Estimation of Buffer Size in Computer Networks

Saleh A. Khawatreh
Computer Engineering Dept., Faculty of Engineering
Al-Ahliyya Amman University

Bashar E. A. Badr
Electrical Engineering Dept., Faculty of Engineering
Al-Ahliyya Amman University

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 research, 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 are made through computer simulation program.

The analysis presented in this paper may help in designing reliable networks by making sure, early in the design stages, 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 the 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]. All previous models [9,10,11,12,13,14] 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 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.

2. THE MODEL
Let V1, V2...Vn be nodes in the network sending messages to a central node R as shown in Fig. 1. Let Q1, Q2…, Qn be n queues of buffers maintained by R and used for storing messages arriving from V1, V2….., Vn respectively as shown in Fig. 2.

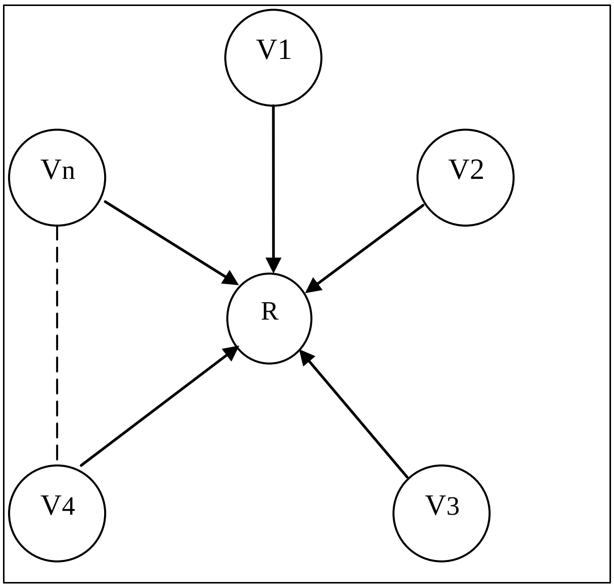

Fig. 1 System Configuration

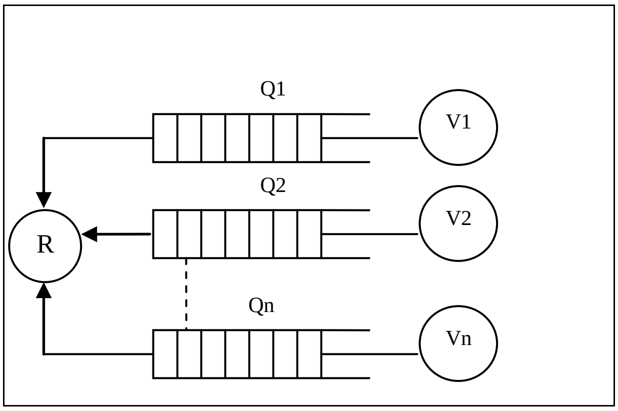

Fig. 2 Buffer Queues

Let t1, t2……, tn be the minimum interarrival times between successive messages arriving from nodes V1, V2…., Vn respectively and c1, c2….., cn be the maximum processing times arriving from nodes V1, V2….., Vn respectively.

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

$$lfi = ci/ti \quad \text{.. (1)}$$

lfi is called the load factor of Vi.

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

lfi < 1 ……………………. (2)

Let RT(i,m) denote the worst case response time for message m received from node Vi. In the system under consideration RT(i,m) can be defined as follows:

RT(i,m)=DWi(m)+DPi(m)+ci………. (3)

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

The model is based on the following assumptions:

a) Each message has a unique priority level associated with it which is the priority of its sending node.

b) Each node has a unique priority level associated with it.

c) c)Each node has a queue where it stores arriving messages.

d) d)A node can generate new messages or process arriving messages in its queue.

e) e) Priority assignments are from 1 to n with lower numbers indicating higher priorities.

f) f) A message of priority i arriving at a node executing at priority level j with j>i will

g) preempt that execution.

3. FEASIBILITY THEOREM:

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

(d/ti) *ci

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

(d/ti) *ci < d or ci/ti < 1

Generalizing this to all Q1, Q2..., Qn, we have

lfi < 1 for all i=1 to n.

Since R is assumed to be monolithic processor multiplexed among all arriving messages, which means that during any given interval d, R may service requests from many queues, the finiteness of all Qi then must depend on all lfi as stated in the following theorem.

Theorem 1:

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

$$\sum_{i=1}^{n} lfi < 1 \quad \ldots\ldots\ldots\ldots\ldots.. \quad (4)$$

Proof:

In order that Qi be finite, R must complete processing before any new messages arrive during any given time interval. The maximum number of messages arriving during d from any node Vi is (d/ti). R must service all requests arriving during d from all nodes before any new ones arrive. Hence

$$\sum_{i=1}^{n} (d/ti) *ci < d \quad \text{or} \quad \sum_{i=1}^{n}(lfi) < 1$$

Theorem2:

For Qi to be finite it is sufficient that the following condition be satisfied:

$$\sum_{j=1}^{i} cj/tj < 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

Proof:

Processing of arriving messages from node Vi is affected by the arrival of messages from higher priority nodes. For any given interval, d, processing for all arriving messages from node Vi and from the nodes at priorities higher than Vi must be completed before the arrival of any new messages. Hence

$$\sum_{j=1}^{i} (d/tj) *cj < d$$

$$\sum_{j=1}^{i} (cj/tj) < 1 \quad \text{or} \quad \sum_{j=1}^{i} lfj < 1$$

Theorem 3:

Let WRTi=max RT(i,m) over all m.

If Qi is finite, then WRTi is finite.

Proof:

If Qi is finite, then no message waits indefinitely in Qi before processing.

Assume that WRTi is infinite then DPi must also be infinite which means that processing of a message arriving from Vi does not complete within any interval d. Hence

(d/ti) *ci > d or ci/ti > 1

Which implies by theorem 2 that Qi is infinite.

Theorem 4:

If ∑ (cj/tj) < 1 for j=1 to (i-1) then the delay encountered of a message of priority

i (DPi) during processing is finite.

Proof:

Processing of an arriving message from node Vi is preempted by the arrival of messages from higher priority nodes which are V1 to V(i-1). During any interval d, the maximum number of arrivals of such messages is

$$\sum_{j=1}^{i-1} d/tj$$

The maximum processing time they require is (d/tj) *cj. By hypothesis we have

$$\sum_{j=1}^{i-1} c_j/t_j < 1 \quad \text{or} \quad \sum_{j=1}^{i-1} (d/t_j) * c_j < d$$

Which implies that processing of these messages can be completed before any new arrivals. Hence DPi is finite.

4. PROCESSING DELAYS

Let t1 be the time at which processing of a message mi arriving from node Vi is preempted and t2 be the time at which processing of mi is resumed. To examine what could happen during the open-ended interval A= [t1, t2), we apply the fact that once processing of messages of priority i is preempted, the point of resumption is independent of the order of arrival of messages with higher priorities. It is dependent on the number of arriving messages during the preemption interval. For this reason we define the term Ar(A,j) as follows:

$$Ar(A,j)=[t2/tj]-[t1/tj] \quad \ldots\ldots\ldots\ldots\ldots.. \quad (6)$$

Where A is the interval between t1 and t2. Ar computes the maximum number of message arrivals from node Vj during the interval A.

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

$$Ar(A,j)*cj \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \ldots. \quad (7)$$

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

$$DP(A,i)=\sum_{j=1}^{i-1} Ar(A,j)*cj \quad \ldots\ldots\ldots\ldots \quad \ldots.(8)$$

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

5. PROCESSING DELAY COMPUTATION:

Lemma 1:

If DPi(m) =0 for all m, then DWi(m) =0.

Suppose that a message m1 arriving from node Vi such that DWi(m1) ≠0. This implies That WRTi, m2)>ti. As stated in the feasibility theorem, ci is less than ti which implies that WRT (i, m2) >0, then DPi(m2) =0. As a contrary to this result that if

DP (i, m2) =0 then DW (i, m2) =0.

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

A) A message mi arriving from node Vi needs ci processing time. Let A be initially the interval [0, ci). If Ar(A,j) for j=1,i-1 is null then DPi=0 and by lemma 1 DW=0.

Let PTi=ci where PTi is the processing time needed for a message arriving from node Vi.

b) Assume now that Ar(A,j)≠0 for some j=1,i-1.Messages mk for which Ar(A,k)≠0 where k<i cause processing delay to mi. To compute this delay accurately we consider the interval A= [t1, t2) with t1=ci and t2=ci+DPi. DPi being:

$$DPi= \sum_{k=1}^{i-1} Ar(A,k)*ck$$

This is the interval by which processing of mi is extended because of delays caused by higher priority messages. Hence

$$Pti=t2=Dpi+ci$$

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

c) If during the new interval A= [t1, t2), Ar(A,j) = 0 for all j=1 to i-1 then as stated in (a)

PTi=t2 else a new interval must be considered as in step (b)

with

$$First(Anew) = last(Aold)$$

$$Last(Anew) = last(Aold) + DPi(Aold)$$

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

6. ESTIMATION OF BUFFER REQUIREMENTS:

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

The number of messages that are arriving from node Vi during the time WRTi is (PTi/ti). To accommodate these number of messages we need at least ki buffers. Where ki is:

$$ki= PTi/ti \quad \ldots\ldots\ldots.. \quad . (9)$$

7. 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.

8. 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 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.

9. REFERENCES

[1] Tanenbaum, A. S., "Computer Networks" Prentice-Hall, 2012.

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

[3] Tanenbaum, A. S.," Network Protocols" Computing Surveys, Vol. 13, No. 4, December 1981.

[4] A. Mahjoub, "On the Static Analysis of Distributed System Performance" Computer journal 27(3), 1994, PP. 201-208.

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

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

[7] Mcquillan, J. M., and Cref, V. G. "a Practical View of Computer Communications Protocols", IEEE, Computer Society, 1978, PP. 76-85.

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

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

[10] Cheong, V. E., and HirsChhiem, R. A." Local Area Networks" Wiley, 1983.

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

[12] C. Kellett, R. Shorten, D. Leith. "Sizing internet buffers, Active queue management, and the Lur'e problem". Proceedings of IEEE Conf on Decision and Control, 2006.

[13] Y. Ganjali, N. McKeown. "Update on Buffer Sizing in Internet Routers". Computer Communications Review, October 2006.

[14] Silberschatz, A., Galvin P. B., and Gagne, G. " Operating System Concepts " John Wiley, 2016.