Performance Evaluation 18 (1993) 133–140
North-Holland

Buffer size requirements under longest queue first *

H.R. Gail, G. Grover, R. Guérin, S.L. Hantler
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

Z. Rosberg
IBM Israel, Haifa 32000, Israel

M. Sidi
Technion, IIT, Haifa 32000, Israel

Received 1 August 1991
Revised 8 October 1992

Abstract

Gail, H.R., G. Grover, R. Guérin, S.L. Hantler, Z. Rosberg and M. Sidi, Buffer size requirements under longest queue first, Performance Evaluation 18 (1993) 133–140.

A model of a switching component in a packet switching network is considered. Packets from several incoming channels arrive and must be routed to the appropriate outgoing port according to a service policy. A task confronting the designer of such a system is the selection of policy and the determination of the corresponding input buffer requirements which will prevent packet loss. One natural choice is the Longest Queue First discipline, and a tight bound on the size of the largest buffer required under this policy is obtained. The bound depends on the channel speeds and is logarithmic in the number of channels. As a consequence, Longest Queue First is shown to require less storage than Exhaustive Round Robin and First Come First Served in preventing packet overflow.

1. Introduction

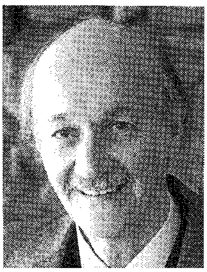
Technological advancements have brought about new switching fabrics that can support various types of traffic, including real-time traffic such as voice conversations, video sessions and computer-to-computer data transfer. Some of these switching fabrics employ packet switching techniques. In order to reduce the nodal processing overhead necessary for each packet in conventional packet switching networks, part of the switching functions are off-loaded onto high-speed specialized hardware that will be called the switching component.

The need to support real-time and high-speed traffic that has a delivery time bound suggests the use of limited (finite) buffering in the switching component. The reason is that with unlimited buffers, the delay of a packet that enters the system cannot be bounded. However, the limited buffering may cause packet loss, which must be minimized in order to provide a reasonable quality of service. The subject of this paper is the analysis of service policies for the traffic coming into a switching component in a packet switching network and the determination of the size of the finite buffers needed to insure operation without packet loss.

The traffic arrives into the switching compo-

Correspondence to: H.R. Gail, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA.
* This work was done while Z. Rosberg and M. Sidi were at the IBM T.J. Watson Research Center.

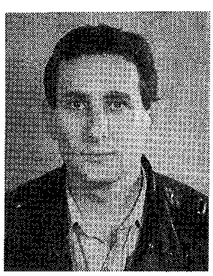
H. Richard Gail received the B.A. degree in mathematics from the University of California, Riverside, the M.A. degree in mathematics from the University of California, Los Angeles, and the Ph.D. degree in engineering from the University of California, Los Angeles in 1983. Since 1984 he has been employed at the IBM Thomas J. Watson Research Center in Yorktown Heights, New York as a Research Staff Member. During the fall of 1990 he was a Visiting Researcher in the Computer Science Department of the Institute of Mathematics of the Federal University of Rio de Janeiro. His areas of interest include applied probability, queueing theory, and the modeling and analysis of computer systems and computer networks.



George Grover is engaged in communications research at the IBM Thomas J. Watson Research Center. He has been involved in the design of algorithms for designing near least cost physical networks, and in the design of deadlock free flow control and of critical synchronization processes for the IBM System 36 APPN, a peer to peer data network switch. Since 1979 he has worked extensively in the design of SNA (IBM's System Network Architecture) networking functions. Previously, he participated in assembler, compiler and operating system design and development activities in conjunction with IBM's System 360, the Stretch computer, and the 7950—a special purpose extension of the Stretch computer; and in technical planning activities relating to advanced technology, and security and privacy.

Roch Guérin received the "Diplôme d'Ingénieur" from the École Nationale Supérieure des Télécommunications, Paris, France, in 1983, and the M.S. and Ph.D. from the California Institute of Technology, both in electrical engineering, in 1984 and 1986, respectively. Since August 1986 he has been with IBM at the Thomas J. Watson Research Center, Yorktown Heights, New York, where he now manages the Network System Design Group in the High Performance Computing and Communication Department. His current activities include analysis, architecture, and deployment of high-speed networks and applications. His research interests are in the area of performance analysis and modeling of communications systems, with emphasis on congestion control, bandwidth management, dynamic routing, and their interactions in high-speed networks. Dr. Guérin is a member of Sigma Xi, the IEEE Communications and Information Theory Societies, and is an editor for the *IEEE Transactions on Communications*.

Sidney L. Hantler received the B.S., M.S. and Ph.D. degrees in mathematics from the University of Michigan, where he did research in functions of several complex variables. In 1974 he joined the IBM T.J. Watson Research Center, where he is manager of the Stochastic Analysis Group.



Zvi Rosberg received the B.Sc., M.A. and Ph.D. degrees from the Hebrew University of Jerusalem, in 1971, 1974 and 1978, respectively. From 1972 to 1978 he was a senior systems analyst in the Central Computing Bureau of the Israeli government. From 1978 to 1979 he held a research fellowship at C.O.R.E., Catholic University of Louvain, Belgium. From 1979 to 1980 he was a visiting assistant professor at the Department of Business Administration, University of Illinois. From 1980 to 1990 he held a position in the Computer Science Department, Technion, Israel. In 1990 he joined IBM Israel as a Research Staff Member in the Science and Technology Department, where he currently holds a position of a Program Manager of Communication Networks. From 1985 to 1987 he was on leave of absence in IBM Thomas J. Watson Research Center, Yorktown Heights, USA. Since 1980 he held summer research positions in the IBM Thomas J. Watson Research Center, Yorktown Heights, the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, and the Department of Electrical Engineering and Computer Science, University of California, Berkeley. His main research interest and development activities include the areas of analysis and algorithms in probabilistic models for communication network and computing systems, models and tools for network design, performance evaluation, control of queueing systems and applied probability.

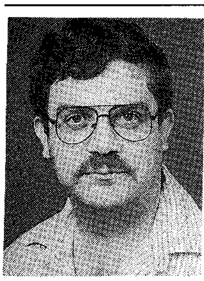
nent from several incoming communication channels. Instead of using a standard stochastic model for the entering traffic, we use a more adequate model that reflects the continuous flow of bits along the channels [4–6]. Thus, through every active incoming channel, bits arrive at a constant rate (the transmission channel capacity) and are stored in the corresponding input buffer. The task of the switching component is to serve these bits and route them to the appropriate outgoing ports. The service rate is assumed to be at least as large as the aggregate arrival rate, since otherwise arrival patterns that will cause at least one of the finite buffers to overflow may be easily constructed. The data unit is a variable length packet that consists of not more than L bits. Since packet switching is employed, there is a restriction of serving only complete packets, and for efficiency and practical purposes no preemption is allowed. This implies that a packet cannot be switched to an outgoing port unless the whole packet resides in the input buffer (i.e., the whole packet has already arrived), and once the switching of a packet starts, it cannot be interrupted.

One problem that a designer of such a switching component faces is to determine the service (switching) policy of the packets arriving through each of the incoming channels, and the size of the input buffers needed to reduce potential losses. This problem was first considered in [4], where the Exhaustive Round Robin (ERR) service policy was proposed and analyzed. The ERR discipline was also considered in [11], and in addition a policy called Gated Round Robin (GRR) was introduced in that paper. The First Come First Served (FCFS) discipline was studied in [2]. When the channel speeds are not equal, both ERR and FCFS have been shown to require that the size of the largest buffer increases without bound linearly with the number of channels in order to prevent packet overflow, not a very desirable property.

In this paper we consider another service policy, the Longest Queue First (LQF) discipline. According to this policy, when service of a packet is complete, the next packet to be served is taken from the buffer with the largest number of bits. We derive an upper bound on the size of the largest buffer required at the input channels to guarantee no packet loss, and show that this bound grows logarithmically (not linearly) with the number of incoming channels. We also construct an example that shows the bound is tight.

2. Model description and background material

Consider a switching component with N input channels, each with a corresponding finite input buffer. Let S_i, $1 \leqslant i \leqslant N$ be the transmission rate of channel i (in bits/s). Bits arriving through the ith channel are stored in its buffer, and if the buffer is full, they are lost. The channel can either be on (receiving bits) or off, and bits arrive gradually into the buffer instead of instantaneously. This gradual input or noninstantaneous input model of arrivals has been used extensively in the analysis of switching systems [1] as well as dams [7]. The data unit is a packet rather than a bit, and every packet consists of a variable number of bits with a maximal length of L bits. We do not include any specific statistical assumptions about the packet lengths or the on/off process of arrivals. Such deterministic models have been



Moshe Sidi received the B.Sc., M.Sc. and the D.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, Israel, in 1975, 1979 and 1982, respectively, all in electrical engineering. From 1975 to 1981 he was a Teaching Assistant and a Teaching Instructor at the Technion. In 1982 he joined the faculty of the Electrical Engineering Department at the Technion. During the academic year 1983–1984 he was a Post-Doctoral Associate at the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, Cambridge, MA. During 1986–1987 he was a visiting scientist at IBM, Thomas J. Watson Research Center, Yorktown Heights, NY. He received the New England Academic Award in 1989. He is a coauthor of the book *Multiple Access Protocols: Performance and Analysis* (Springer, Berlin, 1990). Currently, he serves as the Editor for Communication Networks of the *IEEE Transactions on Communications*, as the Associate Editor for Communication Networks and Computer Networks of the *IEEE Transactions on Information Theory* and as an Editor of the *IEEE/ACM Transactions on Networking*. His research interests are in queueing systems and in computer communication networks.

used, not only for telecommunications applications [4–6], but also in the analysis of various service policies for real-time scheduling in manufacturing systems [8–10].

A server (switch) serves the packets residing in the input buffers at a rate of S (bits/s). If $S < \Sigma_{i=1}^{N} S_i$, then arrival patterns can be constructed so that given any set of finite buffer sizes at least one of the buffers will overflow. Therefore, we clearly require that $S \geqslant \Sigma_{i=1}^{N} S_i$. The server is restricted to serve only complete packets. Thus, if a buffer contains only part of the packet, that packet cannot be served. In addition, packets are served in a nonpreemptive manner, i.e., once the service of a packet starts, it cannot be interrupted. Thus, packet fragments cannot be served. Furthermore, we assume that the server is work-conserving, i.e., it is not idle if there is a complete packet at some buffer. Finally, we assume that initially each buffer contains no more than L bits, so that there are at most NL total bits in the system at any time.

The problem of introducing a service policy and determining the buffer sizes that insure operation without any packet loss was first studied in [4]. The service policy considered there was Exhaustive Round Robin (ERR). Under this policy the input buffers are served in a round robin manner, and once the service of a buffer starts it is exhaustive, i.e., every complete packet in the buffer is served. It has been shown that if the bit arrival rates on all channels are the same, input buffers (for each channel) that can contain $3.35L$ bits are sufficient to insure operation without loss. When the rates are not the same, the size of the largest buffer required depends on the arrival rates and grows linearly with the number of incoming channels. Finally, it has been shown in [4] that when the arrival rates are equal, the buffer sizes should be at least $(2 - 1/N)L$ in order to avoid loss.

Improvements in the bounds for ERR in the equal speed case and bounds for a new policy called Gated Round Robin (GRR) have recently appeared (see [11]). Although GRR is similar to ERR in that channels are served in a round robin fashion, the service at each channel is given in a gated rather than an exhaustive manner. In [11] it is shown that the upper bound for ERR with equal speed channels can be tightened to $3.307L$, while a lower bound for this case is $3.051L$. In the same paper an upper bound of $3L$ to prevent packet loss was found for GRR, again in the case of equal speed channels.

Another natural service policy to consider is First Come First Served (FCFS). It is easy to show that under FCFS the largest buffer storage required also grows linearly with N. This was first demonstrated in [2], and we now briefly review that argument. Define $Q_i(t)$ to be the amount of bits in storage at time $t \geqslant 0$ at channel i. The largest amount of bits in storage for a channel will occur just before service begins at the channel. Consider an arbitrary packet of length $\delta \leqslant L$ whose final bit arrives at channel i when there are B bits from other packets in the system. Thus $B + \delta \leqslant NL$, since there are never more than NL bits in the system. This packet must wait for a time B/S until it is served, where $S \geqslant \Sigma_i S_i$ is the speed of the server. The amount of bits accumulating at channel i during this time is at most $S_i B/S$, so that the queue size at i just prior to service of the packet at an instant t is

$$\begin{aligned} Q_i(t) &\leqslant \frac{S_i B}{S} + \delta \leqslant \frac{S_i}{S}(NL - \delta) + \delta \\ &\leqslant L\left\{1 + (N-1)\frac{S_i}{S}\right\}. \end{aligned}$$

Note that this upper bound is attained for $B = NL - L$, $\delta = L$, and a continuous flow of bits into the system. As is the case for ERR, linear behavior with N occurs. With equal speed channels $(S_i/\Sigma_j S_j = 1/N)$ we obtain the upper bound $Q_i(t) \leqslant (2 - 1/N)L$.

3. Longest queue first

In the previous section we have seen that two natural choices for a service policy, ERR and FCFS, exhibit behavior that is linear in terms of the number of channels. That is, the size of the largest buffer required increases linearly with N without bound. Another promising candidate is the Longest Queue First (LQF) policy. It seems reasonable to want to serve the channel that has the most bits in storage, and therefore decrease its queue. In effect, this gives a higher priority to channels with more bits in storage, which are likely to be the faster input channels. We will show that LQF does have better behavior than

ERR and FCFS. The largest buffer storage needed increases with N without bound, but the behavior is logarithmic instead of linear. An upper bound on the queue size at each channel will be derived, and then a particular choice of system parameters will yield an example that shows the upper bound is attained.

We now describe the operation of the LQF policy. At the end of each service period, the channel to be served is chosen by the following rule.

The next channel to be served is any channel that has the largest number of bits in storage among those with a full packet. If no channel has a full packet waiting, then the server becomes idle.

3.1. Queue size upper bound

We will analyze this policy by finding an upper bound on the queue size for any set of channels at any time t. For a set $\mathscr{J} \subset \{1,\ldots, N\} = \mathscr{N}$, define $S_{\mathscr{J}} = \sum_{i \in \mathscr{J}} S_i$ and $Q_{\mathscr{J}}(t) = \sum_{i \in \mathscr{J}} Q_i(t)$. We seek constants $C_{\mathscr{J}}$ such that $Q_{\mathscr{J}}(t) \leqslant C_{\mathscr{J}}$ for $t \geqslant 0$ and $\mathscr{J} \subset \mathscr{N}$. Define

$$S_{\mathscr{J},k} = \max_{\substack{\mathscr{I} \supset \mathscr{J} \\ |\mathscr{I}| = k}} S_{\mathscr{I}} \quad \text{for } k = |\mathscr{J}|,\ldots, N \tag{1}$$

(thus $S_{\mathscr{J},|\mathscr{J}|} = S_{\mathscr{J}}$). Note that if $\mathscr{J} \subset \mathscr{K}$ and $k \geqslant |\mathscr{K}|$, then $S_{\mathscr{J},k} \geqslant S_{\mathscr{K},k}$. Define

$$C_{\mathscr{J}} = |\mathscr{J}|\, L \left\{ 1 + \sum_{k=|\mathscr{J}|}^{N-1} \frac{S_{\mathscr{J},k}/S}{k} \right\} \tag{2}$$

(thus $C_{\mathscr{N}} = NL$). Since $S_{\mathscr{J},k} \leqslant \sum_{i=1}^{N} S_i \leqslant S$ for all $\mathscr{J}$ and k, we have

$$C_{\mathscr{J}} \leqslant |\mathscr{J}|\, L \left\{ 1 + \sum_{k=|\mathscr{J}|}^{N-1} \frac{1}{k} \right\}. \tag{3}$$

Recall that the behavior of LQF will be examined under the assumption that each channel initially has at most L bits in storage. Note that this is also the case whenever a new busy period begins after the server has been idle. This insures that the total amount of bits in the system at any time $t \geqslant 0$ is at most NL. Another consequence of this assumption is that the initial queue size satisfies $Q_{\mathscr{J}}(0) \leqslant |\mathscr{J}|\, L \leqslant C_{\mathscr{J}}$ for all $\mathscr{J} \subset \mathscr{N}$. We will now prove the following lemma.

Lemma 3.1. *Under the Longest Queue First policy*

$$Q_{\mathscr{J}}(t) \leqslant C_{\mathscr{J}} \quad \textit{for } \mathscr{J} \subset \mathscr{N}, \quad t \geqslant 0. \tag{4}$$

Proof. Let τ_n, $n = 1, 2, \ldots$ be the time when the nth packet is taken into service ($\tau_0 = 0$). We will prove by induction on n that (4) holds for $0 \leqslant t \leqslant \tau_n$. As noted above, the result holds for $t = 0 = \tau_0$. So assume it holds for τ_n, and let $\tau_n < t \leqslant \tau_{n+1}$. Let $\mathscr{J} \subset \mathscr{N}$. If the server is idle at time t, then no buffer can have L or more bits at this instant. Thus, $Q_{\mathscr{J}}(t) < |\mathscr{J}|\, L \leqslant C_{\mathscr{J}}$. Otherwise, the server must be busy at time t, say serving channel j. This channel must have been served continuously during the interval (τ_n, t), because $t \leqslant \tau_{n+1}$. Therefore, $\delta \overset{\text{def}}{=} t - \tau_n \leqslant L/S$. If $j \in \mathscr{J}$, then

$$Q_{\mathscr{J}}(t) \leqslant Q_{\mathscr{J}}(\tau_n) + (S_{\mathscr{J}} - S)\delta \leqslant Q_{\mathscr{J}}(\tau_n) \leqslant C_{\mathscr{J}}$$

by the induction hypothesis.

Consider now the case of $j \notin \mathscr{J}$. Then $Q_{\mathscr{J}}(t) \leqslant Q_{\mathscr{J}}(\tau_n) + S_{\mathscr{J}}\delta$. First suppose that no buffer at τ_n had more than L bits. Then

$$Q_{\mathscr{J}}(t) \leqslant |\mathscr{J}|\, L + S_{\mathscr{J}}\delta \leqslant |\mathscr{J}|\, L \left\{ 1 + \frac{S_{\mathscr{J}}/S}{|\mathscr{J}|} \right\}.$$

Since $S_{\mathscr{J},|\mathscr{J}|} = S_{\mathscr{J}}$ and $|\mathscr{J}| \leqslant N - 1$, we conclude from (2) that $Q_{\mathscr{J}}(t) \leqslant C_{\mathscr{J}}$.

Next suppose that at least one buffer had more than L bits at τ_n. We need to show

$$Q_{\mathscr{J}}(\tau_n) \leqslant C_{\mathscr{J}} - S_{\mathscr{J}}\delta. \tag{5}$$

Suppose not, so that

$$Q_{\mathscr{J}}(\tau_n) > C_{\mathscr{J}} - S_{\mathscr{J}}\delta. \tag{6}$$

Since the channel with the longest queue at τ_n must have had more than L bits, it had a full packet. Thus the channel, say j, chosen for service at τ_n under the LQF policy had the maximal queue size among all channels. That is,

$$Q_j(\tau_n) \geqslant Q_i(\tau_n) \quad \text{for } i = 1,\ldots, N. \tag{7}$$

Summing (7) over $i \in \mathscr{J}$, we obtain

$$|\mathscr{J}|\, Q_j(\tau_n) \geqslant Q_{\mathscr{J}}(\tau_n).$$

Using (6), we have

$$Q_j(\tau_n) > \frac{C_{\mathscr{J}} - S_{\mathscr{J}}\delta}{|\mathscr{J}|}. \tag{8}$$

Define $\mathscr{K} = \mathscr{J} \cup \{j\}$. Adding (6) and (8) and then using $\delta \leqslant L/S$ yields

$$Q_{\mathscr{K}}(\tau_n) > \frac{|\mathscr{K}|}{|\mathscr{J}|}(C_{\mathscr{J}} - LS_{\mathscr{J}}/S). \tag{9}$$

We now claim that

$$\frac{|\mathscr{K}|}{|\mathscr{J}|}(C_{\mathscr{J}} - LS_{\mathscr{J}}/S) \geqslant C_{\mathscr{K}}. \qquad (10)$$

From (2) we need only show

$$|\mathscr{K}|L\left\{1+\sum_{k=|\mathscr{J}|}^{N-1}\frac{S_{\mathscr{J},k}/S}{k}\right\} - |\mathscr{K}|L\frac{S_{\mathscr{J}}/S}{|\mathscr{J}|} \geqslant |\mathscr{K}|L\left\{1+\sum_{k=|\mathscr{K}|}^{N-1}\frac{S_{\mathscr{K},k}/S}{k}\right\}.$$

Since $S_{\mathscr{J},|\mathscr{J}|} = S_{\mathscr{J}}$, the above inequality reduces to

$$|\mathscr{K}|L\left\{1+\sum_{k=|\mathscr{K}|}^{N-1}\frac{S_{\mathscr{J},k}/S}{k}\right\} \geqslant |\mathscr{K}|L\left\{1+\sum_{k=|\mathscr{K}|}^{N-1}\frac{S_{\mathscr{K},k}/S}{k}\right\},$$

which holds since $\mathscr{J} \subset \mathscr{K}$. This proves (10). From (9) and (10), we obtain

$$Q_{\mathscr{K}}(\tau_n) > C_{\mathscr{K}}$$

contradicting the induction hypothesis. Therefore, (5) holds, and the lemma follows. □

We can now prove the following theorem, which gives an upper bound on queue size at each channel.

Theorem 3.2. *Under the Longest Queue First policy*

$$Q_j(t) \leqslant L\left\{1+\sum_{k=1}^{N-1}\frac{1}{k}\right\} \quad j=1,\ldots, N, \ t \geqslant 0. \qquad (11)$$

Proof. This follows immediately by specializing the results of (3) and Lemma 3.1 to the case of a single channel, that is, $\mathscr{J} = \{j\}$. □

This theorem illustrates that the size of the largest buffer increases at most logarithmically with respect to N for the LQF policy. In fact, if the buffer at a channel is large enough to hold $L\{2+\ln(N-1)\}$ bits, then the buffer will never overflow, regardless of the relative speeds of the various channels. For equal speed channels $(S_i/\Sigma_j S_j = 1/N)$, note that we obtain the bound $Q_j(t) \leqslant (2-1/N)L$, which is identical with the equal speed upper bound for FCFS. Thus a buffer that can contain two maximal length packets will suffice to prevent overflow. In addition, note that the bounds for ERR, GRR, FCFS and LQF are all independent of N in the equal speed case.

3.2. *Queue size lower bound*

In order to show that the buffer size behavior is indeed logarithmic, we will exhibit a system for which such behavior is attained. This example will be constructed under the assumption of infinitesimal length packets, that is, the ratio between the length L of the longest packet and the length of the shortest packet can be made arbitrarily large. In a recent extension of this work, we have shown that similar examples exhibiting logarithmic behavior can be constructed as long as this ratio is "large enough". However, in those cases the examples become more complex, and thus we will use the above simplifying assumption here. For notational convenience, we order the channels so that $S_1 \leqslant \cdots \leqslant S_N$. We will find time instants $0 < t_1 < t_2 < \cdots < t_N$ such that at t_i, buffers $i, \ldots, N$ have an equal number of bits, say X_i. During the interval (t_i, t_{i+1}), the server will first serve a packet from channel i of (maximum) length L bits, and then spend the remainder of the interval equalizing the queue length at channels $i+1, \ldots, N$ by serving infinitesimal length packets from channels $i+2, \ldots, N$. Finally, for a specific choice of the speeds S_i, the amount of bits at time t_N in the buffer of the fastest channel will be shown to increase logarithmically with N.

Formally, we construct a worst case scenario as follows. We assume that the channel speeds and the server speed satisfy $\Sigma_i S_i = S$. We also assume that at $t=0$ each buffer has at most L bits in storage and that there is a continuous flow of bits into the system. Using infinitesimal length packets, we can construct a time $t_1 > 0$ such that all channels have $X_1 = L$ bits in queue, and the first (slowest) channel has a maximum length packet. The L bit packet at channel 1 is served, and then queues $3, \ldots, N$ are equalized with queue 2 by serving infinitesimal length packets from these queues. After equalization at time $t_2 > t_1$, buffers $2, \ldots, N$ will have an equal number $X_2 > X_1$ of bits, buffer 1 will have less than X_2 bits and buffer 2 will have a maximal length packet of size

L. Continuing in this manner, we see that channel i will never be served during the interval (t_{i+1}, t_N).

To determine the value of X_i, we analyze the behavior of the system during the interval (t_i, t_{i+1}). At t_i queues $i, \ldots, N$ have X_i bits, and queues $1, \ldots, i-1$ have less than that amount. First, a packet of L bits from channel i is served, which takes time L/S. The amount of additional bits in queues $i+1, \ldots, N$ after this service is $S_{i+1}L/S, \ldots, S_N L/S$. We now spend a time T_i to equalize queues $i+1, \ldots, N$. This may be done under the LQF policy by serving infinitesimal length packets, since the slower channels $1, \ldots, i$ have smaller queue length than the faster channels. Note that queue $i+1$ is not served during this time period. When equalization occurs, we set $t_{i+1} = t_i + L/S + T_i$. The total amount of bits entering queue $i+1$ during (t_i, t_{i+1}) is $S_{i+1}(L/S + T_i)$, while the total number of bits entering queues $i+2$ through N is $\Sigma_{j=i+2}^N S_j(L/S + T_i)$. The total amount of bits leaving queue $i+1$ during (t_i, t_{i+1}) is 0, while the total number of bits leaving queues $i+2$ through N is ST_i. At t_{i+1} queues $i+1, \ldots, N$ are equal, and so

$$S_{i+1}(L/S + T_i) = \frac{\Sigma_{j=i+2}^N S_j(L/S + T_i) - ST_i}{N-(i+1)}.$$

Since $\Sigma_i S_i = S$, we obtain

$$(N-i-1)S_{i+1}(L/S + T_i) = L - \sum_{j=1}^{i+1} S_j(L/S + T_i)$$

or

$$(L/S + T_i)\left\{(N-i)S_{i+1} + \sum_{j=1}^{i} S_j\right\} = L.$$

Thus we have

$$X_{i+1} - X_i = S_{i+1}(L/S + T_i) = \frac{L}{(N-i) + \Sigma_{j=1}^i S_j/S_{i+1}}.$$

Summing for $i = 1, \ldots, N-1$ and using $X_1 = L$, we obtain

$$X_N = L\left\{1 + \sum_{i=1}^{N-1} \frac{1}{(N-i) + \Sigma_{j=1}^i S_j/S_{i+1}}\right\} \quad (12)$$

which is $Q_N(t_N)$, the number of bits in buffer N at time t_N.

We now make a particular choice for the channel speeds S_i. Let $C \geqslant 1$ be a constant. Pick S_i so that $S_{i+1} \geqslant C\Sigma_{j=1}^i S_j$. We see that this may be done by choosing $S_2, \ldots, S_N$ in turn in terms of S_1 which satisfy the above inequalities, and then determining the value of S_1 through the constraint $\Sigma_i S_i = S$. For such a choice of channel speeds we have $\Sigma_{j=1}^i S_j/S_{i+1} \leqslant 1/C$, so that (12) becomes

$$Q_N(t_N) \geqslant L\left\{1 + \sum_{i=1}^{N-1} \frac{1}{(N-i) + 1/C}\right\} = L\left\{1 + \sum_{i=1}^{N-1} \frac{1}{i + 1/C}\right\} \stackrel{\text{def}}{=} \text{LB}(C).$$

We have constructed an example of a system for which the buffer size required at the fastest channel is at least LB(C), thus exhibiting the promised logarithmic behavior. An interesting case is obtained by letting $C \to \infty$. Recall from (11) that an upper bound on maximal buffer size is $\text{UB} \stackrel{\text{def}}{=} L\{1 + \Sigma_{i=1}^{N-1} 1/i\}$. Noting that $\lim_{C\to\infty}\text{LB}(C) = \text{UB}$, the above construction shows that the bounds we have obtained are tight. That is, we have the following theorem.

Theorem 3.3. *Given $\epsilon > 0$, a set of channel speeds can be chosen so that the corresponding system, when operating under the Longest Queue First policy, requires a buffer size at the fastest channel of at least* UB $- \epsilon$ *to prevent packet overflow.*

4. Discussion

In this paper we have introduced and analyzed the Longest Queue First (LQF) policy for servicing packets that reside in the input buffers of a switch. According to this policy, when the service of a packet is complete, the next packet to be served is taken from any buffer with the largest number of bits among those that contain a full packet. If there is no full packet in any buffer, then the server becomes idle. We derived an upper bound on the size of the largest buffer required at the input channels to guarantee no packet loss, and we showed that this bound grows logarithmically with the number of incoming channels. We also constructed an example that shows this bound is tight.

The advantages of logarithmic growth for the LQF service policy compared to the linear growth for the ERR and FCFS service policies are obvious. The size of the largest buffer that guarantees no loss is much smaller with LQF than with ERR and FCFS. However, to gain this advantage, the switch should be capable of determining the number of bits in each of the incoming buffers at the end of service of each packet. With ERR the switch is a bit simpler, since data about queue lengths of all buffers is not needed.

In a subsequent paper [3], a service policy is introduced that guarantees no loss if each input buffer can accommodate only two maximal length packets, for any number of channels and for any set of transmission rates. However, the implementation of this policy is more complicated than LQF, since it requires knowledge of the transmission rates of the various channels in addition to the capability of determining the number of bits in each of the buffers at a service completion.

References

[1] D. Anick, D. Mitra and M.M. Sondhi, Stochastic theory of a data-handling system with multiple sources, *Bell System Tech. J.* **61**(8) (1982) 1871–1894.

[2] A. Birman, P.C. Chang, J.S.C. Chen and R. Guérin, Buffer sizing in an ISDN frame relay switch, IBM Research Report, RC 14386, August 1989.

[3] A. Birman, H.R. Gail, S.L. Hantler, Z. Rosberg and M. Sidi, An optimal service policy for buffer systems, IBM Research Report, RC 16641, March 1991.

[4] I. Cidon, I. Gopal, G. Grover and M. Sidi, Real time packet switching: a performance analysis, *IEEE J. Selected Areas Comm.* **6** (1988) 1576–1586.

[5] R.L. Cruz, A calculus for network delay, Part I: network elements in isolation, *IEEE Trans. Inform. Theory* **37**(1) (1991) 114–131.

[6] R.L. Cruz, A calculus for network delay, Part II: network analysis, *IEEE Trans. Inform. Theory* **37**(1) (1991) 132–141.

[7] D.P. Gaver and R.G. Miller, Limiting distributions for some storage problems, in: Arrow, Karlin and Scarf, eds., *Studies in Applied Probability and Management Science* (Stanford Univ. Press, Stanford, CA, 1962).

[8] P.R. Kumar and T.I. Seidman, Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems, *IEEE Trans. Autom. Control* **35**(3) (1990) 289–298.

[9] S.H. Lu and P.R. Kumar, Distributed scheduling based on due dates and buffer priorities, *IEEE Trans. Autom. Control* **36**(12) (1991) 1406–1416.

[10] J.R. Perkins and P.R. Kumar, Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems, *IEEE Trans. Autom. Control* **34**(2) (1989) 139–148.

[11] G. Sasaki, Input buffer requirements for round robin polling systems, *Proc. 27th Annual Allerton Conference on Communication, Control, and Computing*, 1989, pp. 397–406.