Theoretical Performance Analysis of Sliding Window Link Level Flow Control for a Local Area Network

E.Arthurs
G.L.Chesson
B.W.Stuck

Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

A transmitter sends packets over a local area network to a receiver. The receiver has a finite amount of storage capacity for buffering messages. A *sliding window protocol* is employed to insure all packets are delivered without error in order of arrival. Mean throughput rate and packet delay statistics are studied as a function of model parameters. Simulation data are presented.

1. Introduction

At the present time there is great interest in communication over logical abstractions of physical circuits (so called *virtual* circuits) within local area networks. *Protocols* control the transfer of data from transmitters to receivers over the virtual circuit to insure no loss of data, and take appropriate actions in event of a wide variety of failures. References on this subject are found elsewhere (e.g., Tanenbaum, 1981, pp.187-196).

Examples of such mechanisms are stop-start protocols where the transmitter stops until the receiver acknowledges receipt of the message (e.g., Binary Synchronous Communications (IBM)), Digital Equipment's product line for Digital Network Architecture (Wecker,1980; Tanenbaum, 1981, pp.172-174), IBM's product line for Systems Network Architecture framework(Green, 1979; Atkins, 1980), the Defense Advance Research Project Agency Transport Control Protocol (Tanenbaum, 1981, pp.373-377) or CCITT's X.25 (Tanenbaum, 1981, pp.167-172).

In our opinion, at the present time there is a great deal of guidance required to engineer such systems to achieve predictable traffic handling characteristics (cf the current literature: Bux, Kuemmerle, Truong, 1980; Easton, 1980; Fayolle, Gelenbe, Pujolle, 1978; Kleinrock, 1978A, 1978B; Reiser, 1979; Sunshine, 1976, 1977; Traynham, Steen, 1977; Yu, Majithia, 1979; Luderer, Che, Marshall, 1982; Luderer, Che, Haggerty, Kirslis, Marshall, 1981).

Here we report on a novel technique for upper and lower bounding mean throughput rate based only on zero contention mean packet processing and transmission times. Furthermore, the bounds obtained are the *best* possible bounds, i.e., they are *achievable* given only mean times for packet handling under no load. The control of the flow of data for the protocol adopted here is found in the open literature (Knuth, 1981; Tanenbaum, 1981, pp.148-164) and is called a *sliding window link level flow control protocol.* It is representative of a great many protocols currently in use, each of which differs in detail in terms of error handling, but not in terms of pacing the flow of data between the transmitter and receiver (cf Schwartz, 1982, for an analysis of a protocol with the same control of flow of data but a different acknowledgement strategy). Most importantly, the analysis is *tractable,* i.e., it can be checked quickly in actual field operations.

We test the sensitivity of our *assumptions* as well as numerical *parameters* by comparing mean value analysis results against results based on a Jackson network analysis and on simulation results. In all cases, the results suggest that buffering *two* packets at the receiver buys virtually all the benefit of buffering more than two packets, and is superior to buffering only *one* packet at the receiver. This is well known to many digital systems engineers: the contribution here is rigorous analysis to substantitate this folk lore.

2. Model Description

The system that motivated this work is described elsewhere (Fraser, 1979; Chesson, 1979, 1980; Luderer, Che, Haggerty, Kirslis, Marshall, 1981; Luderer, Che, Marshall, 1982). An illustrative hardware block diagram is shown in Figure 1. A number of devices are interconnected via *Data Terminal Equipment (DTE)* to a *local area network* switching system. The devices communicate with one another in three phases: first, a full duplex virtual circuit is set up between two devices, second, communication exchange takes place, and third, the full duplex virtual circuit is taken down. Communication takes place as follows:

[1] One device, called the transmitter, takes a variable length message, decomposes it into a series of fixed length packets, appends a variety of control bits to each packet, and transmits it over the network. The transmitter holds each packet in local storage until an acknowledgement is received that the packet was correctly received.

[2] The receiver processes each packet, removes control bits, and groups packets into messages, plus sends an acknowledgement of proper receipt back to the transmitter.

In a local area network, the fraction of packets that are lost or garbled by the network is typically quite small; the impact on traffic handling characteristics of lost packets is ignored from this point on.

In a well engineered local area network, the packet delay due to the network is typically quite small compared to the packet handling at the sender and receiver; from this point the local area network delay for each packet is assumed to be an additive constant delay. There are a variety of other phenomena, such as hardware and software failures, higher level protocols, and so forth that are ignored from this point on in the interest of brevity.

2.1 Flow Control Policy

Initially, the transmitter starts a packet sequence counter, denoted C, at zero. Messages are transmitted in order of arrival; packets within

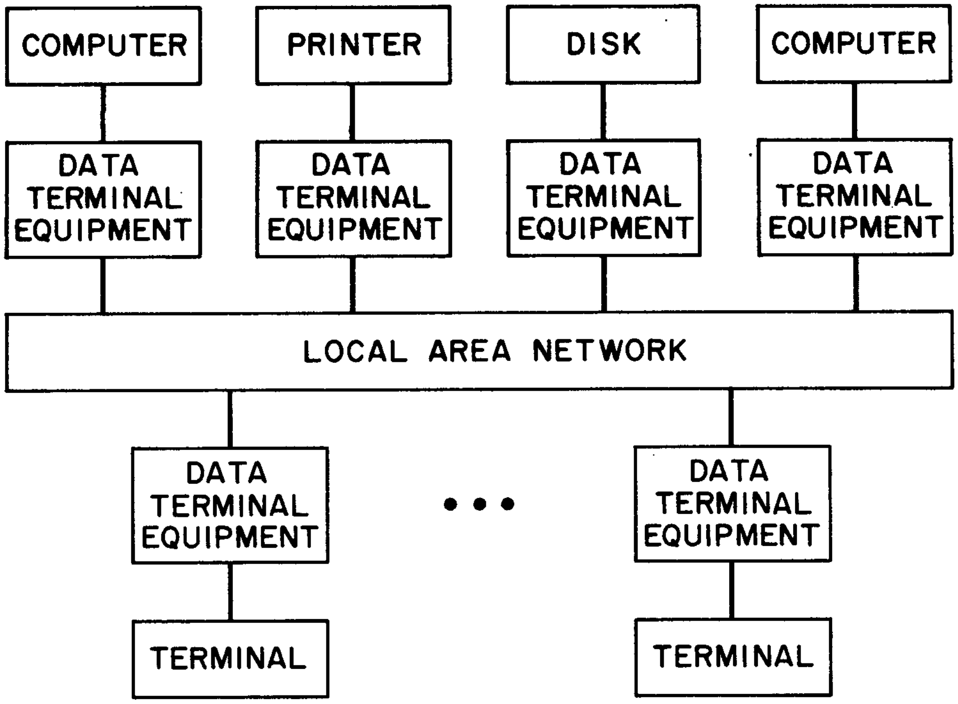

Figure 1.Hardware Block Diagram

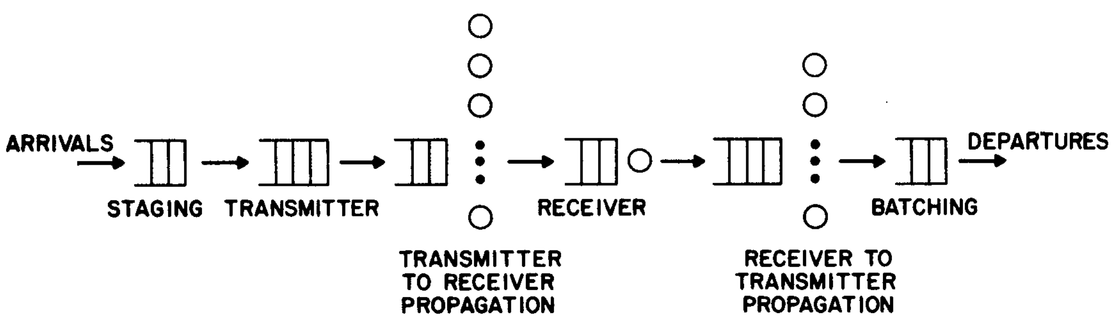

Figure 2.Queueing Network Block Diagram

messages are transmitted in order over the virtual circuit. Each packet has a sequence number that is used to pace the flow of data from the source to the sink. Each time the transmitter sends a packet, C is incremented by one; each time the transmitter receives an acknowledgement, C is decremented by one and flushes this packet from its transmitter buffer. The maximum number of packets that can be buffered by the receiver is called the *window* denoted W. The largest the transmitter packet sequence counter can be is W: the transmitter knows the receiver can buffer at most this many packets. Each packet holds one receiver buffer; when the packet sequence counter strikes W the transmitter ceases to send messages, until a minimum number of acknowledgements are received. A start/stop protocol would have a window of size one (W=1): the transmitter would send the first packet of a message and wait for a positive acknowledgement before sending the next packet, and so forth. A double buffering protocol would have a window size of two (W=2).

How frequently should the receiver acknowledge packets? Ideally it should be done after each packet; however, if this requires an unacceptable amount of processor time at the transmitter or at the receiver or both, then acknowledgements could be batched. The normal operating regime is where control packets are much less frequent than data packets; here we focus on the data packet processing time alone.

2.2 Queueing Network Model

The queueing model of this system (Figure 2) follows the above description quite closely.

2.2.1 Queues and Servers The system consists of a staging queue (with no server), a transmitter queue (with one server), a transmitter to receiver queue (with W servers), a receiver queue (with one server), and a receiver to transmitter queue (with W servers)

2.2.2 Packet Flow Through Queueing Network A packet migrates from one queue to another: packets arrive at an external queue where they are staged, before migrating to the transmitter queue, then through the transmitter to receiver queue, then to the receiver queue, then finally to the receiver to transmitter queue, before leaving the system; a packet is in the system if it is in the transmitter or receiver queue (waiting or in execution), or in the propagation queue from the transmitter to the receiver and vice versa.

2.2.3 Service Required for Each Step of Packet Communications Each packet requires some processing time by the transmitter, denoted T_{trans}, including both packet processing time and transmission time. Each packet propagates from the transmitter to receiver, in a mean time denoted $T_{trans-rec}$. Each packet requires receiver processing time, denoted T_{rec}. Each receiver acknowledgement packet propagates from the receiver to the transmitter in a time interval denoted $T_{rec-trans}$. The receiver and transmitter processing times are assumed to include the time to handle acknowledgement processing.

2.2.4 Flow Control Policy Arriving packets are stored in the staging queue. If there are less than W packets in the system, the packet immediately enters the transmitter queue; otherwise, the packet waits in the staging queue.

3. Mean Throughput Rate Bounds (cf Reiser, 1979; Fayolle et al, 1978)

The first step in the analysis is to construct a state space for this system, denoted by Ω. If we imagine observing* the system in operation at a given instant of time, say t, we would note at most four kinds of activities in process, one for each of the steps. Let the four tuple $\underline{J}=(J_{trans},J_{trans-rec},J_{rec},J_{rec-trans})$ denote the state of the system, whose components are nonnegative integers, with each component denoting the number of packets either queued or in execution at that instant. The statement that the system is in state $\underline{J}$ at time t then means that at the time of observation, there were *concurrently* in progress J_{trans} packets at the transmitter (both queued and in processing), and so forth. Not all values for $\underline{J}$ are possible. Since we wish to determine the behavior of the system under load, we will assume we have an external staging queue to the system that contains $J_{staging}$ packets, while there can be at most W packets in the system:

$$J_{trans} = \min[J_{staging}, W - J_{trans-rec} - J_{rec} - J_{rec-trans}] \quad (1)$$

Our goal is to determine the maximum mean throughput rate of packet transmission through this model. We assume that there are always sufficiently many packets waiting in the staging queue such that $J_{staging} \geq W$, i.e., there are W packets circulating in the system:

$$W = J_{trans} + J_{trans-rec} + J_{rec} + J_{rec-trans} \quad (2)$$

If this maximum mean throughput rate is unacceptably low, then operating at a lower mean throughput rate (in order to achieve acceptable packet delay) will also be unacceptable.

* As an aside, it is interesting to note that the system state is intrinsically *distributed* in the transmitter, the receiver, and the channel; due to the finite propagation time of signals between the transmitter and the receiver, it is *impossible* for *either* to observe the total system state at a given instant of time, say t

3.1 Analysis

The mean throughput rate is denoted by λ. The mean number of packets in execution in the transmitter and in the receiver equals the mean throughput rate multiplied by the total mean execution time (Little, 1961; Conway, Maxwell, Miller, 1967, pp.18-19). We denote by $E()$ the time average of the argument, and write:

$$\lambda T_{trans} = E[1 - \delta_{0,J_{trans}}] \leq 1 \quad (3a)$$

$$\lambda T_{rec} = E[1 - \delta_{0,J_{rec}}] \leq 1 \quad (3b)$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad (3c)$$

Since only one packet is in execution at the transmitter or the receiver at any one time, the above inequalities immediately give us

$$\lambda \leq \min\left[\frac{1}{T_{trans}}, \frac{1}{T_{rec}}\right] \quad (4)$$

The mean number of packets in propagation from the transmitter to receiver is

$$\lambda T_{trans-rec} = E[J_{trans-rec}] \quad (5a)$$

while the mean number of packets in propagation from the receiver to transmitter is

$$\lambda T_{rec-trans} = E[J_{rec-trans}] \quad (5b)$$

Our goal is to find upper and lower bounds on mean throughput rate, subject to meeting state space constraints.

If we add equations (3a), (3b), (5a), and (5b), we see

$$E[1 - \delta_{0,J_{trans}} + J_{trans-rec} + 1 - \delta_{0,J_{rec}} + J_{rec-trans}] = \lambda[T_{trans} + T_{trans-rec} + T_{rec} + T_{rec-trans}] \quad (6)$$

However, since

$$1 - \delta_{0,J_{trans}} \leq J_{trans} \qquad 1 - \delta_{0,J_{rec}} \leq J_{rec} \quad (7)$$

we find that

$$\lambda[T_{trans} + T_{trans-rec} + T_{rec} + T_{rec-trans}] \leq E[J_{trans} + J_{trans-rec} + J_{rec} + J_{rec-trans}] = W \quad (8)$$

Combining (4) and (8), we find

$$\lambda \leq \min\left[\frac{1}{T_{trans}}, \frac{1}{T_{rec}}, \frac{W}{T_{trans} + T_{trans-rec} + T_{rec} + T_{rec-trans}}\right] \quad (9)$$

To get a lower bound on the mean throughput rate, we realize that

$$\lambda[T_{trans} + \frac{T_{trans-rec}}{W} + T_{rec} + \frac{T_{rec-trans}}{W}] = E[1 - \delta_{0,J_{trans}} + \frac{J_{trans-rec}}{W} + 1 - \delta_{0,J_{rec}} + \frac{J_{rec-trans}}{W}] \geq 1 \quad (10)$$

The lower bound on the right hand side arises from realizing that if $J_{trans}=J_{rec}=0$ then the right hand side is one, while if either J_{trans} or J_{rec} is greater than zero then the right hand side is lower bounded by one. This allows us to show

$$\lambda \geq \frac{W}{W[T_{trans} + T_{rec}] + T_{trans-rec} + T_{rec-trans}} \quad (11)$$

3.1.1 Interpretation of Upper Bound on Mean Throughput Rate The physical interpretation of the upper bound on mean throughput rate is as follows

- If the transmitter is the bottleneck, then

$$\lambda_{upper} = \frac{1}{T_{trans}} \quad (12a)$$

- If the receiver is the bottleneck, then

$$\lambda_{upper} = \frac{1}{T_{rec}} \quad (12b)$$

- If the number of buffers is the bottleneck, then

$$\lambda_{upper} = \frac{W}{T_{trans} + T_{trans-rec} + T_{rec} + T_{rec-trans}} \quad (12c)$$

3.1.2 Interpretation of Lower Bound on Mean Throughput Rate The physical interpretation of the lower bound is that at most one packet at a time is being handled by the system.

3.2 Negligible Local Area Network Delay

As an example, let us assume that $W=1$ and $T_{trans-rec}=T_{rec-trans}=0$, i.e., the local area network delay is negligible compared to the mean time required at the transmitter and receiver to handle a packet, which is typical in *local area networks* (cf Luderer et al, 1981, 1982). If we do so, we see

$$\frac{1}{T_{trans}+T_{rec}} \leqslant \lambda_{max} \leqslant \frac{1}{T_{trans}+T_{rec}} \qquad W=1 \qquad (13)$$

In words, the maximum rate of transmitting packets is the reciprocal of the sum of the mean time spent by the transmitter plus the mean time spent by the receiver.

Increasing the number of buffers from one to two, $W=1$ to $W=2$ always increases the maximum mean throughput rate, and now we see

$$\frac{1}{T_{trans}+T_{rec}} \leqslant \lambda_{max} \leqslant \min\left[\frac{1}{T_{trans}},\frac{1}{T_{rec}}\right] \qquad W>1 \qquad (14)$$

Furthermore, this increase is maximized for $T_{trans}=T_{rec}$, and then the upper bound *doubles* in going from one buffer to more than one buffer. Why is this so? By having more than one buffer, both the transmitter and receiver can simultaneously be filling and emptying a buffer, allowing greater concurrency or parallelism compared with the single buffer case. We also note that allowing more than two buffers, e.g., *infinite* buffers, will not increase the upper bound on the maximum mean throughput rate any further; intuitively, we only have two serially reusable resources, the transmitter and receiver, and double buffering keeps them both busy simultaneously.

The lower bound on mean throughput rate, is identical to the upper bound for $W=1$. Why is this so? There may be significant fluctuation about the mean values shown above, and in the limit of one big swing about the mean value all of the messages will pile up at one stage in the network and nothing will be transmitted until buffers become available.

3.3 Nonnegligible Local Area Network Delay

If the local area network delay is not negligible compared to the packet processing at the transmitter and receiver, the upper bound on mean throughput rate will increase as a linear function of the amount of buffering available at the receiver, until either the transmitter or the receiver becomes a limiting bottleneck. Figures 3A, 3B, and 3C plot these upper and lower bounds, as well as the results of an Jackson queueing network analysis (e.g., Kelly, 1976, 1979), for the special case where

$$T_{trans} = T_{rec} \qquad T_{trans-rec} = T_{rec-trans} \qquad (16)$$

for three different cases, where the propagation delay is much smaller, equal, and much larger than the mean processing time at either the transmitter or receiver.

Introducing a Jackson network model requires additional information about the *fluctuations* about the mean packet service times at each step. This allows us to study the sensitivity of our findings to our *assumptions* as well as to the actual numerical values of model parameters.

The fraction of time the queueing network model predicts the system to be in state $\underline{J}$ is denoted by $\pi(\underline{J})$ where

$$\pi(\underline{J}) = \frac{1}{G} T_{trans}^{J_{trans}} \frac{T_{trans-rec}^{J_{trans-rec}}}{J_{trans-rec}!} T_{rec}^{J_{rec}} \frac{T_{rec-trans}^{J_{rec-trans}}}{J_{rec-trans}!} \qquad (17)$$

The system partition function denoted G is chosen to normalize the probability distribution:

$$\sum_{\underline{J}} \pi(\underline{J}) = 1 \qquad (18)$$

where the summation is carried out over all feasible system states.

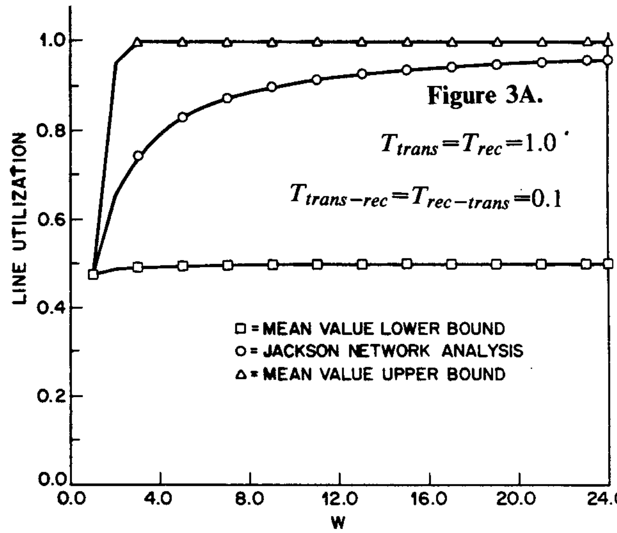

Figure 3A.

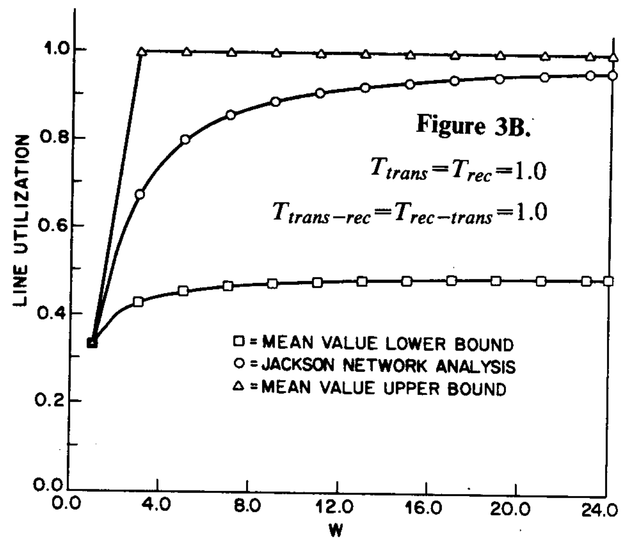

Figure 3B.

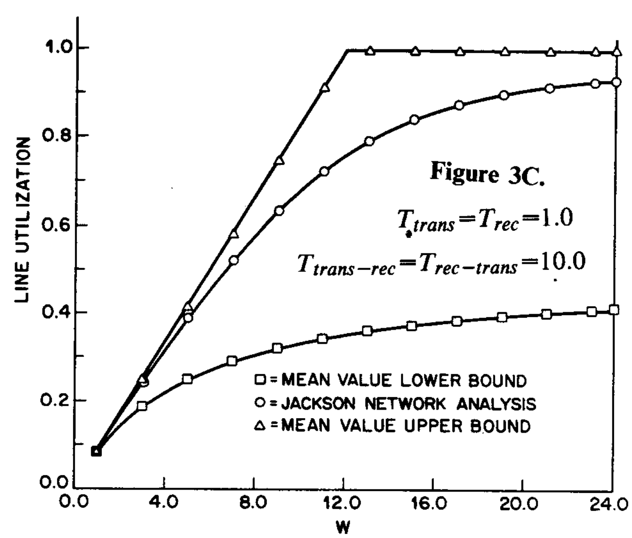

Figure 3C.

Two regimes are evident, one where the buffers are the bottleneck and the mean throughput rate grows linearly in the number of buffers, and one where the receiver is the bottleneck and the mean throughput rate is independent of the number of buffers; we saw exactly these same regimes in bounding mean throughput rate. As is evident from the figures, the Jackson network analysis tracks quite closely the mean value upper bound on throughput rate.. Since the Jackson network analysis assumes the packet processing times are *exponentially* distributed, i.e., with significantly greater fluctuation in processing times than might occur in practice, and since the agreement (at this level of analysis) between the mean value upper bound and the Jackson network is quite close, this suggests using the mean value upper bound as a guide to setting flow control parameters, because it is quite straightforward to analyze.

4. Simulation Results

We complement the above theoretical analysis with simulation results for packet delay. Eleven thousand packets were generated and sent through a GPSS simulation (IBM, 1971; Schriber, 1974; Appendix). The first one thousand were used as a warmup, and data was examined to insure that stationarity had set in before starting to gather and record data that is presented below. The statistics gathered for the next ten thousand packets were segmented into ten groups of one thousand each; the observed variability was felt to be within statistical fluctuations, and the uniformity of the statistics across the ten samples suggested that long term time averaged statistics were in fact meaningful. The packet interarrival times were exponential random variables; the transmitter, receiver, and channel propagation times were constant. The results below are a sample of some of the data gathered. We assume the channel propagation delay is zero here, and the transmitter and receiver require one time unit each per packet.

Table 1--First Two Moments of Packet Delay Statistics

Mean Interarrival Time	*Mean Arrival Rate*	*W=1,B=1* *Mean Delay*	*W=1,B=1* *Standard Deviation*	*W=2,B=1* *Mean Delay*	*W=2,B=1* *Standard Deviation*
6.67	0.15	3.366	4.085	2.088	0.257
4.00	0.25	3.378	3.453	2.169	0.375
2.80	0.36	4.248	3.194	2.292	0.546
2.20	0.45	4.983	3.316	2.442	0.741
1.70	0.59	--	--	2.745	1.080
1.30	0.77	--	--	3.748	2.052
1.10	0.91	--	--	7.141	4.562

Table 2--Percentiles of Packet Delay Statistics

Mean Interarrival Time	*Mean Arrival Rate*	*W=1,B=1* *90th Percentile*	*W=1,B=1* *95th Percentile*	*W=2,B=1* *90th Percentile*	*W=2,B=1* *95th Percentile*
6.67	0.15	6.395	10.610	2.393	2.722
4.00	0.25	8.145	10.635	2.710	2.930
2.80	0.36	8.670	10.675	2.915	3.400
2.20	0.45	9.615	11.295	3.365	3.955
1.70	0.59	--	--	4.025	4.820
1.30	0.77	--	--	6.585	7.965
1.10	0.91	--	--	11.300	13.557

As is clear from the above tables, to within statistical fluctuations the delay characteristics for $W=2,B=1$ and $W=1,B=1$ appear to differ greatly. Furthermore, the delay statistics for $W=2,B=1$ and $W=7,B=1$ (which are not presented here in the interest of brevity) appear to be virtually identical, to within fluctuations. The main difference in the delay characteristics for the double buffering and infinite buffering case was the startup transient in each: as congestion rises, both systems will always be started up, and the impact of this transient is apparently negligible for the numbers investigated above.

5. Summary

Granted the modeling assumptions described earlier for a single virtual circuit connecting one device to another over a local area network, it appears that double buffering (W=2) window flow control offers performance superior to single buffering (W=1) and performance comparable to larger (W>2) buffer sizes. This was suggested from mean value bounds on mean throughput rate, and substantiated from a Jackson network analysis and a simulation analysis.

Appendix--GPSS Simulation

We have been asked to include a more detailed description of the simulation referred to in the body of the text. Figure A is a source code listing of the GPSS simulation program. The transmitter and receiver each require a mean amount of processing per packet. The sequence of transmitter and receiver packet processing times are constant. The channel propagation time has a given mean value; the sequence of channel propagation times are constant. The packet interarrival times are independent exponentially distributed random variables with a given mean interarrival time The remaining parameters are the window size and the batch size.

The table below summarizes the parameters, the source code statement that defines each parameter, and the value of each parameter in Figure A.

GPSS Simulation Parameter Summary

Parameter	*Statement*	*Value*
$E(T_{trans})$=E(TRANS)	10	100
$E(T_{rec})$=E(REC)	20	100
$E(T_{trans-rec})=E(T_{rec-trans})$= E(TRANS-REC)=E(REC-TRANS)	16	0
W=WINDOW	3	7
B=ACKNOWLEDGEMENT BATCH SIZE	5	1
MEAN INTERARRIVAL TIME	1	280

References

[1] J.Atkins, *Path Control: The Transport Network of SNA,* IEEE Transactions on Communications, **28** (4) 527-538 (1980).

[2] W.Bux, K.Kuemmerle, H.Linh Truong, *Balanced HDLC Procedures: A Performance Analysis,* IEEE Trans. Communications, **28** (11), 1889-1898 (1980).

[3] G.L.Chesson, *DATAKIT Software Architecture,* 20.2.1-20.2.5, Proceedings International Communications Conference, Boston, Massachusetts, 1979.

[4] G.L.Chesson, A.G.Fraser, *DATAKIT Network Architecture,* COMPCON'80, 25-28 February 1980, San Francisco, California, IEEE Catalog Number 1491-0/80, pp.59-61

[5] R.W.Conway, W.L.Maxwell, L.W.Miller, **Theory of Scheduling,** Addison-Wesley, Reading, Massachusetts, 1967; Little's formula, pp.18-19.

[6] M.C.Easton, *Batch Throughput Efficiency of ADCCP/HDLC/SDLC Selective Reject Protocols,* IEEE Trans.Communications, **28** (2), 187-195 (1980).

[7] G.Fayolle, E.Gelenbe, G.Pujolle, *An Analytic Evaluation of the Performance of the "Send and Wait" Protocol,* IEEE Trans.Communications, **26** (3), 313-319(1978).

[8] A.G.Fraser, *DATAKIT--A Modular Network for Synchronous and Asynchronous Traffic,* 20.1.1-20.1.3, Proceedings International Communications Conference, Boston, Mass., 1979.

[9] P.E.Green, *An Introduction to Network Architecture,* IBM Systems Journal, **18** (2), 202-222 (1979).

[10] IBM, *General Information--Binary Synchronous Communications,* **IBM Systems Reference Library,** GA27-3004-2.

[11] IBM, *MSP/7 Binary Synchronous Communications: Program Logic Manual,* GY34-0012-1.

[12] IBM, *IBM General Purpose Simulation System Users Manual,* **IBM Systems Reference Library,** White Plains, NY, 1971, GH20-0326-04.

[13] F.P.Kelly, *Networks of Queues,* Advances in Applied Probability, **8,** 416-432 (1976).

[14] F.P.Kelly, **Reversibility and Stochastic Networks,** Wiley, Chichester, 1979.

[15] L.Kleinrock, *Principles and Lessons in Packet Communications,* Proceedings IEEE, **66** (11), 1320-1329(1978;A).

[16] L.Kleinrock, *On Flow Control,* Proc.Int.Conf.Communications, Toronto, Canada, 27.2-1--27.2-5, June 1978;B.

[17] D.E.Knuth, *Verification of Link-Level Protocols,* BIT, **21,** 31-36 (1981).

[18] J.D.C.Little, *Proof of the Queueing Formula* $L=\lambda W$, Operations Research, **9** (3), 383-386(1961).

[19] G.W.R.Luderer, H.Che, W.T.Marshall, *A Virtual Circuit Switch as the Basis for Distributed Systems,* Journal of Telecommunication Networks, **1,** 147-160 (1982).

[20] G.W.R.Luderer H.Che, J.P.Haggerty, P.A.Kirslis, W.T.Marshall, *A Distributed UNIX System Based on a Virtual Circuit Switch,* ACM Operating Systems Review, **15** (5), 160-168 (8th Symposium on Operating Systems Principles, Asilomar, 14-16 December 1981), ACM 534810.

[21] M.Reiser, *A Queueing Network Analysis of Computer Networks with Window Flow Control,* IEEE Trans.Communications, **27** (8), 1199-1209(1979).

[22] T.J.Schriber, **Simulation Using GPSS,** Wiley, NY, 1974.

[23] M.Schwartz, *Performance Analysis of the SNA Virtual Route Pacing Control,* IEEE Transactions on Communications, **30** (1), 172-184 (1982)

[24] C.A.Sunshine, *Factors in Interprocess Communication Protocol Efficiency for Computer Networks,* AFIP NCC, pp.571-576(1976).

[25] C.A.Sunshine, *Efficiency of Interprocess Communication Protocols for Computer Networks,* IEEE Transactions on Communications, **25,** 287-293(1977).

[26] A.S.Tanenbaum, **Computer Networks,** especially Section 4.2, pp.148-164, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[27] K.C.Traynham, R.F.Steen, *SDLC and BSC on Satellite Links: A Performance Comparison,* Computer Communication Review, 1977

[28] S.Wecker, *DNA: The Digital Network Architecture,* IEEE Transactions on Communications, **28** (4), 510-525 (1980).

[29] L.W.Yu, J.C.Majithia, *An Analysis of One Direction of Window Mechanism,* IEEE Trans.Communications, **27** (5), 778-788(1979).

```
*LOC    OPERATION  A,B,C,D,E,F,G
*
*  FULL DUPLEX LINK LEVEL HIGH/LOW FLOW CONTROL
*  WITH CHANNEL PROPAGATION DELAYS
*
        SIMULATE
 1      FUNCTION   RN3,C24
0         0          .1         .104       .2         .222
.3        .355       .4         .509       .5         .69
.6        .915       .7         1.2        .75        1.38
.8        1.6        .84        1.83       .88        2.12
.9        2.3        .92        2.52       .94        2.81
.95       2.99       .96        3.2        .97        3.5
.98       3.9        .99        4.6        .995       5.3
.998      6.2        .999       7          .9997      8
*
* INTERARRIVAL TIMES IID EXPONENTIAL ARRIVALS
*
 1      GENERATE   280,FN1
*
* QUEUE 1--TRANSMITTER QUEUE
*
 2      QUEUE      1
 3      TEST E     Q3,K7,5
 4      LOGICS     1
 5      TEST E     Q3,K6,7
 6      LOGICR     1
 7      GATE LR    1
 8      SEIZE      1
 9      DEPART     1
 10     ADVANCE    100
 11     RELEASE    1
 12     TABULATE   4
*
* QUEUE 2--FORWARD AND REVERSE CHANNEL
*
 13     QUEUE      2
 14     SEIZE      2
 15     DEPART     2
 16     ADVANCE    0
 17     RELEASE    2
 18     TABULATE   5
*
* QUEUE 3--RECEIVER
*
 19     QUEUE      3
 20     ADVANCE    100
 21     SEIZE      3
 22     DEPART     3
 23     RELEASE    3
 24     TABULATE   6
 4      TABLE      M1,100,10,40
 5      TABLE      M1,100,10,10
 6      TABLE      M1,200,50,40
 25     TERMINATE  1
*
*  GENERATE 10000 EVENTS
*
        START      1000
```

Figure A.GPSS Simulation Source Code