Buffer Management as Inventory Control*

Özalp Babaoğlu

TR 84-600
April 1984

Department of Computer Science
Cornell University
Ithaca, New York 14853

*Partial support for this work was provided by the National Science Foundation under grant MCS 82-10356.

Buffer Management as Inventory Control*

Özalp Babaoğlu

Department of Computer Science
Cornell University
Ithaca, New York 14853
U. S. A.

ABSTRACT

We consider computer subsystems that use buffering as a mechanism to enhance communication performance between two or more components exhibiting a short-term speed mismatch. Often, a fixed size buffer is inserted in such a communication path since it can improve performance by dampening speed variations. We formalize this buffer management problem within the framework of inventory control theory. We show that among all admissible policies for controlling such communications, the structure of the optimal one is analogous to the ***reorder point/order up to level*** policy that arises in the single commodity, continuous review, inventory control problem. This confirms the appropriateness of the intuitive and often-used ***high water mark/low water mark*** policy for buffer management. Given this policy structure, we derive expressions for the optimal parameter values. We discuss extensions of these results whereby policy parameters are dynamically estimated based on current observations of the communication characteristics. An algorithm to generate the optimal ordering decisions (and resulting costs) when communication patterns are known *a priori* is developed as a useful benchmark for evaluating the goodness of on-line policies.

April 26, 1984

* Partial support for this work was provided by the National Science Foundation under Grant MCS 82-10356.

1. Introduction

A prevalent view of a computing system is as a collection of communicating components. This view is valid for most levels of abstraction in such a system. For example, at the lowest level, there are hardware components (CPUs, input/output devices and processors, communication channels, etc.) exchanging data. At the level of the operating system and various network protocols, primitives are provided for data movement between logical (processes) and physical (processors) resources. And, at the user level, a popular paradigm for structuring systems is as a collection of asynchronous processes communicating through messages. Systems or languages that encourage this approach include UNIX [Ritchie and Thompson, 1974], CSP [Hoare, 1978], Ada [Ada, 1979], and Thoth [Cheriton et al., 1979].

A particularly attractive aspect of this structuring scheme is that it extends to distributed computing environments in a natural manner. If the only interaction between processes consists of exchanging messages, there is no need for them to be executing on the same physical processor (site) provided the underlying communication system supports the required primitives.

It has long been observed that the performance of such subsystems can be improved by introducing buffering in the communication path. For example, most hardware input/output devices and channels include some amount of buffer storage so that data transmission can be overlapped with reading (writing). Many operating and database systems enhance file system performance by using large amounts of primary memory as a disk cache. Empirically, a large percentage of the file access requests by user processes can be satisfied from this cache if the file manager is able to detect file reference patterns and issue disk read operations in anticipation [Powell, 1977]. Memory manager components of operating systems usually maintain blocks of free memory so that most of the allocation requests can be satisfied from these regions [Babaoğlu and Joy, 1981]. Nodes in a store-and-forward communication network have to provide local buffer space for temporary storage of packets as they advance along their routes towards destination nodes [Lam, 1976]. Finally, the buffering provided by a message-based inter-process communication mechanism is primarily dictated by the desired semantics [Andrews and Schneider, 1983].

In most of the examples cited above, the buffer storage is usually provided as a *pool* that is shared by the various communication pairs that are active simultaneously. Quite often, the number and behavior of such pairs vary significantly over time. The buffering mechanism associated with a subsystem has to be able to cope effectively with such system dynamics.

Undoubtably, the buffering problem that has received the most attention in the literature has been that of store-and-forward communication networks. Here, a given node may be on the routes of several active connections at any given time, and therefore, the local buffer becomes an important source of global contention and has important packet throughput implications. These studies have devised and analyzed flow control schemes suitable for limiting congestion; thus preventing system throughput from collapsing under increasing input load [Lam and Reiser, 1979; Gerla and Kleinrock, 1980; Kermani, 1981] or provably avoiding deadlock states [Toueg and Ullman, 1981]. The proposed policies regulate traffic through a node based on the current state of the local and/or global buffer allocation. The only study of a flow control policy outside of the store-and-forward communication network environment is due to Reiser [1983]. In this work, Reiser analyzes a message buffering system inserted between a Poisson arrival process and a Poisson server process. The policy employed operates by turning off arrivals when the fixed size buffer is full, and turning it back on when the number of occupied buffers falls below a fixed "resume level." The resulting mechanism is often referred to as the "high water mark/low water mark" scheme. The analysis concludes that a resume level of one half the buffer size is a good rule of thumb for reducing the message delay times in the buffer as well as the frequency of arrival process on/off cycles without significantly degrading throughput (as compared to a system with a resume level very close to the buffer size). The common feature of all of these studies is the analysis of a control policy in an environment where the structure of the policy is assumed to be known *a priori.*

In this paper, we examine the buffer management decision problem without assuming a particular structure for the governing policy. The next section introduces precisely the context within which the buffering mechanism exists. The *Buffer Management Problem* is formally defined in Section 3. Section 4 uses Markov decision theory to derive the structure of the optimal policy among the class of all admissible policies. Once the optimal policy structure has been established, the policy parameters are optimized with respect to the cost function. We also discuss the possible formulations of the same problem as instances of an inventory control system and a queueing system with a bulk server. In Section 5, we extend the earlier results to incorporate nonstationarity of the relevant stochastic assumptions. As these extensions are trivial to implement, the practicality of the proposed policies in dealing effectively with system dynamics is maintained. An algorithm to generate the optimal schedule, when the communication patterns are known *a priori*, is developed in Section 6. We suggest that the goodness of various on-line policies developed in the earlier sections should be judged ultimately by comparing their results with the look-ahead policy results of this section. A discussion of the problem of selecting the optimal buffer size in systems where this is possible and further extensions concludes the paper.

2. The System Model

The system interaction we are interested in studying can be modeled as a *producer/consumer* pair. The producer process is the source of the items (bytes, lines, blocks, etc.) that are being communicated, while the consumer is another process that generates demand for these items. All items produced have to be eventually communicated to the consumer. The focus of our attention is a buffer inserted into this communication path that is capable of holding a finite number of items after they are produced and before they are consumed. The demands generated by the consumer can be satisfied from this buffer (in the order they are produced) as long as it is not empty. Both the size of an item and the size of the buffer (in units of items) are assumed to be fixed.

Formally, we make the following assumptions about the behavior of the consumer process:

(C1) Each demand event generated is for a single unit of the items.

(C2) The demand events constitute a stationary Poisson process with rate μ.

(C3) Demands that encounter an empty buffer are backlogged and persist until they are satisfied.

Note that assumptions C2 and C3 together imply that the demand is generated by an infinite population. In other words, the single consumer models the aggregate behavior of a very large number of processes; each cyclically requesting an item and then computing with it when it becomes available. Individually, a component process blocks when no items are available in the buffer. However, other processes continue to generate demand such that the global demand pattern remains Poisson with a constant rate.

The behavior of the producer is assumed to satisfy the following:

(P1) Whenever the producer is requested to generate x additional items[1], all x items are placed in the buffer after a random length of time[2], called the *lead time*. The lead time is independent of the number of items produced.

(P2) The sequence of lead times are mutually independent and identically distributed random variables with common (known) distribution function F. Furthermore, they are independent of the demand process.

(P3) At any given time, the producer can only be working on the production of a single order. In other words, no additional orders can be received until the current one is delivered, nor can the size of the current order be augmented beyond its original size.

(P4) It is not possible for the producer to generate the same item twice. That is, the execution of the producer proceeds only forward in time and it is impossible to back up to a past state and regenerate a subsequence of items.

For most instances of real producers, assumption P1 adequately models their execution. For example, consider a node in a packet-switching network as a producer. Here, the lead time captures the random delays such as queueing time, protocol execution, interference from other network activities, and process scheduling that are associated with the request for more packets. It is true that for this and most other examples, the lead time is *not* independent of the number of items requested. However, the component of the lead time that depends on

[1] We will call this event "receiving an order for x items."

[2] We will call this event "delivery of an order."

the request size (in this example, the packet transmission time) is usually very small with respect to the total delay and can be neglected.

We make assumption P2 solely for analytical tractability. P3, on the other hand, is dictated by systems where there is a single producer process that responds to requests in a strictly sequential manner such that the lead times of two consecutive orders cannot overlap. Clearly, one could envision generalizations where the lead times are not mutually independent or where several producer processes execute with overlapping lead times.

Finally, the requirement that the consumer eventually receive every item that has been produced, together with assumption P4, implies that items cannot be lost due to a buffer full condition at delivery times. Since the buffer is the only storage facility for items between production and consumption, the size of a single order has to be bounded by the buffer size, even if the system has backlogged items.

3. The Buffer Management Problem

We begin with some preliminary concepts and terminology drawn from Markov decision theory [Chap. 6, Ross, 1970].

Let S_t represent the state of the producer/consumer system at time t. From our earlier description, the only possible control mechanism that can be invoked on this system consists of placing orders for additional items to the producer. At time t, the set of all possible order sizes constitutes the ***action space.*** Given a buffer size m, the ***buffering policy*** can be informally described as a rule for deciding when and how many additional items to order. In chosing an action, a policy may base its decision on any arbitrary collection of information such as the history of the system up to the present, some set of randomizing events, or a collection of events that will occur some time in the future. Policies that exclude the inclusion of future events in their decision making are called ***on-line.*** ***Stationary policies*** constitute an important subclass of on-line policies that base their decisions only on the current system state and do not permit randomization. Formally, a stationary buffering policy, π, is a function mapping the space of system states to the action space, defined for all instants of time t. That is,

$$\pi \colon S_t \rightarrow \{0,1,2,\ldots,m\}$$

where $\pi(S_t) = x$ represents the "place an order for x additional items at time t" action (except when $x=0$ which means "do not order any items"). The range of x is bounded by m since a buffering policy that is consistent with the system model of Section 2 is not permitted to place an order for more than m items. The sequence of ordered pairs (by time)

$$< (t, \pi(S_t)) \colon t \geq 0,\ \pi(S_t) \neq 0 >$$

(i.e., the times orders are placed and their sizes) that the policy π generates for a particular execution is called a ***schedule.*** We say that a policy is ***admissible*** if and only if all of the schedules it generates are consistent with the system model of Section 2.

If a policy includes future events in its decision making information, it is called *look-ahead*. Clearly, only on-line policies are implementable. However, look-ahead policies are useful as benchmarks for evaluating on-line policies.

Finally, we classify policies based on the times they are allowed to operate. The times that a *periodic review* policy is permitted to take actions are restricted to the set $\{t=i\tau:\ i=0,1,2,...\}$, for some fixed positive τ. That is, the policy can be invoked only at fixed intervals of length τ. A policy that can decide on an action at any arbitrary time is called a *continuous review* policy.

We are interested in studying the communication performance enhancements that can result from the presence of the buffering in the producer/consumer system. Intuitively, the role of this buffering is to permit anticipation of future demands for items and overlap their production with current consumption. If we envision demands as customers arriving to pick up single units of the items for subsequent processing, the goal of the buffering mechanism will be achieved if every customer finds an item already in the buffer immediately upon arriving. In this manner, the processing of the items by the constituents of the consumer can proceed at its maximum inherent rate. The penalty associated with not being able to satisfy a demand immediately from the buffer will be precisely the time delay incurred until the item is produced and eventually delivered to the buffer. To this end, we define a *backlog state* to exist if there is at least one unfulfilled demand in the system due to the buffer being empty. Let n be the number of items we wish to communicate from the producer to the consumer through the buffer. Let B denote the total time the system remains in a backlog state during this communication. Clearly, a trivial scheme to minimize B is to allocate a buffer of size n and order all of the items at the outset. However, for an unbounded number of items to be communicated, this scheme requires an unbounded size buffer. For practical reasons, we will consider a fixed buffer size m $(m<n)$ that is imposed on the system by external constraints. The problem of selecting the optimal buffer size in systems where this is feasible is discussed in Section 7.

In order to reason about the system at steady state, we will assume arbitrarily long communication sequences. Consequently, it makes sense to consider the backlog time *per item* rather than the total backlog time as our objective function. For some buffer policy π, let

$$V(\pi) = \lim_{n\to\infty} (B(\pi)/n).$$

The *Buffer Management Problem* (BMP) can now be stated as follows:

> *For an instance of the producer/consumer system with a buffer size* m, *determine an admissible buffer policy* π *such that* $\mathrm{E}[V(\pi)]$ *is minimized.*

We will devote the next section to studying the structure of continuous review, on-line policies that solve the BMP.

4. Structure of Optimal On-Line Policies

It is clear that there is a great deal of similarity between the problem we have posed and the so-called single commodity inventory control problem [Chap. 4, Hadley and Whitin, 1963]. The latter problem is concerned with the operation of a store containing a certain commodity that is subject to demand by customers. The owner of the store is faced with the question of when and how much additional units of the commodity to order from a warehouse such that some suitable operating cost measure is minimized. The natural correspondence between the principal components of the two systems is immediate (item-commodity, buffer-store, producer-warehouse, consumer-customers). However, a significant difference between the two systems is the structure of their respective cost measures. Still, we expect to gain significant insight into our problem by formulating it within the well-established inventory control theory framework. We will also draw freely from the terminology of this field in the presentation of our analysis. We note that studies of optimal policies concerning batch operations in database systems have also exploited inventory parallels [Lohman and Muckstadt, 1977].

At time t, let $z_t \in \{m, m-1, ..., 1, 0, -1, ...\}$ denote the inventory level[3] and Ω_t be an indicator variable with value one if there is an order outstanding, zero otherwise.

LEMMA 1. *The state of the producer/consumer system at time t is completely characterized by the tuple $S_t = (z_t, \Omega_t)$ and π^* is a stationary policy.*

Proof. Since we require that there be at most one outstanding order at any time, the set of states where π^* can be applied (called ***decision states***) are those where $\Omega_t = 0$. Now, for any arbitrary instant in such states, the distribution of the time until the next demand point is exponential with mean $1/\mu$ and it is independent of both the time since the last delivery and the current inventory level. This follows directly from our assumption that the demand process is stationary Poisson.

□

Let π and π' be two admissible policies for the BMP. Let z_t and z_t' denote the inventory levels under each of the two policies, respectively. The following Lemma will aid us in eliminating a large class of admissible policies from consideration for π^*.

LEMMA 2. *If $z_t \geq z_t'$ for all t, then* $\mathrm{E}[V(\pi)] \leq \mathrm{E}[V(\pi')]$.

Proof. This result follows trivially from the definition of our objective function — minimization of the expected backlog time per item. Clearly, for any fixed number of items processed by both policies, the backlog intervals due to π'

---

[3] Negative inventory levels indicate backlog conditions.

(t such that $z_t' < 0$) include those due to π (t such that $z_t < 0$) since $z_t \leq z_t'$.

□

COROLLARY 1. *The action space of π^* consists of the two points* $\{0, \min(m, m - z_t)\}$.

Proof. The first action corresponds to the "do not order" decision. If, however, the policy decides to order at time t, by Lemma 2 the expected backlog time will be minimized by maximizing the order size (since there is no cost associated with the ordering action itself). The largest feasible order size is given by $\min(m, m - z_t)$ due to our constraint that no items be lost upon delivery due to a full buffer.

□

We next try to further characterize the instants when π^* might place an order.

LEMMA 3. *The optimal policy π^* orders either immediately after a delivery or immediately after a demand if no order is outstanding.*

Proof. For contradiction, assume that there exists another policy π' such that $E[V(\pi')] < E[V(\pi^*)]$ and it differs from π^* at only one point in its schedule. Whereas π^* places the ith order at time t, π' places it at time t' such that $t' > t$ and $z_l = z_t$ for all l such that $t \leq l \leq t'$. In other words, π' delays placing the order some arbitrary amount of time beyond that of π^* with no state changes having occurred in the meantime. Since the two inventory levels are the same, the two order sizes have to be the equal. Now, because the lead times are independent of the demand process, the deliveries for π^* and π' will take place at times $t + L$ and $t' + L$, respectively. Therefore, it is the case that $z_l \geq z_l'$ for all l and our initial assumption that $E[V(\pi^*)] > E[V(\pi')]$ contradicts Lemma 2.

□

COROLLARY 2. *Under π^*, the system backlog states are restricted to occur only during order lead times. That is, for any state (z_t, Ω_t), if $z_t < 0$, then $\Omega_t = 1$.*

Proof. If the system ever enters a backlog state with no outstanding order and defers placing an order, the cost can only increase since the order size will never be greater than m.

□

Summarizing our results so far, the optimal policy has the following form:

$$\pi^*(S_t) = \begin{cases} 0, & \text{if } \Omega_t = 1, \\ m, & \text{if } \Omega_t = 0 \text{ and } z_t < 0, \\ ?, & \text{otherwise} \end{cases}$$

where the state is examined only at demand points and delivery instants. We next examine the structure of π^* in the only remaining indeterminate case when $z_t \geq 0$ and $\Omega_t = 0$.

Let $D_k \in \{m, m-1, ..., 1, 0, -1, -2, ...\}$ denote the inventory level immediately after the kth delivery. Then, the sequence $\{D_k: k > 0\}$ constitutes an infinite Markov chain. This follows from the fact that A_k, the kth *lead time demand,* defined as the number of items demanded during the lead time associated with the kth order, are independent and identically distributed random variables when the demand process is stationary Poisson. Therefore, we can write

$$D_k = r_k + Q_k - A_k \tag{4.1}$$

where the *reorder point* r_k is the inventory level at the time the kth order is placed, and Q_k is the kth order size. The key observation is that this Markov chain is *independent* of the structure of the optimal policy in the states $z_t \geq 0$. In other words, given that we are in some non-backlog state, the inventory level at the next delivery point is independent of the inventory level when the order is eventually placed. This can be clearly seen when we write Equation 4.1 as follows:

$$D_k = \begin{cases} m - A_k & \text{if } D_{k-1} \geq 0, \\ D_{k-1} + m - A_k & \text{otherwise.} \end{cases}$$

Let a_i denote the probability that i demand points are generated during an arbitrary lead time. That is, $a_i = \Pr\{A_k = i\}$. We will find it convenient to relabel the states of this Markov chain with the non-negative integers X_k through the transformation $X_k = m - D_k$. Let δ_i, defined as $\delta_i = \lim_{k \to \infty} \Pr\{X_k = i\}$, be the limiting state probabilities of this Markov chain. It is obvious that the above-defined Markov chain is irreducible and aperiodic. The following Lemma states the condition required for the chain to be also positive recurrent.

LEMMA 4. *For any lead time distribution F with mean $\bar{L}$, if $\mu \bar{L} < m$, then the limiting probabilities δ_i exist.*

Proof. If the limiting probabilities exist, they constitute a positive solution to the system of equations $\delta_i = \sum_j \Pr\{X_k = j \mid X_{k-1} = i\} \delta_j$, $i \geq 0$ such that $\sum_{i=0}^{\infty} \delta_i = 1$. These equations can be written as:

$$\delta_i = a_i \sum_{j=0}^{m-1} \delta_j + \sum_{j=0}^{i} \delta_{m+j} a_{i-j}, \qquad i \geq 0. \tag{4.2}$$

Let $G(s) = \sum_{i=0}^{\infty} \delta_i s^i$ and $H(s) = \sum_{i=0}^{\infty} a_i s^i$ be the generating functions of δ_i and a_i, respectively. Note that the expected lead time demand can be obtained from the generating function by evaluating the first derivative at $s = 1$. That is,

$$\mathrm{E}[A_k] = \mu \bar{L} = H'(s)\,|_{s=1}. \tag{4.3}$$

Multiplying Equation 4.2 by s^i and summing over i, we obtain

$$G(s) = H(s)\sum_{j=0}^{m-1}\delta_j + \frac{G(s)H(s)}{s^m} - H(s)(\frac{\delta_0}{s^m} + \frac{\delta_1}{s^{m-1}} + \frac{\delta_2}{s^{m-2}} + ... + \frac{\delta_{m-1}}{s})$$

or

$$G(s)(s^m - H(s)) = H(s)[\, s^m \sum_{j=0}^{m-1}\delta_j - \sum_{j=0}^{m-1}\delta_j s^j\,].$$

Solving for $G(s)$ yields

$$G(s) = \left[\frac{H(s)}{s^m - H(s)}\right]\sum_{j=0}^{m-1}(s^m - s^j)\delta_j \;.$$

Now, taking the limit of both sides as $s \to 1$ and noting that $G(1)=H(1)=1$, we have

$$\lim_{s\to 1} G(s) = 1$$

$$= \lim_{s\to 1}\left[\frac{H(s)(\delta_0 m s^{m-1} + \sum_{j=1}^{m-1}(m s^{m-1} - j s^{j-1})\delta_j) + H'(s)\sum_{j=0}^{m-1}(s^m - s^j)\delta_j}{m s^{m-1} - H'(s)}\right]$$

$$= \frac{\sum_{j=0}^{m-1}(m-j)\delta_j}{m - \mu\bar{L}} \tag{4.4}$$

where we have used l'Hopital's rule to evaluate the right limit and substituted Equation 4.3 in the denominator.

Now, since $\delta_i \geq 0$, and the numerator of Equation 4.4 is a positive weighted sum of the first m limiting probabilities, the stability condition for this system clearly requires $m > \mu\bar{L}$.

□

We will define a *cycle* to be the time interval between two consecutive deliveries. Having established that the inventory level at the beginning of a cycle has a steady state probability distribution, we will minimize our objective function—expected backlog time per item—within an arbitrary cycle. Given that the reorder point has an arbitrary value r within a cycle, it is easy to show that the time the system will remain in a backlog state until the delivery, $BC(r)$, has an expected value given by

$$\mathrm{E}[BC(r)] = \begin{cases} \bar{L} - \int_0^\infty (1-F(x))P(r,\mu x)\,dx & \text{if } r \geq 0, \\ \bar{L} & \text{otherwise} \end{cases}$$

where $F(x)$ is the lead time distribution with mean $\bar{L}$ and

$P(r,\mu x)=\sum_{i=0}^{r} e^{-\mu x}(\mu x)^i/i!$ is the Poisson distribution with mean μx. Furthermore, the number of items delivered at the end of the cycle, $QC(r)$, when the reorder point is r is exactly the order size

$$QC(r) = \begin{cases} m-r & \text{if } r\geq 0, \\ m & \text{otherwise.} \end{cases}$$

Suppose we are told that the inventory level at the beginning of a cycle (at steady state), D, is equal to i. Now, without loss of generality, let us assume that the policy π^* is such that the inventory level, when it finally decides to place an order, has some arbitrary value r_i. By Corollary 2, if $i<0$, we order immediately and $r_i=i$. Else ($i\geq 0$), we know that $i\geq r_i\geq 0$. Conditioning on the transformed states X where $X=m-D$ (so that the state labels match those we used in the proof of Lemma 4), we obtain the following expression for the cost of a cycle:

$$CC(r_0,r_1,\ldots) = \mathrm{E}[BC/QC]$$

$$= \sum_{i=0}^{\infty} \mathrm{E}[BC(r_i)/QC(r_i) \mid X=i]\delta_i$$

$$= \sum_{i=0}^{m} \left[\frac{\delta_i}{m-r_i}(\bar{L}-\int_0^{\infty}(1-F(x))P(r_i,\mu x)dx) \right] + \sum_{i=m+1}^{\infty} \delta_i \frac{\bar{L}}{m} \tag{4.5}$$

Note that given $X=i$, the reorder point r_i is a constant and therefore we could take the term $QC(r_i)$ out of the expected value in the first sum. Also, the second sum is independent of the reorder points.

What remains to be done in completing the structure of π^* is the identification of the r_i's such that Equation 4.5 is minimized subject to the constraints $i\geq r_i\geq 0$. We will first solve the unconstrained problem by examining a set of $m+1$ first difference equations. Let $\Delta_1 CC_i$ denote the first difference of the per cycle cost with respect to r_i. That is,

$$\Delta_1 CC_i(r_0,r_1,\ldots,r_m) = CC(r_0,r_1,\ldots,r_i,\ldots,r_m) - CC(r_0,r_1,\ldots,r_i-1,\ldots,r_m).$$

Now, recall our discussion about the independence of the steady state probabilities δ_i from the policy employed in the non-backlog states. The implication is that, since the same δ_i hold no matter what the choice of reorder points is, the (unconstrained) reorder points cannot be functions of the initial inventory level i. In other words, there is a single solution to the set of the first difference equations whose prototype is

$$\Delta_1 CC(r) = \frac{\bar{L} - \int_0^{\infty} P(r,\mu x)(1-F(x))dx \; - (m-r)\int_0^{\infty}(1-F(x))e^{-\mu x}\frac{(\mu x)^r}{r!}dx}{(m-r)(m-r+1)}.$$

Let r^* be the set of positive integers such that $\Delta_1 CC(r^*) \leq 0$ and $\Delta_1 CC(r^* + 1) \geq 0$. To ensure that the cost function indeed attains a minimum at these points, we also have to show that the second difference evaluated at each r^* is positive. This can be easily shown to be equivalent to requiring the following

$$\int_0^\infty \frac{e^{-\mu x}(\mu x)^{r^*}}{r^*!} \left(\frac{r^*}{\mu x} - 1\right)(1-F(x))dx \geq 0.$$

Unfortunately, for an arbitrary lead time distribution F, we cannot characterize the optimal reorder points r^* any better. To continue the analysis further, let us assume that the lead is exponentially distributed with mean $\bar{L}$. Then, the per cycle cost equation and its first and second differences become

$$CC(r_0, r_1, \ldots, r_m) = \sum_{i=0}^{m} \frac{\delta_i \bar{L}}{(m-r_i)} \rho^{r_i+1} + \sum_{i=m+1}^{\infty} \frac{\bar{L}}{m} \delta_i$$

$$\Delta_1 CC(r) = \frac{\bar{L}\delta_i \rho^r}{(m-r)(m-r+1)} (m\rho - r\rho + \rho - m + r) \tag{4.6}$$

$$\Delta_2 CC(r) = \bar{L}\delta_i \rho^{r-1} \frac{((m-r+2)(m-r+1)\rho^2 - 2(m-r+1)(m-r)\rho + (m-r)(m-r+1))}{(m-r)(m-r+1)(m-r+2)} \tag{4.7}$$

respectively, where $\rho = \mu\bar{L}/(1+\mu\bar{L})$.

LEMMA 5. *If the lead time distribution is exponential with mean $\bar{L}$, then the per cycle cost function $CC(r)$ is convex and has a minimum at $r^* = \lfloor m - \mu\bar{L} \rfloor$ for all possible system parameters.*

Proof. To prove the convexity result, it suffices to show that $\Delta_2 CC(r) \geq 0$ for all r, μ and $\bar{L}$. Note that, by Equation 4.7, when $\rho = 0$, $\Delta_2 CC(r) > 0$ since $r < m$ and $\rho > 0$. Through simple algebra, we can demonstrate that the quadratic in ρ of Equation 4.7 has no real roots for any $r < m$. Therefore, $\Delta_2 CC(r) > 0$ always. Now, since the cost function is convex, we know that it has a single minimum in the interval $0 \leq r < m$. This minimum occurs at the smallest integer r^* such that $\Delta_1 CC(r^*) \leq 0$ and $\Delta_1 CC(r^* + 1) \geq 0$. From Equation 4.6, we obtain $r^* = \lfloor m - \mu\bar{L} \rfloor$.

□

Observe that for a stable and nondegenerate system ($\mu\bar{L} < m$ and $\mu\bar{L} \neq 0$), the optimal reorder point is bounded above by $m-1$ and below by zero. This result has a particularly simple intuitive interpretation: If the mean lead time demand is very small, small size orders will be placed frequently. If, on the other hand, the mean lead time demand is very large, the placement of the order will be delayed as long as possible so as to make its size large. For intermediate cases, the mean order size is exactly the mean lead time demand.

Incorporating the constraints $0 \leq r_i \leq i$ on the reorder points, which we have ignored up to now, can be accomplished as follows:

$$r_i = \begin{cases} r^* & \text{if } i \geq r^* \\ i & \text{otherwise.} \end{cases}$$

Finally, the complete optimal policy π^* is obtained by putting together the results of Lemmata 1-5.

THEOREM 1. *In a producer/consumer system with buffer size m, stationary Poisson demand with rate μ, and exponential lead times with mean $\bar{L}$, the policy that minimizes the long run backlog time per item is given by*

$$\pi^*(S_t) = \begin{cases} 0, & \text{if } \Omega_t = 1, \\ 0, & \text{if } \Omega_t = 0 \text{ and } z_t > \lfloor m - \mu\bar{L} \rfloor, \\ m - z_t, & \text{if } \Omega_t = 0 \text{ and } 0 \leq z_t \leq \lfloor m - \mu\bar{L} \rfloor, \\ m, & \text{if } \Omega_t = 0 \text{ and } z_t < 0. \end{cases}$$

where the only relevant system states are those at demand points and deliveries.

It is interesting to compare our policy with the well-known $<R,r>$ policy developed for the infinite horizon, single commodity, unit demand, random lead times, continuous review inventory problem with complete backlogging. The two policy parameters, R and r, are the "order up to quantity" and the "reorder point," respectively. The structure of the policy is quite similar to ours in that an order is placed to bring the inventory level up to R at the instant of the first demand that brings the level down to r. There are, however, some important differences. First, in our system, the order size is bounded by the buffer size and the cost function we are trying to minimize is quite different. Second, we are restricted to having at most one outstanding order. Because of this constraint, our environment exhibits a curious mixture of periodic review and continuous review characteristics. In effect, the lead time intervals impose periods during which no action can be taken, making state knowledge worthless (even though it is available). On the other hand, during the periods from the time of a delivery to the placement of an order, the state is examined continuously to detect the reorder point.

The BMP could be posed as an equivalent queueing problem with a "bulk" server through a simple translation of terminology [pp. 137-139, Kleinrock, 1975].

Consider an infinite capacity queue where customers arrive according to a stationary Poisson process with rate μ. The single service station operates as follows: If the queue is not empty, a group of up to m customers are given service at the same time. The service time distribution F is independent of the batch size and the demand process. The equivalence of the BMP and this "bulk" server queueing system is obvious. Service periods correspond to lead times, customers account for the empty buffer slots created by demand such that an empty queue represents a full buffer and a queue length greater than m represents a backlog state. In fact, the random variable X_k introduced in the previous section is precisely the queue length after the kth service interval in this system. Kleinrock derives the steady state distribution for the number of customers in the system under the simplification that a new arrival joins the batch currently in service (where the service time distribution is assumed to be exponential) as long as the total number of customers in service is not greater than m. Note that, in the BMP, this simplification corresponds to the ability to augment the size of an outstanding order up to m items before it is delivered.

5. Coping with Nonstationarity

The stochastic optimality of the policy developed in the previous section is contingent on the demand process being stationary Poisson. The optimality argument collapses even if the Poisson assumption is retained but the rate is made time dependent. In consumer instances typically found in computer systems, demand rates are often functions of time. The intensity of activity generated by certain subcomponents usually exhibits cyclic patterns by hours, days, weeks, etc. In addition to this, the activity within a single cycle tends to consist of distinct phases. For example, the demands placed on a disk device serving the file system vary on a 24 hour basis with high intensity during working hours and low intensity at other times [Smith, 1981]. Within a given day, the file access patterns are very much a function of the particular set of programs that happen to be executing at any time. In inventory control terminology, the demand process can be classified as being "seasonal" with "shock" changes at random times. Clearly, similar time dependencies could be applicable to the production lead times as well.

In practice, even if we know that the demand process and lead time distribution are adequately modeled as Poisson and exponential, respectively, we need to estimate their parameters. However, from Theorem 1, the policy π^* is a function of the demand process and the lead time distribution only through the *mean lead time demand* expression $(\mu\bar{L})$. Therefore, it suffices to estimate this quantity alone. Now, assuming that the duration of each system phase within the seasonal cycles is sufficiently long so that steady state arguments hold, the problem of accommodating nonstationarity reduces to that of estimating the mean lead time demand. A wealth of techniques have been proposed for the more general problem of forecasting from time series data [Box and Jenkins, 76]. As most of these techniques require lengthy statistical analysis, they are not appropriate for our environment where the parameter estimation has to be done in real time with an insignificant amount of overhead.

Gross and Craig have studied the problem of forecasting specifically within the domain of inventory control [Gross and Craig, 1974]. In particular, they report results comparing maximum likelihood, exponential smoothing and Bayes forecasting schemes in estimating the per period demand (the inventory model is periodic review with zero lead time). Their results, based on simulation studies, conclude that for demand patterns with long range trend or shock changes, the "exponential smoothing with trend" scheme performs better than the others. This is especially attractive, since the scheme is computationally trivial. Given two smoothing constants α and β, both between zero and one, the estimate for the mean demand during the kth lead time, $\hat{A}_k$, is given by

$$\hat{A}_k = S_k + T_k$$

where

$$S_k = \alpha A_{k-1} + (1-\alpha)(S_{k-1} + T_{k-1}),$$

$$T_k = \beta(S_k - S_{k-1}) + (1-\beta)T_{k-1}$$

and the initial conditions are $S_1 = A_1$ and $T_1 = 0$.

6. An Optimal Look-Ahead Policy

When the on-line policy is extended as proposed in the previous section, it becomes extremely difficult to make a statistically meaningful statement about the optimality of the resulting policy. In fact, even for systems where the relevant stochastic processes are stationary but some of the other assumptions are questionable (e.g., demand process is not Poisson or the lead times are not independent and identically distributed), it is again very difficult to asses the goodness of the proposed on-line policy.

The ultimate measure for the goodness of an on-line policy that is stochastically optimal under some set of assumptions is its relative performance compared to a look-ahead policy for various real problem instances. This methodology has been often used in evaluating on-line algorithms for page replacement in a virtual memory systems [Belady, 1966]. Demonstrating favorable comparison of an on-line policy with its look-ahead counterpart over a large problem domain is much more effective than trying to statistically validate all of the assumptions over the same domain.

In this section, we present an algorithm to generate the optimal look-ahead policy for the BMP. To make the comparison fair with respect to the on-line results, we require that orders can only be placed at demand points or at delivery times (as they were in the optimal on-line policy). Note that, since in the look-ahead case the times of the future demands are known *a priori*, an optimal schedule could place an order at arbitrary times. However, we conjecture that the solution to the problem with unconstrained order times is computationally intractable.

The input data for the algorithm consists of the two sets $\{t_1, t_2, \ldots, t_n\}$ and $\{L_1, L_2, \ldots, L_n\}$ deterministically characterizing the consumer and producer behaviors, respectively. These are positive real numbers where t_i is the time of the ith demand and L_i is the lead time associated with the ith order, if it is placed. Without loss of generality, we will assume that $t_1=0$. Note that no policy will ever need to place more than n orders. Since the number of items is fixed, for a given buffer size m, our objective is to construct a look-ahead policy such that the total backlog time is minimized. We summarize the constraints for the problem:

- All n demands must be satisfied eventually.
- At any given time, there can be at most one outstanding order.
- Orders can only be placed at instants when demands or deliveries occur.
- The size of an order is bounded by the buffer size.

To avoid dealing with special cases at the boundary, we introduce an artificial $n+1$st demand point at infinity (i.e., $t_{n+1}=\infty$). This demand will make sure that a nonzero backlog does not persist indefinitely after t_n.

Consider some time interval $[t,t_j]$, beginning with a demand point or a delivery at time t and continuing upto a demand point at time t_j such that $1\leq j\leq n+1$ and $t\leq t_j$. Let $B^k_{t,j}(Z,p)$ denote the minimum backlog time attainable in this interval given that the pth order is placed at time t when the inventory level is Z, and no more than k additional orders are placed within the interval. At the beginning of the problem (i.e., at time 0), we assume that the buffer is empty and an order with zero lead time is placed for zero items. Consequently, we will abbreviate $B^k_{0,j}(0,0)$ as simply $B^k_{0,j}$. Let

$$\pi^k_{0,j}=<(\sigma_1,\gamma_1),(\sigma_2,\gamma_2),\ldots,(\sigma_q,\gamma_q)>$$

be the (optimal) schedule that achieves $B^k_{0,j}$. Note that $|\pi^k_{0,j}|=q\leq k$. Our goal is then to obtain $\pi^n_{0,n+1}$ and the corresponding backlog time $B^n_{0,n+1}$.

Now, assume we know the last element in the optimal schedule, $\pi^k_{0,j}$, for the subproblem defined in $[0,t_j]$. In particular, let $t=\sigma_q$ be the time when the last order is placed in this schedule, let $\theta=\sigma_q+L_q$ be the time the last order is delivered, and let b be the smallest integer such that $t_b\geq t$. In other words, t_b is the instant of the first demand point after the last order. The *last event* in this interval occurs at time $l=\theta$ if $\theta>t_{j-1}$, and at time $l=t_j$ otherwise. Then, the minimum backlog time within $[0,t_j]$ can be written as

$$B^k_{0,j} = \min(\, B^{k-1}_{0,b} + B^0_{t,j}(Z^{k-1}_{0,b},\ |\pi^{k-1}_{0,b}|+1) - (t_b-t)\cdot\psi(Z^{k-1}_{0,b}<0),\ B^{k-1}_{0,j}\,) \quad (6.1)$$

where $Z^k_{0,j}$ denotes the inventory level as seen by the jth demand when the schedule $\pi^k_{0,j}$ is applied within this interval and $\psi(I)$ is an indicator function with value one if the Boolean expression I is true, and zero otherwise. Note that we have corrected the double accounting of the backlog time (if any) within $[t,t_b]$ where the two intervals overlap. Also, we have to allow for the possibility that the minimum backlog time is attained by not utilizing the option of increasing the number of orders that are placed. As we do not know $\pi^k_{0,j}$ *a priori*, the first term of the minimization above will have to be obtained by trying all possible locations for the last order point within $[0,t_j]$ and selecting the one resulting in the minimum backlog time. Let

$$B = \min_{1 \le i \le j} (B_{0,i}^{k-1} + B_{l_i,j}^{0}(Z_{0,i}^{k-1}, |\pi_{0,i}^{k-1}|+1) - (t_i - l_i)\cdot\psi(Z_{0,i}^{k-1}<0), \tag{6.2}$$

$$B_{0,i}^{k-1} + B_{t_i,j}^{0}(Z_{0,i}^{k-1}, |\pi_{0,i}^{k-1}|+1))$$

where l_i is the time of the last event in $[0,t_i]$. The two terms of the minimization represent the ordering decisions at the last delivery point and the demand point, respectively. Note that there could be several last order times that result in the same backlog time B. If this is the case, we select the largest of these since delaying the last order as long as possible in a schedule (while not increasing the backlog time) is sufficient to attain the largest possible inventory level at the end of the interval (i.e., $Z_{0,j}^{k}$). In turn, this will ensure optimality within future intervals since $B_{t,j}^{k}(Z,p)$ is clearly a decreasing function of Z. Finally, substituting into Equation 6.1, we obtain

$$B_{0,j}^{k} = \min(B, B_{0,j}^{k-1}) \ . \tag{6.3}$$

If this comparison results in a tie, we chose the first option (increasing the number of orders allowed) since this will ensure the largest possible inventory level at the end of the interval.

We structure the dynamic programming computation to operate on an $n+1$ by $n+1$ table with the columns indicating the maximum number of orders permitted (k, $0 \le k \le n$), the rows corresponding to the end points of the subintervals ($[0,t_j]$, $1 \le j \le n+1$), and each entry containing the appropriate $\pi_{0,j}^{k}$, $B_{0,j}^{k}$, and $Z_{0,j}^{k}$. Given an initially empty buffer and the problem input, we obtain the entries for the first column as follows:

$$\pi_{0,j}^{0} = \langle\rangle \ ,$$

$$B_{0,j}^{0} = t_j \ ,$$

$$Z_{0,j}^{0} = 1-j \ , \quad 1 \le j \le n+1$$

where $\langle\rangle$ denotes the empty sequence.

Since computing $B_{0,j}^{k}$ requires only values from the previous column (see Equation 6.3), we proceed filling the table from left-to-right, bottom-to-up. Take an arbitrary entry in the jth row, kth column. For any given i, $1 \le i \le j$, let $Z = Z_{0,i}^{k-1}$ and $p = |\pi_{0,i}^{k-1}|+1$. Let d be the smallest integer such that $t + L_p < t_d$. In other words, t_d is the first demand point after the delivery of the pth order placed at t. Then, we can determine the $B_{t,j}^{0}$ as required by Equation 6.2 as follows:

$$B_{t,j}^{0}(Z,p) = \infty \qquad \text{if } L_p > t_j - t$$

since the lead time extending beyond the end of the segment would imply two outstanding orders at t_j, thus violating one of the constraints. Otherwise (i.e., $L_p \le t_j - t$), if i is the smallest integer such that $t_i \ge t$,

$$B_{t,j}^{0}(Z,p) = \max(0, t+L_p - \max(t, t_{i+Z}))$$
$$+ \max(0, t_j - t_{\max(d,\, d+\min(m,\, m+Z-d+i))})$$
$$+ (t_d - t - L_p)\cdot\psi((m+Z-d+i < 0) \wedge (d \neq n+1))$$

This expression is easily obtained by enumerating all possible cases giving rise to backlog times in $[t, t_j]$ with an initial inventory level Z and an order with L_p lead time placed at t.

Let t be the order time minimizing Equation 6.2 after the ties are broken as discussed earlier. Let b the smallest index such that $t_b \geq t$. Then, the remaining two elements of the (j,k)th table entry are determined as

$$Z_{0,j}^{k} = \begin{cases} Z_{0,j}^{k-1} & \text{if } B > B_{0,j}^{k-1} \\ Z_{0,b}^{k-1} - j + b + \min(m,\, m - Z_{0,b}^{k-1} + d - b) & \text{otherwise} \end{cases}$$

and

$$\pi_{0,j}^{k} = \begin{cases} \pi_{0,j}^{k-1} & \text{if } B > B_{0,j}^{k-1} \\ \pi_{0,b}^{k-1} \,||\, \langle t, \min(m,\, m - Z_{0,b}^{k-1} + d - b)\rangle & \text{otherwise} \end{cases}$$

where " || " denotes the sequence concatenation operator.

There is one last detail we have to be concerned with. Let $\xi = Z_{0,n+1}^{n}$ be the inventory level seen by our artificial demand. If $\xi < 0$, there is residue backlog in the system and the schedule $\pi_{0,n+1}^{n}$ has to be augmented so that all demand is satisfied. This can be accomplished simply by placing $\lceil \xi / m \rceil$ additional orders of size m each (except for the last one) in succession. The final total backlog time, $B_{0,n+1}^{n}$ will also have to be augmented by the sum of the lead times associated with these orders.

The overall time complexity of the proposed algorithm is clearly $O(n^3)$ since the table contains n^2 entries and the elements of each entry require $O(n)$ comparisons to generate.

Comparison of the costs resulting from look-ahead and on-line policies remains to be seen for actual producer/consumer instances.

7. Selecting the Buffer Size

Suppose we are allowed to select the buffer size, m, rather than it being externally imposed. This is commonly the case in systems where individual buffers are part of a global pool and the partioning among them can be varied. We will assume that once selected, the buffer size remains fixed. In this environment, the BMP has to resolve not only the buffering policy but also the buffer size.

This aspect of the problem can be incorporated into our formulation by explicitly accounting for the storage space invested in the buffer for the duration of the communication. In particular, we will redefine our objective function to be

$$V(m,\pi) = \lim_{n \to \infty} \frac{1}{n}[B(m,\pi) + hmT(m,\pi)]$$

where h is a system constant relating the cost of one additional unit of storage allocated towards the buffer to the equivalent backlog time units and $T(m,\pi)$ is the total elapsed time to transfer the n items.

For a given buffer size, it is easy to show that Lemmata 1-4 of Section 4 continue to hold for the new objective function. Therefore, the optimal policy with respect to the new objective function will have the same structure as before. To derive expressions for the optimal reorder points and the optimal buffer size, we can once again consider the expected cost per cycle (with deliveries delimiting cycles) and solve the set of equations obtained by taking first differences with respect to the reorder points and the buffer size. However, with this new objective function, obtaining a closed form solutions for these parameters requires an explicit solution for the limiting probabilities for the embedded Markov chain of the inventory level at the beginning of cycles as defined in Section 4. We are currently unable to exhibit such a solution.

8. Discussion and Conclusions

Our results confirm the appropriateness of the intuitive and often-used "high water mark/low water mark" scheme for governing buffer systems. We have shown that for the system model capturing the operation of a producer/consumer pair communicating through a fixed size buffer, the structure of the policy that is optimal with respect to minimizing the long-run time the system remains in a backlog state per item is particularly simple. A single reorder point ("low water mark" or "resume level") governs when an order should be placed after a delivery. Order size is the maximum number of items required to fill the buffer, but never exceeding the buffer size ("high water mark"). We were able to demonstrate the convexity of the cost function (thus the single reorder point) only under the assumptions of stationary Poisson demand and exponential lead times. The structure of the optimal policy when either of these assumptions is relaxed remains an open problem.

The study we have presented in this paper could be extended in several other ways. For example, it is conceivable for the system to have a bound on the number of backlogged items at any time. In other words, the consumer could model a *finite* population of size M such that the demand rate, μ, is function of the system state as follows:

$$\mu(z_t) = \begin{cases} \mu, & \text{if } z_t > M \\ 0, & \text{if } z_t \leq M. \end{cases}$$

The assumption that each demand is for a single unit of the items could easily be relaxed so that multiple items are requested with some known probability distribution. Also, we could remove the restriction that the order size is bounded by the buffer size if we allow the producer to regenerate lost items due to the buffer full condition upon delivery. In this case, the independence of the lead times would no longer hold since the lost items would have to generated (incurring a lead time delay) before any additional orders could be placed.

In incorporating any of the extensions we have proposed, it should be worthwhile to try to exploit the natural analogy between our problem and some appropriate instance of the inventory control problem.

Acknowledgments

I am grateful to Nesim Erkip and Richard Koo for extensively discussing this material with me and to Fred Schneider for his careful reading of an earlier version of this paper.

References

Ada, "Preliminary Ada reference manual." *SIGPLAN Notices*, **14**, 6 (June 1979).

Andrews, G. R., and Schneider, F. B. "Concepts and notations for concurrent programming." *ACM Comput. Surv.* **15**, 1 (March 1983), 3-43.

Babaoğlu, Ö., and Joy, W. N. "Converting a swap-based system to do paging in an architecture lacking page-referenced bits." In *Proc. 8th ACM Symp. Operating Systems Principles* (Asilomar, California., December 1981). ACM, New York, pp. 78-86.

Belady, L. A. "A study of replacement algorithms for a virtual storage computer." *IBM Systems J.* **5**, 2 (1966), 78-101.

Box, G. E. P., and Jenkins, G. M. *Time Series Analysis: Forecasting and Control.* Holden-Day, San Francisco, 1976.

Cheriton, D. R., Malcom, M. A., Melen, L. S., and Sager, G. R. "Thoth, a portable real-time operating system." *Commun. ACM* **22**, 2 (February 1979), 105-115.

Gerla, M., and Kleinrock, L. "Flow control: a comparative survey." *IEEE Trans. Commun.* **COM-28**, 4 (April 1980), 553-576.

Gross, D., and Craig, R. J. "A comparison of maximum likelihood, exponential smoothing and Bayes forecasting procedures in inventory modelling." *Int. J. Prod. Res.* **12**, 5 (1974), 607-622.

Hadley, G., and Whitin, T. M. *Analysis of Inventory Systems.* Prentice-Hall, Englewood Cliffs, 1963.

Hoare, C. A. R. "Communicating sequential processes." *Commun. ACM* **21**, 8 (August 1978), 666-677.

Kermani, P. "Analysis of a feedback scheme for congestion control in computer networks." Tech. Rep. RC-8645, IBM T. J. Watson Research Center, January 1981.

Kleinrock, L. *Queueing Systems, Volume I: Theory.* John Wiley & Sons, New York, 1975.

Lam, S. S. "Store-and-forward buffer requirements in a packet switching network." *IEEE Trans. Commun.* **COM-24**, 4 (April 1976).

Lam, S. S., and Reiser, M. "Congestion control of store-and-forward networks by input buffer limits—an analysis." *IEEE Trans. Commun.* **COM-27**, 1 (January 1979), 127-134.

Lohman, G. M., and Muckstadt, J. A. "Optimal policy for batch operations: backup, checkpointing, reorganization, and updating." *ACM Trans. Database Syst.* **2**, 3 (September 1977), 209-222.

Powell, M. L. "The DEMOS file system." In *Proc. 6th ACM Symp. Operating Systems Principles* (West Lafayette, Ind., November 1977). ACM, New York, 1977, pp. 33-42.

Reiser, M. "Queueing and delay analysis of a buffer pool with resume level." In A. K. Agrawala and S. K. Tripathi (Eds.), *Performance '83.* North-Holland, Amsterdam, 1983, pp. 25-32.

Ritchie, D. M., and Thomson, K. "The UNIX timesharing system." *Commun. ACM* **17**, 7 (July 1974), 365-375.

Ross, S. M. *Applied Probability Models with Optimization Applications.* Holden-Day, San Francisco, 1970.

Smith, A. J. "Analysis of long-term file reference patterns for application to file migration algorithms." *IEEE Trans. Softw. Eng.* **SE-7**, 4 (July 1981), 403-417.

Toueg, S., and Ullman, J. D. "Deadlock-free packet switching networks." *SIAM J. Comput.* **10**, 3 (August 1981), 594-611.