

2



# Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

RECEIVED

LAWRENCE  
BERKELEY LABORATORY

MAR 12 1984

LIBRARY AND  
DOCUMENTS SECTION

## Computing Division

Presented at the Computer Networks Symposium,  
Silver Spring, MD, December 13, 1983

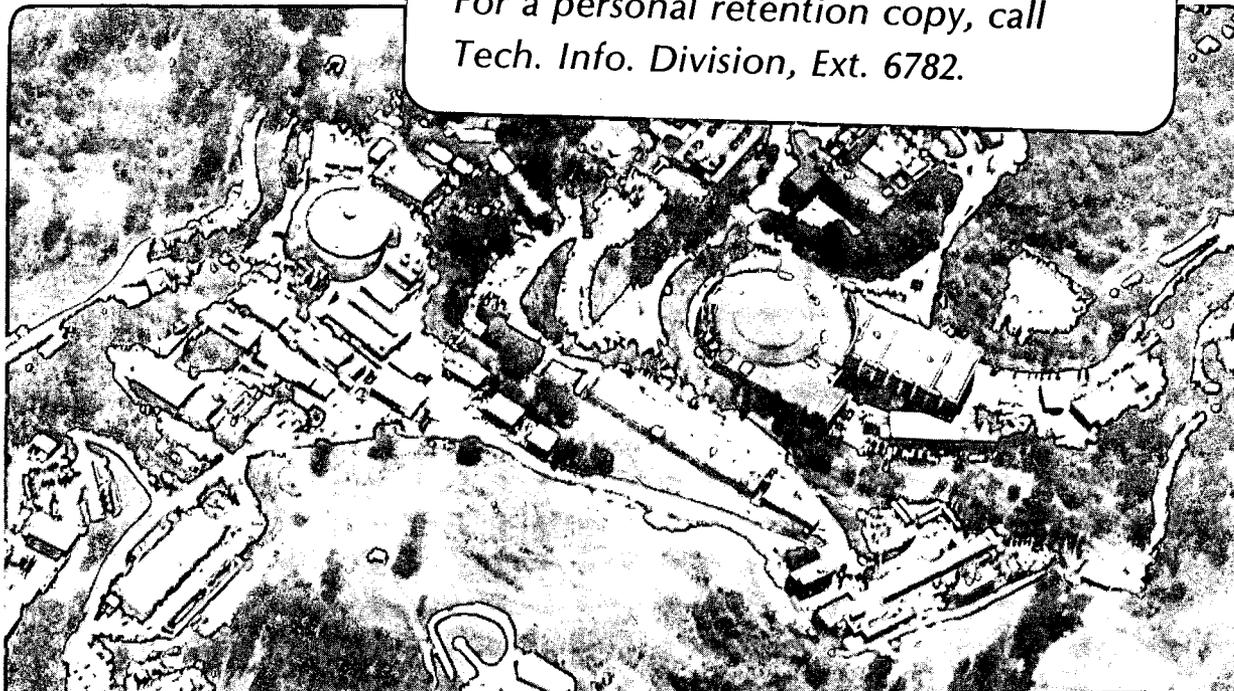
TOKEN RING LOCAL AREA NETWORKS  
A Comparison of Experimental and Theoretical  
Performance

J. Sventek, W. Greiman, M. O'Dell,  
and A. Jansen

June 1983

### TWO-WEEK LOAN COPY

*This is a Library Circulating Copy  
which may be borrowed for two weeks.  
For a personal retention copy, call  
Tech. Info. Division, Ext. 6782.*



LBL-16254

## **DISCLAIMER**

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

TOKEN RING LOCAL AREA NETWORKS

A Comparison of Experimental and Theoretical Performance

J. Sventek, W. Greiman, M. O'Dell, A. Jansen

Lawrence Berkeley Laboratory  
University of California  
Computer Science Department

June 1983

Presented at the Computer Networks Symposium, Silver Spring, MD.,  
December 13, 1983.

This work was supported by the Applied Mathematical Sciences Research  
Program of the Office of Energy Research of the U.S. Department of  
Energy under Contract DE-AC03-76SF00098.

# Token Ring Local Area Networks

## A Comparison of Experimental and Theoretical Performance<sup>†</sup>

*J. Svontek, W. Greiman, M. O'Dell<sup>‡</sup>, A. Jansen<sup>§</sup>*

*Computer Science and Mathematics Department  
Lawrence Berkeley Laboratory  
Berkeley, CA 94720*

LBL-16254

### *Abstract*

Many existing performance models for local area networks are based on the assumptions of random traffic patterns and infinite buffer capacity. All real network implementations violate the infinite capacity assumption, while an increasingly popular class of local area network applications violates the randomness assumption. This paper describes the experimental measurements of the characteristics of a commercially available token ring network. It is shown that the violation of both assumptions invalidates model predictions, and that the utility of all access control mechanisms for such applications must be re-evaluated. Finally, it is shown that congestion is a very real problem even in small "simple" local area networks. Schemes for controlling interface congestion may require rather novel approaches to the problem.

### **1. Introduction**

The design and analysis of local network protocols has largely concentrated on "link access" protocols like Ethernet [Metc76], with analytic analysis and simulation providing powerful tools for exploring protocol behavior. The general procedure is to create an analytic model of the protocol under study, derive an approximate analytic solution, and then verify it by comparison with simulation studies [Bux81b, Jafa80, Toba82]. These modeling efforts have shown that critical user-visible characteristics of a local network (e.g. stability and fairness) depend strongly upon the access control mechanism used by the interconnection technology.

Most network protocol models, whether for analytic or simulation purposes, make several simplifying assumptions. Two of these assumptions are of particular interest to this discussion:

---

<sup>†</sup>This work was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research of the U.S. Department of Energy under Contract DE-AC03-76SF00098.

<sup>‡</sup>Current address: Group L Corporation, 481 Carlisle Drive, Herndon, VA 22070.

<sup>§</sup>Current address: EDP-Center, University of Oslo, Pb 1059, Blindern, Oslo 3, NORWAY.

- (1) Each receiver on the network is assumed to have infinite buffer capacity and no processing dead time, implying that a receiver can actually receive all messages successfully placed onto the medium. This permits the models to focus on the fairness and stability of the access control mechanism with respect to each transmitter's ability to place messages on the communication medium.
- (2) The distribution of packet lengths is assumed to be exponential; the distribution of packet arrivals is assumed to be Poisson, which results if there is no correlation between the packets. This absence of correlation is essential to permit closed form analytic solutions for the characteristics of a network.

In any modeling effort, it is necessary to validate the models by comparison with the performance of actual networks. In particular, it is necessary to show that either the model assumptions hold true, or that the network performance is independent of the assumptions for the application regime which dominates the use of the communication medium. The measurements described in this paper attempt to compare the model predictions with a real token ring network, where the dominant application of the network is to provide communication between several disk-less work stations and a file server.

The choice of this application regime was motivated by several factors:

- (1) An increasing number of "disk-less workstations" are becoming commercially available.
- (2) The packet size distribution of such applications is expected to be highly non-random.
- (3) The use of the network for paging/swapping by the client systems, coupled with the expected persistence and optimized network access of the client operating systems in such cases, should present a heavy, non-random (in time) load to the network.

The remainder of the paper describes the performance measurements conducted for a commercially available 10-Mbit token ring for use in such an environment. Section 2 contains a description of an idealized token ring. The specifics of the measured hardware are outlined in section 3, followed by the experimental design in section 4. Section 5 consists of the results of the measurements. In section 6, we summarize the results of the measurements and describe their impact upon local area networks in general. Section 7 offers some proposals for addressing the problems discovered in the course of this work.

## **2. Properties of an Ideal Token Ring**

In a token ring, access to the transmission channel is controlled by sequentially passing a permission token around the ring. When a station wishes to transmit, it waits until it receives the token, at which time it transmits its message before sending the token on to the next station. Each station is responsible for removing its own packets from the ring.

The performance model which describes this operation is that of a multi-queue system with a non-exhaustive cyclic server [Kueh79]. The basic features of such a system are:

- (1) The throughput vs. offered-load curve (see figure 1) increases linearly until 100% channel utilization is reached. Beyond 100%, the curve remains constant. This latter feature illustrates that the token ring is stable when confronted by more than 100% offered-load.

- (2) The cyclic polling system enforced by the access method guarantees that each node has equal access to the medium. As a result, the access method is fair.
- (3) The polling nature of the access method tends to eliminate randomness on the ring when the offered load is high.

### **3. The PRONET[Prot82] Local Network Interface**

The ring interface hardware used in these experiments is the PRONET local network interface, manufactured by Proteon Associates, Inc. When 2 or more hosts are interconnected via these interfaces and a wire center, a star-shaped ring [Salt80] results. Such a network provides one-way baseband communication between nodes at 10 Mbits/second.

Some of the salient features of the interface are:

- (1) Full-duplex Direct Memory Access is provided by the device, thus permitting concurrent transmit and receive DMA operations.
- (2) The receiver portion of the device is singly buffered. After a packet has been received, the interface is busy until the packet has been transferred to host memory.
- (3) The transmitter portion of the device is singly buffered, i.e., DMA cannot overlap transmission on the ring.
- (4) The transmit buffer must be reloaded to retransmit a packet.
- (5) If a packet is not accepted by its destination because the receiver buffer is not yet empty, a bit indicating "message refused" is set at the end of the packet. This bit is returned to device state registers permitting software to inspect the outcome of the transmission attempt.

### **4. Experimental Configuration**

The test configuration for measuring the performance of the PRONET interfaces consisted of a single DEC PDP-11/34 and two DEC VAX-11/780's. The PDP-11/34 was configured with core memory and a memory cache. The PRONET Unibus interface accesses Unibus memory directly on the 11/34, while it accesses VAX memory through the intervention of the "buffered data paths" in the VAX Unibus Adapter (UBA).

Measurements of raw hardware event timings were obtained by placing probes on the Unibus backplane and measuring the time intervals on a calibrated oscilloscope.<sup>1</sup> For the measurements requiring software, stand-alone programs (written in assembly language) were used. The network interface was the only device active in each machine during the tests. The programs busy-waited for each transfer to complete, thus avoiding any interrupt latency. When transmitting, each system simply transmitted uniform size packets as quickly as possible, thus producing a deterministic arrival distribution similar to the continuously queued sources of [Shoc80]. When receiving, each system continuously waited for a packet to arrive. Upon arrival, the message was copied into memory, and the device was re-enabled for the next message. The DMA copy to memory was initiated by the device and the program looped on a status bit until the receive-copy composite was finished.

Additional capabilities were provided in the software on the 11/34 to drive some of the measurements:

---

<sup>1</sup>Proteon Associates graciously provided the logic and timing diagrams necessary for locating test points and understanding detailed interface operation.

- (1) The software could optionally delay a specified time after successfully completing each transfer. This permitted the code to simulate a fixed processing interval required for each packet.
- (2) When receiving packets, a history of the source host of each packet was kept to permit the fairness of the system to be determined.

## 5. Performance Measurements

### 5.1. Throughput

Since the interface is singly buffered in each direction, with no overlap of DMA and transmission on the ring, the time to transmit (or receive) data is

$$\text{software time} + \text{DMA time} + \text{ring transfer time}$$

The ring transfer time is about 0.80  $\mu\text{sec}/\text{byte}$  for large messages (10Mbits/sec with small header and trailer). By placing probes on the Unibus backplane, the DMA times were measured for transfers to memory (receive) and from memory (transmit). The results are:

Machine	Direction	Time ( $\mu\text{sec}/\text{byte}$ )
11/780	Transmit	2.23
11/780	Receive	2.23
11/34	Transmit	1.17
11/34	Receive	0.99

Note that the VAX has a very slow DMA rate even though the buffered data path was used. The VAX receive and transmit times are equal since the buffered data path masks memory access delays. On the PDP 11/34, receive and transmit times are unequal since there is no buffering between memory and device and different bus requests are made for read/write access to memory. As one can easily see, the DMA times are greater than the ring transfer time in all cases.

The maximum transfer rate can be calculated from the above measurements. Assuming no software overhead and large packets, the maximum transmit rates are:

Machine	DMA + Ring Time ( $\mu\text{secs}/\text{byte}$ )	Transmit Rate (Mbits/sec)
11/780	3.03	2.64
11/34	1.97	4.06

These rates are upper bounds since no software overhead was assumed. The following rates were measured sending 512 byte messages using the test programs described above:

Machine	Transmit Data rate (Mbits/sec)	Receive Data Rate (Mbits/sec)
11/780	2.56	2.56
11/34	3.84	4.23 (est)

The 11/34 receive data rate is estimated from the zero software overhead values because we were unable to drive the 11/34 to a receive data rate limit. Two points are worth noting about this data:

- (1) The software times in these test programs are small by design. In real networks, the software times dominate.
- (2) For both types of CPU's, the maximum bandwidth achievable by a single machine does not remotely approach total utilization of the medium. This inability to receive each message on a loaded network represents a clear violation of the infinite capacity assumption.

## 5.2. Stability

Measurements of throughput vs. offered-load were made for the 11/34 sending to one of the 11/780's (see figure 2). Note that the throughput curve experiences a sharp dip at the point where the 11/34 exceeds the maximum receive rate of the 11/780. From this point on, every other message is missed by the receiver, since the receiver's ring interface is busy DMA'ing the last message into host memory.

The dip in the throughput curve indicates that this ring implementation is not stable with respect to a single transmitter and receiver. While several transmitters would be able to originate a constant amount of traffic once the medium was completely saturated, consistent with the usual model predictions, the successful traffic to any particular receiver would experience the type of instabilities seen in figure 2. Such instabilities can only be avoided if each receiver is able to receive every message on the medium.

## 5.3. Fairness

In an attempt to measure the fairness of the ring, both 11/780's transmitted 512-byte packets continuously to the 11/34. A measurement run consisted of starting the 11/780's and then receiving 10,000 packets as counted by the 11/34. The software in the 11/34 kept a history of the sender of each packet received (indicated by the second byte of the message, as inserted by the transmitter's hardware). Several measurement runs were conducted, with fixed delays of varying sizes introduced after each received packet in the 11/34. The delay models a fixed processing overhead for each packet. If the access method is truly fair, then one would expect that for a given run, 50% of the packets should originate from each host, independent of the processing delay. A quick perusal of figures 3 and 4 will indicate that this is not so for a large number of delay values, where one of the transmitters is always unsuccessful in having its messages received.

The observed starvation behavior can be explained in the following manner. The polling nature of the ring's access method causes packets from the 11/780's to alternate. The upper portion of figure 3 shows this synchronization of packets ("convoys") to scale. The singly buffered nature of the network interface causes the receiver to be unable to read the second message in the convoy, since it is busy copying the first message into host memory. Defining the transmission time of a packet on the wire as  $t$  and the delay between packets from a

particular host as  $d$ , then the only receiver delay regions ( $D$ ) in which a fair distribution will be detected are those when

$$(n * d) < D < ((n * d) + t) \quad n=1,2, \dots$$

i.e. when the delay is just long enough to cause the receiver to miss the start of the message from the previously successful transmitter, but just short enough to pick up the start of the trailing message of the other transmitter. This description easily extends to cover the situation when the number of transmitters is larger than 2, or the number of receive buffers is larger than 1.

One should also note the multi-valued nature of the function just preceding each transition region in figure 3. The lower branch of the function in these cases results when the initial read for a particular run locks onto the second packet of a convoy. The delay values in these regions are such that the receiver will continue to receive the second packet in the convoy, thus resulting in complete starvation of the first transmitter. Since the position of the convoy with respect to the receiver is random for the first read in a particular run, one would expect the relative population of the two branches to be

$$(d - t) / t$$

As a result, performance predictions for a system operating in these regions have very little value.

## 6. Discussion

The data presented above indicate that very counter-intuitive behavior occurs when the assumptions of infinite buffer capacity and random traffic are violated. The following discusses these results in the context of the specific ring interface measured, as well as for networks in general. A solution to these problems for token rings is proposed in section 7.

### 6.1. Finite Capacity

When discussing the finite capacity problem, two parameters determine the behavior of the network when there is sustained traffic: the receiver delay ( $D$ ), which is the time necessary for the interface to be ready for another message from the network, and the transmit time for a packet on the wire ( $t$ ). Two particular operating situations can be characterized:

$D < t$ :

In this situation, it takes the transmitter longer to originate the message than it does for the receiver to copy it to memory and be ready for another message. By providing two receive buffers and permitting DMA to overlap message reception, a receiver can see all messages on the ring. Previous low-speed local networks (eg. 3 Mbit Ethernet) fall into this category.

$D > t$ :

Since the reception DMA is longer than the transmission time, no amount of buffering will permit the receiver to handle sustained traffic. The ring interfaces measured fall into this category. Note that if the traffic can be characterized as bursty as opposed to sustained, then an optimal number of buffers can be provided to smooth out the traffic distribution (on the average).

While the current interfaces could be pushed into the  $D < t$  category via engineering changes, the finite buffer problem scales with the current trends in

local networks. The attractiveness of higher speed local nets guarantees that the network speed will be faster than the host channel speeds - i.e.  $t \ll D$ .

The messages lost due to finite capacity will generally trigger timeout and retransmission mechanisms in higher level protocols. Moreover, the coupling between protocol layers can be quite non-linear. Most importantly, no amount of upper-level protocol effort can produce throughput in the face of low-level starvation. See [Donn79] and [Nabi83] for studies of these problems.

It should also be apparent that this particular problem is independent of the access control mechanism. All that is required is that the traffic be bursty enough to more than exhaust a receiver's message buffers. Independent measurements of such anomolous behavior for 10 Mbit Ethernet interfaces are reported in [Nabi83].

## 6.2. Unfair Traffic Distributions

The grossly unfair distributions displayed in figure 3 can be attributed to two causes:

- (1) lack of multi-buffering in the receiver interface
- (2) the non-random ordering of the messages on the ring

The previous discussion has shown that no amount of buffering will guarantee stable performance under all traffic conditions. It is therefore necessary to concentrate on the non-random ordering of packets on the ring.

The regularity of the traffic offered by the continuously queued servers, coupled with the synchrony of the ring access control mechanism, forces the messages in a convoy to be identically ordered. Since limited buffering on the receiver's interface will only permit  $n$  consecutive messages to be received from the convoy, the remaining messages in the convoy will always be missed by the receiver. This situation leads to the extreme starvation seen for many values of receiver delay seen in figure 3.

Broadcast mechanisms, with the randomization inherent in their collision resolution protocols, should not be quite so susceptible to this pathology. It is feasible, however, that several identical, continuously queued servers could synchronize in time such that the collision resolution protocol would not be invoked, but the resulting traffic could still overflow the limited buffering of a single receiver.

For example, assume that the total offered load to the wire is ~50% of capacity, and that the resulting load is >100% of the capacity of a single receiver. The number of collisions (randomizing events) would be expected to be low, resulting in constant ordering of messages as they pass the receiver. If the interface is subject to finite buffering, then the messages from the same hosts will be missed each time, in analogy with the ring situation. Again, it is important to note this interface congestion anomaly is not induced by high total load, but by high loads with respect to a single receiver.

## 7. Proposals for Improving Performance

Please note the problems seen in these experiments are not an indictment of the specific hardware we used. While some problems do result from specific hardware design decisions, these decisions did not, by themselves, produce the anomalies seen. On the contrary, we believe these results show that congestion control is a real, serious problem in the design of even "simple" local area networks.

### 7.1. Receiver-Access Protocols

Historically, most local network design, analysis and modeling has been done for *link-access* protocols. While the value of such efforts cannot be overemphasized, this study shows that stable link access characteristics, while necessary, do not guarantee adequate network performance in the face of interface congestion. Therefore, we propose that after a new protocol design has been shown *link-access stable*, efforts be applied to designing an accompanying *receiver-access* protocol to be *receiver-access stable*. Receiver-access stability implies transmitters will not see suddenly-decreasing throughput or delivery unfairness in the presence of highly asymmetric (congested) conditions like those in this experiment. A receiver-access protocol embodies the congestion control strategy used by each transmitter to optimize the chances that its packets successfully placed onto the medium will actually be successfully received. This is a somewhat different view in that congestion control is usually seen as a global issue instead of relating to a single host. In most local networks, congestion of the network medium is handled by the fairness and stability of the basic link-access protocol. Interface congestion is controlled with the receiver-access protocol.

Analyzing receiver-access protocols implies modeling both the receiver implementation and the strategy used by the transmitter for providing "best effort". Note that this is *not* a call for guaranteed-reliable link-level protocols, but it is an acknowledgement of a specific kind of unreliability and the desire to reduce its impact. Receiver-access protocols will quite likely complicate models to the point of analytic intractability, thereby requiring simulation and measurements of real implementations. Only in this way can reasonable guarantees be made about the behavior of a real network. As a first attempt, we now present an analysis of the experimental system from the receiver-access point of view.

### 7.2. An Example of a Poor Receiver-Access Protocol

The initial receiver-access protocol implemented in the driver software for these tests and the production local network which uses the ring interfaces is very simple-minded, and consequently very poor. Recall the "refused" bit indicates that the recipient of a packet is an active member of the ring, but did not take the packet off the wire because its input buffer was not empty. The first-order strategy implemented is to simply, and immediately, retransmit a packet when it returns marked "refused". We thereupon discovered one design flaw in the interface: the transmit buffer must be reloaded via DMA to do the retransmission. A holding register for the initial byte count would allow retransmission of a refused packet without the overhead of the DMA. The critical error, however, is the retransmission strategy, not the need for the additional DMA.

By repeatedly attempting to transmit the packet until the number of refusals reached some threshold, we guaranteed the creation of the convoy traffic pattern which induces the anomalies described above.

### 7.3. A Better Receiver-Access Protocol

Provided with the knowledge that some formal mechanism is needed to stabilize receiver access and control interface congestion, we examined the problem to see how it can be accomplished in the framework of the interfaces at hand, and how the interface design could be improved to make this easier. Some of the following discussion could apply to an Ethernet-style network, but the scheme described below relies critically upon the back-channel signal

provided by the "refused" bit.

Assuming the absence of message priorities, which will be discussed later, the main problem is one of fairness, or insuring transmitters all have equal likelihood their messages are missed. This requires a mechanism for "stirring" the convoy so transmitters trade positions in fair ways.

One challenging characteristic of a token ring is that each transmitter's observed frequency of message origination varies with the load on the ring. This is not surprising, but the extrema of these values differ by a factor of over 200 for large rings. To reorder the convoy, each sender must wait a random time interval based upon one convoy rotation time, or the original packet order will be preserved. This means a simple scheme based upon a fixed convoy rotation time would wait far too long for small rings and reduce the available throughput. Providing the interface with a mechanism which can provide software with a measurement of the "speed" of the last convoy rotation would allow more accurate estimation of the waiting times necessary to reorder the convoy. Schemes for providing this mechanism are currently under study.

Other improvements would arise from relocating the "refused" bit. In the current ring implementation, the refused bit is at the end of a message; relocating it to the front of the packet would allow large packets to be terminated immediately upon receipt of the refused bit. This relocation helps in two ways: less bandwidth is used for useless packets, and the abrupt termination helps break the synchrony of the convoy.

The receiver-access protocol which evolves from this analysis is essentially the Ethernet CSMA/CD algorithm with some definitions changed [Metc76]. The "refused" bit is taken to mean "collision" at the receiver. The convoy rotation timer in the interface is used to estimate the loading of the ring and is used in formulating the initial random backoff interval, with successive "refusals" exponentially scaling the back-off interval. Providing a countdown timer in the interface to process these delays would make the software simpler.

At present, this receiver-access protocol has not yet been implemented; a software implementation is being designed to support further experiments.

#### **7.4. Congestion Control and Multiple Buffers**

We now return to the issue of multiple buffering in receivers. Many current interface designs subscribe to the idea that having "enough" buffers in the receiver will alleviate the problems described in this study. While having more buffers allows longer convoys to be received without incapacitating congestion, disaster is merely postponed. Given a network with no receiver-access protocol, when the service times of the packets queued in receiver interface buffers exceed their transmission times, the receiver *will* fall behind with the eventual result being lost packets from interface congestion. Packets lost because the receiver is too congested to accept it are no different from those lost because the checksum was corrupted by failing hardware. Providing massive buffering may well reduce the likelihood of debilitating congestion to levels acceptable for *some* applications, but buffering alone *cannot* be the basis for meaningful guarantees about network behavior.

One other possibility to consider is that excessive buffering could exacerbate interactions with higher-level protocols by preserving stale data packets in an already congested interface. For instance, many datagram-based or real-time protocols (packet voice, for instance) want the most recent version of a retransmitted packet, and not one which has been delayed in an over-buffered receiver interface due to congestion. Designers of systems with low latency requirements should evaluate the receiver buffering issue with extreme care.

### **7.5. Implications for Rings with Message Priorities**

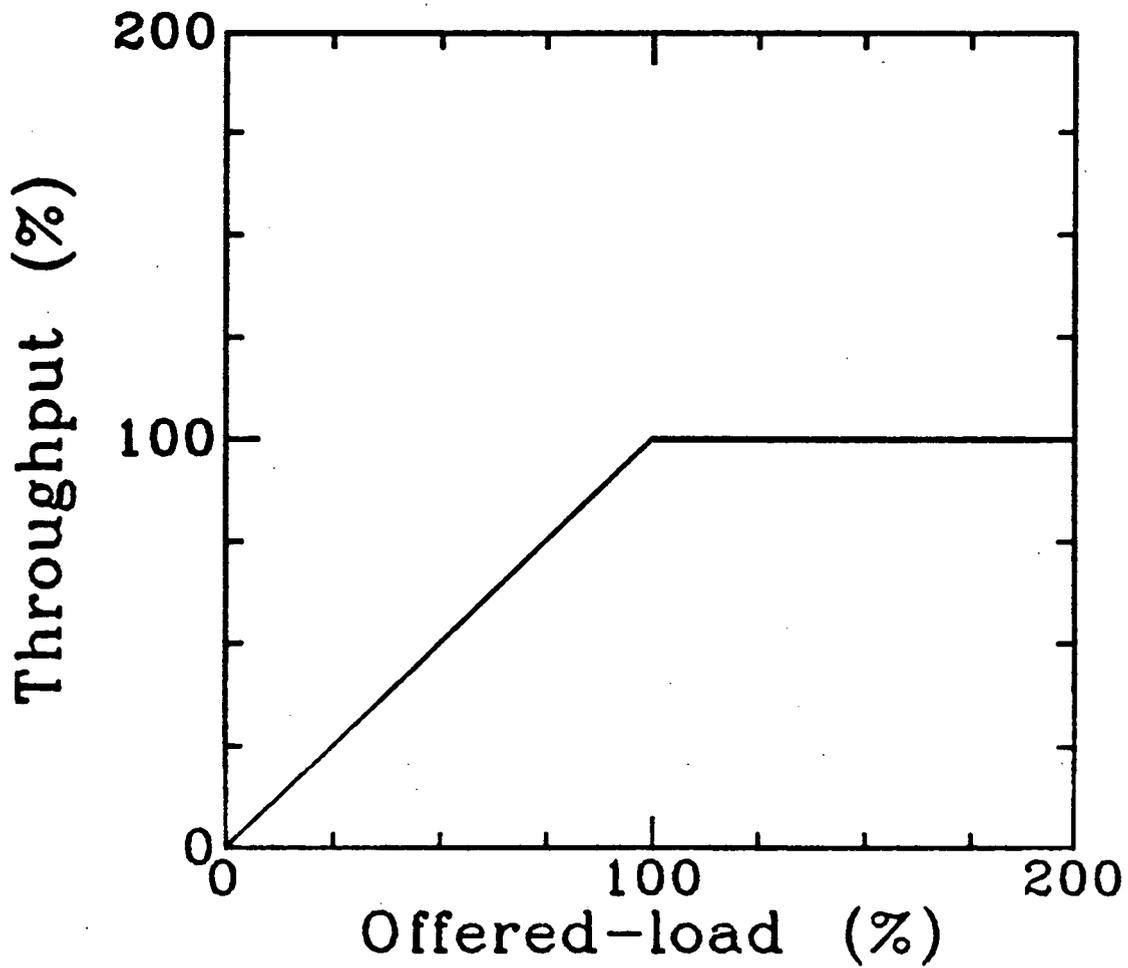
Several ring designs include provisions for message priorities [Bux81a] which are intended to provide a better quality of service for high priority messages. The results of this study cast some doubt on the implementability of such differentiated service. Serious problems arise when the traffic rate of low priority messages into one interface becomes sufficient to saturate the finite internal buffering of the interface. This could easily produce starvation of a high priority source with little possibility of resolution. The only fix we see at present would be to design the interfaces to discard low priority messages from the internal buffers when a higher priority message arrives. Even then, there are no absolute guarantees, but the likelihood of prompt delivery is improved.

### **8. Conclusions**

This study revealed the unsettling fact that current performance models do not appear to predict the performance of real-world network implementations, particularly in regimes prone to interface congestion. The reasons for the discrepancies were analyzed, and schemes for improving modeling, congestion control protocols, and interface hardware architecture were presented. We believe these proposals indicate directions of further fruitful research.

## References

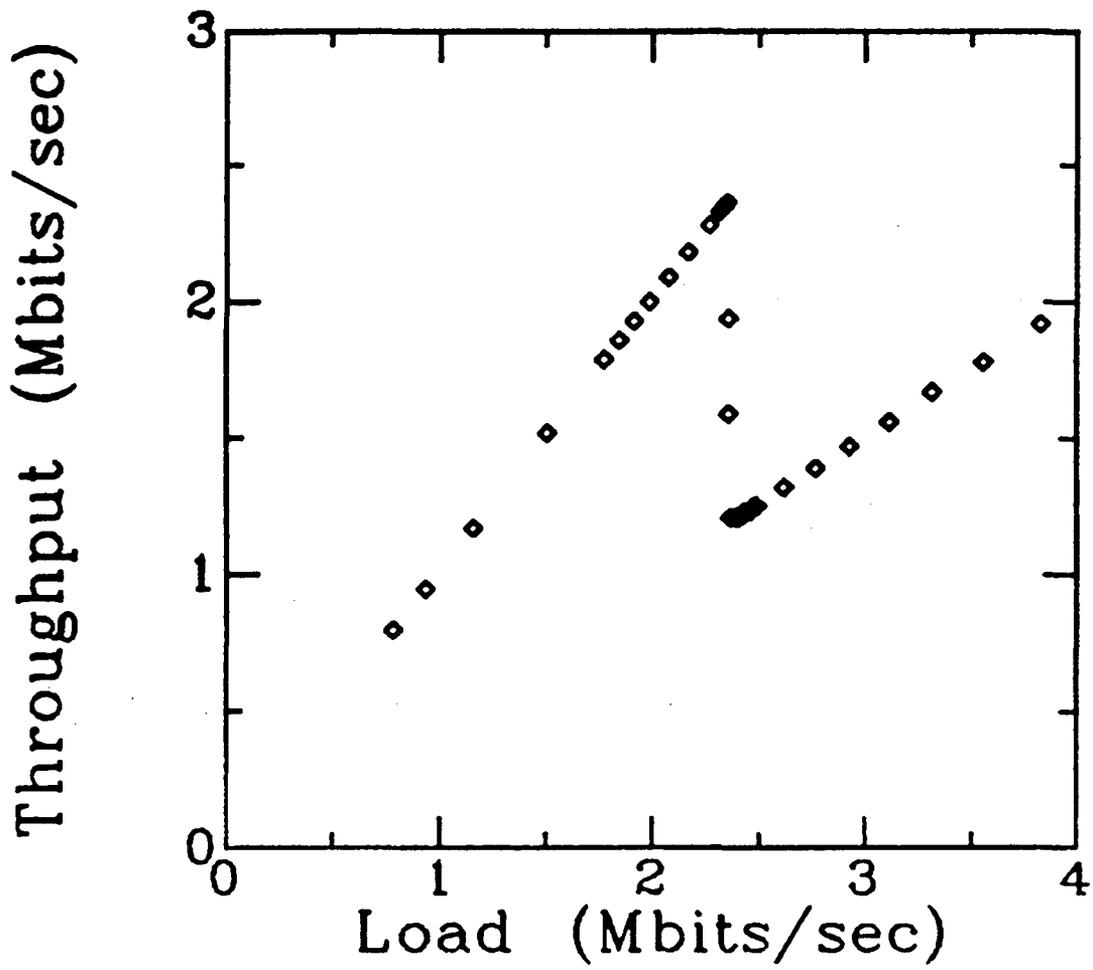
- Bux81a  
Bux, W., F. Closs, P. Janson, K. Kummerle, and H.R. Muller, "A Reliable Token-Ring System for Local-Area Communication," Report RS 1095 (#39517), IBM Zurich Research Laboratory, 1981.
- Bux81b  
Bux, W., "Local-Area Subnetworks: A Performance Comparison," *IEEE Trans. Commun.*, Vol. 29, October 1981, pp. 1465-1473.
- Donn79  
Donnelley, J.E. and J.W. Yeh, "Interaction Between Protocol Levels in a Prioritized CSMA Broadcast Network," *Computer Networks*, Vol. 3, 1979, pp. 9-23.
- Jafa80  
Jafari, H., T.G. Lewis, and J.D. Spragins, "Simulation of a Class of Ring-Structured Networks," *IEEE Trans. on Computers*, Vol. 29, May 1980, pp. 385-392.
- Kueh79  
Kuehn, P.J., "Multiqueue Systems with Nonexhaustive Cyclic Service," *Bell Syst. Tech. J.*, Vol. 58, March 1979, pp. 671-698.
- Metc76  
Metcalf, R.M. and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, Vol. 19, July 1976, pp. 395-404.
- Nabi83  
Nabielsky, J., "Interfacing to the 10 Mps Ethernet: Observations and Conclusions," Mitre Corporation report, March 1983.
- Prot82  
Proteon Associates, Inc., "Operation and Maintenance Manual for PRONET UNIBUS HSB," August 1982.
- Salt80  
Saltzer, J.H. and K. Pogran, "A Star-Shaped Ring Network with High Maintainability," *Computer Networks*, Vol. 4, 1980, pp. 239-244.
- Shoc80  
Shoch, J.F. and J.A. Hupp, "Measured Performance of an Ethernet Local Network," *Communications of the ACM*, Vol. 23, December 1980, pp. 711-721.
- Toba82  
Tobagi, F.A., "Carrier Sense Multiple Access with Message-Based Priority Functions," *IEEE Trans. Commun.*, Vol. 30, January 1982, pp. 185-200



XBL 842-574

Figure 1

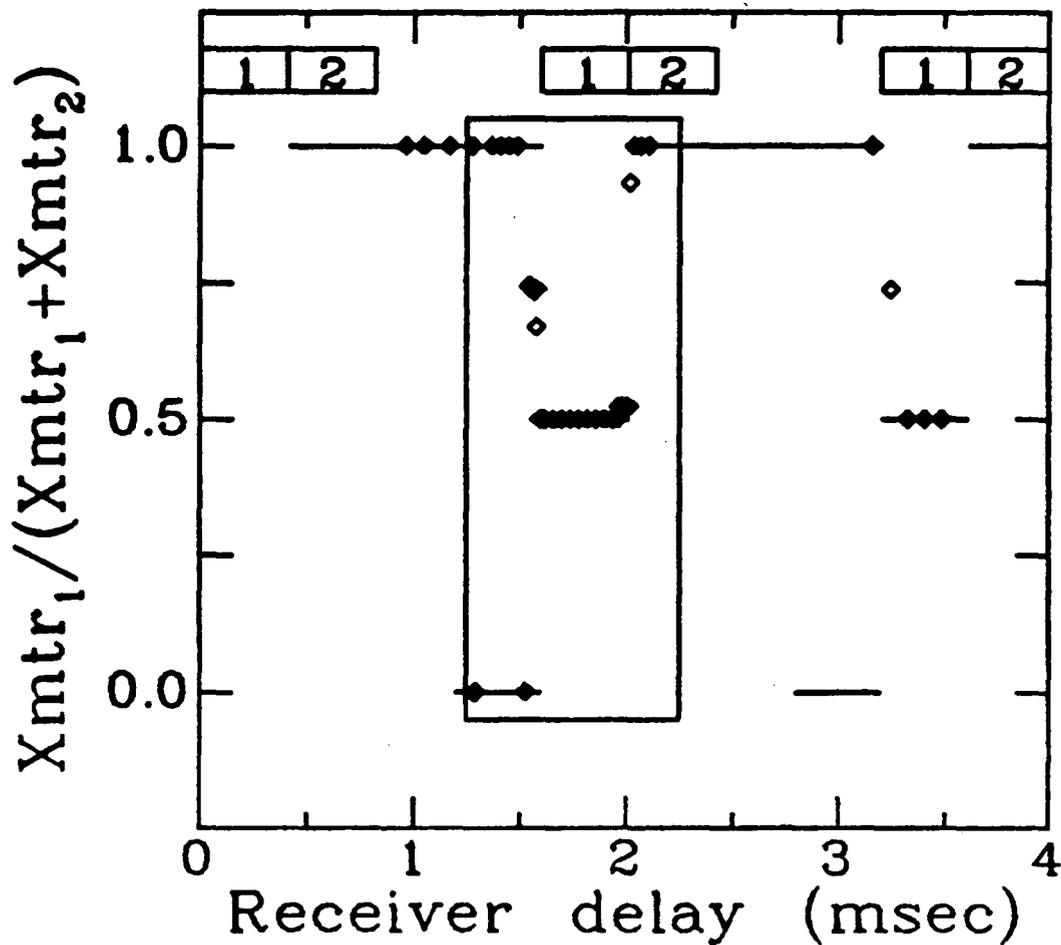
Throughput vs. offered-load for an ideal token ring.



XBL 842-575

Figure 2

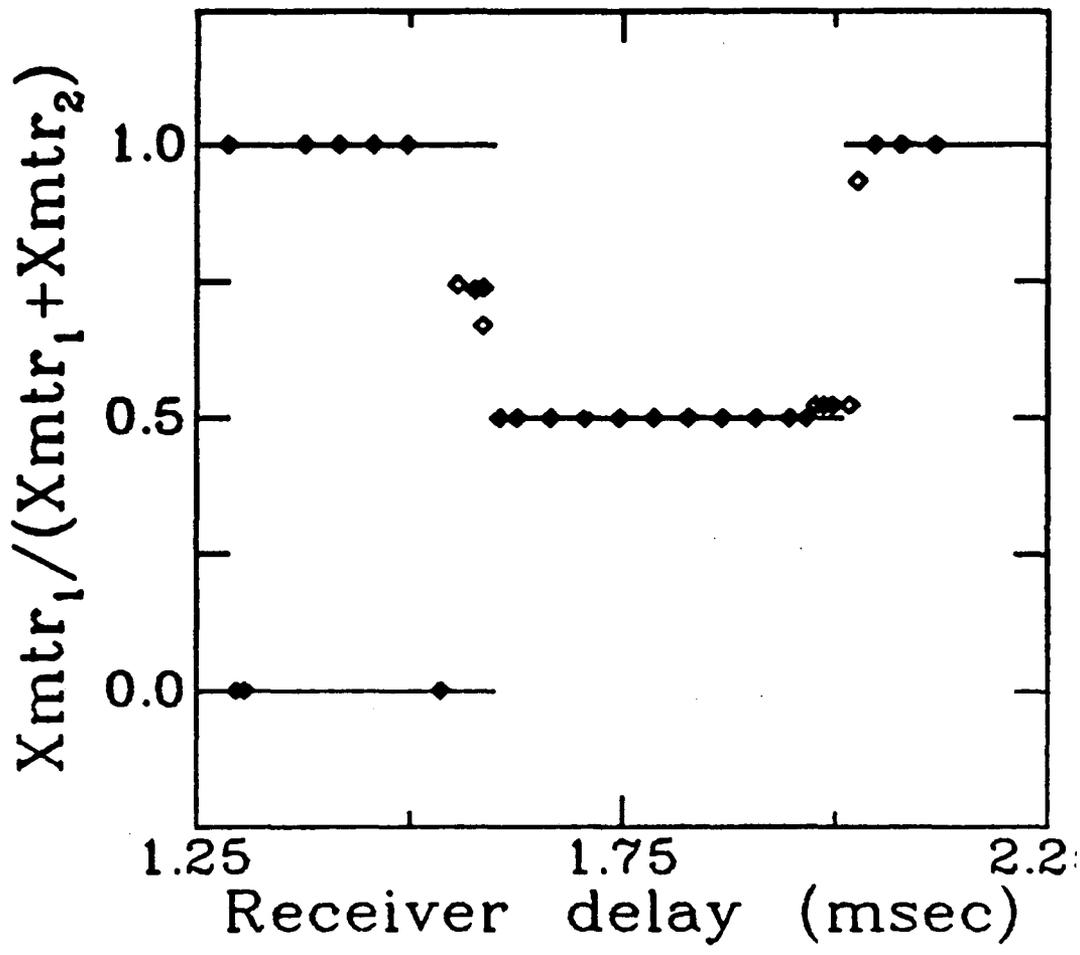
Throughput vs. offered-load for a single transmitter and single receiver pair. The severe transition at 2.3 Mbits/sec is due to the transmitter overrunning the receiver.



XBL 842-576

Figure 3

Fraction of packets received from transmitter 1 vs. the receiver's processing delay for each received packet. The solid curves are the expected values from analysis of the convoy pattern displayed at the top of the figure. The multi-valued nature of the function prior to each transition region is due to the initial conditions of the receiver for a particular run, as described in the text.



XBL 842-577

Figure 4

An expansion of the first transition region of figure 3.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT  
LAWRENCE BERKELEY LABORATORY  
UNIVERSITY OF CALIFORNIA  
BERKELEY, CALIFORNIA 94720