International Journal of Soft Computing Applications
ISSN: 1453-2277 Issue 4 (2009), pp.48-56

http://www.eurojournals.com/IJSCA.htm

Estimation of Buffer Requirements in Computer Networks

Saleh A. Khawatreh
Arab Academy for Banking & Financial Sciences, Jordan
E-mail: skhawatreh@yahoo.com

Khalid Khanfar
Arab Academy for Banking &Financial Sciences, Jordan
E-mail: kkhanfar@hotmail.com

Abstract

In computer networks, a message passes through several nodes to reach its destination. A message is delayed by different sources such as link bandwidth and buffer limitations.

In this paper, a mathematical model is implemented to compute the optimal number of buffers that should be available for each node so that none of the messages is lost. This model is based on a priority assignment strategy where processing of a message is preempted by the arrival of messages from higher priority nodes. The load generated by each node is measured by a load factor which is defined as the ratio between the maximum time needed to process the arriving message and the minimum interarrival time between messages.

A case study is made on a star network in which a central node receives messages from n other nodes. The relation between the amount of buffer space needed and the load factor is made through a computer simulation program.

The analysis presented in this paper may help in designing reliable networks especially in early design stages. This can be achieved by making sure that a sufficient amount of buffer space is provided to avoid message loss and unnecessary delays thereby increasing the network throughput.

Keywords: Computer networks, buffer space, delay, priority, load factor, interarrival time, response time.

1. Introduction

A message will pass through several nodes to reach its destination according to the routing algorithm that is applied for the network. A great deal of literature dealing with the subject of network performance already exists [1,2,3,4,5,6,7,8]. These studies have traditionally treated the problem of network performance from a purely mathematical point of view. Queuing theory models have generally been the major tool for the analysis of network performance [9,10]. Although these schemes give a somewhat accurate estimate of a network performance, they do not guarantee that such performance would be maintained even in extreme situations. The correlation between network performance and network load is usually dominated by probabilistic factors [9,10,11,12,13,14,15]. All previous models which are discussed in the previous chapter assume that all messages require the same class of service and therefore message priority was not considered. Also it is found that none of the previous models which are discussed in chapter two addressed the effect of buffering limitations on the overall network performance. We make an attempt to approach the problem of network performance evaluation from a different angle. We attempt to make this problem tractable by investigating the seemingly close relationship between the capacity of the network in terms of message buffering and the traffic load that can be handled by the network without loss of performance.

2. Previous Models

In this section, some of the most relevant literature on the estimation of the buffer size which is directly related to the matter of this research will be presented.

The work of Schweitzer and Lam [9] on the use of input buffer limits for congestion control in store-and-forward networks gives a method of estimating the ratio of buffers needed for the storage of the input messages to the total number of the buffers in any given node such that the throughput would be the maximum.

In [9], it is assumed that each node in the network deals with two types of messages: input messages generated by the node itself and transit messages received from the adjacent nodes. It is assumed that each node has a pool of buffers, some of which are occupied by the input messages generated locally and some are occupied by the transit messages. The network is assumed to be homogenous and of an arbitrary topology.

The limit of the input buffer is defined to be the ratio of the number of buffers occupied by the input messages to the total number of the buffers available for each node. i.e.

Input buffer limit = N_I/N_T (1)

Where

N_I= number of buffers for input messages.

N_T= total number of buffers available in each node.

The analytical model which was suggested later in [9] assumes the Poisson arrivals of messages and exponential message length. These assumptions are taken into consideration because of their effect on processing times. The model also assumes that the internode communication is reliable and therefore there is no consideration of link failure.

In [9], it is shown that the limit of the input buffer is always less than the ratio of input message throughput to total message throughput of the node. i.e.

$N_I/N_T < R$ (2)

This gives a systematic method of achieving maximum throughput. It is, however, shown that congestion free traffic cannot be guaranteed.

R can be computed using the following formula:

$R = T/(T + K(1-B))$ (3)

Where

T = node throughput rate in message per second.

K = rate of arriving transit messages from adjacent nodes in messages per seconds.

B= equilibrium loss probability of transit messages.

R can be expressed in the number of hops h that are traversed by the message from its source to reach its destination by the following formula:

$R = 1/(l+h)$ (4)

A shortcoming of this work is that it does not address message loss. In fact it is possible that messages could be lost on arrival due to the lack of the buffer space and this is because of the way it estimates the buffer requirements of a node. Another observation regarding this model is that it does not deal with message priorities nor, does it consider message delays.

The results presented in [9] are estimative and cannot be considered as upper limits, for the model from which they are derived is a probabilistic one.

The early work of Bricualt and Delgalvis [10] gives a method approximating the number of buffers that are necessary to serve all requests without exceeding a certain waiting time. The waiting time is defined to be the interval of the time between the instant at which a buffer is requested and the instant at which a buffer is reserved. The work of Bricualt and Delgalvis distinguishes between waiting time and holding time, the latter being the interval of time which starts when a buffer is reserved and ends when it is released.

In [10], it is assumed that each node has a pooled number of buffers which are used for assembly and disassembly of received messages. A buffer is reserved to be used by a customer according to its request, then after the service is finished the buffer is released so it can be used by another customer. It is assumed that all buffers are homogeneous.

The solution suggested in [10] is based on a Poisson distribution of buffer requests and exponential distribution of service times.

The Bricualt and Delgalvis paper expresses the buffer utilization u as a function of the average buffer request rate r, the average holding time t and the number of buffers of constant size P. i.e.

$$u=(r*t)/P \tag{5}$$

The Bricualt and Delgalvis paper then goes into a complex procedure to show that p can be calculated through an iterative process by assuming that the average holding time is constant.

Although such solutions are useful as they provide a handle on the improvement of the performance of the network, they however do not address some of the important issues of this research mentioned earlier. By their very nature they are probabilistic and hence lead to estimated results which indicate only mean results.

The work of Beheshti, Ganjali, Goel, and McKeown [16] estimates the buffer size that must be available at each router for different topologies. In a tree topology network with Poisson traffic arrivals, packet drop rate of Є can be achieved if every router in the network has a buffer size B, where

$$B \geq \log_{1/\rho}\left(\frac{n}{\in}\right) \tag{6}$$

Where *n* is the maximum number of buffering nodes on each route and ρ is the load factor which it should be less than one.

In [16], if a network of arbitrary topology with Poisson traffic arrivals, packet drop rate of Є can be achieved if the size of each buffer in the network is B, where

$$B \geq 4\log_{1/\alpha}\left(\frac{n}{(1-\alpha)\in}\right) \tag{7}$$

Where *n* is the maximum number of routers on every route, and ρ is the load factor which it should be less than one, and $\alpha = \rho e^{1-\rho}$.

The work of Singh, Lohia and Srivastava [17], estimating the memory size and the waiting time using the queuing theory. To measure the queue length at each time instance $t_1,t_2,\ldots\ldots\ldots,t_n$, let the queue size at $t_1,t_2,\ldots\ldots\ldots,t_n$ be $Q_1,Q_2\ldots\ldots..,Q_n$ respectively. Then, the average queue length

$$Q(L) = \frac{1}{n}\sum_{i=1}^{n} Q(i) \tag{8}$$

$$W(t)=1/\mu[Q(i)]. \tag{9}$$

The size of memory to provide minimal data flow is given as:

$$M=Q(L)+Variance \tag{10}$$

In case of M/M/1 queue model,

$$Variance=Q(L) \tag{11}$$

$$M=2Q(L) \tag{12}$$

The work of Sujoy Ghose, Rajeev Kumar, Nilanjan Banerjee, and Raja Datta [18] suggests using a genetic algorithm to find the minimum buffer size. In order for a genetic algorithm to minimize the buffer size that should be available at each node, a shortest path routing algorithm is used. It gives good results but it does not use priority assignment strategy and it does not apply the genetic algorithm on an adaptive routing algorithm.

In [19], the minimal buffering requirements for different adaptation policies are studied. A minimum buffer requirement equation for TCP-Friendly protocol is developed as a function of the average of the achievable transmission rate, RTT and packet size, as shown in the following equation:

$$\Delta = \frac{\alpha}{18MSS} \times R^2 \times RTT^2 \qquad (13)$$

where Δ is the minimum buffer size for decreasing the delay for Additive Increase and Multiplicative Decrease (AIMD), MSS is the packet size, R is the average achievable transmission rate, RTT is the round trip time and α is the increasing parameter of the AIMD protocol AIMD(α, β), with the TCP-Friendliness increase/decrease relationship

$$\alpha = 3(1-\beta)/(1+\beta \qquad (14)$$

Research in [15] also shows that the adaptation policies that maximize throughput are not suitable for interactive applications with high bit rates or long round trip times (RTTs) due to the long delay caused by large buffer size.

3. The Model

Let $B_1, B_2, \ldots, B_n$ be the sending nodes in the network. These nodes are sending messages to a central node R as shown in Figure 1. Let $Q_1, Q_2, \ldots, Q_n$ be queues of the buffers that are maintained by R and these queues are used for storing messages that are arriving from nodes $B_1, B_2, \ldots, B_n$ respectively as shown in Figure 2.

Let t_n be the minimum interarrival time between the successive messages that are arriving to node R from node B_n and c_n be the maximum processing times for messages that are arriving from node B_n.

The routing algorithm applied in this model is a simple nonadaptive routing algorithm where each node is sending data to the central node R through a direct one hop link. If a failure occurs in the direct route between any sending node and the central node then there will be no transmission between this node and the central node.

Figure 1: System Configuration (Source:[4])

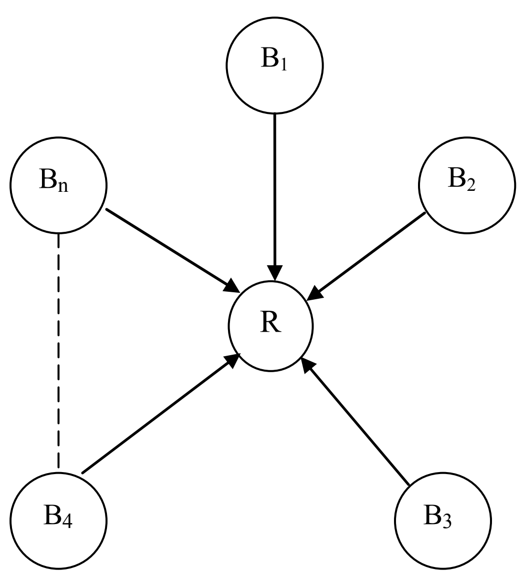
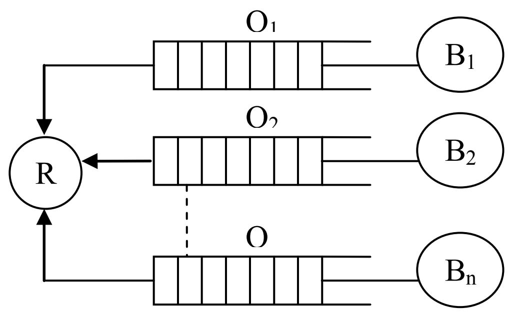


Figure 2: Buffer Queues (Source:[4])



We define the load generated by a node Bi in the network as follows:

$lf_i = c_i/t_i$ (15)

lf_i is called the load factor of node B_i.

Clearly for Q_i to be finite, the following condition must be satisfied:

$Lf_i < 1$ (16)

Let RT(i,m) denotes the worst case of the response time for the message m that is received from the node B_i. Within the system that is under consideration, RT(i,m) can be defined as follows:

$RT(i,m)=DW_i(m)+DP_i(m)+c_i$ (17)

Where DW_i is the delay that is caused by the processing of a message of the same priority and DPi is the delay that is encountered by those messages during the processing time (that is due to the preemption by higher priority messages). It is clear that if RT(i,m) < 1 for all m then the length of Q_i needs not to exceed one.

The model is based on the following assumptions:

a) Each node has a unique priority level that is associated with it.
b) Each node has a queue where it stores its arriving messages.
c) A node can generate new messages or process arriving messages in its queue.
d) Priority levels are from 1 to n with the lower numbers indicating higher priorities.
e) Each message has a unique priority level that is associated with it. This priority level is the same priority level of its sending node.
f) A message of a priority i arriving at a node executing a message of a priority level j with j>i will preempt that execution.

4. Feasibility Theorem

It was pointed earlier that in order for Q_i to be finite, the load factor lf_i must be less than 1. To see this, consider a certain time interval d. The maximum number of messages received by R from node B_i during this time interval is (d/t_i). Since each of these messages requires ci processing time from R, a maximum time of

$(d/t_i)*c_i$

is needed to process these messages. In order for Q_i to be finite, R must complete the processing of these messages within the interval d. Hence we have the following inequality:

$(d/t_i)*c_i < d$ or $c_i/t_i < 1$

Generalizing this to all Q_1, Q2,…..,Q_n, we have

$lf_i < 1$ for all i=1 to n.

Since R is assumed to be monolithic processor multiplexed among all arriving messages and during any given interval d, R may serve requests from many queues. The finiteness of all queues then must depend on all lf_i as stated in the following theorems.

Theorem 1:

A necessary and sufficient condition that all queues Qi for i= 1 to n, be finite is that

$$\sum_{i=1}^{n} lf_i < 1. \tag{18}$$

Proof:

In order for Q_i to be finite, R must complete the processing before any new messages arrive during any given time interval. The maximum number of messages arriving during d from any node B_i is (d/t_i). R must serve all requests arriving during d from all nodes before any new ones arrive. Hence

$$\sum_{i=1}^{n} (d/t_i)*c_i < d \tag{19}$$

$$\sum_{i=1}^{n} (lf_i) < 1. \tag{20}$$

Theorem 2:

If $\sum (c_j/t_j) < 1$ for j=1 to (i-1) then the delay encountered of a message of priority i (DP_i) during processing is finite.

Proof:

The Processing of an arriving message from node B_i is preempted by the arrival of messages from the higher priority nodes which are B_1 to $B_{(i-1)}$. During any time interval d, the maximum number of arrivals of such messages is

$$\sum_{j=1}^{i-1} d/t_j \tag{21}$$

The maximum processing time that the arriving messages from nodes B_1 to $B_{(i-1)}$ during interval d is

$\sum (d/t_j)*c_j$ for j=1 to (i-1). The hypothesis that we have

$$\sum_{j=1}^{i-1} (d/t_j)*c_j < d \tag{22}$$

and

$$\sum_{j=1}^{i-1} c_j/t_j < 1 \tag{23}$$

Imply that the processing of these messages can be completed before any new arrivals. Hence DP_i is finite.

5. Processing Delays

Let t_1 be the time at which the processing of a message m_i that is arriving from the node B_i is preempted and t_2 be the time at which the processing of the message m_i is resumed. To examine what could happen during the open-ended interval $A=[t_1,t_2)$, we apply the fact that once the processing of the messages each of priority i is preempted, the point of resumption is independent of the order of the arrival of these messages with higher priorities. But the point of resumption depends on the number of arriving messages during the preemption interval. For this reason we define the term Ar(A,j) as follows:

$$Ar(A,j)=[t_2/t_j]-[t_1/t_j] \tag{24}$$

Where A is the interval between t_1 and t_2. Ar computes the maximum number of messages that arrive from the node B_j during the interval A.

Arriving messages from node B_j during the interval A require the following amount of processing time:

$Ar(A,j)*c_j$ (25)

Hence, the total processing time required for all arriving messages from nodes with priorities higher than the priority of B_i during the interval A is:

$$DP(A,i)=\sum_{j=1}^{i-1} Ar(A,j)*c_j \quad (26)$$

This clearly represents the maximum processing delay that can be experienced by the messages that are arriving from B_i.

The maximum processing time (PT_i) needed for the messages that are arriving from B_i during the interval A is calculated according to following formula:

$$PT_i=t_2=Dp_i+c_i \quad (27)$$

6. Processing Delay Computation

Lemma 1

If $DP_i(m) = 0$ for all m, then $DW_i(m) = 0$.

Suppose that a message m_1 arrives from node B_i such that $DW_i(m_1) \neq 0$. This implies That $WRT(i,m_2)>ti$. As stated in the feasibility theorem, c_i is less than t_i this implies that $WRT(i,m_2) > 0$, and then $DP_i(m_2) = 0$. As a contrary to this result that if $DP(i,m_2) = 0$ then $DW(i,m_2) = 0$.

To compute the delay time experienced by the preemption of the arrival of messages from higher priority nodes, the following procedure is presented:

a) A message m_i arriving from node B_i needs c_i processing time. Let A initially be the interval $[0,c_i)$. If $Ar(A,j)$ for j = 1 to (i-1) is null then $DP_i = 0$ and by lemma 1 $DW_i = 0$.

Let $PT_i=c_i$ where PT_i is the processing time needed for a message arriving from node B_i.

b) Now, assume that $Ar(A,j) \neq 0$ for some j = 1 to (i-1). Messages m_k for which $Ar(A,k) \neq 0$ where k<i cause processing delay to m_i. To compute this delay accurately we consider the interval $A=[t_1,t_2)$ with $t_1=c_i$ and $t_2=c_i+DP_i$. DP_i is computed by the following formula:

$$DP_i=\sum_{k=1}^{i-1} Ar(A,k)*c_k \quad (28)$$

DP_i is the interval by which the processing of m_i is extended because of the delay caused by higher priority messages. Hence

$$PT_i=t_2=DP_i+ c_i \quad (29)$$

Therefore a convenient way of defining the interval A is to consider it as an open-ended interval representing the time by which processing is extended due to the delays caused by higher priority messages.

c) If during the new interval $A= [t_1,t_2)$, $Ar(A,j) = 0$ for all j=1 to i-1 then as stated in (a) $PT_i=t_2$ else a new interval must be considered as in step (b) with

$$First(A_{new})= last(A_{old}) \quad (30)$$

$$Last(A_{new})=last(A_{old})+ DPi(A_{old}) \quad (31)$$

The feasibility theorem guarantees that this process terminates, otherwise PT_i would be infinite.

7. Estimation of Buffer Requirements

To avoid any message loss we provide a certain number of buffers, k_i, where the messages from node B_i are stored and served in first-come-first-served order.

To accommodate the number of messages that are arriving from node B_i during the time WRT_i we need at least k_i buffers. Where k_i is calculated as follows:

$$k_i= PT_i/t_i \quad (32)$$

8. Computer Analysis

The analytical solution derived in the previous section can be easily expressed through an algorithmic procedure. The purpose of program implementation is to enable us to examine and analyze the effect of interarrival and processing time variance on response times and buffer requirements. The output data of this program are the worst case response time and the maximum number of buffers needed for each node, the output results are plotted to show various relationships among the three main parameters: Load factor, response time and number of buffers.

The output that is produced by the program is used for analysis purposes. In particular we examine the variations in response times and buffer requirements as functions of the system load factor and the interarrival and processing times.

We have divided the analysis into three stages as follows:

First stage: load factor variance.

In this stage the system load factor is varied and its effect on the response time and buffer requirements is examined. The program runs with a different input values. The results are plotted to illustrate these effects.

Second stage: Interarrival time variance.

In this stage the iterarrival time is varied to see its effect on the response time and on the b required buffers. The interarrival time is varied for each node separately to study the effect of each priority level on the response time and on the number of required buffers.

Third stage: Processing time variance.

In this stage the processing time is varied to see its effect on the response time and on the number of required buffers. The processing time is varied for each node separately. The output results are analyzed in this section to see the effect of variations of the input data on the number of estimated buffers.

9. Conclusion and Future Work

It is seen from the analysis that the response time and the number of buffers are maximum when the system load factor is nearly unity. The response time and the number of needed buffers decrease with the decrease of the load factor until they reach a constant value. It is seen from the analysis that the variation of the interarrival time has a direct effect on the response time and on the number of required buffers which is expected because the number of arrivals is dependent upon the interarrival time. If the computation time is changed even by a small value, there will be a significant effect on the response time and on the number of required buffers which means that if the existing processor is replaced by a faster one then response time might become much smaller and consequently the number of required buffers will also decrease.

The model derived for the star network can be extended to be applied on a fully connected network. Also the number of buffers that should be available at each node can be computed with different routing algorithms either adaptive or nonadaptive from which the best routing algorithm is estimated.

References

[1] A. S. Tanenbaum, “Computer Networks,” Prentice-Hall, 2003.

[2] F. S. H. Bakry, ”Extended Computer Networks for Engineers and Scientist,” Journal of Eng. Sci., College of Engineering, King Saud University, Vol. 8, No. 1, PP. 91-98, 1982.

[3] A. S. Tanenbaum, ”Network Protocols,” Computing Surveys, Vol. 13, No. 4, PP. 453-489, December,1981.

[4] A. Mahjoub, “On the Static Evaluation of Distributed System Performance,” Computer Journal Vol. 27, No.3, PP. 201-208, 1984.

[5] D. W. Davies and D. L. A Barber, “Communication Networks for Computers,” Wiley, 1980.

[6] F. F. Kuo, “Protocols and Techniques for Data Communication Networks,” Prentice-Hall, 1981.

[7] S. Al-Qahtani, T. Helmy and K. Salah, ”Performance Evaluation of a QoS-Aware Mechanism for Polling-Based High-Speed Network Interfaces,” Journal of Computer Science Vol.1, No. 4, PP. 461-470, 2005.

[8] S. S. Lam and M. Reiser, “Congestion Control of Store-and-Forward Networks by Input Buffer Limits Analysis,” IEEE Transactions on Communication, Vol. Com-27, No. 1, PP.122-130, January 1979.

[9] P. J. Schweitzer and S. S. Lam, “Buffer overflow in a Store-and-Forward Networks Nodes,” IBM J. Res. Develop. November, PP. 58-67, 1976.

[10] J. P. Bricault and I. Delgalvis, ”On Teleprocessing System Design Part III: An Analysis of a Request-Queued Buffer Pool,” IBM Systems Journal, Vol. 5, No. 3, PP.148-157, 1966.

[11] H. M. Deitel, “An Introduction to Operating Systems,” Addison Wesley, 2000.

[12] A. Silberschatz, P. B Galvin and G. Gagne, “Operating System Concepts,” John Wiley, 2006.

[13] S. Floyd, ”High Speed TCP for Large Congestion Windows,” IETF RFC 3649, December 2003.

[14] D. Leith and R. Shorten, ”H-TCP Protocol for High-Speed Long Distance Networks,” Second International Workshop on Protocols for Fast Long-Distance Networks, Pfldnet, 2004.

[15] T. Kelly, “Scalable TCP: Improving Performance on High speed Wide Area Networks,” Computer Communication Review, Vol. 33, No. 2, PP. 83-91, April 2003.

[16] N. Beheshti, Y. Ganjali, A. Goel and N. McKeown, “Obtaining High Throughput in Networks with Tiny Buffer,” In Proceedings of 16th International Workshop on Quality of Service (IWQoS), Enschede, Netherlands, PP. 64-69, June 2008.

[17] L. K. Singh, R. M. Lohia and R. Srivastava, "Estimation of Buffer Size of Internet Gateway Server via G/M/1 Queuing Model," PWASET, Vol. 21, PP. 474-482, 2007.

[18] S. Ghose, R.Kumar, N. Banerjee, and R. Datta, " Multihop Virtual Topology Design in WDM Optical Networks for Self-Similar Traffic," Photonic Network Communications, Vol.10, No.2, 199–214, 2005.

[19] K. Li, C. Krasic, J. Walpole, M. H. Shor and C. Pu, “The Minimal Buffering Requirements of Congestion Controlled Interactive Multimedia Applications,” in IDMS ’01: Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems, PP. 181–192, 2001.