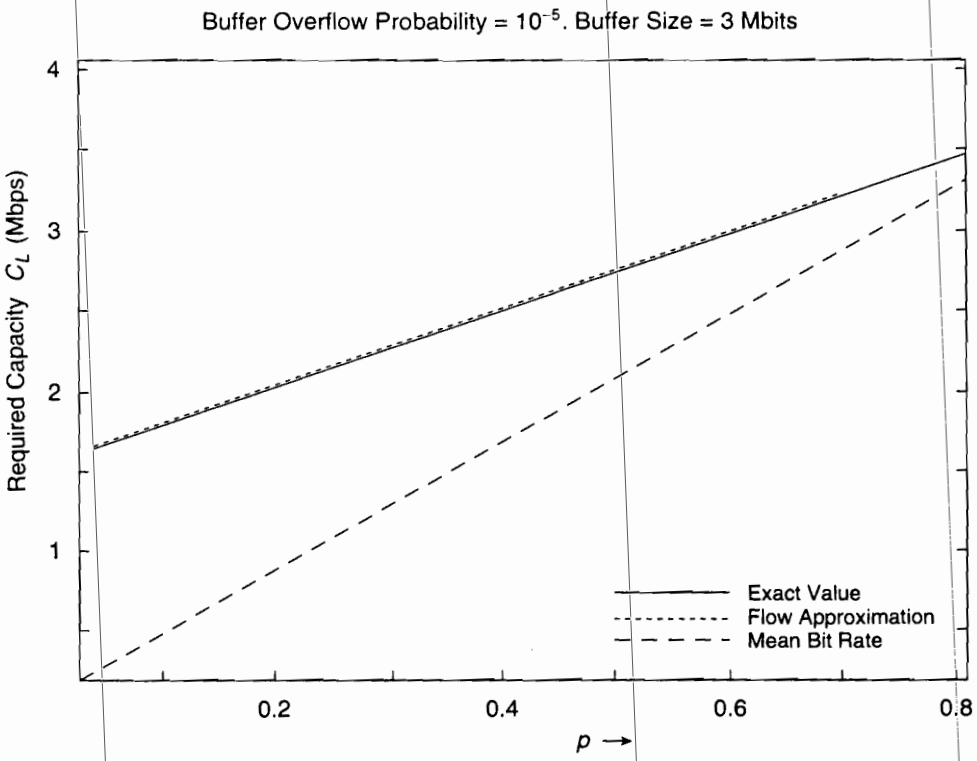


FIGURE 4–10 ■ Equivalent capacity for a single source with 4 Mbs peak rate and 100 ms mean burst period (from [Guer 1991], Figure 3. © 1991 IEEE).

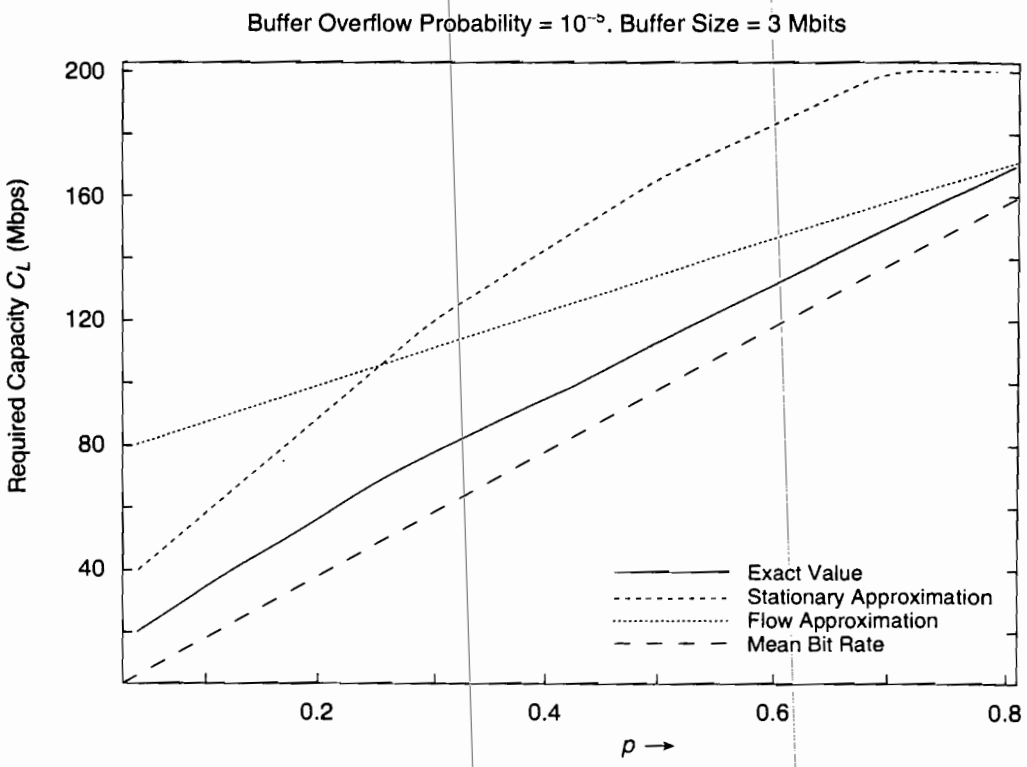


FIGURE 4–11 ■ Equivalent capacity for 50 sources, each with 4 Mbs peak rate and 100 ms mean burst period (from [Guer 1991], Figure 5. © 1991 IEEE).

analysis, to which equation (4–22) is the asymptotic approximation. The capacity is plotted as a function of p, as indicated. Note that for this single source case, equation (4–27) provides an excellent approximation. Figure 4–11 shows the equivalent capacity C_L required for $N = 50$ such on-off sources. Note that here, equation (4–27) only provides relatively good results for $p \geq 0.6$. The capacity found using equation (4–27) is much too conservative for p below that value. This corresponds to $\rho \geq 0.83$ for that case. This agrees with our earlier statement that assuming $A_{N\rho}N = 1$ in going from equation (4–22) to (4–25) should result in values of capacity that are too conservative for ρ^N small.

Figure 4–12, also taken from [Guer 1991], shows the results for another example, this time for $N = 5$ sources, each with a 40 Mbps peak rate. Here, equation (4–27) provides results intermediary between those of Figures 4–10 and 4–11. Note that here equation (4–27) (the "flow approximation" curve) provides a relatively good approximation to the full fluid-flow analysis for $p \geq 0.4$, or $\rho \geq 0.67$. This is as expected.

Guérin et al have proposed as a "good" approximate measure of equivalent capacity C_L the minimum of C_L, as calculated from equation (4–27), and

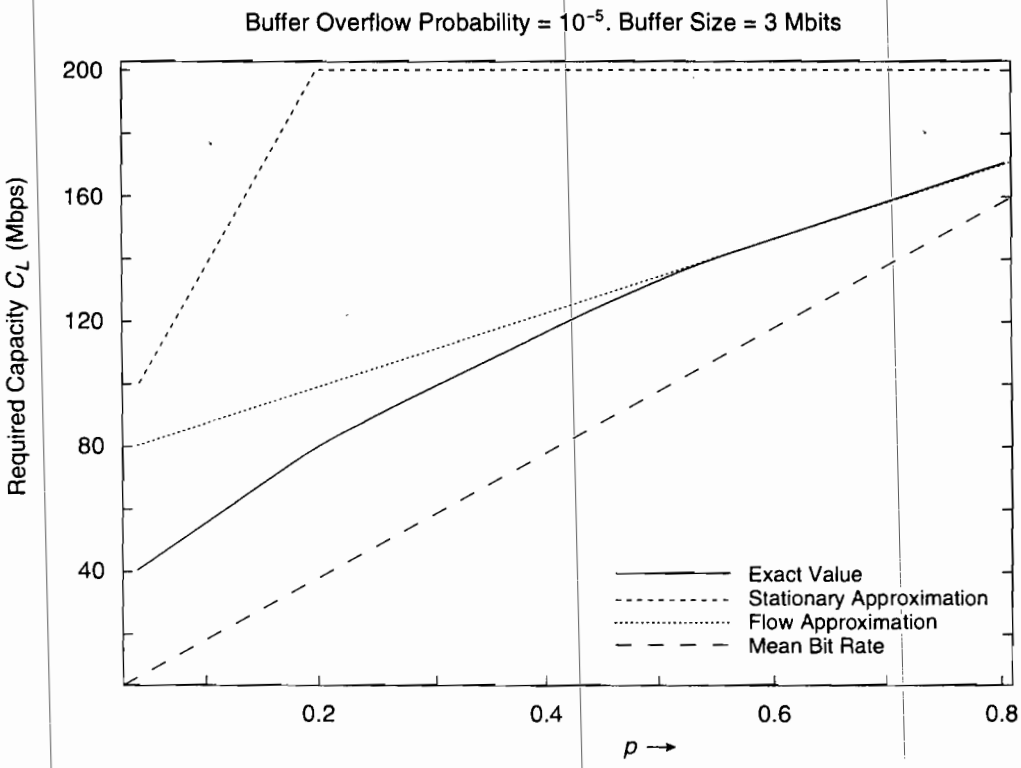


FIGURE 4–12 ■ Equivalent capacity for five sources, each with 50 Mbs peak rate and 100 ms mean burst period (from [GUER 1991], Figure 4. © 1991 IEEE).

the value obtained using our earlier simple measure of equivalent capacity [equation (4–12)]. They call this earlier value the "stationary approximation," and it is so labeled in the curves of Figures 4–11 and 4–12. It is clearly an inappropriate approximation for the $N = 5$ source case of Figure 4–12. This is as to be expected since we assumed $N >> 1$ in obtaining equation (4–12). In Figure 4–11, for $N = 50$ sources, however, it does provide a better approximation than the flow approximation of equation (4–27) for $p \leq 0.25$. Using either approximation, whichever is smaller, thus does provide a closer fit to the capacity obtained using the complete fluid-flow analysis over the complete range of variation of p.

Specifically, letting

$$C_{LS} = mR_p + \sigma R_p \sqrt{-\ln(2\pi) - 2\ln\varepsilon}$$

from equation (4–12), and labeling C_{LF} the fluid approximation value of equivalent capacity from equation (4–27), [GUER 1991] propose as a measure of equivalent capacity

$$C_L = \min[C_{LS}, C_{LF}] \tag{4–29}$$

The discussion thus far has focused on the loss probability P_L as a measure of quality of service. As noted earlier, however, the choice of buffer size x implicitly provides a bound on the maximum wait time in the buffer. Consider the example just discussed in which x was chosen as 3 Mbits. For the single-source case of Figure 4–10, with C_L varying from 1.4 Mbps with $p = 0$ to the peak value of 4 Mbps with $p = 1$, the maximum buffer delay (wait time) ranges from 2.1 to 0.75 sec. This number might be considered excessive in some applications, particularly since this is only one buffer in a number that might be encountered as the traffic moves through a network. But note also that with $p << 1$, the source "off" time $1/\alpha$ is very long, so that 2 sec, in the case of some image transmission applications, for example, might not be considered too long.

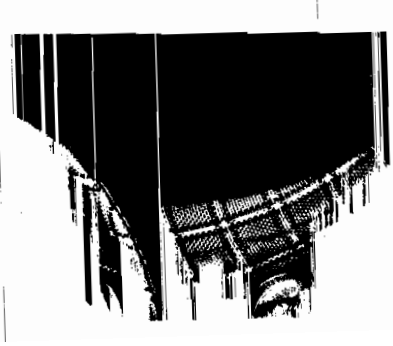
But more importantly, for broadband networking one would presumably use much higher bit rates with statistical multiplexing invoked. The results of Figures 4–11 and 4–12 are then more significant. Note that in these cases, the capacities range from 20–200 Mbps, with the corresponding maximum buffer delays ranging from 15–150 msec. These numbers appear much more tolerable for bursty source applications such as images and file transfers with the mean burst intervals of 100 msec and 10 msec chosen for Figures 4–11 and 4–12, respectively.

All of the discussion to this point has focused on equivalent or effective capacity and admission control for multiplexed homogeneous sources. We conclude this section with some brief extensions to the case of multiplexed heterogeneous sources. First note that our simple expression equation (4–12), for the equivalent capacity of multiplexed sources, lends itself quite readily to heterogeneous sources of various types, since it only involves two moments: the mean bit rate and standard deviation. The basic assumption is that the central limit theorem applies, that is, that the histogram of the bit rates of the traffic sources summed (multiplexed) approaches a normal distribution. If so, one simply adds mean bit rates and variances of the bit rate distributions to obtain m and σ^2 directly.

The second, fluid approximation to the equivalent capacity can be readily extended to heterogeneous bursty (on-off) sources as well. To show this, we first rewrite equation (4–27) in an equivalent form. Replacing the on-state probability p in that expression by $\alpha/(\alpha + \beta)$, one readily shows that the equivalent capacity C_L can be written in the following form:

$$C_L/N = \frac{R_p}{2} - \frac{(\alpha+\beta)x}{2\ln(1/P_L)} + \sqrt{\left(\frac{R_p}{2} - \frac{(\alpha+\beta)x}{2\ln(1/P_L)}\right)^2 + \frac{\alpha R_p x}{\ln(1/P_L)}} \tag{4–27a}$$

Now consider N heterogeneous on-off sources multiplexed together, each with different peak bit rates, mean "on" (burst) times, and mean "off" intervals. For



the ith source, $1 \le i \le N$, these are R_p^i, $1/\beta_i$, and $1/\alpha_i$, respectively. We shall show in chapter 6 that the equivalent, or effective, capacity C_L required for the multiplexed sources accessing a single buffer is given by

$$C_L = \sum_{i=1}^{N} \left[\frac{R_p^i}{2} - \frac{(\alpha_i + \beta_i)x}{2\ln(1/P_L)} + \sqrt{\left[\frac{R_p^i}{2} - \frac{(\alpha_i + \beta_i)x}{2\ln(1/P_L)}\right]^2 + \frac{\alpha_i R_p^i x}{\ln(1/P_L)}} \right] \qquad (4\text{–}30)$$

The assumption here is that each source requires the same QoS parameter P_L. This is consistent with the FIFO service discipline we have been assuming throughout. Capacities defined in this manner thus add.

The concept of equivalent or effective capacity can be generalized to a variety of source models, including those such as the Markov-modulated models used to describe voice and video signals, as discussed at length in Chapter 3. We shall return to a detailed discussion of effective capacity in Chapter 6, after developing the analytical tools required to present this concept properly. We note here that, in general, the effective capacity is found to provide a conservative estimate of the bandwidth requirements of various sources. This is in agreement with the conclusions of [GUER 1991]. Yet it is a useful tool in many situations because of its additive property, as shown by equation (4–30). Given any set of multiplexed sources, one calculates the capacity of each source, and simply adds the capacities, assuming each source requires the same QoS. Details appear in Chapter 6, where we present both the benefits and deficiencies of this technique.

4.2 ■ ACCESS CONTROL—LEAKY BUCKET TECHNIQUE

In the previous section we discussed admission control in the context of B-ISDN. The objective there was to determine whether or not a call requiring a virtual connection (VC) with specified QoS should be admitted to a network. A comprehensive admission control admits the call if it is found that the quality of service of existing connections on the source-destination path is not affected. The difficulty of evaluating performance end-to-end along a complete virtual path (VP), forced us to focus on a more restricted class of admission controls, those exerting control at the access point only. (More generally, the results obtained apply within the network at nodal switching points if the traffic characteristics of both the new connection and those already in existence may be assumed known and statistically independent of one another.)

Assuming a call has been admitted and the connection established, it is necessary to monitor and control the traffic actually generated by the call to ensure it conforms to the traffic descriptors originally specified. This is required because of the statistical nature of the traffic generation process. It is also required because of the possibility of malicious behavior on the part of

some users. This procedure of controlling the traffic during the period of a connection is variously referred to as Usage Parameter Control (UPC), access control, credit management, or traffic "policing". Users found violating their connection "agreement" (the traffic descriptors specified during the admission phase) may have arriving cells dropped to avoid network congestion or adverse impact on the QoS of the traffic of other users. Proposals have also been made to "tag" cells of user traffic violating the agreement. These cells are then allowed into the network, to be dropped later should congestion be encountered.

Alternatively, entering traffic may be "shaped" or smoothed to reduce any adverse impact on the network. There are two possibilities here: to first buffer the traffic and then read cells out at a smoothed, more regular rate; or, as another possible policy, to control the flow of traffic at the source to prevent it from congesting the network. We describe the first method briefly in this section. The second method has, in the past, been somewhat controversial, since investigators feel that at the high bit rates proposed for ISDN, B-ISDN flow control cannot be exerted effectively. A source located some distance away from the control would not be able to react quickly enough to flow control signals, and might continue to overrun or congest the network.

Despite these concerns, the ATM Forum has proposed a feedback control mechanism be adopted for the control of available bit rate (ABR) traffic. This mechanism, which controls the rate of transmission of cells at the source along a given virtual circuit, has been found by simulation to potentially provide appropriate congestion control over wide-area networks. We discuss this mechanism in Chapter 7.

In Chapter 7 we also describe briefly another traffic case in which flow control *is* feasible. This is the case of multiplexed video and bursty, nonreal-time, sources. Recall from our discussion of video traffic in Chapter 3 that we noted that compressed (VBR) video may retain frame-to-frame correlation for the order of 10–20 frames. This implies being able to predict video traffic intensity over more than one frame interval, or 30 msec. This interval is long enough to allow flow control signalling back to the source of the bursty traffic. (Round-trip propagation delay across the United States is 40–50 msec, which is just in the range of video correlation time). As noted above, we defer discussion of possible flow control procedures to a later chapter. In this section we focus on "open loop" access control. Each connection requires its own access controller and possibly a traffic shaper before being multiplexed with other connections for entry into the network. The general picture appears as shown in Figure 4–13. In the material following, we at first discuss access control without shaping, then include it through the use of an access control buffer.

A variety of techniques have been proposed and compared for access control of ATM networks [RATH 1991]. We focus here on the one most commonly discussed in the literature, the "leaky bucket" policing mechanism. The Usage Parameter Control technique proposed by the ATM Forum, called the Generic

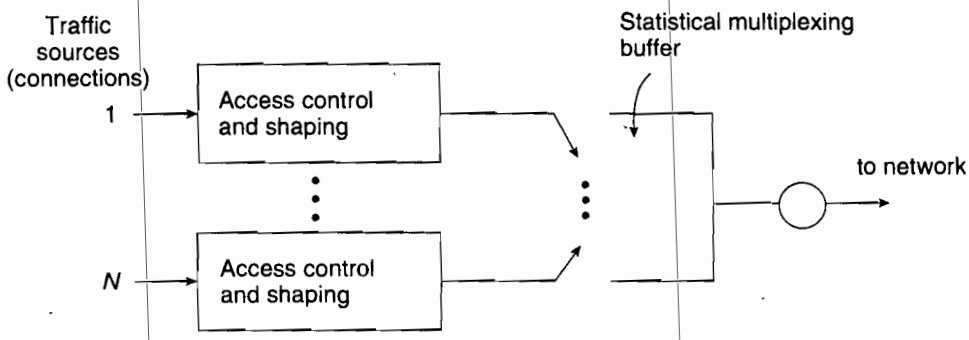


FIGURE 4–13 ■ Access control in B-ISDN networks.

Cell Rate Algorithm, has a number of equivalent representations, one of which is the leaky bucket technique. By generalizing some of the features of this technique, including making it adaptive or dynamic, one can come up with a number of related control procedures as well. The leaky bucket control method, in its basic form, is quite simple and would appear to be particularly appropriate for bursty sources of the on-off type that we encountered both in the previous section and in Chapter 3. It does two things—it restricts the average cell throughput into the network to a specified maximum value, and allows bursts of cells up to some maximum number. (Depending on the burst length allowed, the peak rate may thus be quite higher than the average rate). It was first proposed as a possible traffic control procedure by Jonathan Turner [TURN 1986].

The algorithm is very simple and may be described and/or implemented in several equivalent ways. One method proposed involves incrementing a counter C by 1, periodically every D sec, until a maximum value of M is reached. The counter is decremented by 1 for every cell transmitted. Transmission is interrupted and cells are dropped whenever the counter reaches 0. (Alternatively, in the "violation tagging" mode noted above, cells are "tagged"; that is, a bit is set, when the counter is at 0. We discuss the cell dropping mode only in the material following). The average rate of transmission is then limited to a maximum of $r = 1/D$ cells/sec. The counter setting M allows a maximum burst of M cells to be transmitted. There are thus two parameters to adjust in this algorithm. Because cells may be occasionally dropped, (the counter is 0 when a cell arrives), the average throughput is less than its maximum possible value r. Increasing the burst size M allowed reduces this effect, but slows down the system response to congestion.

[RATH 1991] Rathgeb, E. P., "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks," *IEEE JSAC, 9,* 3 (April 1991): 325–334.

[TURN 1986] Turner, J. S., "New Directions in Communications (or Which Way in the Information Age?)," *IEEE Commun. Mag., 24,* 10 (Oct. 1986): 8–15.

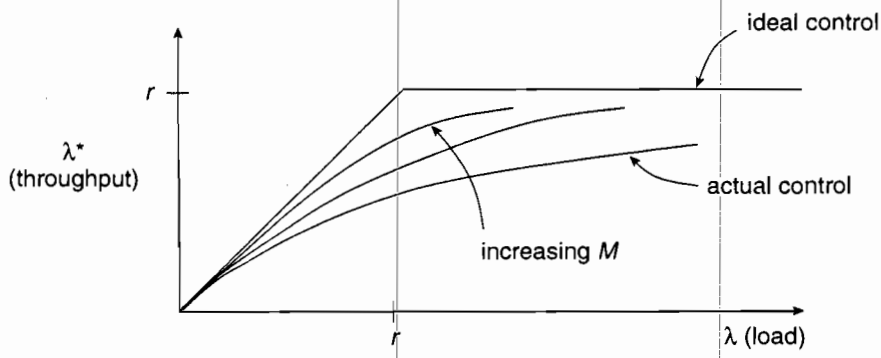


FIGURE 4–14 ■ Throughput-load curve, leaky bucket algorithm.

An ideal control curve would have the throughput, in cells/sec, equal the traffic load until that load reached r. As the load continued to increase beyond r, the throughput would stay at r. This curve is shown sketched diagrammatically in Figure 4–14. The load is represented by the symbol λ, the throughput by the symbol λ^*.

A second representation of the algorithm has the counter decremented periodically until it reaches 0. It is incremented by 1 whenever a cell is transmitted, until a maximum value of M is reached, at which time no cells can be transmitted. (This again ignores violation tagging). A third representation involves the use of a "token pool" buffer as shown in Figure 4–15. A cell must have a token waiting to be transmitted. Tokens are generated once per D sec, as shown, and wait in the buffer, until the buffer fills. At this time no further tokens are generated. It is left for the reader to show that all three representations are identical. As indicated in Figure 4–15, the average throughput λ^* differs from the load λ because of possible cell loss. The parameter P_L in Figure 4–15 represents, as it did in the previous section, the cell loss probability. Variations of this basic algorithm include, in addition to the possibility of violation tagging already mentioned, adding a cell smoothing buffer, adapting both the token arrival rate r (the counter increment rate) and the maximum token buffer (counter) size M to the traffic load and/or congestion conditions in the network, as well as other possibilities. A simple modification of the algorithm to handle variable-length packets instead of fixed-size cells has been proposed as well [BALA 1990]. In this procedure a given block of octets within a packet requires a token. Longer packets thus require more tokens to be transmitted.

We shall focus first on the basic algorithm, showing how one obtains the $\lambda^* - \lambda$ (throughput-load) curve analytically. We do this first through a very

[BALA 1990] Bala, K., et al., "Congestion Control for High-Speed Packet-Switched Networks," *IEEE Infocom '90,* San Francisco, (June 1990): 520–526.

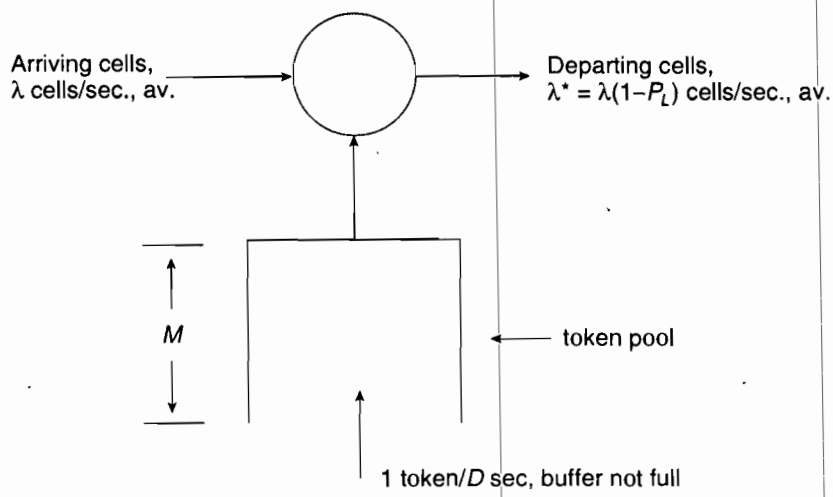


FIGURE 4–15 ■ Leaky bucket algorithm.

simple approximate technique, appropriate only for Poisson cell arrivals. This serves to reinforce the leaky bucket concept and provides simple, first-order results. We then provide a more detailed discrete-time analysis of the leaky bucket algorithm using a simplified version of an approach proposed in the literature. Finally, in the next section, we generalize the algorithm to include a data shaping buffer and show how the fluid-flow analysis of Chapter 3 can be used to provide useful performance results for on-off, bursty traffic in this case.

The simple approximate analysis of the leaky bucket procedure is based on the closed queueing network model shown in Figure 4–16. A little thought will indicate that this model captures the essence of the leaky bucket procedure diagrammed in Figure 4–15. Cells are assumed generated at a Poisson rate λ, as indicated by the upper server. But these will only be generated if the upper queue has at least one "occupant" or token. That queue increases at an average rate $r = D^{-1}$, as shown. (Note that this rate is also assumed Poisson with this model, rather than the desired periodic incrementing of the counter in the leaky bucket algorithm). The closed network has M tokens circulating in it, as shown. At most M cells can thus be served in succession. If all M circulating tokens are in the lower queue, the upper queue is empty, and cell service is blocked. This two-queue model of the leaky bucket algorithm and the analysis based on it are due to Wong and Kramer [WONG 1988].

[WONG 1988] Wong, L-N, and M. Kramer, "A Performance Analysis of a 'Leaky Bucket' Access Control Scheme for Broadband MANs," unpublished paper, Nynex Science and Technology, August 1988.

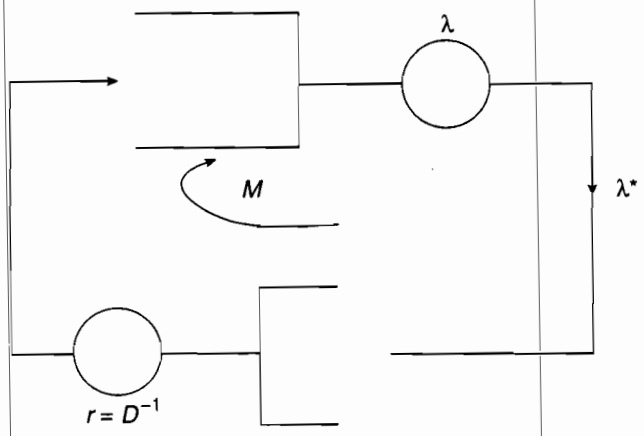


FIGURE 4–16 ■ Closed queueing network model, leaky bucket, *M/M/1* approximation.

The throughput λ^* of this system is now written simply as

$$\begin{aligned}\lambda^* &= \lambda(1-p_0) \\ &= \lambda(1-P_L)\end{aligned} \tag{4–31}$$

where $p_0 = P_L$ is the probability the upper queue is empty or, from Figure 4–16, the probability the lower queue is full. But either queue behaves as a finite $M/M/1$ queue, of maximum occupancy M, under the Poisson assumptions made above. In particular, from finite $M/M/1$ analysis, the probability the lower queue is full is

$$P_L = \rho^M(1-\rho)/[1-\rho^{M+1}] \tag{4–32}$$

with $\rho \equiv \lambda/r = \lambda D$.

Inserting equation (4–32) in (4–31), and simplifying, we have

$$\lambda^* = \lambda\left[\frac{1-\rho^M}{1-\rho^{M+1}}\right] \tag{4–33}$$

This may also be written in normalized form as

$$\lambda^*/r = \lambda^* D = \rho\left[\frac{1-\rho^M}{1-\rho^{M+1}}\right] \tag{4–33a}$$

Equation (4–33a) has been plotted in Figure 4–17. Note that it has the expected form sketched previously in Figure 4–14. As M increases, the throughput-load characteristic approaches the ideal characteristic in which the throughput λ^* equals the load λ for $\lambda \leq r$, and saturates at the maximum value of r for $\lambda >> r$.

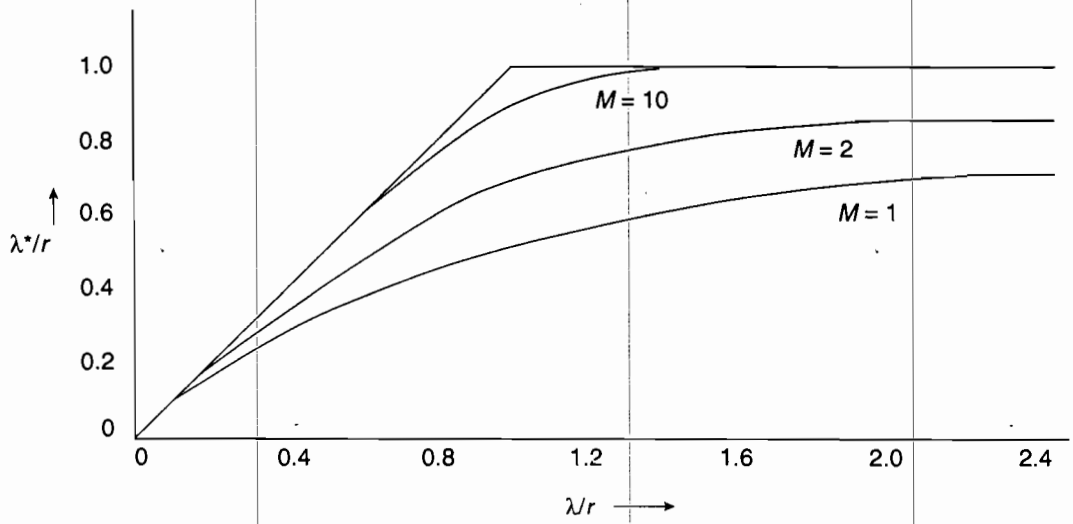


FIGURE 4–17 ■ Throughput-load characteristic, model of Figure 4–16.

A number of more sophisticated analyses of the leaky bucket algorithm have appeared in the literature [SIDI 1989], [BERG 1990], [HEYM 1992], [BUTT 1991]. They differ in the way in which they model the leaky bucket technique and in the types of traffic and traffic models they can handle. Interestingly, the curve for $M = 10$ in Figure 4–17 tracks the equivalent curve in [BERG 1990] reasonably well [BERG 1990, Figure 3]. The critical design parameter may not be average throughput shown in Figure 4–17, however, but the loss probability P_L. For small values of P_L, the approximate solution may not be accurate enough and the more sophisticated analyses may be required. This point is pursued later, at the end of the next section.

We now proceed with two more sophisticated, albeit relatively simple, analyses of the leaky bucket algorithm. The first is a Markov chain discrete-time analysis due to Sidi et al. [SIDI 1989]. The second, described in the next section, uses the fluid-flow approximation generalized to finite buffers and ap-

[SIDI 1989] Sidi, M., et al., "Congestion Control Through Input Rate Regulation," *Proc. IEEE Globecom 89,* Dallas, TX, Nov. 1989: 1764–1768.

[BERG 1990] Berger, A. W., "Performance Analysis of a Rate Control Throttle Where Tokens and Jobs Queue," *Proc. IEEE Infocom 90,* San Francisco (June 1990): 30–38.

[HEYM 1992] Heyman, D. F., "A Performance Model of the Credit Manager Algorithm," *Computer Networks and ISDN Systems, 24* (1992): 81–91.

[BUTT 1991] Butto, M., et al., "Effectiveness of the 'Leaky Bucket' Policing Mechanism in ATM Networks," *IEEE JSAC, 9,* 3 (April 1991): 335–342.

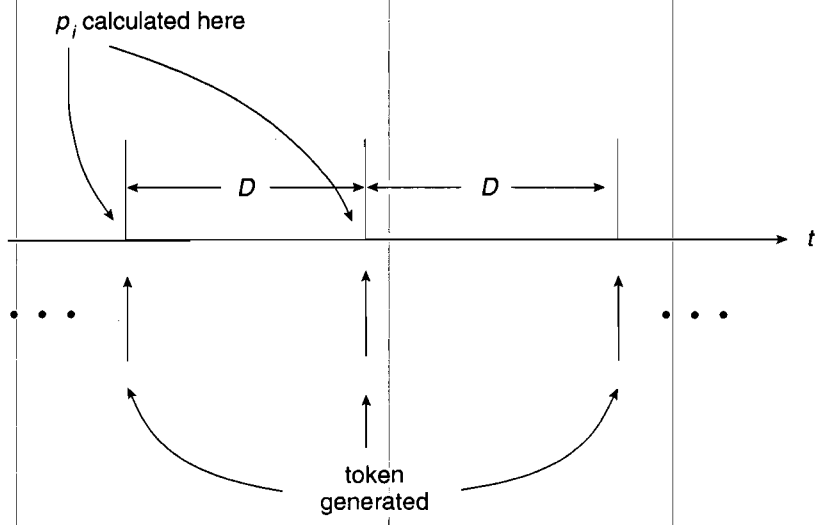


FIGURE 4–18 ■ Discrete-time calculation, leaky bucket algorithm.

propriate to bursty on-off sources or Markov-modulated models of traffic [ELWA 1991].

The Markov chain discrete-time analysis proceeds as follows. Let i be the number of tokens in the token buffer pool of Figure 4–15. We shall calculate the steady-state data throughput λ^* by first calculating the steady-state probability p_i that i tokens are in the pool just *before* a token is generated. Recall that tokens are generated periodically every D sec. Token generation times are thus spaced D sec. apart, as shown in Figure 4–18. Let α_j be the probability j cells (packets) arrive in any D sec interval. Arrivals are assumed independent from interval to interval. (The Markov chain model arises from this assumption). As an example, if the arrivals are Poisson with average rate λ,

$$a_j = (\lambda D)^j e^{-\lambda D}/j! \qquad j = 0,1,2,\ldots \tag{4–34}$$

If the arrival process is Bernoulli, with probability $a_1 = p$ that one cell arrives in a D sec interval, we have

$$a_1 = p \qquad a_0 = 1 - p \tag{4–35}$$

Other models can be invoked as well. But note that the on-off traffic models discussed previously are not captured easily with this analysis.

[ELWA 1991] Elwalid, A. I., and D. Mitra, "Stochastic Fluid Models in the Analysis of Access Regulation in High Speed Networks." *Proc. IEEE Globecom 91,* Phoenix, Dec. 1991: 1626–1632. (A more detailed version appears in *Communication Systems,* Special issue of *Queueing Systems,* D. Mitra and I. Mitrani (eds.), *9* (1991). Switz.: J. C. Baltzer A. G., Basel, 29–63.)

In the steady state one expects the probability p_i that there are i tokens in the token buffer just before a token arrival (Figure 4–18) to be the same in any D sec interval. We can thus equate the probability p_i at the end of one interval to the sum of the probabilities of the various states the token buffer could have been in, exactly one interval earlier. This gives rise to a set of discrete-time balance equations. As an example, consider the state $i = M$; that is, the state for which the token buffer is full. For the buffer to be in this state just before a possible token arrival, there can only have been two possibilities one interval earlier: the buffer was in state $(M - 1)$, a token was added and *no* cells arrived during the entire interval, *or* the buffer was in state M, no token was added and again *no* arrivals took place. We must thus have as the discrete-time probability balance equation for this case,

$$p_M = (p_{M-1} + p_M)a_0 \tag{4–36}$$

Consider state $M - 1$ now. Repeating the same kind of argument, it is left for the reader to show that the balance equiation is now given by

$$p_{M-1} = p_{M-2}a_0 + p_{M-1}a_1 + p_M a_1 \tag{4–37}$$

More generally, allowing any number j of data cells to arrive in an interval, with its corresponding probability of arrival a_j, the probability p_i the system is in state i just before a token arrival is given by the set of balance equations

$$p_i = \sum_{j=i}^{M} p_{j-1}a_{j-i} + p_M a_{M-i} \qquad 1 \le i \le M \tag{4–38}$$

and

$$p_0 = \sum_{j=0}^{M-1} p_j \bar{a}_j + p_M \bar{a}_{M-1} \tag{4–39}$$

Here

$$\bar{a}_j \equiv 1 - \sum_{k=0}^{j} a_k = \sum_{k=j+1}^{\infty} a_k \tag{4–40}$$

It is again left for the reader to verify these equations.

One can now solve these equations recursively for p_i, and from this, as we shall show shortly, one obtains the desired data throughput as a function (implicitly) of D, M, and, of course, the traffic arrival statistics a_j. The recursive technique proceeds as follows:

Choose some value of $p_M < 1$. Then from equation (4–36), we have

$$p_{M-1} = p_M(1 - a_0)/a_0 \tag{4–41}$$

From equation (4–37), one then finds

$$p_{M-2} = \{[p_{M-1}(1 - a_1)] - p_M a_1\}/a_0 \tag{4–42}$$

Proceeding in this manner, we have, recursively,

$$p_i = \left[p_{i+1} - \sum_{j=1}^{M-1-i} p_{i+j} a_j - p_M a_{M-1-i} \right] / a_0, \qquad 0 \le i \le M-2 \tag{4–43}$$

But recall that the value of p_M was arbitrarily chosen. We now collect all the p_i's generated and calculate the sum

$$S = \sum_{i=0}^{M} p_i \tag{4–44}$$

Dividing all the p_i's through by S to obtain the desired probability normalization, we have, finally,

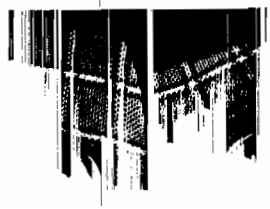
$$p_i = p_i/S \qquad 0 \le i \le M \tag{4–45}$$

Given p_i, the probability the token buffer of Figure 4–15 is in state i, how does one now determine the data throughput from this? Recall that to be transmitted, each cell (packet), on arrival, must find a token waiting; otherwise it is dropped. Say the token buffer is in state M. Then at most, M cells can be transmitted. (No token can be added since the token buffer state has been defined just prior to a possible token arrival. This arrival is blocked if the buffer is full). Exactly M cells will be transmitted if M or more cells arrive in the interval D sec long. If fewer arrive they will all be transmitted. Consider a token buffer state $k \le (M - 1)$ now. Then the number of cells that can be transmitted in one token generation interval varies from 1, with arrival probability a_1, to at most $(k + 1)$, since an additional token will be added during that interval. (The buffer is no longer full). Putting this all together, we get as the throughput $\lambda^* D$, the average number of cells transmitted in a D sec interval,

$$\lambda^* D = p_M \left[\sum_{i=0}^{M} i a_i + M \sum_{i=M+1}^{\infty} a_i \right] + \sum_{k=0}^{M-1} p_k \left[\sum_{i=1}^{k+1} i a_i + (k+1) \sum_{i=k+2}^{\infty} a_i \right] \tag{4–46}$$

This is comparable to the normalized load λ^*/r given by equation (4–33a) for the earlier approximate analysis.

Consider some examples. Take $M = 1$ first. One token at a time only is allowed in the token buffer. Hence individual data cells only may be trans-



mitted. If more than one arrives in the interval D, one will be transmitted and the others dropped. From equations (4–38) to (4–40), or (4–41) to (4–45), we find that

$$p_0 = 1 - a_0 \qquad p_1 = a_0 \qquad M = 1 \tag{4–47}$$

From equation (4–46) we find the throughput to be given by

$$\lambda^* D = (1 - a_0)(p_0 + p_1) = 1 - a_0 \qquad M = 1 \tag{4–48}$$

For the case of Poisson arrivals [equation (4–34)], one then has

$$\lambda^* D = 1 - e^{-\lambda D} = 1 - e^{-\rho} \qquad M = 1 \tag{4–48a}$$

with $\rho \equiv \lambda D$.

This is sketched, as a function of the normalized load $\rho = \lambda D = \lambda / r$, in Figure 4–19. Also shown for comparison is the comparable curve found using the approximate analysis, as previously plotted in Figure 4–17. Note that the earlier curve provides a conservative estimate of the throughput. This is found to be true in general.

The second simple example is that for $M = 2$. One finds here, using the equations derived above, that

$$\begin{aligned} p_0 &= (1 - a_1 - a_0)/(1 - a_1) \quad p_1 = a_0(1 - a_0)/(1 - a_1) \\ p_2 &= a_0^2/(1 - a_1) \qquad M = 2 \end{aligned} \tag{4–49}$$

Using these values of the token buffer probabilities to calculate the normalized throughput from equation (4–46), it is easy to show that

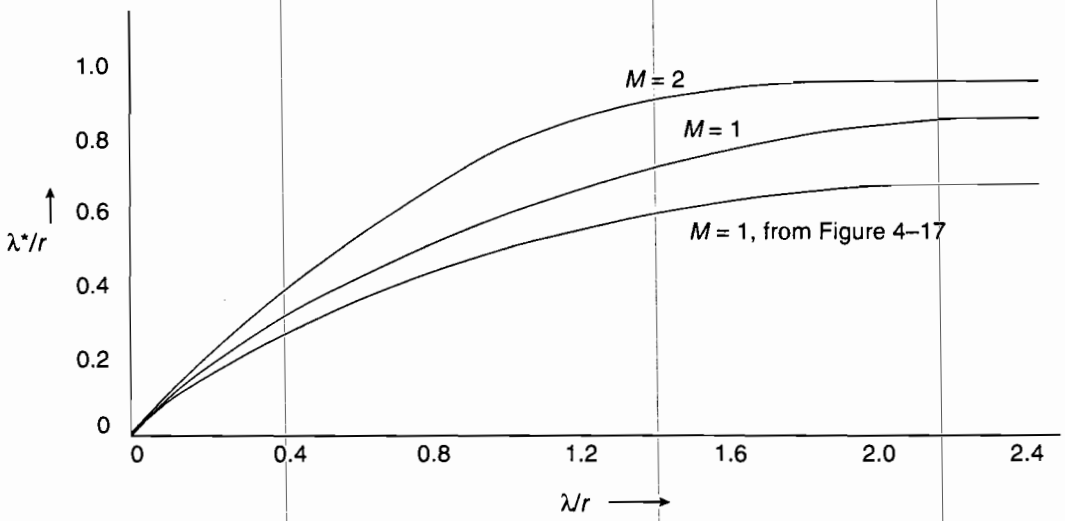


FIGURE 4–19 ■ Leaky bucket throughput characteristic.

$$\lambda^* D = \lambda^*/r = 1 - \frac{a_0^2}{(1-a_1)} \qquad M = 2 \tag{4–50}$$

Taking Poisson arrivals again as an example,

$$\lambda^*/r = 1 - \frac{e^{-2\rho}}{(1-\rho e^{-\rho})} \qquad \rho \equiv \lambda D \qquad M = 2 \tag{4–50a}$$

The curve for this case is also plotted in Figure 4–19. By comparison with Figure 4–17, one can again see that the approximate analysis provides a conservative estimate of throughput here.

Although we have focused on the average throughput in the analysis above, since that is one useful measure of performance controlled by the leaky bucket technique, the loss probability is of particular interest. Recall that this was the measure of QoS on which we focused in the previous section on admission control. Ideally, as noted in discussing the form of the throughput-load curve sketched in Figure 4–14, one would like a control to track the load exactly until the maximum average throughput r is reached, and then saturate at that value. This implies zero loss probability for $\lambda \leq r$, and then the value, from equation (4–31), required to have $\lambda^* = r$ beyond that point. Specifically, the ideal $P_L - \lambda$ curve would have the form

$$\begin{aligned} P_L &= 0 \qquad \lambda/r \leq 1 \\ P_L &= 1 - 1/(\lambda/r) \qquad \lambda/r \geq 1 \end{aligned} \tag{4–51}$$

In normalized form, letting $\rho \equiv \lambda/r$, we can also write

$$P_L = 1 - \frac{1}{\rho} \qquad \rho \geq 1 \tag{4–51a}$$

How well does the leaky bucket mechanism perform compared to this ideal characteristic? The performance using the approximate model of Figure 4–16 is readily obtained from equation (4–32), the explicit expression for the loss probability for this model. Note first, as expected, that for large token pool size M, the loss probability is small for $\rho < 1$. At $\rho = 1$, the reader can readily ascertain that

$$P_L = \frac{1}{(M+1)} \qquad \rho = 1$$

while for $\rho >> 1$,

$$P_L \to 1 - \frac{1}{\rho} \qquad \rho >> 1$$

as desired for the ideal characteristic. By increasing the token pool size M, then, one can approach the ideal characteristic more and more, but again at the cost of slower response time of the control. This is expected of course, since we noted the same behavior with the throughput characteristic.

The loss probability performance using the discrete-time model of [SIDI 1989] can also be obtained quite readily from the throughput equation of (4–46). Using this together with equation (4–31), one can calculate the loss probability directly as

$$P_L = 1 - \frac{\lambda^* D}{\lambda D} = 1 - \frac{\lambda^* D}{\rho} \tag{4–51b}$$

with ρ again defined as λD. Sidi et al have carried out this calculation for the case of Poisson arrivals [equation (4–34)] and their results are reproduced, for $\lambda D \equiv \rho < 1$, in Figure 4–20. Note that the control behaves as expected. The larger the token pool size M, the closer the control comes to the ideal control just described. But, as already noted, larger values of M result in correspondingly longer times required to detect violations of the control, however [BUTT 1991], [RATH 1991]. There is thus a tradeoff between rapidity of response and the ability to detect violations.

One approach proposed to overcome this problem is to introduce a "smoothing" buffer as part of the leaky bucket mechanism. This extended form of the leaky bucket procedure is shown schematically in Figure 4–21.

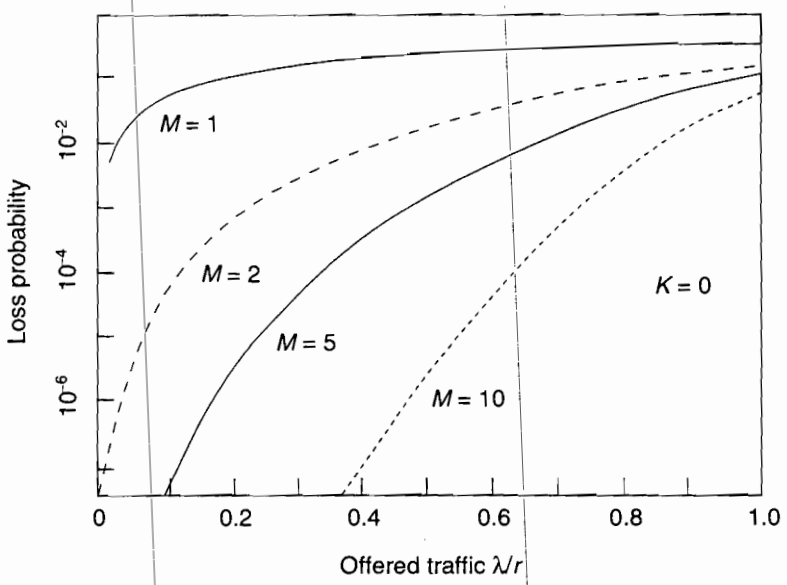


FIGURE 4–20 ■ Loss probability, leaky bucket (from [SIDI 1989], Figure 4. © 1989 IEEE).

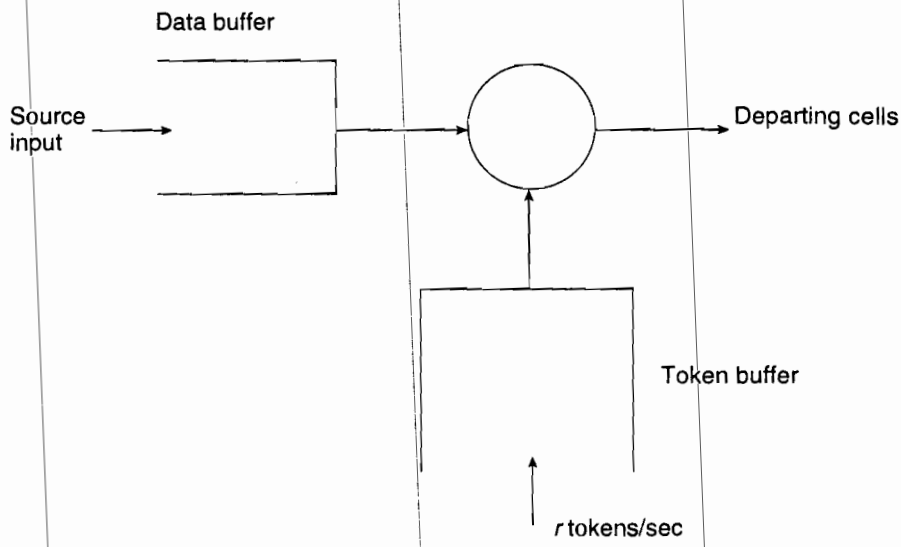


FIGURE 4–21 ■ Leaky bucket access control with data "shaping" buffer.

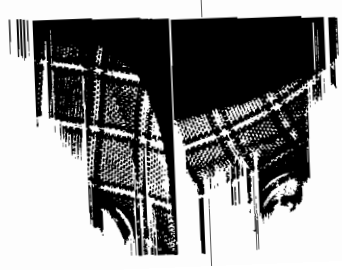
Bursts of data cells are queued with this scheme instead of being dropped if tokens are not immediately available. The traffic leaving the leaky bucket controller is thus "smoothed" before accessing the network. As will be shown in the analysis in the section following, the loss probability now turns out to depend on the *sum* of the buffers. This has been noted a number of times in the literature [SIDI 1989], [BERG 1990], [ELWA 1991]. One can use a smaller token buffer and increase the data buffer size correspondingly to attain the same loss probability performance. The price paid, however, is additional delay introduced in the data path.

4.3 ■ LEAKY BUCKET WITH DATA BUFFERING–FLUID ANALYSIS

We concluded the last section by noting the leaky bucket can be extended by adding a data buffer. This serves to buffer and smooth out bursts of cells arriving, enabling the control to be exerted more quickly with a smaller value of token buffer (or maximum counter value) required. The price paid of course is some delay introduced in buffering incoming cells.

We now quantify these remarks by carrying out an analysis of the extended leaky bucket mechanism of Figure 4–21. We use a fluid model approach for this purpose, repeating, and extending, the analysis of section 3.8.

Figure 4–22 portrays a fluid analysis version of the extended leaky bucket mechanism of Figure 4–21. The traffic source is, most generally, represented by an N-state Markov-modulated fluid source such as that introduced



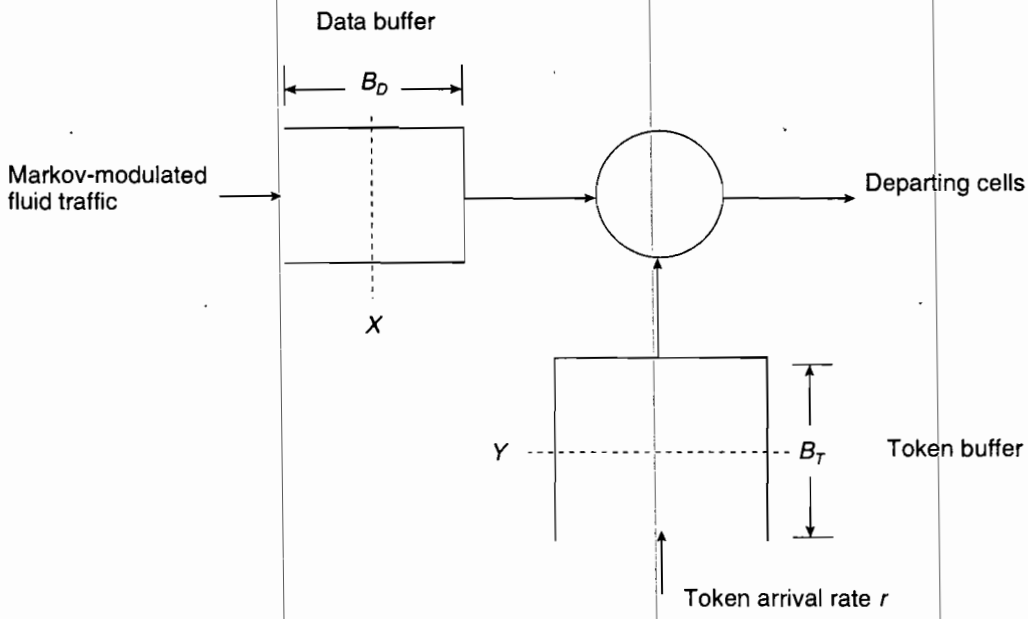


FIGURE 4–22 ■ Fluid-flow analysis, leaky bucket of Figure 4–21.

in section 3.8. The traffic model, equivalent to that presented in Figure 3–39, is shown schematically in Figure 4–23. We repeat some of the analysis of section 3.8 for completeness here. Transitions between states are governed by an underlying Markov chain, only a few of which are shown in Figure 4–23. This chain is again represented by an $N \times N$ infinitesimal generating matrix M, defined (as it was previously in Chapter 3 by equation (3–47)) by

$$M = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1j} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2j} & \cdots & \mu_{2N} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \mu_{i1} & \mu_{i2} & \cdots & \mu_{ij} & \cdots & \mu_{iN} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \mu_{N1} & \mu_{N2} & \cdots & \mu_{Nj} & \cdots & \mu_{NN} \end{bmatrix} \tag{4–52}$$

(Compare with the $B \times B$ matrix of equation (3–134) as well.) The diagonal rate parameter μ_{ii} is again given as

$$\mu_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{N} \mu_{ij} \qquad 1 \leq i \leq N \tag{4–53}$$

Row sums are thus all zero. The steady-state probability π_i that the Markov chain is in state i is then again given by solving the matrix equation

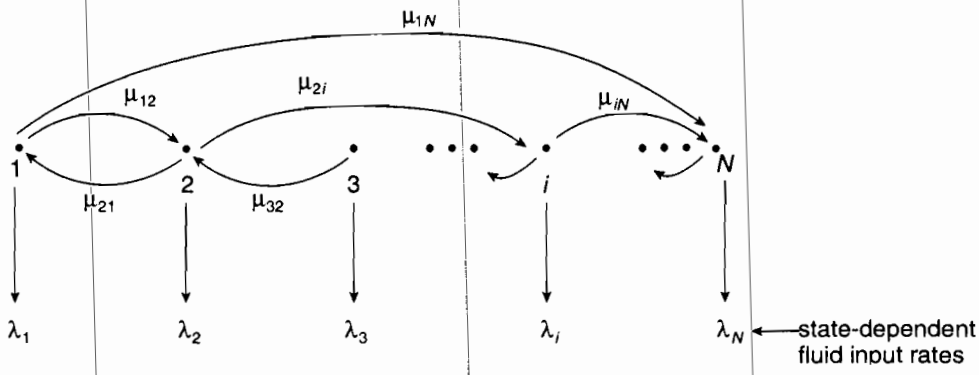


FIGURE 4–23 ■ Markov-modulated fluid source model (fluid analog of MMPP, Figure 3–11).

$$\boldsymbol{\pi} M = 0 \tag{4–54}$$

with

$$\boldsymbol{\pi} \equiv [\pi_1, \pi_2, \ldots \pi_i, \ldots, \pi_N] \tag{4–55}$$

the row vector of the N steady-state probabilities. (A separate normalization condition $\sum_{i=1}^{N} \pi_i = 1$ is needed to ensure the π_i's represent probabilities).

As in Figure 3–39, the input traffic rates shown in Figure 4–23 are constant flows, as appropriate to a stochastic fluid source. As a special case, the one on which we shall focus later, let the traffic source be modeled as the simple on-off source of Figure 4–1. This is reproduced here as Figure 4–24. Then, for this case,

$$M = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix} \tag{4–56}$$

and

$$\boldsymbol{\pi} = \left[\frac{\beta}{\alpha+\beta}, \frac{\alpha}{\alpha+\beta} \right] \tag{4–57}$$

In this model $\lambda_1 = 0$, $\lambda_2 = R_p$. We use the same notation used previously in Figure 4–1 to preserve continuity.

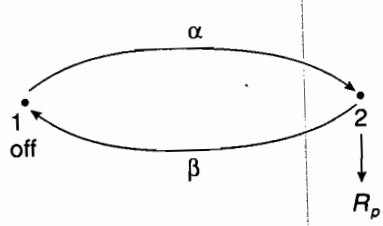


FIGURE 4–24 ■ Simple on-off fluid source.

Returning to Figure 4–22 (the model of the leaky bucket algorithm augmented with a finite data smoothing buffer), note that we have changed the notation somewhat from that used previously. The maximum token pool is now given as B_T tokens rather than the symbol M used previously. The finite data buffers shown can accommodate, at most, B_D data cells. The objective is now to find the steady-state occupancy statistics of the two buffers, X and Y for the data and token buffers respectively, from which performance measures of interest can be found.

It turns out, as noted earlier, that the performance of the extended leaky bucket algorithm depends only on the *sum* of the buffer sizes shown, $B = B_T + B_D$. (Note again that the token buffer is only a model representation. The data buffer, on the other hand, is a real buffer.) This is due to the fact that the two random variables X and Y are related to one another. In fact, as shown below, one doesn't find the statistics of X and Y separately, but, rather, jointly, by defining an equivalent "virtual" queue representation from which the statistics of either one can be found [ELWA 1991]. To demonstrate this, note two facts from the leaky bucket technique:

1. The data buffer can be occupied only if the token queue is empty; otherwise the data cells would be transmitted, one per token. Hence we must have

$$Y = 0, \qquad X > 0 \tag{4–58}$$

2. Conversely, the token buffer can be occupied only if the data buffer is empty. If data cells were queued, they would each capture a token and be transmitted. Hence we also have

$$X = 0, \qquad Y > 0 \tag{4–59}$$

Conditions of equations (4–58) and (4–59) are both encapsulated in the single condition

$$XY = 0 \tag{4–60}$$

Following the approach of Elwalid and Mitra [ELWA 1991], we now define the following single "virtual buffer" random variable

$$W \equiv X - Y + B_T \tag{4–61}$$

Note that because of the conditions on the random variables X and Y,

$$0 \le X \le B_D, \qquad 0 \le Y \le B_T, \qquad XY = 0$$

W has the following properties:

1. $$0 \le W \le B_T + B_D = B \tag{4–62}$$
2. In the range $0 \le W \le B_T$,

$$X = 0, \qquad W = B_T - Y \tag{4–63}$$

3. In the range $B_T \le W \le B = B_T + B_D$,

$$Y = 0, \qquad W = X + B_T \tag{4–64}$$

Since X and Y appear at disjoint ranges of W, one need only find the statistics of W. From these, one can find the statistics of X and Y, using equations (4–64) and (4–63) respectively. The range of W and its relation to Y are diagrammed in Figure 4–25.

Because of the stochastic fluid process assumed driving the data buffer (Figure 4–22), X and Y, and hence W, are all continuous r.v.'s. To find the statistics of W, and from these, those of X and Y, if desired, we proceed as previously in our fluid-flow analysis. We dispense here with the initial incremental time analysis used to find the equivalent probability distribution in Chapter 3 (see equation (3–13) for example), and proceed directly to the steady-state analysis as was done in section 3.8. Thus define the joint probability

$$F_i(x) \equiv \text{Prob. } [W \le x, S = i] \tag{4–65}$$

with $S = i$ the state of the input source Markov chain, $1 \le i \le N$. Note from the definition of W that this implies

$$F_i(B) = \pi_i \tag{4–66}$$

Proceeding as was done in Chapter 3, one then shows that the desired solution for $F_i(x)$, $1 \le i \le N$, is given by the solutions to our familiar set of linear differential equations now specialized to this problem,

$$(\lambda_i - r)\frac{dF_i(x)}{dx} = \sum_{j=1}^{N} \mu_{ji} F_j(x) \qquad 1 \le i \le N \tag{4–67}$$

Here μ_{ij} is the transition rate between states i and j of the underlying N-state Markov chain (Figure 4–23), λ_i is the input fluid rate at state i, and

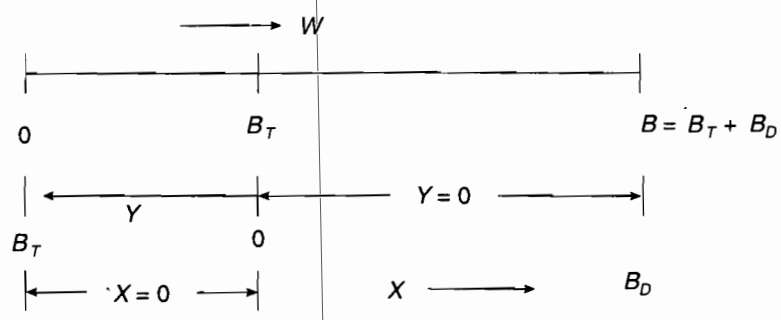


FIGURE 4–25 ■ "Virtual buffer" variable W.

$\mu_{ii} = -\sum_{j=1, j\neq 1}^{N} \mu_{ij}$. This is to be compared with equation (3–138). Details are left to the reader.

More compactly, in vector form, we have our familiar equation

$$\frac{d\boldsymbol{F}(x)}{dx} D = \boldsymbol{F}(x)M \tag{4–68}$$

with

$$\boldsymbol{F}(x) \equiv [F_1(x), F_2(x), \ldots, F_N(X)] \tag{4–69}$$

M the $N \times N$ matrix of equation (4–52), and

$$D \equiv \text{diag. } [\lambda_i - r] \tag{4–70}$$

(Compare with equation (3–139) to equation (3–141). Note that the token arrival rate r, the counter increment (or decrement) rate in the leaky bucket procedure, takes the place of the capacity C_L or μ in our earlier work using fluid analysis. As noted much earlier in Chapter 3, we clearly must have $\lambda_i \neq r$, all i. The solution to equation (4–68) is again given by sums of exponentials:

$$\boldsymbol{F}(x) = \sum_{j=1}^{N} a_j \boldsymbol{\Phi}_j e^{z_j x} \tag{4–71}$$

with $(z_j, \boldsymbol{\Phi}_j)$ the (eigenvalue, eigenvector) pair satisfying the eigenvalue equation

$$z_j \boldsymbol{\Phi}_j D = \boldsymbol{\Phi}_j M \qquad 1 \leq j \leq N \tag{4–72}$$

The a_j's, $1 \leq j \leq N$, are again constants to be determined by invoking N boundary conditions.

Note a basic difference beween this solution and that found earlier using the fluid source process: Here the "buffer" occupancy r.v. W is *bounded* ($0 \leq W \leq B = B_T + B_D$), whereas previously we had always assumed the buffer was infinite. The variable x (the continuous-valued state of W), is thus finite as well. Hence we must retain *all* the eigenvalues, positive as well as negative, in the solution of equation (4–71). This is where the analysis here deviates from that of section 3.8. As previously, one eigenvector is known. Since $\boldsymbol{\pi} M = 0$, one eigenvalue of equation (4–72) must again be zero. Calling this eigenvalue z_1, its associated eigenvector $\boldsymbol{\Phi}_1 = \boldsymbol{\pi}$. Hence equation (4–71) can be simplified to

$$\boldsymbol{F}(x) = a_1 \boldsymbol{\pi} + \sum_{j=2}^{N} a_j \boldsymbol{\Phi}_j e^{z_j x} \tag{4–71a}$$

We now need N boundary conditions to find the unknown constants a_j, $1 \leq j \leq N$. (Why is a_1 not equal to 1, as in our earlier fluid analysis?)

To establish these, we note that, just as in our discussion of statistical multiplexing in Chapter 3, as well as our admission control analysis earlier, some of the Markov chain states must be underload or "emptying" states; the others must be overload or "filling" states. (What is "filling" or "emptying" here?) Specifically, with r taking the place of our link capacity C_L in the previous discussions, there must exist some Markov states i for which

$$\lambda_i - r < 0\text{: emptying states;}$$

and some i for which

$$\lambda_i - r > 0\text{: filling states.}$$

(Why must this be so? What would happen if $r < \lambda_i$, all i; or $r > \lambda_i$, all i?) Hence all N states of the Markov chain modulating the source arrival rate can be divided into two disjoint sets,

$$S_F \equiv \{i \varepsilon N | \lambda_i - r > 0\}\text{: filling states}$$

and

$$S_E \equiv \{i \varepsilon N | \lambda_i - r < 0\}\text{: emptying states}$$

We have said nothing about the relative size of the λ_i's in Figure 4–23 thus far. Without loss of generality, assume that the Markov chain states are chosen such that the λ_i's monotonically increase with the state number. Hence we have

$$\lambda_1 < \lambda_2 < \lambda_3 < \ldots \lambda_l < \lambda_{l+1} < \ldots \lambda_N \qquad (4\text{–}73)$$

Let λ_l correspond to the largest source rate such that the system is in the emptying range. Then we must have $(\lambda_l - r) < 0$, $(\lambda_{l+1} - r) > 0$ and $\lambda_l < r < \lambda_{l+1}$. State l is thus equivalent to L_u in our earlier work, while $(l + 1)$ is equivalent to L_o.

Consider the l emptying states now, $i \varepsilon S_E$. Since the data arrival rate is less than the token arrival rate in these states, $\lambda_i < r$, the tokens are tending to collect. The data queue of Figure 4–22 is tending to empty ($X \to 0$) and the token queue is tending to fill ($Y \to B_T$). Hence

$$W = X - Y + B_T \to 0$$

For these states then, the probability the "virtual queue" is full tends to zero, or we have

$$\text{Prob. } [W = B, S = i] = \text{Prob. } [X = B_D, S = i] = 0 \qquad i \varepsilon S_E \qquad (4\text{–}74)$$

But note that

$$\text{Prob. } [W = B, S = i] = P[W \leq B, S = i] - P[W \leq B^-, S = i]$$

From equation (4–74) we thus have

$$P[W \leq B^-, S = i] = P[W \leq B, S = i] = \pi_i$$

In our probability distribution function notation, then,

$$F_i(B^-) = \pi_i \qquad i\varepsilon S_E \tag{4–75}$$

This provides l of the N boundary conditions.

Consider the remaining $(N - l)$ filling states, $i\varepsilon S_F$. For these states, with $\lambda_i > r$, the token buffer of Figure 4–22 tends to empty ($Y \to 0$), the data buffer tends to fill ($X \to B_D$), and

$$W = X - Y + B_T \to B_D + B_T = B$$

For these states, then, the probability that the virtual queue is empty must be zero. We thus have

$$P[W \le 0^+, S = i] = F_i(0^+) = 0 \qquad i\varepsilon S_F \tag{4–76}$$

Since $S_E \cup S_F = N$, equations (4–75) and (4–76) provide the necessary boundary conditions from which to find the N unknown constants a_i, $1 \le i \le N$, of equation (4–71a). The N equations to be solved for the N unknown a_i's may be written out in scalar form as follows:

1. $i\varepsilon S_E$:

$$F_i(B^-) = \pi_i = a_1 \pi_i + \sum_{j=2}^{N} a_j \Phi_{ji} e^{z_j B^-} \tag{4–77}$$

2. $i\varepsilon S_F$:

$$F_i(0^+) = 0 = a_1 \pi_i + \sum_{j=2}^{N} a_j \Phi_{ji} \tag{4–78}$$

(To get these equations, the jth eigenvector $\mathbf{\Phi}_j$ has been defined as $\mathbf{\Phi}_j \equiv [\Phi_{j1}, \Phi_{j2}, \ldots \Phi_{ji} \ldots \Phi_{jN}]$).

We now apply this analysis to the simplest example, the basic on-off source of Figure 4–24. This could be representative of a voice source or an image source, depending on the choice of parameters. But note that this is precisely the example used in section 3.3 in our first discussion of the fluid source model. For this example there is of course only one eigenvalue z to be found. Using the results of the earlier analysis (as a check, see equations (4–23) and (4–24) in the admission control analysis of section 4.1), we have immediately for this case

$$z = -\frac{(\alpha + \beta)}{(R_p - r)}(1 - \rho) \tag{4–79}$$

where now

$$\rho \equiv \frac{R_p}{r}\left(\frac{\alpha}{\alpha + \beta}\right) = \frac{R_p p}{r} \tag{4–80}$$

$p \equiv \alpha/(\alpha + \beta)$.
(Compare with equations (3–25a), (3–108) and (3–111), and (4–22) to (4–24)). Recall again that the token parameter r here corresponds to C_L in section 4.1. Note that the *average* load pR_p here corresponds to the average load λ (the Poisson arrival rate), on which attention was focused in the previous section. The parameter $\rho \equiv pR_p/r$ defined here then corresponds to the normalized load λ/r appearing in the previous two models. Recall that we used the symbol ρ to represent the normalized load in those models as well.

The single eigenvector $\mathbf{\Phi} = [\Phi_1, \Phi_2]$ appearing in equation (4–78) for this on-off traffic model was found as well in section 3.3. See equation (3–30) for that result. In terms of the notation used here, we have

$$\Phi_1/\Phi_2 = (R_p - r)/r = \left(\frac{R_p}{r} - 1\right) \tag{4–81}$$

Letting $\Phi_2 = 1$ arbitrarily (we could just as well have set $\Phi_2 = r$, $\Phi_1 = (\lambda - r)$), we get, for this example,

$$\boldsymbol{F}(x) = a_1\boldsymbol{\pi} + a_2\left[\left(\frac{R_p}{r} - 1\right), 1\right]e^{zx} \tag{4–82}$$

The two unknown constants a_1 and a_2 are found using the boundary conditions of equations (4–77) and (4–78). In this simple example, with $N = 2$ states only, state one with $\lambda_1 = 0$ is the emptying state and state two with $\lambda_2 = R_p$ is the filling state. We must thus have $0 < r < R_p$ for the fluid analysis to provide a stationary solution in this case. This implies that the single eigenvalue z of equation (4–79) will be negative if the parameter ρ defined by equation (4–80) is less than 1. This corresponds to the region $\lambda < r$ in Figures 4–14, 4–17, and 4–19, since the λ there is the average (Poisson) arrival rate, comparable to ρ here, as noted above.

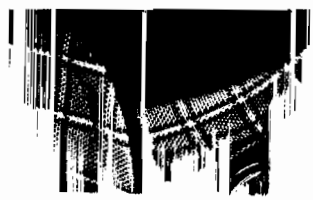
From equation (4–77) we have, using equation (4–82),

$$F_1(B^-) = \pi_1 = a_1\pi_1 + a_2\left(\frac{R_p}{r} - 1\right)e^{zB^-} \tag{4–83}$$

with $\pi_1 = (1 - p) = \beta/(\alpha + \beta)$.
Similarly, from equations (4–78) and (4–82) we have

$$F_2(0^+) = 0 = a_1\pi_2 + a_2 \tag{4–84}$$

Here $\pi_2 = p = \alpha/(\alpha + \beta)$.
Note that the parameter $B = B_T + B_D$, the *sum* of the two buffer sizes, enters naturally here, as noted earlier. It is this parameter that will play a role in determining the throughput and loss probability performance of the extended leaky bucket procedure. Solving equations (4–83) and (4–84) simultaneously for a_1 and a_2, we get



$$a_1 = 1/\left[1-\frac{\alpha}{\beta}\left(\frac{R_p}{r}-1\right)e^{zB}\right] \tag{4–85}$$

and

$$a_2 = -\pi_2 a_1 = -a_1\, a/(a+\beta) \tag{4–86}$$

(Note that we have dropped the – from B^- since it is no longer needed here.) Using equations (4–85) and (4–86) in equation (4–82), and following the notation of [ELWA 1991], we finally get for the two probability distribution functions in this case of a single on-off source,

$$F_1(x) = \pi_1 \Delta(x)/\Delta(B) \tag{4–87}$$

and

$$F_2(x) = \pi_2(1 - e^{zx})/\Delta(B) \tag{4–88}$$

with

$$\Delta(x) \equiv 1-\frac{\alpha}{\beta}\left(\frac{R_p}{r}-1\right)e^{zx} \tag{4–89}$$

$\pi_1 = \beta/(\alpha+\beta)$, $\pi_2 = 1 - \pi_1 = \alpha/(\alpha+\beta)$, and z given by equation (4–79).

All the performance parameters of interest can be obtained from $F_1(x)$ and $F_2(x)$. We shall obtain these after first developing performance parameters for the general N-state Markov chain model.

Consider the average throughput λ^*, in cells/sec, first. This can be calculated in a number of equivalent ways. One way is to say that this is the average load, less the cell loss when the data buffer is full. This may be written as

$$\lambda^* = \sum_{i=1}^{N} \lambda_i \pi_i - \sum_{i=1}^{N} (\lambda_i - r) P[X = B_D,\ S = i] \tag{4–90}$$

The first term on the right-hand side is of course the average load, averaged over all N states. The second term is the average cell loss, also averaged over the N states. But note that the $(N - l)$ filling (overload) states only will be included in the sum since we have already seen that in the emptying (underload) states the data buffer cannot be full and cells cannot be lost [see equation (4–74)]. The probability $P[X = B_D,\ S = i]$ that the data buffer is full with the system in overload state i, is readily calculated as

$$\begin{aligned} P[X = B_D, S = i] &= P[W = B,\ S = i] \\ &= P[W \le B,\ S = i] - P[W \le B^-,\ S = i] \\ &= \pi_i - F_i(B^-) \end{aligned} \tag{4–91}$$

This is similar to the analysis following equation (4–74) above.

We thus have, from equation (4–90),

$$\lambda^* = \sum_{i=1}^{N} \lambda_i \pi_i - \sum_{i=1}^{N} (\lambda_i - r) \quad [\pi_i = F_i(B^-)] \tag{4–90a}$$

This can be rewritten in somewhat different form by expanding the second sum and noting that the first sum then cancels:

$$\lambda^* = r + \sum_{i=1}^{N} F_i(B^-)(\lambda_i - r) \tag{4–90b}$$

An alternate approach is to focus on average token throughput. Since every cell requires a token for transmission, the average token throughput must equal the average cell throughput. We thus have, immediately,

$$\lambda^* = r - \sum_{i=1}^{N} (r - \lambda_i) F_i(0) \tag{4–90c}$$

The second term here represents a mean token "loss" due to a full token buffer, since

$$P[Y = B_T, S = i] = P[W = 0, S = i]$$

(See equation (4–63) or Figure 4–25.) This implies, comparing equations (4–90b) and (4–90c), that

$$\sum_{i=1}^{N} (\lambda_i - r) F_i(B^-) = \sum_{i=1}^{N} (\lambda_i - r) F_i(0) \tag{4–92}$$

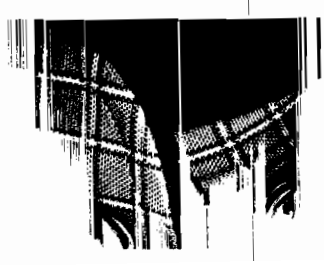
More generally, Mitra has shown [MITR 1988] that the following identity holds:

$$\sum_{i=1}^{N} F_i(x)(\lambda_i - r) = \text{constant}, \quad \text{all } X \tag{4–93}$$

Consider the on-off source of Figure 4–24 as an example. Recall again that $\lambda_1 = 0$, $\lambda_2 = R_p$, $\rho \equiv R_p p / r$, $p = \alpha/(\alpha + \beta)$ in this case. Then it is readily shown, using equations (4–87) and (4–88) in either (4–90b) or (4–90c), that the normalized throughput for this example is given by

---

[MITR 1988] Mitra, D., "Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer," *Adv. Appl. Prob., 20* (1988): 646–676.



$$\lambda^*/r = 1 - (1 - \rho)/\Delta(B) \tag{4–94}$$

The term $\Delta(B)$ is again found from equation (4–89). It is also left as an exercise for the reader to demonstrate identity (4–93) in this case.

The cell loss probability P_L is easily obtained in this case from equation (4–94), using equation (4–31):

$$P_L = 1 - \lambda^*/\sum_{i=1}^{N} \lambda_i \pi_i = 1 - \frac{\lambda^*/r}{\rho} \qquad \rho \equiv \frac{R_p p}{r} \tag{4–95}$$

Note that this is identical to equation (4–51b), as it should be. As already noted, pR_p in the on-off case and more generally $\sum_{i=1}^{N} \lambda_i \pi_i$ for the N-state Markov chain modulating traffic model, correspond to the average Poisson arrival traffic rate λ in the models discussed previously.

How can we use these results for the on-off case for design purposes? Consider the range $\rho = pR_p/r < 1$ first. We must also ensure that $R_p > r$; that is, that there exists a filling or overload region of operation. Calculations can thus only be carried out for $\rho > p$. Recall that this is the range of operation in which we would like the loss probability to be very small. From equations (4–94) and (4–95), using equation (4–89) to evaluate Δ(B), we then get

$$P_L \doteq \left(\frac{1}{\rho} - 1\right)[1 - \Delta(B)] = \left(\frac{1}{\rho} - 1\right)\frac{\alpha}{\beta}\left(\frac{R_p}{r} - 1\right)e^{zB} << 1 \tag{4–96}$$

Since $z < 0$ in this range [See equation (4–79)], the loss probability decreases exponentially with $B = B_T + B_D$. We take two examples now.

Example 1 [ELWA 1991]:

This is a voice-like source. Let $1/\beta = 1$ sec, $1/\alpha = 1.86$ sec, $p = 0.35$. Take $\rho = 0.7$. Then $R_p/r = 2$. From equation (4–96) it is then readily found that

$$\log_{10} P_L = -0.38\left(\frac{B}{R_p}\right) - 0.636 \qquad \rho = 0.7 \tag{4–97}$$

For voice we have $R_p = 170$ cells/sec when in talk spurt (see Chapter 3). Table 4–1 shows the values of B required for various values of P_L. Note that doubling the value of B changes P_L by two magnitudes in this case. As the value of ρ is increased, approaching $\rho = 1$, correspondingly larger values of B are required.

Example 2: A very bursty image source [BUTT 1991].

Let the average burst interval $1/\beta = 0.5$ sec, while the average "off" time is $1/\alpha = 11$ sec. Then $p = \alpha/(\alpha + \beta) = 0.0435$. Let the image transfer rate during the burst interval be $R_p = 5200$ cells/sec. (Note that this is then 30 times the rate required for a voice source). An average image thus consists of

TABLE 4–1 ■ Leaky Bucket Control, Voice-like Source, 170 cells/sec, average talk spurt = 1 sec, average silent interval = 1.86 sec, $\rho = 0.7$, $r = 85$

P_L	B (cells)
10^{-9}	3550
10^{-7}	2700
10^{-5}	1850
10^{-3}	1000

2600 cells, or 114,400 octets (915,200 bits), if 44 octets of payload appear in each cell, as proposed for the class 3/4 ATM Adaptation Layer. (The actual ATM transfer rate is 2.2 Mbps, using 53-octet ATM cells).

Let us (naively, it will appear) choose $\rho = 0.7$ as the nominal operating point. Since $\rho = pR_p/r$, this implies the token (or counter) increment rate is $r = R_p/16.1 = 323$ tokens/sec. This appears to be a reasonable number. Using equation (4–96), we then calculate the "buffer" size B required for a given value of cell loss probability P_L. For images, this number is usually much smaller than the value allowed for voice. Say, specifically, that we require $P_L = 10^{-9}$ at the nominal operating point. For the number chosen here we then find $B \doteq 152{,}000$ cells! This is clearly an exceptionally large number, even if B can be split between the data buffer of size B_D and the token pool of size B_T. The problem is that with bursty traffic, with a relatively high peak rate when transmitting, one needs a much larger token (or counter) increment rate. This requires increasing r considerably, or, equivalently, reducing the operating value of ρ.

[BUTT 1991] have developed some design rules for choosing the two leaky bucket parameters r and B. They consider a number of control scenarios involving two types of traffic—voice and the bursty, image-like source, we have been discussing. It turns out the proper choice of the two parameters depends on what the leaky bucket technique is designed to control and how tight a control is required. If, for example, it is the peak data transfer rate R_p that is to be controlled, then it turns out the token increment rate r should be close to R_p. This implies operating at a value of $\rho \sim p \ll 1$, *not* the relatively high value of $\rho = 0.7$ chosen above. Increasing the rate r brings the value of B required down considerably.

The design rule they propose for this case of packet rate control is the following: Choose the loss probability P_L required at the normal operating point. As an example, let it be $P_L = 10^{-9}$. Choose a fractional increase in the data transfer rate beyond which the system should not go. The loss probability should thus increase considerably beyond that point. (Note that this approach is similar in philosophy to the comment made in the previous section

that an ideal control is one that has zero loss up to a particular operating point and then clamps the data rate to a maximum average value beyond that point.) Consider, as an example, the following design rule [BUTT 1991]:

1. $P_L = 10^{-9}$ at the normal operating values of α_0, β_0, and R_{p0}.
2. $P_L \geq 0.07$ for $R_{pi} \geq 1.2\, R_{p0}$, α_0, β_0 unchanged.

(Note the tightness of control here: a 20% increase in the peak rate R_p results in more than eight magnitudes of increase in the loss probability P_L!) Using equation (4–96), we get two equations to be solved for the desired design values of r and B.

In particular, for the example under consideration here, we find

$$r \doteq 0.99\, R_{p0} = 5150 \text{ tokens/sec}$$

and

$$B \doteq 0.077\, R_{p0} = 400 \text{ cells}$$

The operating point turns out to be $\rho \doteq p$, as noted above. Note, in particular, the drastic decrease in the value of B required.

The discerning reader may have been wondering why the buffer sizes obtained here differ so much from the values used in the previous section (see Figures 4–17, 4–19, and 4–20, with the maximum value of M chosen as 10). There are two reasons: 1. The traffic source models are clearly different; 2. The controls were, implicitly, not as tight as the one chosen here. Consider, for example, Figure 4–20, taken from [SIDI 1989]. Note that the control becomes tighter as the token pool size M (comparable to B here) increases. This is of course as to be expected and agrees with the throughput-load curves of Figures 4–17 and 4–19. But note from Figure 4–20 that with $M = 10$, the loss probability increases by three orders of magnitude as the load increases by 50%. Although this is a relatively tight control, it is not as stringent as the one proposed above with P_L increasing by eight orders of magnitude for a 20% increase in the load.

To show the impact of tightening the control in a very simple manner, consider the first, very approximate leaky bucket model described in the previous section. Just as we showed that we should choose $r \sim R_p$ for bursty on-off sources, pick $r \sim \lambda$ as the desired operating point for the Poisson model. In particular, let $\rho = \lambda/r = 0.9$. Say we again want $P_L = 10^{-9}$ at this load. Then, from equation (4–32) we readily find that $M = 184$ is the token buffer size required. A 10% increase in load to $\rho = 1$ then increases the loss probability to $P_L = 1/(M+1) \doteq 0.005$ in this case, a $10^5 - 10^6$ increase in loss probability. Another 10% increase, to $\rho = 1.1$, increases P_L to $P_L \doteq 1 - 1/\rho \sim 0.1$, just the 10^8 increase in loss probability required in the on-off example. So note that one gets comparable results with these two very different models.

Say a still tighter control is desired with the Poisson approximate model. Let $\rho = \lambda/r = 0.99$ as the normal operating point, with $P_L = 10^{-9}$ desired at this point. Then, from equation (4–32), we find $M = 1600$. If the load now increases 1%, to $\rho = 1$, P_L increases to $P_L = 1/(M + 1) \sim 6 \times 10^{-4}$. If it increases 20% to $\rho = 1.2$, the loss probability increases to $P_L \doteq 1 - 1/\rho = 0.17$, again an increase of 10^8. So it appears that the tightness of control required is principally responsible for the size of the buffer required.

The discussion above related to peak rate control. Similar design rules for the leaky bucket algorithm can be adopted for the control of other parameters [BUTT 1991]. Consider, for example, the control of the average on-time interval $1/\beta$ of a bursty on-off source, or, equivalently, the average number of cells transmitted by the source at a fixed peak rate R_p. The design rule might then be to limit any increase in $1/\beta$ to some tolerable value. For increases beyond this value, the loss probability P_L should again increase substantially.

Consider the following example of this control approach. Let a bursty on-off source have an average nominal transmission interval (on-time) of $1/\beta_0 = 1$ sec, while it is nominally off for $1/\alpha_0 = 10$ sec. During the on-time, it transmits at a peak rate of $R_p = 10{,}000$ cells/sec. (This corresponds to a 4.24 Mbps transmission rate if 53-octet cells are used. The actual data transmitted in the 1-sec, average, on-time interval, using a figure of 44 octets of data per cell, is 3.52 Mbits.) Say it is desired to have $P_L = 10^{-6}$ for these normal source characteristics, but to have P_L increase to at least 10^{-4} (two orders of magnitude) if the average on-time increases to 1.5 sec. (a 50% increase). Then it is left to the reader to show that the appropriate values of r and B are $r = 5700$ tokens/sec and $B = 60{,}000$ cells, respectively.

We have thus far described the performance of the leaky bucket technique with a shaping or smoothing data buffer (Figure 4–21), independently of the size of the data buffer. All the results, thus far, have been shown to depend on the *total* "buffer" size $B = B_T + B_D$. Recall that B_T is not really a buffer but the token pool size in the token model (Figure 4–21), or the maximum counter size in the leaky bucket counter implementation. Given the proper choice of B and r discussed above, how does one now decide on the distribution between the actual data buffer B_D and the token pool (or maximum counter value) B_T? This is where the "smoothing" or "shaping" introduced by the data buffer, as noted earlier, comes into play.

Consider the extreme case in which there is no token buffer (or, equivalently, no counter) for the leaky bucket. All buffering is then done by the data buffer. As a burst of up to B cells arrives at the leaky bucket controller, it is stored in the data buffer, to be read out at the fixed rate of r cells/sec. The "burstiness" of the source has then been "absorbed" by the leaky bucket data buffer, and the output process is much more regular. In essence, the introduction of the data buffer has reduced the variation in time of the input (traffic) process. This is measured by the coefficient of variation c_v^2, defined to be the ratio of the variance to the first moment (mean) squared. A deterministic

process has $c_v^2 = 0$, c_v^2 for a Poisson process is 1, and a bursty on-off process has $c_v^2 = \sigma^2/m^2 = p(1-p)/p^2 = (1-p)/p$, with p again defined as the probability the source is in its on-state.

Elwalid and Mitra [ELWA 1991] have calculated the first and second moments of the leaky bucket output process, from which the output coefficient of variation is readily obtained. The first moment for the on-off bursty source considered in detail in this section is of course the average throughput λ^* as shown to be given earlier by equation (4–94). This depends on the *total* buffer size $B = B_T + B_D$, as noted a number of times. The second moment is found, however, to depend on *both* B_D and B_T. In particular, with $B = B_T + B_D$ kept fixed, the output coefficient of variation is found to *decrease* with increasing data buffer size B_D. Similar results are obtained for more complex sources modeled as three-state Markov chains [ELWA 1991, detailed version]. These results validate the intuitive comment made above that introducing a data buffer "smooths" the output process, reducing its variation.

Consider Example 1, the voice-like source model mentioned briefly earlier. Recall that this had the following parameters: $R_p = 170$ cells/sec, $1/\beta = 1$ sec, $1/\alpha = 1.86$ sec, $p = 0.35$, $\rho = 0.7$, and hence $r = 85$ tokens/sec. For this example, the input coefficient of variation $c_v^2 = (1-p)/p = 1.86$. Elwalid and Mitra have plotted the output coefficient of variation as a function of the token pool size B_T (hence B_D as well) for $P_L = 2.26 \times 10^{-5}$ and $B = 1700$ cells.[1] For $B_T = 0$ (that is, $B_D = 1700$ cells), the output coefficient of variation is reduced from 1.86 to 0.4. The trade-off, however, is that the data traffic experiences delay. Specifically, for $B_D = 1700$ cells and a peak rate of 170 cells/sec, the maximum delay becomes 10 sec. The average delay, as calculated by Elwalid and Mitra, is 2.17 sec [ELWA 1991]. This is clearly intolerable for real-time voice with maximum end-to-end delays, source to destination, limited to 100 msec. But for certain nonreal-time data traffic with these characteristics, such delays might be acceptable.

Both the maximum and average delay decrease, of course, as B_T increases (B_D decreases), with B fixed. But the output coefficient of variation increases rapidly as well. In the example calculated by Elwalid and Mitra, B_T can be no more than 10% of B if the output c_v^2 is to be kept relatively low. For $B_T = 170$ tokens and $B_D = 1700 - 170 \doteq 1500$ cells, the output coefficient of variation is about 1 (still considerably less than its maximum value of 1.86), while the average delay reduces by about a factor of two to 1 sec [ELWA 1991].

This chapter has focused on the leaky bucket algorithm as a method of carrying out Usage Parameter Control or policing of traffic sources admitted to an ATM network. This is the technique most commonly discussed in the lit-

[1]Note that Elwalid and Mitra use the "unit of information" formulation discussed in Chapter 3. The unit of information is the number of cells transmitted in an average on-interval. For $R_p = 170$ cells/sec and $1/\beta = 1$ sec, this is just 170 cells. Their buffer sizes are all shown normalized to the unit of information. Hence $B = 10$ in this example, using their notation.

erature and is the basis of the Generic Cell Rate Algorithm proposed by the ATM Forum, as already noted. But it is clear from our discussion that not all source types lend themselves to leaky bucket control. This mechanism has two control parameters only, and although adaptive versions incorporating additional adjustable parameters have been proposed in the literature, the control of more complex traffic types may require completely different approaches to the policing or access control function. In particular, Skelly has proposed a so-called Worst-case Histogram Control method for policing VBR video traffic to ensure it conforms to its agreed-on traffic descriptors [SKEL 1992]. This method applies the video histogram representation of section 3.7 to the access control problem in this case.

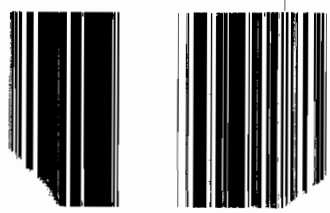
In particular, the proposal here is to supply the network admission controller with the video histogram parameters. (Recall from section 3.7 that eight such numbers were found to provide a good representation of the video signal.) In the case of stored video, such as might be available on tape, the numbers can be measured beforehand and be made available to the network during the admission control procedure. In the case of video being produced in real time, these parameters could be measured at the beginning of the filming. The control procedure than consists of continuously monitoring these parameters as the video signal is generated and transmitted into the network. A window of measurement has to be chosen, both for the admission procedure and for the subsequent control phase. Measurements are made offline of these parameters as the window is scanned across a segment of the signal and the worst-case parameter measurements are selected and compared with the agreed-on parameters. (The phrase "worst-case" here implies the largest parameters or those with the highest bit rates.)

A relatively long window provides more accurate control, but suffers from a longer delay in detecting violations of the agreed-on traffic descriptors. A short window provides more rapid detection of violations, but tends to approach peak-rate control as the window gets smaller.

Two methods of dealing with violations can be used, as noted earlier. Cells can be dropped if a violation is detected, or cells can be tagged for future dropping further into the network wherever congestion might be detected. Simulation of this technique has shown that cell dropping is more effective than cell tagging in detecting violations, and that the technique can do so successfully in a multiplexed environment without harming conforming sources [SKEL 1992].

■ PROBLEMS

4–1 a. Verify that the two probabilities P_L and ε given by equations (4–4) and (4–5), respectively, may be approximated by equations (4–4a) and (4–7).



b. Show that P_L and ε are related by equation (4–8).

c. Derive equation (4–10) for ε, using the approach suggested in equations (4–9) to (4–9b). From this, show the normalized capacity C is given by equation (4–12). For $\varepsilon = 10^{-5}$ show C is given by equations (4–14) and (4–14a).

d. Repeat **c.** using P_L in place of ε, and find the capacity expressions equivalent to equations (4–12), (4–14), and (4–14a). (Here $P_L = 10^{-5}$). How do the two sets of expressions compare?

e. Consider the case of N multiplexed on-off sources. Take $p = 0.02$ and $R_p = 4$ Mbps. Plot $C_L = CR_p$ vs. N for the three separate cases $\varepsilon = 10^{-4}, 10^{-5}, 10^{-6}$. For each case, include curves for peak and average capacity assignments superimposed on the same figure.

f. Consider the case $\varepsilon = 10^{-5}$, $p = 0.02$, $R_p = 4$ Mbps, $C_L = 200$ Mbps. What is the maximum number of sources allowed (This is the admission policy.) How does this change if $C_L = 100$ Mbps?

4–2 Consider the multiplexing gain defined by equation (4–20). Plot this vs. p on a log-log scale for ε (or P_L) $= 10^{-7}$ and $C = 10$, 100, and compare with Figure 4–9. Calculate $\rho = G_e p$ and plot vs. p as well.

4–3 **a.** Using the fluid-analysis loss-probability approximation of equation (4–25), show the equivalent capacity is given by equation (4–27).

b. A particular image source generates traffic at a peak rate of $R_p = 5200$ cells/sec (What is the rate in Mbps?) Its average burst interval is $1/\beta = 0.5$ sec (What is the average image size, in octets? Don't forget to take off the 5 octets of header per cell.) $N = 100$ such sources are multiplexed. Calculate and plot the capacity C_L, in cells/sec, required as the average "off" interval $1/\alpha$ is varied from a very large to a very small value. The buffer size chosen is $x = 3900$ cells. $P_L = 10^{-6}$ is required. Find the maximum buffer delay as well. Repeat for different values of x and P_L of your own choosing.

4–4 Calculate the equivalent capacity for the examples of problem **4–3b.** above using the simple estimate of equation (4–12). Superimpose on the same plots as those of problem **4–3b.** and plot the smaller of the two, as proposed by [GUER 1991], equation (4–29). Discuss your results.

4–5 N bursty on-off sources are to be multiplexed together at an ATM access port. Each source has exponentially-distributed on- and off-times, with average values of 1 sec and 10 sec, respectively. When "on," a source transmits at its peak rate of 5 Mbps. The outgoing link capacity of the multiplexer is 100 Mbps.

a. Find the number of sources that may be accommodated if (1) peak-rate allocation is used; (2) average rate allocation is used. What is the probability of loss with peak-rate allocation?

b. Find the number of sources that may be multiplexed if the probabil-

ity of loss is $P_L = 10^{-6}$. P_L is approximated by the average time the multiplexer is in the overload region.

c. Repeat **b.** if approximate fluid-flow analysis is used, with the probability of loss defined as $P_L = P$[buffer occupancy $> x$], x chosen such that the maximum buffer delay is 100 msec. Compare all four values of N in **a., b., c.**

d. Repeat **a., b., c.,** if the average off-time is reduced to 5 sec. Compare with the previous case of 10 sec.

4–6 Three different representations of the leaky-bucket algorithm, two using counter implementations, the third using the concept of a token pool, are described in section 4.2. Show all are equivalent.

4–7 This problem focuses on the analysis of the leaky bucket access control carried out by [SIDI 1989], as summarized in section 4.2.

a. Using steady-state balance arguments, as described in the text, show the steady-state probability p_i that there are i tokens in the token buffer of Figure 4–15 is given by equations (4–38) to (4–40).

b. From **a.** above, validate the recursion relations of equations (4–41) to (4–43).

c. $M = 5$ is the maximum burst size. The arrivals are Poisson [equation (4–34)]. Find p_i, $i = 1$ to 5, for a number of values of $\lambda D \equiv \lambda/r$, $0 < \lambda D < 2.5$. *Note:* You'll have to use a normalization technique such as the one proposed in equations (4–44) and (4–45).

d. Show the normalized throughput $\lambda^* D$ (or λ^*/r), the average number of cells getting through the leaky bucket regulator in a D-sec interval, is given by equation (4–46).

e. Plot the normalized throughput-load curve, $\lambda^* D$ vs. λD (or λ/r vs. λ/r), for $M = 5$, using the results of **c.** above. Superimpose the curve obtained using the much simpler approximation of equations (4–31) to (4–33a) and compare.

f. Using equation (4–51b) and the results of **e.** above, plot the loss probability P_L as a function of λ/r for the same example and over the same range of normalized load. Compare with Figure 4–20. (But note that that figure only has λ/r up to 1).

4–8 Starting with the "virtual buffer" r.v. W, as defined by equation (4–61), show that the joint probability $F_j(x)$, as defined by equation (4–65), satisfies the differential equation (4–67). Compare with the differential equation obtained in problem **3–22a**. Show the solution to equation (4–67) is given by equation (4–71a). Why is a_1 not equal to 1 in this case? Carry out your own analysis to show that the equations providing the N unknown boundary conditions appearing in equation (4–71a) are given by equations (4–77) and (4–78).

4–9 This problem focuses on the leaky bucket fluid-flow analysis for the on-off (bursty source) model:

a. Show that for this model the joint probability vector $F(x)$ is given by equation (4–82), with the eigenvalue given by equation (4–79).

b. Show that the two constants a_1 and a_2 in equation (4–82) are given by equations (4–85) and (4–86). Show that the two components of $F(x)$ in this case are given by equations (4–87) and (4–88).

c. Determine the normalization throughput λ^*/r for this case and show it is given by equation (4–94). Calculate the loss probability P_L and verify it is given by equation (4–95). Show that for very small values, P_L is approximated by equation (4–96).

4–10 Consider an on-off source with the following characteristics: the average on-time $1/\beta = 0.35$ sec. The average off-time $1/\alpha = 0.65$ sec. The peak rate of transmission $R_p = 100$ cells/sec. (What traffic source has these characteristics?)

a. Find the leaky bucket token increment rate r for $\rho \equiv pR_p/r = 0.5, 0.7, 0.9$.

b. Calculate and plot P_L vs. B for each of the three values of ρ above. Take $0.1 \geq P_L \geq 10^{-9}$. Use a log scale in plotting P_L.

c. Take $\rho = 0.5$ as the access control operating point. B is chosen to have $P_L = 10^{-5}$ at that point. Calculate P_L if R_p increases to 120 cells/sec, and then to 140 cells/sec. Repeat if $\rho = 0.7$ is chosen as the operating point. Which point provides tighter control? What are the tradeoffs with respect to B and r?

4–11 A human user interacts through a network with an image source file. Each image averages 800,000 bits (100,000 octets) in size. The image source, during admission control, is allocated 2 Mb/s transmission rate in transmitting images across the UNI to the network access buffer. The time between images requested by the human user is estimated to average 10 sec. An exponential on-off model is assumed to be valid.

a. A leaky bucket policing mechanism is to be designed to have $P_L = 10^{-9}$ for the traffic descriptors noted above, but to increase to 0.1 if the transmission rate increases to 2.4 Mb/s. Find r and B to satisfy these conditions.

b. With r and B as chosen above, determine the sensitivity of P_L to changes, first, in image length; second, to changes in average off time. Are there better operating points for these parameters?

4–12 **a.** Starting with equation (4–90) as the leaky-bucket throughput, show one obtains equation (4–90b).

b. Demonstrate why equation (4–90c) also represents the throughput. This implies identity (4–93) must hold. Show that equation (4–94) is the throughput in the on-off source case. Demonstrate that equation (4–93) does in fact hold in the on-off source case.

c. Show that the leaky bucket loss probability in the range of small

values, for the on-off source case, is approximated by equation (4–96).

4–13 Each on-off source in problem **4–5** is connected to a leaky-bucket access controller before being multiplexed at the access port. Each leaky bucket has a data buffer holding at most B_D cells. Fluid-flow analysis is to be used to design the leaky-bucket control, that is, to choose the token (counter) increment rate r and $B = B_D + B_T$.

a. It is desired to have $P_L = 10^{-6}$ for the normal source characteristics as described in problem **4–5 a–c**. Work with cells/sec so that the normal peak source rate (on-state) is 10,000 cells/sec, assuming a cell size of 500 bits for simplicity. Find r and B so that $P_L \geq 0.02$ if the source cell rate in the on-state increases by more than 10%.

b. Repeat **a.** if the average on-time interval is to be controlled: find r and B so that $P_L \geq 10^{-4}$ if the average on-time increases to 1.5 sec.

4–14 The fluid-flow leaky bucket analysis in the text is specialized to the case of on-off sources. Choose a more complex Markov chain to model the source and derive equations for the throughput and cell loss probability. *Note:* An example of a three-state model appears in problem **4–15** following.

4–15 A leaky bucket mechanism controls access to a network. The source to be controlled is modeled as a three-state Markov process: $\mu_{12} = \mu_{23} = 5$/sec.; $\mu_{21} = \mu_{32} = 10$/sec. All other transition rates are zero. The source generation rates in each of the three states is given by 100,000 cells/sec when in state one, 200,000 cells/sec when in state two, and 300,000 cells/sec when in state three. The token generation rate is $r = 250{,}000$ tokens/sec. There is no data buffer.

a. Find the steady state probability that the source is in each of its three states. Find the average load in cells/sec emitted by the source.

b. Find and plot the throughput λ^* of the leaky bucket controller for various values of the token pool size B_T. The particular choices of B_T are part of the problem to be solved: They should cover small, moderate, and large values of B_T, that is, at least three values of B_T. Use fluid-flow analysis.

c. Find and plot the cell loss probability P_L for the same values of B_T as in **a.**