ANALYSIS OF SHARED STORAGE IN A COMPUTER COMMUNICATION NETWORK

U. KÖRNER
Lund Institute of Technology, Lund, Sweden

ABSTRACT

Consider a node in a packet switching computer communication network with a finite number of buffers in the output queue system. It is well-known that if load increases too much, packets intended for the busiest outgoing transmission line tend to hog the buffers. The paper deals with a restricted buffer sharing policy called "Sharing with Minimum Allocation" (SMA) which reduces the effects described above. The SMA policy also increases the total throughput and consequently decreases the number of lost packets per time unit.

A queueing model of the packet switch is defined and solved, which gives us the blocking probabilities for packets bound for the different outgoing transmission lines. A cost function containing the different blocking probabilities is set up and used to find the optimal allocation of dedicated buffers within the total output queue system under the SMA policy. The number of lost packets per time unit under the SMA policy is compared to that under the CS policy ("Complete Sharing").

INTRODUCTION

During the second half of the seventies, the growth in computer communication networks has been intensive. The need for small, nationwide and multinational, both private and public, computer communication systems has brought about the development of packet switching networks, such as the Arpanet (1), Cyclades (2), EPSS (4), Teletinet (3), EIN (5) and Transpac (6). The major goal especially for the private networks is to achieve resource sharing among the different network users.

In the packet switching computer network host computers are connected to the so called communication sub-network, consisting of high speed transmission lines and switching computers. The sub-network provides the delivery of packets from for instance host A to host B according to the Store-and-Forward principle. In host A a message is decomposed into packets, each of which has a maximum length and these packets are delivered to node 1. To each packet a header is detached with information of among other things the origin and the destination nodes. When an entire packet is received at node 1, it is placed in an output queue waiting for the, by a routing procedure chosen, transmission line to become free. After reaching for instance node 3, the Store-and-Forward principle is repeated until the packet reaches host B via node 5. When all the packets within the message have been delivered, possibly over different paths in the sub-network, the packets are reassembled and so the user at host A can gain a resource at host B. There are of course a number of different implementation details in different networks. Further details of the packet switching principles can be found in (7).
policy has been used. Kamoun (9) discusses in his Ph.D. Thesis five different restricted buffer sharing policies and Irland has treated one of them (10). Rich and Schwartz (11) discuss a technique of sharing extra storage buffers among a number of output lines in a single packet node. An article by Schweitzer and Lam (12) describes a more detailed model of a packet switch using the theory of networks-of-queues.

A MODEL OF THE PACKET NODE AND THE PROBLEM IN ESSENCE

As discussed in (10) the execution of the routing procedure is much faster than the transmission times and thus the routing procedure is omitted from the queueing model in the figure below or better: the figure below shows a model of the packet node after routing has been done. In general there are n outgoing transmission lines and we call a packet, which will be sent over output line j a j-packet. Furthermore, we assume that the output queue system consists of M buffers each of which can hold one packet irrespective of the length of the packet. The buffer sharing policy, analysed here, is generally called "Sharing with Minimum Allocation", (SMA) and works as follows:

We dedicate mj buffers to j-packets ($\forall j = 1, \ldots, n$). The remaining $M - \sum_{j} m_j$ buffers can be used by any packet finding its own number of dedicated buffers filled. Observe that we dedicate a number of buffers not specific buffers.

In this paper we address ourselves to the problem of finding an optimal allocation of the number of buffers dedicated to j-packets $\forall j = 1, \ldots, n$, for the SMA policy. This optimal allocation corresponds to a minimization of the number of lost packets per second. The SMA policy is compared to the CS policy with respect to the number of lost packets per second, since the CS policy being a common but, in many applications, not the best policy.

STABLE STATE SOLUTION

Let $k = (k_1, k_2, \ldots, k_n)$ denote the state $k_j$ j-packets in the output queue system ($\forall j = 1, \ldots, n$). Feasible states within the state space $\Omega$ are discussed in appendix A. The states $k$ in $\Omega$ form an n-dimensional ergodic Markov chain. To obtain the stable state solution $P(k_1, k_2, \ldots, k_n)$ one can solve the global balance equations for the birth-death process. This can be done with an n-dimensional z-transform. The stable state solution

$P(k_1, k_2, \ldots, k_n) = P(0)^{n-k_j} \lambda_j^n j=1$,

where $P(0) = P(0,0,\ldots,0)$ and

$\rho_j = \frac{\lambda_j}{\sum_{j} \lambda_j}$

can also be obtained from some of the local balance equations. The solution thereby obtained must of course satisfy the global balance equations. The conservation relation $\sum_{j} P(k_j) = 1$ gives us the normalization constant $P(0)$.

$P(0) = \sum_{j} \rho_j k_j j=1^n$

BLOCKING PROBABILITIES

Let $k_j^*$ denote the number of j-packets in the common buffers ($k_j^* = \max(0, k_j - m_j)$) ($k_j - m_j$). An arriving j-packet is lost if $k_j^* \leq M$ and $\frac{n k_j^*}{M} = M - \sum_{j} k_j^*$. The probability of losing an arriving j-packet, $B_j$, can be written

ITC-9

KÖRNER-2
\[ B_j = \sum_{k=0}^{m_j} \sum_{i+j} P(k_1, k_2, \ldots, k_{j-1}, k_j + k, k_{j+1}, \ldots, k_n) \]

where \( m_j = M - \sum_{i} m_i \)

In the case when we have two outgoing transmission lines (n=2), \( B_1 \) and \( B_2 \) are respectively given by the sum over the states belonging to the dotted lines in the figures below.

\[ B_1 = \mathcal{I} P(k); \quad \phi_1 \text{ the dotted line} \]

\[ B_2 = \mathcal{I} P(k); \quad \phi_2 \text{ the dotted line} \]

When \( \rho_1, \rho_2 \) and the number 1 are all distinct, \( B_1 \) and \( B_2 \) are given by the following expressions:

\[ B_1 = \frac{m_1 + m_0}{(1 - \rho_2)} \rho_1 \rho_2 \quad \frac{m_1 m_2 + m_0 m_2}{(\rho_1 - \rho_2)} P(0, 0) \]

\[ B_2 = \frac{m_2 + m_0}{(1 - \rho_1)} \rho_2 \rho_1 \quad \frac{m_2 m_1 + m_0 m_1}{(\rho_2 - \rho_1)} P(0, 0) \]

where

\[ P(0, 0)^{-1} = \frac{m_1 m_2 + m_0 (1 - \rho_1)}{(1 - \rho_1)(1 - \rho_2)} + \]

\[ \frac{m_2 m_1 + m_0 (1 - \rho_2)}{(1 - \rho_1)(1 - \rho_2)} \]

In the case when we have three outgoing transmission lines (n=3), \( B_1 \) is given as the sum over the states belonging to the shadowed surface in figure 6.

THE COST FUNCTION

In order to obtain an optimal allocation of the number of dedicated buffers, we use a cost function containing "the number of lost j-packets per second", \( \forall j = 1, 2, \ldots, n \). A minimization of the cost function, with respect to the number of dedicated buffers, gives us the optimal allocation. The cost function \( f_1(m) \) gives us a minimum value of the number of lost packets per second but however the given allocation can favour for instance \( j_1 \)-packets before \( j_2 \)-packets. The function \( f_2(m) \) leads to that the number of lost packets per second will be better spread among the different types of packets. In (15) we used weight factors in the cost function

\[ f_3(m_1, m_2, \ldots, m_n) = \frac{m}{\lambda} \left( \sum_j m_i B_j \right)^2 \]

where \( \lambda = \lambda_j \)

SAMPLE RESULTS

In this section numerical results are presented for the two cases when our node has two (n=2)
and three \((n=3)\) outgoing transmission lines. The results are based upon the two cost functions 
\[
f_1(m) = \frac{1}{n} \sum_{j=1}^{n} \lambda_j B_j \text{ and } f_2(m) = \left( \frac{1}{n} \sum_{j=1}^{n} \lambda_j B_j \right)^2.
\]
The total number of buffers in the output queue system is ten \((M=10)\) when \(n=2\) and fifteen \((M=15)\) when \(n=3\). The table below (Table 1) shows the optimal allocation of \(m_1\) and \(m_2\) with \(\rho_2\) as variable and \(\rho_1\) as parameter. The cost function \(f_1(m)\) tends to

\[
p_1 = 0.3 \quad p_1 = 0.5 \quad p_1 = 0.7 \quad p_1 = 0.9
\]

\[
\begin{array}{cccccccc}
\rho_2 & m_1 & m_2 & m_1 & m_2 & m_1 & m_2 & m_1 & m_2 \\
0.5 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0.6 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 \\
0.7 & 0 & 0 & 1 & 1 & 2 & 2 & 2 & 2 \\
0.8 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\
0.9 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\
1.0 & 2 & 1 & 2 & 1 & 2 & 2 & 3 & 3 \\
1.1 & 1 & 0 & 1 & 1 & 2 & 2 & 3 & 3 \\
1.2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Table 1
give the smallest traffic more dedicated buffers than the bigger one. It is done to prevent the bigger one from hogging too many buffers. Table number 2 shows the optimal allocation under the

\[
p_1 = 0.3 \quad p_1 = 0.5 \quad p_1 = 0.7 \quad p_1 = 0.9
\]

\[
\begin{array}{cccccccc}
\rho_2 & m_1 & m_2 & m_1 & m_2 & m_1 & m_2 & m_1 & m_2 \\
0.5 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0.6 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\
0.7 & 1 & 2 & 1 & 1 & 2 & 2 & 2 & 2 \\
0.8 & 0 & 2 & 1 & 2 & 1 & 2 & 2 & 2 \\
0.9 & 0 & 2 & 1 & 1 & 3 & 3 & 3 & 3 \\
1.0 & 0 & 2 & 1 & 3 & 2 & 2 & 3 & 3 \\
1.1 & 0 & 3 & 1 & 4 & 2 & 4 & 2 & 2 \\
1.2 & 1 & 6 & 1 & 4 & 2 & 4 & 3 & 3 \\
\end{array}
\]

Table 2
cost function \(f_2(m)\).

Figure 7 shows the optimal allocation in the case where one of the input streams is extremely high. Part a and b in the figure shows the allocation according to cost function \(f_1(m)\) and \(f_2(m)\) respectively.

We define a relative cost reduction \(f'(m)\) where

\[
f'(m) = \frac{f(0,0)-f(m_1^*,m_2^*)}{f(0,0)}
\]

\(m_1^*\) and \(m_2^*\) are the optimal allocation according to the SMA policy. Figure 8 shows \(f'(m)\) as a function of \(\rho_2\) with \(\rho_1\) as parameter.
Figure 9 shows the relative cost reduction as a function of $p$ ($p = p_1 = p_2$) and with the blocking probability as a parameter. The figure shows that the cost reduction decreases with an increasing number of buffers.

Figure 10 shows the relative cost reduction as a function of $p_3$ and with $p_1$ and $p_2$ as parameters.

CONCLUSIONS

It is of course impossible to give prominence in general to one buffer sharing policy before another. Even though one of them gives a lower rate of the number of lost packets than the others, the implementation of the former could be more complicated than that of the others.

However, as seen in the previous sections, a restricted buffer sharing policy like the SMA could not only increase the total throughput and decrease the number of lost packets per second but also prevent packets, bound for one of the outgoing transmission lines, from hogging the buffers.

The allocation of the dedicated number of buffers depends on the traffic parameters. If the mean value of one of the stochastic traffic parameters changes, is in that case the previous allocation better than the Complete Sharing policy? Figure 11 showing the cost function $f_1(m_1, m_2)$ indicates, as the minimum is rather smooth, a positive answer to the question.

Do we achieve results in the same direction, as shown in the previous section under the SMA policy, if we make non-exponential assumptions? One thing is for sure, the stable state solution does not show up in the simple product form as in the exponential case. However, simulations with non-exponential packet lengths, have indicated corresponding cost reductions but not always the same number of dedicated buffers.
APPENDIX A

The state space \( \Omega \) is an \( n \)-dimensional space within the following \( t \) boundaries of different dimensions. The boundaries are all called "planes" independent of their dimensions in the \( n \)-dimensional space.

There are \( n \) planes

\[
 k_j = 0 \quad \forall \; j = 1, 2, \ldots, n
\]

Furthermore there are \( \binom{n}{1} = 1 \) plane of the form:

\[
 k_j = M^{(1)} \quad \forall \; j = 1, \ldots, n
\]

there are \( \binom{n}{n-1} = n \) planes

\[
 \sum_{j=1}^{n} k_j = M^{(2)} \quad \forall \; l_1 = 1, 2, \ldots, n
\]

there are \( \binom{n}{n-2} = \frac{n(n-1)}{2} \) planes

\[
 \sum_{j=1}^{n} k_j = M^{(3)} - m^{(2)} \quad \forall \; l_1 = 1, 2, \ldots, n
\]

In general there are \( \binom{n}{n-k} \) planes

\[
 \sum_{j=1}^{n} k_j = M^{(k)} - \sum_{l=1}^{k-1} m^{(l)} \quad \forall \; l_i = 1, 2, \ldots, n; \quad j \neq l
\]

there are \( \binom{n}{1} = n \) planes

\[
 k_j = M^{(n)} - \sum_{l=1}^{n-1} m^{(l)} \quad \forall \; j = 1, \ldots, n
\]

Take the 3-dimensional case as an example

First \( k_1 = 0, \; k_2 = 0, \; k_3 = 0 \)

then \( k_1 + k_2 + k_3 = M \; \binom{3}{1} = 1 \)

\[
 k_1 + k_2 = M - m_3 \quad \binom{2}{1} = 3
\]

\[
 k_1 + k_3 = M - m_2 \quad \binom{2}{1} = 3
\]

\[
 k_2 + k_3 = M - m_1 \quad \binom{2}{1} = 3
\]

\[
 k_1 = M - m_2 - m_3 \quad \binom{3}{1} = 3
\]

\[
 k_2 = M - m_3 - m_1 \quad \binom{3}{1} = 3
\]

\[
 k_3 = M - m_1 - m_2 \quad \binom{3}{1} = 3
\]

See figure 6

REFERENCES


