INT. J. SYSTEMS SCI., 1986, VOL. 17, NO. 10, 1381–1387

Performance evaluation by window control mechanisms

J. DAS†

In this paper an attempt has been made to evaluate the performance of a computer communication network by controlling the window parameters. This includes a simple deterministic model and provides a measure of the window size that gives the best throughput in a store-and-forward communication network.

1. Introduction

Flow control is the set of mechanisms whereby a flow of data can be maintained within limits compatible with the amount of available resources. The objectives of flow control have been defined as the provision of adequate end-to-end service (throughput and delay) and the minimization of the relevant network resources. In high-speed local nets, the control commands suffer little delay in transit, the flow control mechanism that would require somewhat more processing power is based upon granting 'credits' for transmission. The receiver grants credits for a certain amount of data to the sender so that both know exactly how much data will be exchanged. The number of credits provided by the receiver is frequently called the 'window size' when the credits are expressed relative to packet sequence number.

In order to keep the network traffic within desirable limits, flow control, among other things, should be equipped with mechanisms to throttle some resources (Pouzin 1976). Flow control procedures are required to protect network resources during traffic overload periods. Such procedure should not impair network performance when the network has excess capacity. Isarithmic flow control procedures place a limit on the total number of packets that are allowed in the network and are managed on a network-wide or global basis. Another way to limit the number of packets in transit is on an end user-to-end user connection basis. This can be done with window mechanisms that use counters together with scheduling functions to control the number of packets in the network. Window mechanisms are also used to control flows between adjacent nodes at the link level and at the packet level.

A variety of control techniques have been suggested to either prevent congestion or alleviate it, as it begins to develop. End-to-end control or window mechanisms are a common form of control mechanism. Most end-to-end flow control that belongs to global control mechanisms use a variant of the credit throttling technique and are usually described in terms of window size (Cerf and Kahn 1974, Belsnes 1975) where the number of unacknowledged messages (or packets) are limited to the window size. A variant of this technique was originally used in the French Cyclades network (Zimmerman 1975) and the ARPANET VDH (very distant host) connection (Bolt, Beranek and Newman Report 1973). The throughput a user can obtain in a packet-switched network depends on the rate at which he can launch packets in the network. The rate at which packets can be launched in the network is an inverse function of the

---

Received 4 March 1985.

† Electronics and Communication Sciences Unit, Indian Statistical Institute, 203 B T Road, Calcutta-35, India.

response time (which affects all users equally), but directly proportional to the window size. In fact, in a packet-switched network, window size and response time replace the number of user lines and the 'loss' signals of circuit switching.

End-to-end flow control is accomplished through inter-process communication protocols and any attempt to quantitatively study the former should start with the development of an analytical model for the latter. Several authors have discussed techniques for flow control at different levels (Davies 1971, Crowther 1975, Kleinrock 1976). However, due to the complex multivariate environment of flow control, quantitative and analytic results have been lacking. With the exception of simulation studies of flow control (Weber 1964), most authors have focused on the influence of individual tools of flow control on overall performance. In general, however, the complexity of the problem has meant that the interaction of these tools has not been studied analytically.

The purpose of this paper is to develop analytic models for end-to-end communication protocols and to study the window mechanism for flow control in store-forward computer-based communications networks. We deal with a deterministic model in which there are no stochastic fluctuations in the load of the system.

2. Flow control

The set of rules that control the flow of messages inside the network are usually referred to as local control, as opposed to global control, which is the set of control procedures which supervise the admission of messages to the network. Most commonly used global-flow controls can be divided into two categories: end-to-end flow control, used in the ARPANET (McQuillan *et al.* 1972), and isarithmic flow control, as considered in the NPL network (Price 1973). End-to-end flow control is usually described in terms of a window mechanism where the number of 'unacknowledged' messages between a source and destination is limited to the window size. End-to-end flow control is accomplished by interprocess communication protocols and any attempt to quantitatively study the former should start with the development of an analytic model of the latter. Isarithmic flow control is a partial flow control system which suffers from reduced throughput compared to a complete flow control mechanism. In effect, with isarithmic flow control, it is possible to have demands for available resources but insufficient permits, or it is possible to have demands and permits in sufficient qualities to create localized congestion. Network topologies and accompanying routine procedures can be designed to spread a given traffic load out, so that, on average, it will flow smoothly and within specified time limits between the source and destination. The curves in Fig. 1 show network throughput and mean transit delay characteristics versus offered load for networks with and without flow control mechanisms. Transit delay is defined as the time for admittance of a packet to the network from a terminal or application at the origin network node until delivery of the packet to a terminal link queue or application at the destination network node. Without effective flow control, throughput and mean transit delay increase to a saturation region and then flatten, remaining insensitive to load increases.

An important performance measure for a computer communication network is the average source-to-destination message delay T, defined below:

$$T=\sum_{i,j}\frac{\gamma_{ij}}{\gamma}Z_{ij}$$

where γ_{ij} is the average number of messages entering the network per second with

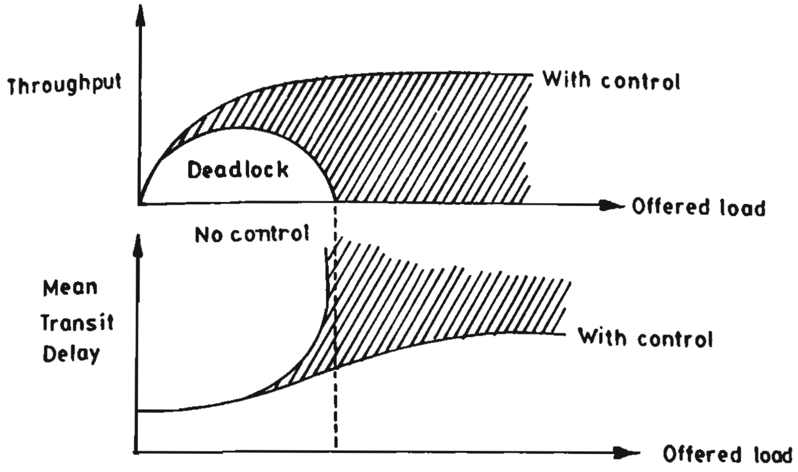

Figure 1. Throughput and transit time versus offered load.

origin i and destination j, γ is the total arrival rate of messages from external sources, and Z_{ij} is the average message delay for messages with origin i and destination j.

For a message switched system, a straightforward application of Little's result (1961) to the queueing model leads to the following expression for T

$$T = T_m = \sum_i \frac{\lambda_i}{\gamma} T_i$$

where T_m is the delay in message switching and T_i is the average delay at node i.

Calculation of T_i is in general non-trivial and very complicated, however, after certain assumptions, we are in a position to reduce the networks of queue models to the model first studied by Jackson (1957). By virtue of his results, each node behaves stochastically independently of the other nodes and similar to a M/M/m system. For the special case of uniform link utilization, all of the nodal delays are identical and we have

$$T_m = T_i \sum_i \frac{\lambda_i}{\gamma}$$

but

$$\Sigma \frac{\lambda_i}{\gamma} = n_h$$

where n_h is the number of hops (or path lengths), so

$$T_m = T_i n_h$$

The above analysis holds for noiseless channels. For noisy channels, the delay calculation is complicated and can be obtained by using an iterative routine.

3. The model

Consider a communication session between a source and destination (say nodes X and Y) in a communication network. The environment under consideration is shown in Fig. 2. (Here we assume that message consists of single packets, so that we do not differentiate between message switching and packet switching.) In this figure the traffic

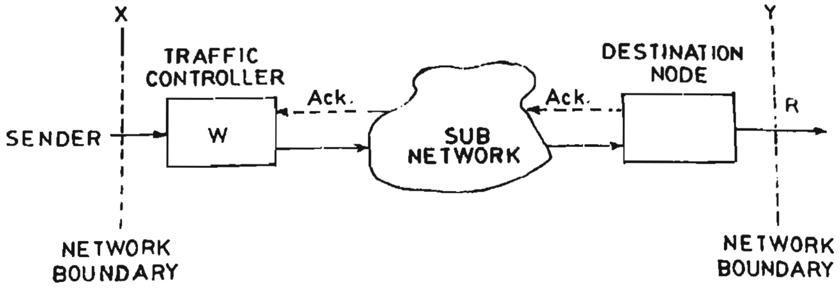

Figure 2. Structure of a network.

controller (TC) in the source node is responsible for regulating the traffic from node X to node Y. It allows a maximum of W (the window size) unacknowledged messages from the sender to be present in the network. The messages are sent towards the destination node and a copy of them is retained in the buffer space provided by TC. When the TC receives an acknowledgement (ACK) for a message, it discards the copy of the message it is storing and accepts a new message from the sender. In the destination node, messages which are received and can be accepted are transmitted to the receiver and ACK from them are sent back to the TC. We assume that the traffic on the path is symmetric (that is, there is the same flow of data from receiver to sender) and that the ACKs are piggybacked on the messages, hence there is no traffic due to ACKs. We also assume that the sender is fast compared to the network, so that whenever the TC can accept a new message, the sender has one ready for transmission. With these assumptions, throughput is mainly determined by the network and we are interested in the maximum throughput delay behaviour of the network within the boundaries shown in Fig. 2. Maximum throughput is the maximum rate at which the network can accept new messages, hence it is the upper boundary of the acceptable input rate of the network. The end-to-end delay is the delay a message experiences from the time it is submitted to the TC until it is delivered to the receiver. This delay consists of the network delay, the delay incurred in the TC and the (sub) network, plus the destination node delay. Another delay measure which is important in our study is the acknowledgement delay (T_a) which is the period from the time a message is delivered to the (sub) network until its ACK is received by the TC. Note that the ACK delay is equal to the round-trip delay only if message transmission is successful. If a message is lost in the network (owing to transmission error or rejection at the destination node) no ACK for the message is received (hence T_a is infinite).

In a store-forward computer network, since packet sequences may be received out of order, the destination node may reject out-of-order packets or messages, or it may store the out-of-order packets or messages, but deliver them in the original order of the receiver. The first solution results in an excessive buffer storage requirement at the destination node. The sequencing problem is more important for the case of multipack messages. In our case, since we have assumed single-packet messages, the sequencing problem is not so critical. In addition, there will not be any reassembly deadlock in the system we study; only rarely may store-forward deadlock occur and we can thus ignore the deadlock problem.

For purely deterministic models, we can assume that the random variables involved are deterministic and the channels are completely noiseless; for every message delivered to the (sub) network TC, an ACK is received (with probability one).

We define the following:

T_a ACK delay (s)
$1/\mu$ message length (bits)
C_1 network channel capacity (bps)
X_1 message transmission time on a channel $(1/\mu C_1)$ (s)
γ_1 maximum discharge rate of messages $(=\mu C_1)$ (msg/s)
λ^* maximum throughput (or acceptance rate of the network (msg/s)
λ input rate of messages to the network (msg/s)
W window size (msg)

Consider Fig. 3. If at time t_0 a message is delivered to the network, the ACK will be received at time t_1 $(=t_0+T_a)$. Now every T_a seconds, one message is accepted by the network and thus for a window size equal to 1 ($W=1$), maximum throughput becomes $1/T_a$ and in general for a window size W the maximum throughput is W/T_a. Figure 4 shows a sketch of maximum throughput as a function of window size. Because the throughput cannot exceed γ_1, the maximum discharge rate, the curve levels off when the window reaches a critical size (W^*). Summing up, we have the following relationship between the maximum throughput and the window size:

$$\lambda^* = \begin{cases} W/T_a, & W \leqslant W^* \\ \gamma_1 & W > W^* \end{cases} \tag{1}$$

where $W^* = T_a\gamma_1$. Note that λ^* is the maximum throughput that the network can carry, the input rate λ is bounded above by λ^*.

In order to obtain a quantitative value for the ACK delay, we make some assumptions about the structure of the network. Figure 5 shows a communication

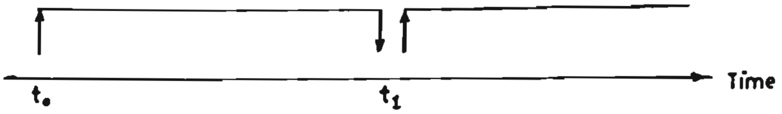

Figure 3. Transmission interval in a network when the window size is 1.

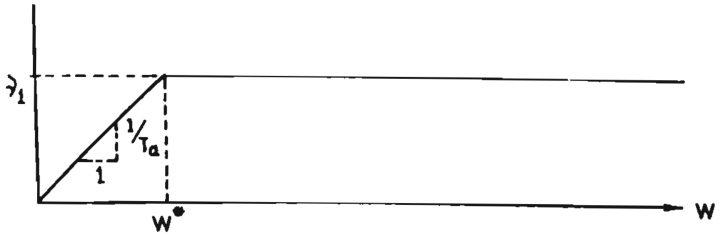

Figure 4. Network throughput as a function of window size (deterministic model).

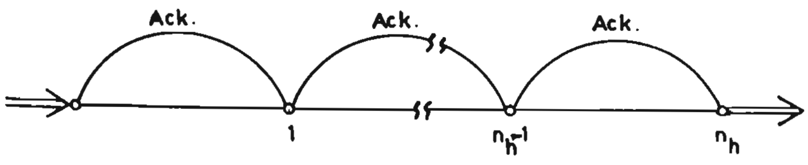

Figure 5. A communication path.

path for n_h hops between source X and destination Y. Because the transmission delay on each channel is X_1, the ACK delay becomes

$$T_a = 2n_h X_1 = 2n_h/\gamma_1$$

and (1) reduces to

$$\lambda^* = \begin{cases} W\gamma_1/2n_h, & W \leqslant W^* \\ \gamma_1, & W > W^* \end{cases} \tag{2}$$

where $W = 2n_h$.

Although the model is crude, it still reveals some interesting characteristics of the system. In this fluid-approximation model, in which there is no stochastic load fluctuations, it shows that the traffic and the throughput of the system can be controlled by the window mechanism. Furthermore, it is not beneficial to increase the window size beyond a certain value. Considering the fact that a large window size means a large number of buffers in the traffic controller and that the buffer storage is a cost item in the overall network cost, the disadvantages of a large window size become clear. With regard to the window size W^*, this deterministic model indicates that a window size of $W > 2n_h$ gives the best throughput, and that the best choice for W^* is $W^* = 2n_h$.

The window size sent by the receiver to the sender represents a promise of buffer space availability at the receiver's end. Two policies have been identified on which this promise can be based. The conservative policy uses the window size as a firm guarantee of buffers actually available to the receiving transmission control protocol (TCP) at the time it generated the message. The optimistic policy bases window-size calculations on some predictive heuristic to give a reasonable estimate of what the receiving protocol expects to possess in the near future. As one would expect, simulation studies have shown that the conservative policy is the best one to follow, especially when buffer-space management is the responsibility of the user process, the policy encouraged by the TCP implementors. In this case, an optimistic policy must try to gauge the demands that the user process is going to make on the TCP and the resources that it will make available.

ACKNOWLEDGMENT

I am grateful to Professor D. Dutta Majumder, Head of the ECSU Division for his valuable suggestions and keen interest in my work, and would also like to acknowledge the help and co-operation of Sri S. Chakravorty and Mrs S. De Bhowmick for preparing the final manuscript.

REFERENCES

BELSNES, D., 1975, *Communications Networks*, edited by D. Barber (London: Online Conference Ltd).

BOLT, BERANEK and NEWMAN Inc., 1973, Specification for the interconnection of a host and an IMP. BBN Report 1882.

CERF, V. G., and KAHN, R. E., 1974, *I.E.E.E. Trans. Comm.*, **22,** 637.

CROWTHER, W. R., 1975, *AFIPS Conf. Proc.*, California, **44,** 161.

DAVIES, D. W., 1971, *Proc. 2nd ACM–I.E.E.E. Symp. on Optimization of Data Communication*, Palo Alto, California.

DAVIES, D. W., and BARBER, D. L. A., 1973, *Communication Networks for Computers* (London: Wiley).

JACKSON, J. R., 1957, *Ops. Res.*, **5**, 518.

KLEINROCK, L., 1976, *Queuing Systems*, Vol. II, *Computer Applications* (New York: Wiley Interscience).

LITTLE, J., 1961, *Ops Res.*, **9**, 383.

MCQUILLAN, J. M., CROWTHER, W. R., COSSELL, B. P., and WALDES, D. C., 1972, *Fall Joint Computer Conf.*, AFIPS Conf. Proc. California, pp. 741–754.

POUZIN, L., 1976, Flow control in data networks: methods and tools. Internal conf. on computer communication proc., Toronto, p. 467.

PRICE, W. L., 1973, *Proc. I.E.E.E. Symp. on Data Network Analysis and Design*, Florida.

RUDIN, H., 1976, Flow control: Introduction. Proc. Int. conf. on Computer Communication, Toronto.

WEBER, J. H., 1964, *Bell System Tech. J.*, **XI,** 978.

ZIMMERMAN, H., 1975, The Cyclades end-to-end protocol. *ACM-I.E.E.E. 4th Data Commun. Symp.*, Canada.