P. J. Schweitzer
S. S. Lam

Buffer Overflow in a Store-and-Forward Network Node

Abstract: Equilibrium behavior of a store-and-forward network node with finite buffer capacity is studied via a network-of-queues model. The positive acknowledgment protocol is explicitly modeled and consumes part of the buffer pool. The principal results are the buffer overflow probability, the mean delays, and the distribution of queue lengths as functions of the buffer capacity and traffic levels.

Introduction

Previous queuing analyses of store-and-forward (S&F) networks assume node independence, Poisson arrivals to each node, and infinite buffer capacity at each node [1, 2]. These assumptions, justified for low traffic levels, lead to convenient decoupled M/G/1 queuing models for each line or channel.

This paper examines the case of finite nodal buffer capacity. Since we retain the assumptions of node independence and Poisson arrivals to each node, our results are valid for moderate traffic levels or nodes with several input lines. To relax these assumptions requires solution of various network functional dependencies.

Our main results are the buffer overflow probability, mean delays, and distribution of queue lengths as functions of the buffer capacity and traffic levels at the S&F node. The main differences between our model and models of finite-capacity statistical multiplexors or demultiplexors [3, 4] are the incorporation of blocking by neighboring nodes and explicit employment of a portion of the buffer pool for packet retention until receipt of positive acknowledgment (ACK); numerical calculations show that these processes may contribute significantly to buffer usage. A similar finite-capacity buffer allocation model without the ACK protocol is under investigation by Irland [5]. For a detailed description of one set of protocols, and for acknowledgment and flow control, see the ARPANET documentation [6, 7].

Loss model description

A node with a pool of N buffers (room for N packets) is considered. This and the next section describe a "loss" model where any arriving messages are lost if all N buffers are full. A subsequent section describes the modifications needed for a "repeat" model, where packets rejected by this node are timed-out by their senders and retransmission is repeatedly attempted until ultimate acceptance.

We assume that there is only one class of messages (no priorities), and that all messages are single packets (no message segmenting or message reassembly). The buffer size must be sufficiently large that there is negligible probability that a packet will not fit in one buffer.

The configuration is shown in Fig. 1. Node 0, under investigation, has one or more input lines and $L \geq 1$ output lines. Lines can be either common carrier facilities or channel attachments to locally-connected terminals and hosts. Thus, some of the traffic at the node can have local origins and/or destinations, and communication between a pair of local devices is permitted. A neighboring node or attached device may have both an input line to, and an output line from, node 0; when this happens, the line and neighboring node or attached device are assumed to be full duplex, because the model treats every incoming and outgoing line from node 0 as independent.

There are λ_i packets/s arriving at node 0 for output line i, $i \leq i \leq L$. We let $\lambda = \lambda_i + \cdots + \lambda_L$ denote the total offered traffic rate and $P_i = \lambda_i / \lambda$ denote the fraction of the traffic headed for line i. All arrivals are assumed Poisson, and all packet lengths are assumed independently and exponentially distributed. The assumption that inter-arrival times and packet lengths are statistically independent rules out the case of very heavy traffic where successive packets can arrive contiguously. These are the same as Kleinrock's independence assumptions [1].

An incoming packet is admitted if one of the N buffers is free; otherwise it is lost. This is assumed to be true for both local and remote sources, and for all line speeds. Hence the P_i describe the branching ratios for both offered and admitted traffic, and

$$\lambda_i^{\text{out}} = \lambda_i(1 - B) = \lambda P_i(1 - B), \qquad i = 1, \cdots, L, \qquad (1)$$

denotes the number of packets/s admitted to line i. Here B is the probability that the node is blocked (all N buffers are full); its calculation is one of the main results of this paper. The node throughput, or total admitted traffic, is

$$\lambda^{\text{out}} = \lambda(1 - B) \text{ packets/s.} \tag{2}$$

An admitted packet is stored in one buffer and queued (via pointers) for service at the node processor. Here the checksum is verified, a routing decision is made, and the header is revised. The packet is then queued for transmission over the appropriate output line. The processor operates at a speed of S_0 bits/s, and serves the queued packets in a first-come first-served (FCFS) order.

Output line i, $1 \leq i \leq L$, operates at the speed of S_i bits/s and transmits the queued packets in FCFS order. When a packet transmission is completed, the buffer is *not* freed. Instead, a copy of the packet is retained at node 0 until either a positive acknowledgment (ACK) is received, or until a pre-specified timeout interval is exhausted. These outcomes occur with probabilities $1 - f_i > 0$ and $f_i < 1$, respectively, where f_i is assumed to be known and constant for every packet sent over line i.

It is assumed that the ACKs sent to node 0 from its neighbors will always be accepted by node 0 without blocking and without increasing the offered load. Consequently, the probability $1 - f_i$ of receiving an ACK is the probability of three events: The packet was transmitted over line i without error; the node or device at the receiving end of line i was not blocked, and generated the ACK promptly enough to arrive before the timeout clock expired; and the ACK was transmitted to node 0 without line error. These three events are usually considered independent; consequently, their probabilities multiply. The model, therefore, computes the blocking probability of node 0 in terms of the blocking probabilities of its neighboring nodes.

We let T_i^{TO} denote the mean timeout interval selected for line i, and T_i^{ACK} denote the mean time until an ACK arrives, given that an ACK rather than a timeout will occur. The associated random variables may have arbitrary distributions, subject only to having rational Laplace transforms. In practice, the timeout distribution will be highly concentrated at T_i^{TO}, and the ACK distribution will be unimodal with support on $[0, T_i^{\text{TO}}]$. The result is that the main quantity of interest is

$$T_i^H \equiv [f_i\, T_i^{\text{TO}}/(1 - f_i)] + T_i^{\text{ACK}},$$

namely the mean total holding time for all timeouts and the one ultimate ACK.

The times-till-ACK for successive packets transmitted over line i are assumed to be statistically independent and identically distributed; the mean ACK time T_i^{ACK} is therefore well-defined. This assumption is valid if the node or device receiving traffic from line i generates ACKs promptly, but if lags occur under heavy traffic conditions, a possible correlation phenomenon would require study by a two-node model.

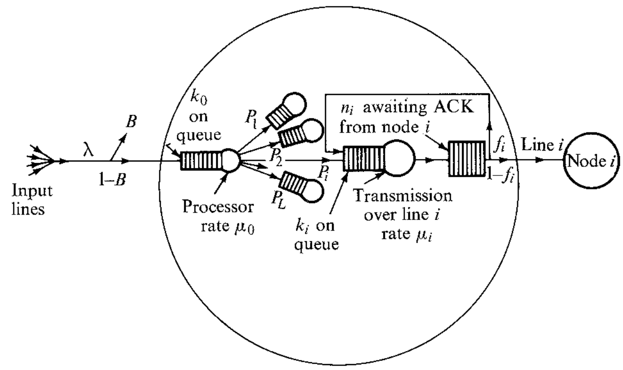


Figure 1 Configuration: Node 0 with L output lines.

If a packet transmitted over line i receives an ACK, the buffer is freed for further use. If the packet is timed-out, it is then placed (by pointer) at the tail of the queue of packets awaiting transmission over line i; the number of occupied buffers remains unchanged.

The model assumes that each admitted packet is permitted an *infinite* number of retransmission attempts; hence, it is eventually transmitted successfully. The average number of transmission attempts over line i will be $1/(1 - f_i)$. The assumption is reasonable if the failure probability per attempt f_i is less than, for example, 0.6–0.8. This is because, on the average, fewer than 2–5 attempts will suffice; any line with f_i exceeding this magnitude would normally be regarded as unusable.

Analytic results for loss model

The above processes in node 0 can be modeled as an open network of queues [8] with $3L + 1$ service centers. The state $(k_0, k_1, \cdots, k_L, m_1, \cdots, m_L, l_1, \cdots, l_L)$ of the network is defined as follows.

k_0 ≡ Number of packets queued for service at the processor, which is modeled as a FCFS single exponential server with rate $\mu_0 = S_0/A$ packets/s, where A denotes the mean number of bits per packet. If $k_0 \geq 1$, the first packet is being processed;

k_i ≡ number of packets queued for transmission over line i, $1 \leq i \leq L$, which is modeled as a FCFS single exponential server having rate $\mu_i = S_i/A$ packets/s. If $k_i \geq 1$, the first packet is being transmitted;

m_i ≡ number of packets transmitted over line i, $1 \leq i \leq L$, and awaiting ACK, which in fact will be successfully ACKed. This service facility is modeled as one without queuing, namely as an infinite number of parallel servers, each with mean service time T_i^{ACK};

l_i $\equiv$ number of packets transmitted over line i, $1 \le i \le L$, and awaiting ACK, which is fact will be timed-out. This service facility is also modeled as having an infinite number of parallel servers, each with mean service time T_i^{TO}.

The following auxiliary variables are also needed.

n_i $\equiv m_i + l_i =$ Number of packets transmitted over line i and awaiting ACK (only n_i is observable, m_i or l_i are not observable), $1 \le i \le L$;

$k_{sum} \equiv \Sigma_{i=0}^{L} k_i =$ total number of packets queued at the processor and output lines;

$n_{\mathrm{sum}} \equiv \Sigma_{i=1}^{L} n_i =$ total number of packets awaiting ACK;

s $\equiv k_{\mathrm{sum}} + n_{\mathrm{sum}} \equiv$ number of occupied buffers;

B $\equiv \Pr[s = N] \equiv$ steady state buffer overflow probability for node 0.

The transcription to the network-of-queues formulation operates as follows. The arrival rate to the system is

$$\lambda(s) = \begin{cases} \lambda & 0 \le s \le N-1 \\ 0 & s = N \text{ (arrivals lost when blocked)}, \end{cases}$$

and all arrivals appear at the processor queue.

The branching probabilities for the model are as follows. When finished at the processor, a packet goes with probability P_i to the queue for transmission line i: k_0 drops by 1 and k_i increases by 1. After transmission over line i, the probabilities are $1-f_i$ and f_i for ACK or TO; k_i drops by 1 and either m_i or l_i, respectively, increases by 1. After arrival of an ACK from line i, m_i drops by 1 and the packet leaves the system. After expiration of the timeout clock for line i, the packet is re-queued for transmission over line i: l_i drops by 1 and k_i increases by 1.

Because each of the $3L+1$ service centers has either a FCFS exponential server or an infinite number of parallel servers whose service time probability density has a rational Laplace transform, local balance conditions hold and the steady-state joint-state probability $P[k_0, \cdots, k_L, m_1, \cdots, m_L, l_1, \cdots, l_L]$ has a product form [8]:

$$P[k_0, \cdots, k_L, m_1, \cdots, m_L, l_1, \cdots, l_L] = P(0) \left[\prod_{i=0}^{L} (\rho_i)^{k_i} \right] \prod_{i=1}^{L} \left[\frac{(a_i)^{m_i}}{m_i!} \frac{(b_i)^{l_i}}{l_i!} \right];$$
$$k_i, m_i, l_i \ge 0; \qquad s \le N, \tag{3}$$

where

$$\rho_0 \equiv \lambda / \mu_0;$$
$$\rho_i \equiv \lambda_i / (\mu_i(1-f_i)) = \lambda P_i / (\mu_i(1-f_i)), \qquad 1 \le i \le L;$$
$$a_i \equiv \lambda_i T_i^{\mathrm{ACK}} = \lambda P_i T_i^{\mathrm{ACK}}, \qquad 1 \le i \le L;$$
$$b_i \equiv \lambda_i f_i T_i^{\mathrm{TO}} / (1-f_i) = \lambda P_i f_i T_i^{\mathrm{TO}} / (1-f_i), \qquad 1 \le i \le L.$$

The normalization factor $P(0)$, which is the probability of an empty node, is determined by the normalization condition

$$1 = \sum_{\substack{k_0, \cdots, k_L, m_1, \cdots, \\ m_L, l_1, \cdots, l_L \ge 0}} P[k_0, \cdots, k_L, m_1, \cdots, m_L, l_1, \cdots, l_L]; \; s \le N. \tag{4}$$

Note that if all offered traffic were accepted, then ρ_0 and ρ_i, $1 \le i \le L$, would represent the server utilizations at the processor and ith output line. Since some of the offered traffic is rejected, the ρ exceed the server utilizations and are permitted to exceed unity. However, in practice the offered traffic should satisfy

$$\lambda < \lambda^* \equiv \min \left[\mu_0, \min_{1 \le i \le L} \mu_i (1-f_i) / P_i \right] \tag{5}$$

so that $\rho_i < 1$ for $i = 0, 1, \cdots$, L. Otherwise one of the FCFS servers (processor or line) will be unable to handle all the traffic offered it, and this will result in a very high buffer overflow probability.

The node output rate satisfies

$$\lambda^{\mathrm{out}} = \lambda(1-B) \le \lambda^* \qquad \text{(if } N \text{ is finite)}, \tag{6}$$

showing that λ^* is the maximum possible throughput of node 0, achievable only when both N and $\lambda \to \infty$. Equation (6) is derived by noting that the processor serves $\lambda(1-B) \le \mu_0$ packets/s, while the ith line transmits $\lambda P_i(1-B)/(1-f_i) \le \mu_i$ packets/s. However, a fraction $1-f_i$ of these transmissions is successful. Together these show that $\lambda(1-B) \le \lambda^*$.

The results tabulated below, and derived in the Appendix, assume for simplicity that $\{1, \rho_0, \rho_1, \cdots, \rho_L\}$ are all distinct [9]. If confluence occurs, one has a choice of the following.

1. Appropriate derivatives of the analytic expressions given below [10];
2. recursive computation of the quantities of interest [10, 11, 12];
3. deliberate perturbation of the μ_i to avoid confluence, and extrapolation of the numerical results as the perturbation approaches zero.

The third approach is the simplest, and perturbations on the order of 0.01–0.1 percent have been found to yield satisfactory results.

The expressions given below are sums or products over the L output lines. The computational effort is modest since $1 \le L \le 4$ in practice, and the expressions are well-behaved away from the confluence of $\{1, \rho_0, \rho_1, \cdots, \rho_L\}$.

It is frequently possible to neglect the queue for the processor, because the processor speed is usually significantly higher than the output line speed. The following expressions can be adapted to the case of infinite processor speed by setting

$$k_0 \leftarrow 0,\ \mu_0 \leftarrow \infty,\ \rho_0 \leftarrow 0,\ \sum_{i=0}^{L} \leftarrow \sum_{i=1}^{L},\ \prod_{i=0}^{L} \leftarrow \prod_{i=1}^{L}$$

$$h(i) \leftarrow \prod_{\substack{j=1\\ j\neq i}}^{L} (1-\rho_j/\rho_i) \qquad \text{for } i=1,\cdots,L$$

(set $h(i)=1$ if $L=1$).

The main results are as follows. The normalization factor $P(0)$ is given by

$$\frac{1}{P(0)} = \sum_{i=0}^{L} \frac{(\rho_i)^{N+1}\, \mathrm{E}_N(\lambda T^H/\rho_i)}{h(i)\ (\rho_i-1)} + \mathrm{E}_N(\lambda T^H)/y_1 \tag{7}$$

where

$$h(i) \equiv \prod_{\substack{j=0\\ j\neq i}}^{L} (1-\rho_j/\rho_i) \qquad i=0,1,\cdots,L$$

$$\mathrm{E}_N(x) \equiv \sum_{i=0}^{N} x^i/i! \qquad N=0,1,2,\cdots$$

$$y_1 \equiv \prod_{i=0}^{L} (1-\rho_i)$$

$$T^H \equiv \sum_{i=1}^{L} P_i\, T_i^H = \text{mean holding time.}$$

The joint distribution $P[k_0,\cdots,k_L,n_1,\cdots,n_L]$ of packets awaiting service or ACK is given by

$$P[k_0,\cdots,k_L,n_1,\cdots,n_L] = P(0)\left[\prod_{i=0}^{L}(\rho_i)^{k_i}\right]\prod_{i=1}^{L}\left[\frac{(c_i)^{n_i}}{n_i!}\right], \qquad s\le N, \tag{8}$$

where

$$c_i \equiv a_i + b_i = \lambda P_i\, T_i^H \qquad 1\le i\le L.$$

The joint distribution of $k=k_{\text{sum}}$ and $n=n_{\text{sum}}$ is given by

$$P[k,n] = P(0)\left[\sum_{i=0}^{L}(\rho_i)^k/h(i)\right](\lambda T^H)^n/n! \qquad k,n\ge 0,\ k+n\le N. \tag{9}$$

The probability that exactly s buffers are occupied is

$$P[s] = P(0)\sum_{i=0}^{L}\frac{(\rho_i)^s}{h(i)}\,\mathrm{E}_s(\lambda T^H/\rho_i) \qquad s=0,1,2,\cdots,N. \tag{10}$$

The correct result $P[s=0]=P(0)$ follows from $\Sigma_{i=0}^{L} 1/h(i)=1$, obtained by setting $z=0$ in Eq. (A1). Of special interest, the node blocking probability is

$$B \equiv \Pr[s=N] = P(0)\sum_{i=0}^{L}\frac{(\rho_i)^N}{h(i)}\,\mathrm{E}_N(\lambda T^H/\rho_i). \tag{11}$$

The node throughput λ^{out} can be calculated from Eq. (2) or from the relationships

$$\lambda^{\text{out}} = \mu_0 \Pr[k_0\ge 1] = \sum_{i=1}^{L}\mu_i(1-f_i)\Pr[k_i\ge 1].$$

The mean queue lengths are given by

$$\mathrm{E}[m_i] = a_i \Pr[s\le N-1] = \lambda P_i\, T_i^{\text{ACK}}(1-B), \qquad 1\le i\le L; \tag{12}$$

$$\mathrm{E}[l_i] = b_i \Pr[s\le N-1] = \lambda P_i f_i\, T_i^{\text{TO}}(1-B)/(1-f_i), \qquad 1\le i\le L; \tag{13}$$

$$\mathrm{E}[n_i] = \mathrm{E}[m_i+l_i] = (a_i+b_i)\Pr[s\le N-1] = \lambda P_i\, T_i^H(1-B), \qquad 1\le i\le L; \tag{14}$$

$$\mathrm{E}[k_i] = P(0)\left[\frac{\rho_i}{1-\rho_i}\frac{\mathrm{E}_N(\lambda T^H)}{y_1} - A_i\,\mathrm{E}_N(\lambda T^H/\rho_i) + C_i\,\mathrm{E}_{N-1}(\lambda T^H/\rho_i) + F_i\right], \qquad 0\le i\le L, \tag{15}$$

where

$$A_i \equiv \frac{(\rho_i)^{N+1}}{h(i)(1-\rho_i)}\left[N+\frac{1}{1-\rho_i}-\sum_{\substack{j=0\\ j\neq i}}^{L}\frac{\rho_j/\rho_i}{1-\rho_j/\rho_i}\right];$$

$$C_i \equiv \frac{\lambda T^H(\rho_i)^N}{h(i)(1-\rho_i)};$$

$$F_i \equiv \sum_{\substack{j=0\\ j\neq i}}^{L}\frac{(\rho_j)^{N+1}\,\mathrm{E}_N(\lambda T^H/\rho_j)}{h(j)(1-\rho_j/\rho_i)(1-\rho_j)}.$$

The mean and mean square number of occupied buffers are given by

$$\mathrm{E}[s] = P(0)\left[\frac{y_2}{y_1}\mathrm{E}_N(\lambda T^H) + \frac{\lambda T^H}{y_1}\mathrm{E}_{N-1}(\lambda T^H) + \sum_{i=0}^{L}\frac{N(\rho_i)^{N+2}-(N+1)(\rho_i)^{N+1}}{h(i)(1-\rho_i)^2}\,\mathrm{E}_N\!\left(\frac{\lambda T^H}{\rho_i}\right)\right]; \tag{16}$$

$$\begin{aligned}\mathrm{E}[s^2] &= P(0)\ [(\lambda T^H)^2\,\mathrm{E}_{N-2}(\lambda T^H) + 2(1+y_2)\lambda T^H\,\mathrm{E}_{N-1}(\lambda T^H) + (3y_2+y_3)\,\mathrm{E}_N(\lambda T^H)]/y_1\\ &+ P(0)\sum_{i=0}^{L}\left[\frac{(N+1)^2(\rho_i)^{N+4}-(2N^2+5N+4)(\rho_i)^{N+3}}{h(i)\ (\rho_i-1)^3} + \frac{(N^2+2N+4)(\rho_i)^{N+2}+(N+1)(\rho_i)^{N+1}}{h(i)(\rho_i-1)^3}\right]\\ &\times \mathrm{E}_N\!\left(\frac{\lambda T^H}{\rho_i}\right),\end{aligned} \tag{17}$$

where $y_2 \equiv \Sigma_{i=0}^{L}\rho_i/(1-\rho_i)$, $y_3 \equiv (y_2)^2 + \Sigma_{i=0}^{L}[\rho_i/(1-\rho_i)]^2$, and $\mathrm{E}_{-1}(x) \equiv \mathrm{E}_{-2}(x) \equiv 0$. The buffer utilization may then be calculated as $\mathrm{E}[s]/N$.

The expected delay D_i at node 0 for a packet headed toward line i, measured from admission to node 0 until

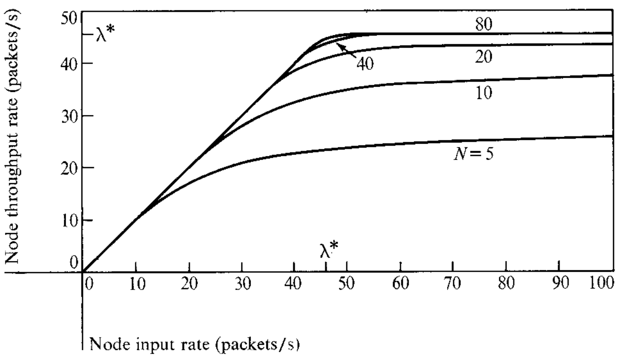


Figure 2 Node throughput vs offered traffic.

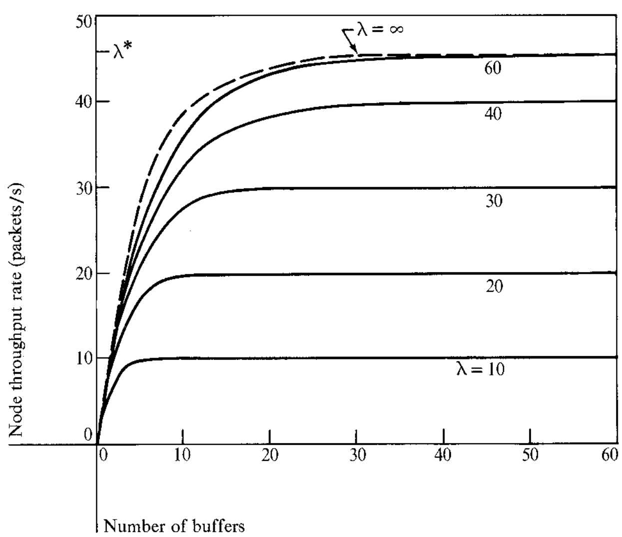


Figure 3 Node throughput vs buffer capacity.

completion of successful transmission over line i, is given by Little's formula [13] as

$$D_i = \mathrm{E}[k_0]/\lambda^{\text{out}} + \mathrm{E}[k_i]/\lambda_i^{\text{out}} + \mathrm{E}[l_i]/\lambda_i^{\text{out}}, \qquad 1 \le i \le L,$$

where the throughput traffic rates λ^{out}, λ_i^{out} are defined by Eqs. (1) and (2), and the expected queue lengths are given by Eqs. (13) and (15). The three components of the delay, described respectively, are 1) a processor delay which, as mentioned above, may frequently be neglected, 2) a queuing and transmission time for an expected $1/(1-f_i)\ \lambda$ transmission attempts over line i, and 3) a wait $\mathrm{E}[l_i]/\lambda_i^{\text{out}} = f_i\ T_i^{\text{TO}}/(1-f_i)$ for an expected $f_i/(1-f_i)$ timeouts.

Buffer behavior as $N \to \infty$ is readily obtained in the non-saturated case where $\lambda < \lambda^*$ because every $(\rho_i)^N \to 0$ and $\mathrm{E}_N(x) \to e^x$. We find $P(0) \to y_1 \exp(-\lambda T^H)$, $B \to 0$, and

$$\mathrm{E}[s] \to y_2 + \lambda T^H = \sum_{i=0}^{L} \rho_i/(1-\rho_i) + \sum_{i=1}^{L} \lambda_i T_i^{\text{ACK}} + \sum_{i=1}^{L} \lambda_i f_i\ T_i^{\text{TO}}/(1-f_i),$$

$$\mathrm{E}[s^2] \to (\lambda T^H)^2 + 2(1+y_2)\ \lambda T^H + 3y_2 + y_3,$$

$$\sigma^2(s) \equiv \mathrm{E}[s^2] - \mathrm{E}[s]^2 \to 2\lambda T^H + \sum_{i=0}^{L}\left[\frac{3\rho_i}{1-\rho_i} + \left(\frac{\rho_i}{1-\rho_i}\right)^2\right].$$

The $3L + 1$ contributions to $\mathrm{E}[s]$ represent the mean number of packets at each of the $3L+1$ service centers. This is consistent with $\mathrm{E}[m_i] \to a_i$, $\mathrm{E}[l_i] \to b_i$, and $\mathrm{E}[k_i] \to \rho_i/(1-\rho_i)$.

To determine buffer behavior as the offered load λ approaches 0 or infinity, assume that the proportions $P_1, \cdots, P_L$ and the mean holding times T_i^H remain constant. Put $\rho_i = g_i\lambda$, where $g_0 \equiv 1/\mu_0$, and where $g_i \equiv P_i/(\mu_i(1-f_i))$, $1 \le i \le L$. Note that the g_i and $h(i)$ are independent of λ, and that Eq. (10) implies

$$P[s] = \lambda^s P(0) q_s = \frac{\lambda^s q_s}{\sum_{j=0}^{N} \lambda^j q_j}, \qquad s = 0, 1, 2, \cdots, N,$$

where $q_s \equiv \sum_{i=0}^{L} (g_i)^s\ E_s(T^H/g_i)/h(i)$ is independent of λ and N. In particular, the ratios $P[s]/P[s']$ scale as $\lambda^{s-s'}$. For light loads this yields, since $q_0 = 1$,

$$P[s=0] = 1 - q_1\lambda + 0(\lambda^2),$$

$$B = q_N\lambda^N + 0(\lambda^{N+1}),$$

$$\lambda^{\text{out}} = \lambda - q_N\lambda^{N+1} + 0(\lambda^{N+2}). \tag{18}$$

Since N is usually large, Eq. (18) suggests that blocking is negligible if λ is below a threshold, but that saturation is rapidly approached as λ advances beyond this threshold. Figures 2 and 4, discussed in the following section, illustrate this behavior.

As $\lambda \to \infty$, only $s = N$ and $s = N-1$ have appreciable probabilities, so that

$$B = \Pr[s=N] = 1 - U_N/\lambda + 0(1/\lambda^2),$$

$$\Pr[s=N-1] = U_N/\lambda + 0(1/\lambda^2), \tag{19}$$

where

$$U_N = q_{N-1}/q_N. \tag{20}$$

The node throughput is

$$\lambda^{\text{out}} = \lambda[1-B] = U_N + 0(1/\lambda). \tag{21}$$

Thus, U_N packets/s is the maximum possible throughput from an N-buffer node, and is achievable only when $\lambda \to \infty$. Equation (6) may be extended to

$$\lambda^{\text{out}} = \lambda(1-B) \le U_N \le \lambda^*$$

(if N is finite). Also, it follows from Eq. (20) that [14]

$$\lim_{N\to\infty} U_N = \left[\max_{0\le i\le L} g_1\right]^{-1} \equiv \lambda^*.$$

Note also the lower bounds $B \ge 1 - U_N/\lambda \ge 1 - \lambda^*/\lambda$ on the blocking probability.

Numerical example

Numerical calculations were performed for a node with $L = 4$ output lines, whose blocking probabilities and branching probabilities are given by $f_1 = f_2 = f_3 = f_4 = 0.05$, and $(P_1, P_2, P_3, P_4) = (0.15, 0.18, 0.40, \text{and } 0.27)$. An infinitely fast processor is assumed, and line speeds are given by $(\mu_1, \mu_2, \mu_3, \mu_4) = (9.6, 9.6, 19.2, \text{and } 50)$ packets/s; these correspond to 1000-bit packets sent over (9600, 9600, 19200, and 50000) bps lines, with the last possibly representing a channel connection. Mean timeout intervals were taken as $(T_i^{\mathrm{TO}}) = (0.6, 0.6, 0.3, 0)$ s, and mean ACK times as $(T_i^{\mathrm{ACK}}) = (0.12, 0.12, 0.06, 0)$ s, roughly the reciprocals of the linespeeds. Maximal possible throughput of the node is given by Eq. (5) as $\lambda^* = 45.6$ packets/s.

Figure 2 plots λ^{out} against λ, with N as the parameter; the computations are based on Eqs. (2) and (11). The curves show that no more than 60–80 buffers are justified, because linespeed rather than buffering becomes the dominant bottleneck. Each curve has an asymptote U_N, given by Eq. (20). Note the sharp turnover in each curve in the vicinity of $\lambda = U_N$: for N moderately large, the behavior is $\lambda^{\mathrm{out}} \sim \lambda$ for $\lambda < 0.7$–0.8 of U_N, and $\lambda^{\mathrm{out}} \sim U_N$ for $\lambda > 1.3\ U_N$.

Figure 3 plots λ^{out} vs N, with λ as a parameter. The upper envelope, where $\lambda \to \infty$, corresponds to U_N of Eq. (20). This presentation is convenient for selecting buffer sizes; e.g., at most 15 buffers are needed if $\lambda = 20$ packets/s. Nothing is gained by using more than about 60 buffers.

Figures 4 and 5 plot the node blocking probability B vs N and λ. Figure 4 exhibits the same turnover behavior as Fig. 2: $B \ll 1$ provided $\lambda < 0.7\ U_N$, and $B \sim (1 - U_N/\lambda)$ for $\lambda > 1.3\ U_N$. Figure 5 has the same convenience as Fig. 3 for the buffer design, e.g., to accommodate $\lambda = 20$ packets/s. At most, about 13 buffers are needed.

Figure 6 plots buffer utilization $\mathrm{E}(s)/N$ vs λ with N as the parameter; the computations are based on Eq. (16). Note that the buffer utilization approaches unity, regardless of N, as λ approaches λ^*. For an $N = 20$-buffer node with 40 packets/s of offered traffic, Fig. 6 shows a buffer utilization of 60 percent. This consists of a buffer utilization (see Eq. (14) and Fig. 4) of

$$\frac{\mathrm{E}(n_{\mathrm{sum}})}{N} = \frac{\lambda T^H(1-B)}{N} = 16 \text{ percent}$$

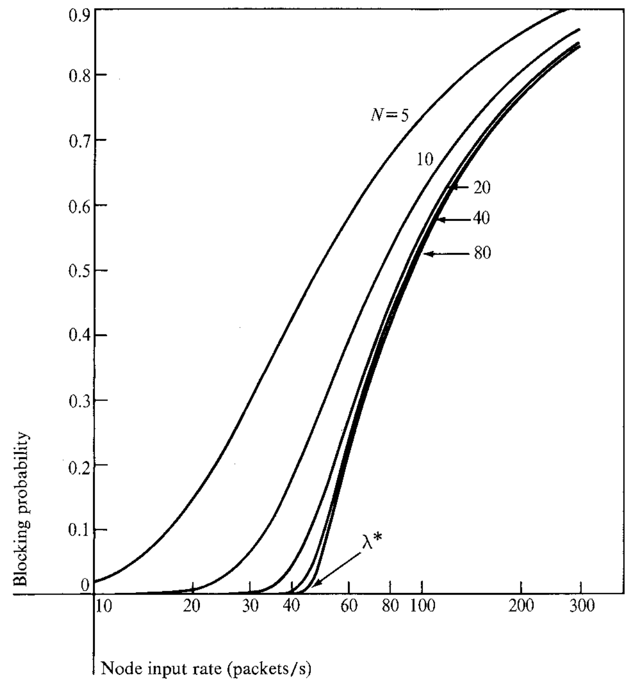
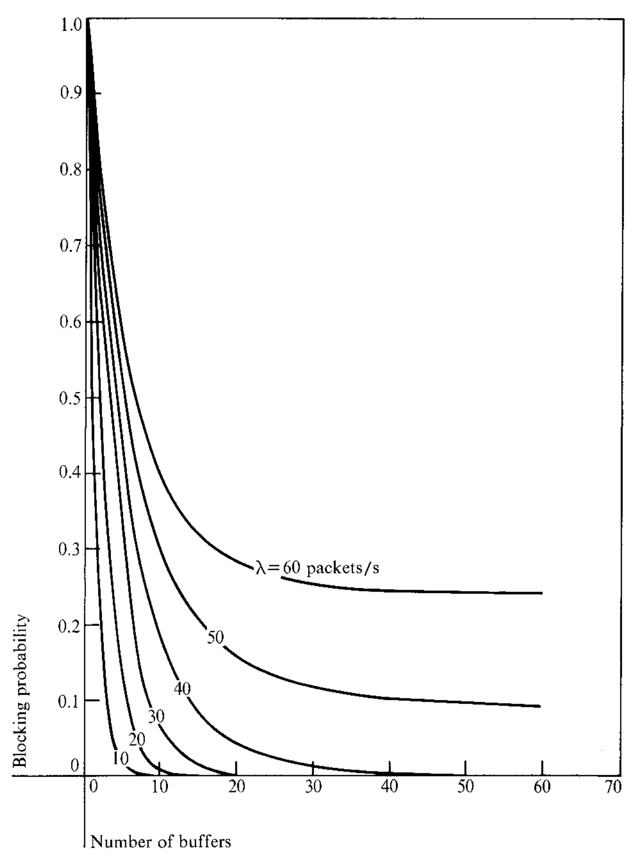


Figure 4 Overflow probability vs offered traffic.

Figure 5 Overflow probability vs buffer capacity.



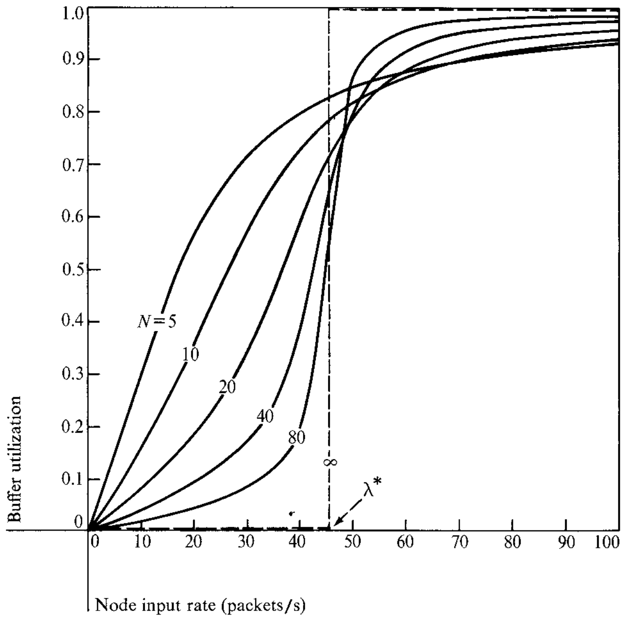


Figure 6 Buffer utilization vs offered traffic.

due to packets awaiting acknowledgments, and a remaining buffer utilization of 44 percent for packets awaiting transmission. Note that the positive acknowledgment protocol contributes significantly to the buffer usage.

Repeat model

In the above loss model, incoming packets rejected by node 0, because of buffer saturation, are permanently lost. In practice, the sending node will timeout and later (repeatedly) attempt retransmission to node 0. If it is assumed that the timeout interval is long compared to all relaxation times at node 0, the effect of the retransmission attempts is to magnify the Poisson arrival stream to node 0 by a factor of $1/(1-B)$.

The following procedure can then be used to determine B and the offered traffic, so that the loss model may still be employed.

1. Given N and the desired net throughputs λ_i^{out}, $1 \leq i \leq L$, on each line emerging from node 0, compute $\lambda^{\text{out}} = \Sigma_{i=1}^{L} \lambda_i^{\text{out}}$ and each $P_i = \lambda_i^{\text{out}}/\lambda^{\text{out}}$;
2. check that $\lambda^{\text{out}} < U_N$; if not, the desired throughputs $[\lambda_i^{\text{out}}]$ cannot be achieved with an N-buffer node.
3. solve the equation $\lambda^{\text{out}} = \lambda(1-B)$, where Eq. (11) shows that the right hand side is a function of λ, for λ, and then compute $1 - B = \lambda^{\text{out}}/\lambda$.
4. the incoming offered traffic for the loss model will be $\lambda = \lambda^{\text{out}}/(1-B)$, of which a fraction $1-B$ will be admitted.

Model extensions and generalizations

1. The present model analyzes one node, and assumes that the overflow probabilities of the neighboring nodes are known. But these in fact depend upon the overflow probability of the node under investigation. In order to analyze a network of nodes, with given traffic rates on every line, an iterative procedure is needed to ensure that all overflow probabilities are self-consistent. A Newton-Raphson procedure, described in [12], has been found satisfactory, and employs the single-node formulation given above. The single-node formulation may be used in isolation, however, if one is investigating the minimal buffer capacities needed to keep all overflow probabilities under a given threshold.
2. Real networks can have non-exponential inter-node arrival patterns, non-exponentially distributed message lengths, multiple message classes with distinct priorities, reassembly of multipacket messages, adaptive routing, and local or end-to-end data flow control protocols. It appears unlikely that these phenomena can be incorporated within the network-of-queues formulation.
3. The present one-node model attempts to capture the congestion at neighboring nodes by static *average* blocking probabilities $f_1, f_2, \cdots, f_L$. In reality, buffer contents at adjacent nodes are positively correlated, and tend to rise and fall together in response to peaks and valleys in the traffic patterns. A multiple-node model is needed for studying such dynamic processes, and determining when the use of static average blocking probabilities will yield adequate predictions.
4. The present model assumes that for a given message, the inter-arrival time, service time at the processor, and service time for each successive transmission attempt are *independently*-distributed random variables. This ignores the fact that these random variables all involve a common message length, hence are dependent. The range of validity of these assumptions must be ascertained, akin to the validation of Kleinrock's independence assumption [1] at low traffic levels and/or under mixing conditions.

Acknowledgment

We thank H. Kobayashi for helpful discussions concerning the network-of-queues model.

Appendix: derivation of analytical results

To derive Eq. (8), add Eq. (3) over all m_i and l_i with $m_i + l_i = n_i$. To derive Eq. (9), note that Eq. (8) implies $P[k, n] = P(0)\, x(k) y(n)$ where

$$x(k) \equiv \sum_{\substack{k_0 \cdots k_L \geq 0 \\ k_0 + \cdots + k_L = k}} \prod_{i=0}^{L} (\rho_i)^{k_i};$$

$$y(n) \equiv \sum_{\substack{n_1,\cdots,n_L \geq 0 \\ n_1+\cdots+n_L=n}} \prod_{i=1}^{L} \frac{(c_i)^{n_i}}{n_i!}.$$

Thus $x(k)$ is the coefficient of z^k in the Maclaurin series expansion of

$$\sum_{k_0,\cdots,k_L=0}^{\infty} \prod_{i=0}^{L} (\rho_i z)^{k_i} = \prod_{i=0}^{L} \frac{1}{(1-\rho_i z)}$$

$$= \sum_{i=0}^{L} \frac{1}{h(i)(1-\rho_i z)}, \tag{A1}$$

where the last step employs a partial fraction expansion, and shows

$$x(k) = \sum_{i=0}^{L} \frac{(\rho_i)^k}{h(i)}. \tag{A2}$$

Similarly, $y(n)$ is the coefficient of z^n in the Maclaurin expansion of

$$\sum_{n_1,\cdots,n_L=0}^{\infty} \prod_{i=0}^{L} \frac{(c_i z)^{n_i}}{n_i!} = \prod_{i=1}^{L} \exp(c_i z) = \exp(z \sum_{i=1}^{L} c_i)$$

$$= \exp(\lambda T^H z).$$

Thus $y(n) = (\lambda T^H)^n/n!$, completing the derivation of Eq. (9).

For later use, we tabulate the value of (A1) and its first two derivatives at $z = 1$:

$$\sum_{i=0}^{L} \frac{1}{h(i)(1-\rho_i)} = \prod_{i=0}^{L} \frac{1}{1-\rho_i} = 1/y_1 \tag{A3}$$

$$\sum_{i=0}^{L} \frac{\rho_i}{h(i)(1-\rho_i)^2} = \sum_{i=0}^{L} \frac{\rho_i}{(1-\rho_i)^2} \prod_{\substack{j=0 \\ j\neq i}}^{L} \frac{1}{(1-\rho_j)}$$

$$= y_2/y_1 \tag{A4}$$

$$2\sum_{i=0}^{L} \frac{(\rho_i)^2}{h(i)(1-\rho_i)^3} = y_3/y_1 \tag{A5}$$

where y_1, y_2, and y_3 are defined in the section on analytic results for loss model.

Equation (10) follows from Eq. (9) because

$$P[s] = \sum_{n=0}^{s} P[s-n, n]$$

$$= P(0) \sum_{i=0}^{L} \frac{(\rho_i)^s}{h(i)} \sum_{n=0}^{s} \frac{(\lambda T^H/\rho_i)^n}{n!}. \tag{A6}$$

Equation (7) follows from Eqs. (4) and (A6) because, after interchanging the summation on n and s,

$$1 = \sum_{s=0}^{N} P[s] = P(0) \sum_{i=0}^{L} \frac{1}{h(i)} \sum_{n=0}^{N} \frac{(\lambda T^H/\rho_i)^n}{n!} \sum_{s=n}^{N} (\rho_i)^s.$$

The innermost sum is $\Sigma_{s=n}^{N} (\rho)^s = (\rho^n - \rho^{N+1})/(1-\rho)$. Therefore,

$$1/P(0) = \sum_{i=0}^{L} \frac{1}{h(i)(1-\rho_i)} [\mathrm{E}_N(\lambda T^H) - (\rho_i)^{N+1} \mathrm{E}_N(\lambda T^H/\rho_i)].$$

The coefficient of $\mathrm{E}_N(\lambda T^H)$ is simplified via Eq. (A3), yielding Eq. (7).

Similarly, to derive Eqs. (16) and (17), employ Eq. (A6) and put

$$\mathrm{E}[s^r] = \sum_{s=0}^{N} s^r P[s]$$

$$= P(0) \sum_{i=0}^{L} \frac{1}{h(i)} \sum_{n=0}^{N} \frac{(\lambda T^H/\rho_i)^n}{n!} \sum_{s=n}^{N} s^r (\rho_i)^s. \tag{A7}$$

The innermost sum is, for $r = 1$ and $r = 2$

$$\sum_{s=n}^{N} s\, \rho^s = \frac{(N+1)\rho^{N+1} - n\rho^n}{\rho - 1} + \frac{\rho^{n+1} - \rho^{N+2}}{(\rho-1)^2} \tag{A8}$$

$$\sum_{s=n}^{N} s^2 \rho^s = \frac{(N+1)^2 \rho^{N+2} - n^2 \rho^n}{\rho - 1} + \frac{(n+1)\rho^{n+1} - (N+2)\rho^{N+2}}{(\rho-1)^2} + \frac{n\rho^n - (N+1)\rho^{N+1}}{(\rho-1)^2} + 2\frac{\rho^{N+2} - \rho^{n+1}}{(\rho-1)^3}. \tag{A9}$$

Insertion of Eqs. (A8) or (A9) into Eq. (A7), summing over n, and employing Eqs. (A3)-(A5) leads to (16) and (17).

The expected queue lengths, Eqs. (12) and (13), are obtained from Eq. (3), as follows:

$$\mathrm{E}[m_j] = \sum_{\substack{k_i m_i l_i \geq 0 \\ s \leq N}} m_j P[k_0,\cdots,k_L, m_1,\cdots,m_L, l_1,\cdots,l_L].$$

Using $m_j P[k, m, l] = a_j P[k_0,\cdots,k_L, m_1,\cdots,m_{j-1}, m_j - 1, m_{j+1},\cdots,m_L, l_1,\cdots,l_L]$ for $m_j \geq 1$, one finds

$$\mathrm{E}[m_j] = a_j \Pr[s \leq N-1] = a_j[1-B] \qquad 1 \leq j \leq L.$$

Similarly, $\mathrm{E}[l_j] = b_j[1-B]$ and $\mathrm{E}[n_j] = c_j[1-B]$ for $1 \leq j \leq L$.

Equation (15) is obtained as follows:

$$\mathrm{E}[k_i] \equiv \sum_{\substack{k_j m_j l_j \geq 0 \\ s \leq N}} k_i P[k_0,\cdots,k_L, m_1,\cdots,m_L, l_1,\cdots,l_L]$$

$$= P(0) \sum_{\substack{k_j m_j l_j \geq 0 \\ s \leq N}} \rho_i \frac{\partial}{\partial \rho_i}\left[\prod_{j=0}^{L} (\rho_j)^{k_j}\right] \times \prod_{j=0}^{L} \left[\frac{(a_j)^{m_j}}{m_j!} \frac{(b_j)^{l_j}}{l_j!}\right]$$

$$= P(0)\, \rho_i \frac{\partial}{\partial \rho_i} \frac{1}{P(0)}.$$

Insertion of Eq. (7) yields Eq. (15) after extended manipulations.

References

1. L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill Book Co., Inc., 1964, out of print; reprinted by Dover Publications, New York, 1972.
2. L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," *AFIPS Conf. Proc., Spring Joint Computer Conference* **35**, 569 (1969).
3. W. W. Chu, "A Study of Asynchronous Time Division Multiplexing for Time-Sharing Computer Systems," *AFIPS Conf. Proc., Fall Joint Computer Conference* **35**, 669 (1969).
4. W. W. Chu, "Demultiplexing Considerations for Statistical Multiplexors," *IEEE Trans. Communications* **COM-20**, 603 (1972).
5. M. Irland, "Queuing Analysis of a Buffer Allocation Scheme for a Packet Switch," *National Telecommunications Conf., Conf. Record* **1**, New Orleans, Louisiana, 1975, p. 24.
6. F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," *AFIPS Conf. Proc., Spring Joint Computer Conference* **36**, 551 (1970).
7. H. Frank, R. E. Kahn, and L. Kleinrock, "Computer Communication Network Design-Ex;erience with Theory and Practice," *AFIPS Conf. Proc., Spring Joint Computer Conference* **40**, 255 (1972).
8. F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *J. ACM* **22**, 248 (1975).
9. F. R. Moore, "Computational Model of a Closed Queuing Network with Exponential Servers, *IBM J. Res. Develop.* **16**, 567 (1972).
10. S. S. Lam, "On An Extension of Moore's Result for Closed Queuing Networks," *Research Report RC 5414*, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1975.
11. J. P. Buzen, "Computational Algorithms for Closed Queuing Networks with Exponential Servers," *Commun. ACM* **16**, 527 (1973).
12. S. S. Lam, "Store-and-Forward Buffer Requirements in a Packet Switching Network," *IEEE Trans. Communications* **COM-24**, 394 (1976).
13. J. D. C. Little, "A Proof of the Queuing Formula: $L = \lambda W$," *Oper. Res.* **9**, 383 (1961).
14. P. Schweitzer, "Maximum Throughput in Finite Capacity Open Queuing Networks," *Research Report RC 5996*, IBM Thomas J. Watson Research Center, Yorktwon Heights, New York, 1976.

Received February 4, 1976; revised April 29, 1976

The authors are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.