PERFORMANCE MODELING AND OPTIMIZATION OF PARING

by

Yaoshuang Qu
Lawrence H. Landweber
Miron Livny

Computer Sciences Technical Report #614

September 1985

PERFORMANCE MODELING AND OPTIMIZATION OF PARING

Yaoshuang Qu, Lawrence H. Landweber, and Miron Livny
Dept. of Computer Sciences

Univ. of Wisconsin - Madison

ABSTRACT

This paper introduces the PaRing, a new token ring architecture, and studies analytical approaches for modeling and optimizing its performance. The PaRing is distinguished from earlier ring architectures by its ability to support concurrent multiple communication paths on a single loop in an efficient and fair manner. A multiple server queueing network model, in which a communication path is viewed as being served by a server, is used to determine packet waiting time for symmetric traffic. The key to the modeling is to convert the multiple server system into multiple independent and identical single server systems. The model is verified by simulation. The performance of the PaRing depends on the number of concurrent paths. An optimization method is developed to maximize the number of concurrent paths.

1. Introduction

This paper presents the PaRing [1], a new token ring local area network recently developed by the authors, and studies analytical approaches for modeling and optimizing its performance characteristics under a given work load. The PaRing utilizes a token for medium access control as in the case with the standard token ring architecture. The important difference from earlier work is the combination of the following characteristics. First, the stations of the PaRing may initiate a transmission not only upon receipt of a token, but also upon receipt of a data packet that is addressed to it. Second, the PaRing uses a *destination removal mechanism,* in which packets are removed by their destination stations rather than by source stations. If a transmission opportunity, i.e., the arrival of either the token or a data packet, comes to a station and the station does not have a packet to send, it either forwards the incoming token or generates and transmits a token while removing the incoming packet. When a station is transmitting its own packet and, at the same time, another packet arrives, the station keeps transmitting and either removes the arriving packet if it is a packet destined for the station or delays it, otherwise. After the station's transmission is completed, the station forwards delayed packets onto the ring. Since a station can remove an incoming packet, which may be transmitted by another station, and transmit its own packet simultaneously, the PaRing supports *concurrent multiple transmissions of packets.* Because of the concurrent multiple transmission, the PaRing has a higher *throughput* and shorter *delay time* than the standard token ring architecture. In addition, the PaRing is also superior to other ring architectures supporting concurrent multiple transmissions such as the Distributed Loop Computer Network (DLCN) [2, 3] because its properties include *automatic acknowledgement and short station latency time.* Automatic acknowledgement, i. e., packet reception acknowledgement performed by the destination station without having to send another packet carrying the acknowledgement information, lets interface hardware detect an undeliverable packet or ring break error and hence reduces the overhead of error detection and recovery and improves protocol performance. The short station latency time, which is only a single bit as compared to twenty four bits for the DLCN, reduces the delay time and increases throughput.

Since the PaRing supports concurrent multiple transmissions, a multiserver queueing system model is developed to predict packet *waiting time,* which is the time interval from the generation of a packet until the start of the transmission of the packet by its source station, for symmetric traffic. Unlike the standard multiple server system, the multiple server system of the PaRIng can be converted into multiple single server systems. Thus, instead of the multiple server system, we solve the single server system.

Within a network environment, communication traffic is not uniform in the sense that a station may be more likely to send/receive packets to/from some stations than others. For the PaRing, the closer destination stations of packets are located to their source stations, the more concurrent multiple transmissions are possible. Therefore, the configuration of stations may affect performance significantly. An analytical method is derived to minimize distances from source stations to destination stations and thereby obtain the optimal station configuration. In addition to the analytical method, a physical implementation of the optimization, using standard ring *wire centers,* is also presented.

The paper is organized as follows. Section 2 gives a description of the protocol of the PaRing, including medium access and packet removal algorithms, automatic acknowledgement, and access fairness. Section 3 develops the performance model. Section 4 derives the performance optimization method. Section 5 provides conclusions.

2. Protocol of PaRing

2.1. Media Access Protocol

Two types of packets constitute the traffic of the PaRing - access control packets and data packets. Unlike the standard token ring, where only an arrival of an access control packet carrying a free token can enable a data packet transmission, the arrival of either one of these packets may trigger the transmission of a packet in the PaRing. In the PaRing each packet has the potential to provide a station with the right to access the shared communication medium. All the information needed to determine whether or not a transmission may be initiated is contained in the header of the packet. A PaRing packet may consist of up to three fields - *Header, Body* and *Tail.* An access control packet consists of only a header whereas a data packet has all three fields. Figure 1 illustrates formats of the header, control packets and the data packet. The PaRing access control scheme utilizes three types of access control packets - *token, invalid packet,* and *error recovery packet.* The packet header has four sub-fields: *flag, identifier, priority,* and *type.* The flag is an unique bit pattern that marks the beginning of the header. The type field distinguishes the three different control packets. The type field along with the identifier distinguishes between control packets and data packets. If both the identifier and type fields are "zero's", the packet represents a token. The priority sub-field is used for access control. More details on this sub-field will be given later.

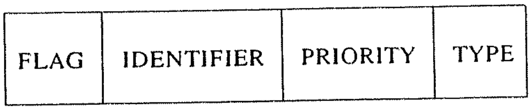
For a data packet, the identifier of the header contains its destination address. The body part has four sub-fields: *monitor, source address, data,* and *data end.* The monitor sub-field (one bit) is used for error detection. The source address is the packet transmitter's address. The data sub-field can be of variable length but cannot exceed an upper bound. The data end is another unique bit pattern, which marks the end of data. The tail part has two sub-fields, *error check* and *acknowledgement* (Ack). Both fields are for error detection. This paper does not describe the error detection and recovery scheme of the PaRing, so the function of the error recovery packet and error detection oriented fields such as the monitor and the error check fields are not explained in this section.

A station with a ready data packet cannot transmit a packet until the arrival of a packet header in which the identifier contains either all "zero's" or the station's address, the priority level is lower than the station's priority level, and the type field is "00". A header that meets the above conditions is either a token or the header of a data packet that is addressed to this station. If an arriving header is addressed to the station but the priority level is not lower than the station's priority level, a new token is generated by the station and transmitted. In both cases, the header is removed before the transmission starts. If it is the header of a data packet, the other parts of the data packet are removed during the transmission. If the station is too busy to remove an arriving data packet, it forwards the packet and sets its acknowledgement bit on the fly. When the arriving header does not meet the above conditions, the entire packet is forwarded by the station.

In order to have good throughput delay performance characteristics, the latency introduced by a PaRing station is one bit time. Since the station can hold only one bit of a transient packet, when a station recognizes an arriving header, the whole header except the last bit has left the station. If the station intends to use the header for transmission, it must remove it before the transmission starts. Instead of physically removing the whole header, the station just changes its last bit and then turn it into an invalid packet. (See Figure 1 (a)) The invalid packet will be removed from the ring when it is delayed by a station.

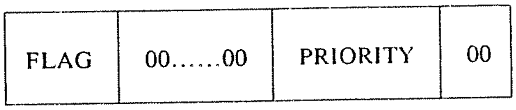
2.2. Automatic Acknowledgment, Access Priority Assignment, and Media Access Fairness

In the PaRing, we regard the communication loop together with the facilities that store delayed packets as an integrated environment. Call this environment *system.* Call the header of the token or of data packet a *valid header.* When the system is initialized, there are a fixed number of valid header(s). Assume that there is only a single valid header at system initialization time. According to the medium access and packet removal algorithms, a station cannot transmit a valid header onto the system without removing the existing valid header from the system, so there is no more than one valid header in the system at any time. This single valid header keeps circulating around the system periodically. When the transmitter of a data packet sees a valid header arriving after it has transmitted the packet and finds that the header is either a token or the header of another data packet, the transmitter knows that its packet has been removed. This feature leads to automatic acknowledgment.

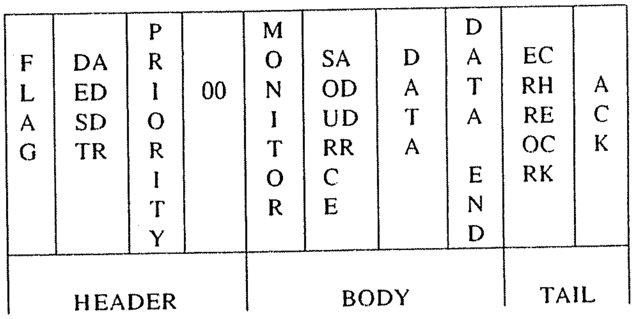


Type:
00 Token or Data Packet
11 Error Recovery Packet
01 or 10 Invalid Packet

(a) Packet Header Format



(b) Token Format



(c) Data Packet Format

Figure 1: Formats of Packet Header, Token, Data Packet

As for the medium access priority assignment, first we discuss how a station is dynamically assigned a priority level, then explain how a data packet and token obtain their priority levels. Once a station has a data packet ready to transmit, it is assigned a medium access priority level, and waits for a transmission opportunity. The priority level is determined by the time constraint of the application the packet is used for. In addition, each station has a timer called a *priority timer*. Whenever the station is assigned a priority level, it initializes the timer. If the station cannot transmit its own packet before the timer expires, it raises the priority level and reinitializes the timer. When the station starts transmission of the packet, it stops the timer and decreases the priority level. Therefore the stricter the time constraint a station has or the longer the station waits for transmission, the higher priority level the station has.

When a data packet that is not addressed to a station is passing through, the station compares its priority level with the packet's level. If the station has a higher level, it records the packet's level, which is called the *original level* of the packet at the station and enters its own level into the packet's priority subfield. All these actions are done on the fly. By doing this, a station may enter its priority level into an invalid packet. However, that does not cause any trouble. When a station removes an arriving valid header, either the header of a data packet or a token, and transmits its own packet, the priority level of the transmitted packet is decided as follows. If the station entered into the previous valid header the same priority level the removed packet has, the transmitted packet inherits the original level of the previous header. Otherwise, the transmitted packet copies the priority level of the removed header. The same principle is used to decide the token's priority level when it is generated by a station. For a station, entering its priority level into a passing packet is, in effect, a request for the next transmission right. Generally, the station entering the highest priority level obtains the next transmission opportunity unless the destination station of the passing packet has a higher priority level than the highest entered priority level. Along with the medium access and packet removal algorithms, the medium access priority assignment method ensures a fair sharing of the medium bandwidth between all stations and satisfaction of time constraints for accessing the medium by different applications.

3. Performance Model

We view the PaRing as a directed graph DG = (V , E), where

$$V = \{ 1, 2, \ldots, N \}$$

is the *node set* and

$$E = \{ (i, i+1) \mid i \in V \} \cup \{(N, 1)\}$$

is the *arc set*. Each station is represented by a node in V, each medium segment linking adjacent stations is represented by an arc in E, and each pair of communication stations, a source station and a destination station, is represented by the path from the source station to the destination station. The path is called a *communication path* and denoted by CP_i^j, where i is the source node and j is the destination node. The workload of the system is characterized by the packet arrival vector $PA[i]$, the traffic matrix T[i, j], and the packet transmission time TT. Each station receives an independent stream of packets at a rate $PA[i]$. The probability that a packet originated from i is destined to j is given by T[i, j]. We assume that the ring is in a steady state, in which the packet arrival rate and the distribution of packet destination nodes of each node do not vary with time. The system will be *symmetricly* loaded if all stations generate the same amount of traffic and packet destination addresses are uniformly distributed, i. e.

$$PA[i] = PA_{ring} / N \text{ and}$$

$$T[i, j] = 1 / N, \text{ for } 1 \le i, j \le N$$

where PA_{ring} is the total packet arrival rate of the system.

Since the PaRing supports concurrent multiple transmissions, a number of communication paths may exist at a given instance. Each of these paths can be viewed as being served by a different server. This property of the PaRing motivated us to develop an multiserver performance model for the PaRing under symmetric load. The model is an approximation model where the key parameter of the model is the number of servers. With respect to the traffic characteristic we use the following assumptions:

(1) Packet arrival interval time at each node is exponentially distributed.

(2) All packets have the same size. In other words, the transmission time of packets, TT, is fixed.

The number of servers can be decided as follows. In the PaRing, in addition to the packet arrival flow, which is an external flow and coming from the outside of the ring and transmitted onto the medium, there is an internal flow at each node. The internal flow comes from the medium and is forwarded onto the medium by the node. Call it the *forward flow*. The packet forward vector, $PF[i]$, can be derived from $PA[i]$ and $T[i, j]$. PF[i] also represents the packet rate passing arc $(i, i+1)$. For the symmetric traffic, the packet forward rate of each node or the packet passing rate of each arc is

$$PF[i] = PA[i] * (N+1) / 2, \qquad (3\text{-}1)$$
$$= PA_{ring} * (N + 1) / 2N.$$

According to (3-1), an arc only delivers a portion of the total packet arrival flow and the remainder of the flow is delivered by other arcs. These arcs may deliver different portions of the flow simultaneously. Therefore, we select the ratio of the total flow to the flow delivered by a single arc as the number of servers:

$$MS = PA_{ring} / PF[i] = 2N / (N+1) \qquad (3\text{-}2)$$
$$= 2 - 2 / (N+1).$$

Note that, in the PaRing, these multiple paths actually are established and terminated sequentially because it takes time to pass the token or packets from node to node and a path cannot be established until its transmission node obtains the token or a packet, which is addressed to it. However, if the token and packet circulation times are much smaller than the packet transmission time, TT, multiple paths can be established and terminated almost at the same time.

In the PaRing, in addition to the transmission and destination nodes, each communication path may have some intermediate nodes between the two nodes. A packet arriving at an intermediate node cannot obtain a service before the communication path, which the node is located at, terminates.

In the multiserver system, this fact can be modeled as follows. Each server has an input queue and can only serve packets of its own queue. Packets arriving at intermediate nodes of a communication path are stored in the queue of a server that represents the path. Thus the multiple server system is decomposed into MS,s independent and identical single server systems. The packet arrival rate and server utilization of each single system are

$$PA_{server} = PA_{ring} / MS = PF[i] \quad \text{and}$$

$$U = PA_{server} * TT = PF[i] * TT.$$

According to Polloczek-Khintchine Formula [5], the packet waiting time is

$$WT = TT * U / (1 - U) \qquad (3\text{-}3)$$

Series of simulations for the packet waiting time have been performed for networks with different number of stations, various sizes of packets, and different packet arrival rates. Simulators are written in DISS [6]. Results of the simulation and modeling are in Table 1 and Figure 2 through 4 where S/N and M/N denote the simulation and the modeling results of a ring with N nodes, respectively. The quality of the model is characterized by the relative difference between the waiting time measured from the simulation, WT_{simul}, and the waiting time computed from the model, WT_{model}: $(WT_{simul} - WT_{model}) / WT_{model}$. The comparison shows that results from the simulation and the model are well matched except for cases when the ring has a large number of nodes, e. g., fifty nodes, and/or a small size of packets, e. g., 128 bits. In the PaRing,

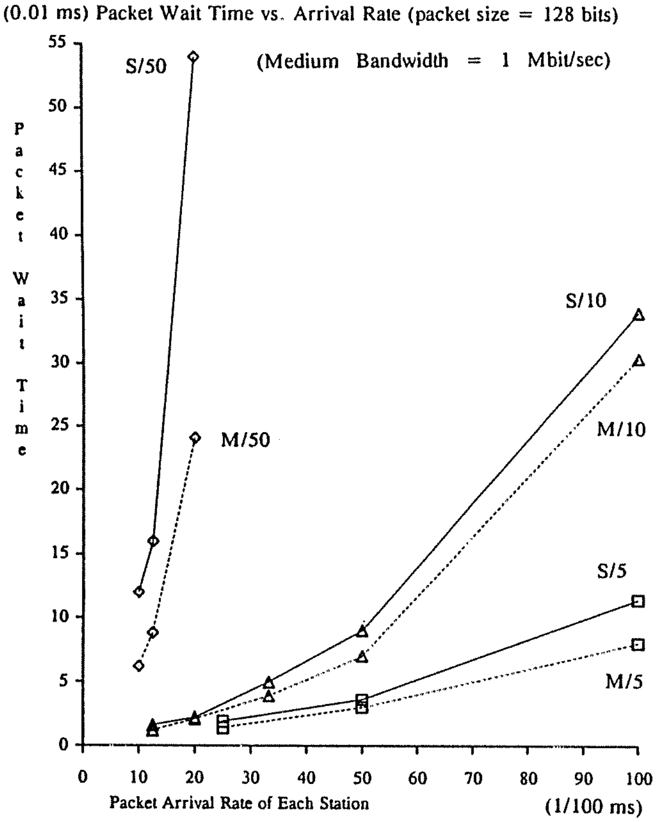


Figure 2

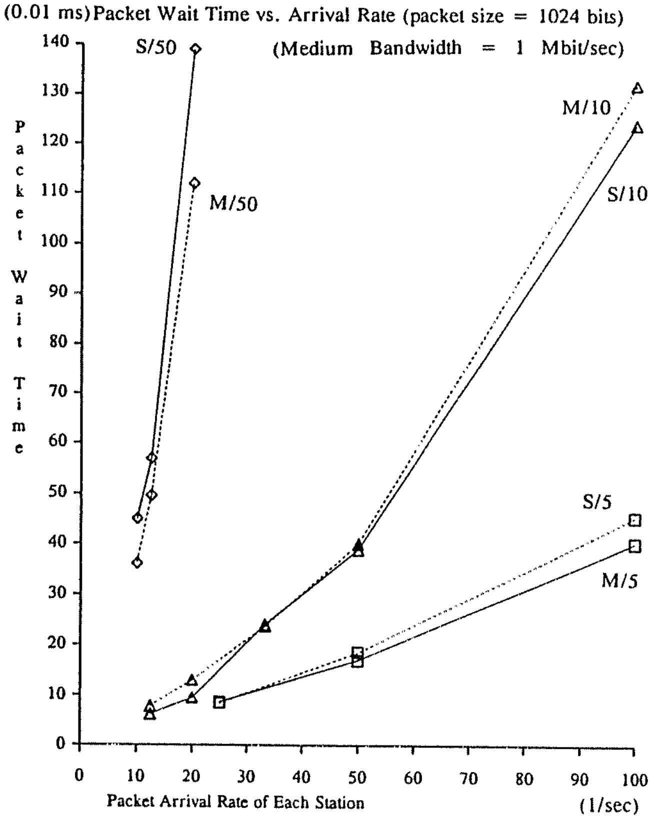


Figure 3

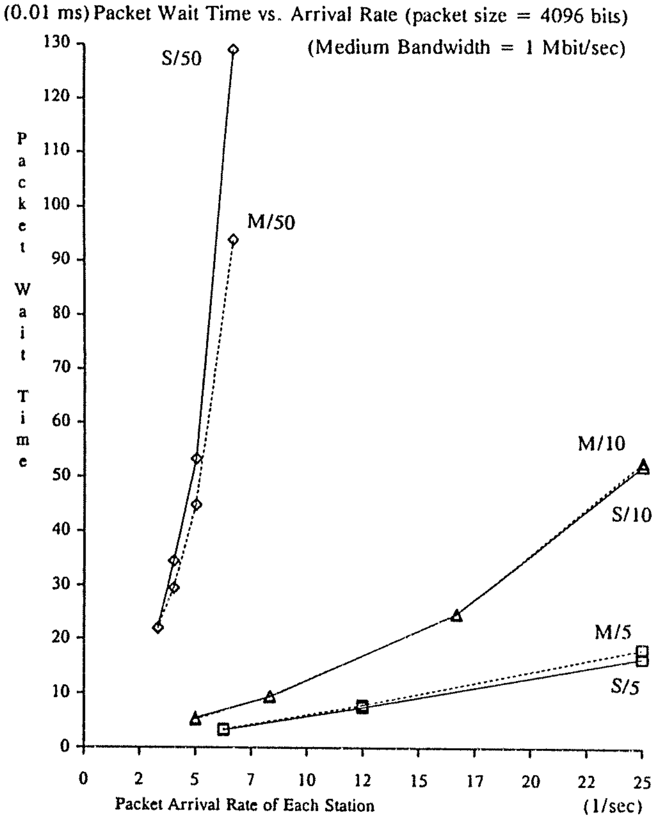


Figure 4

the token and packet circulation times depend on the number of stations, the station latency time, and the medium propagation delay of the ring, so in these two cases token and packet circulation time cannot be ignored. Due to these delays the effective number of multiple servers for these cases is smaller than the value given by (3-2), so the waiting time is longer than the value given by (3-3). When the assumption is met, WT_{simul} is confined by WT_{model} from above. Therefore WT_{model} can be used, as an upper bound, to estimate the delay characteristic of the system.

4. Performance Optimization Method

For an asymmetric traffic, in which a station is more likely to transmit packets to one station than to other stations, the location, or the configuration, of stations affects the performance in the following manner. The closer the corresponding communication stations are, the shorter their communication paths are, and thus the more concurrent paths can be supported. In order to quantitatively describe the closeness of communicating stations, define *communication locality* of the PaRing to be

$$1 / (\sum_{i=1}^{N} PA[i] \sum_{j=1}^{N} T[i,j] * PD(i, j))$$

where $PD(i, j)$ is the *length* of CP_i^j and equal to the number of arcs from i to j. The inverse of the communication locality,

$$\sum_{i=1}^{N} PA[i] \sum_{j=1}^{N} T[i,j] * PD(i, j),$$

is equal to the sum of the lengths of communication paths. If the arc set of the ring is changed, the new arc set may lead to a different communication locality. A larger communication locality implies shorter path lengths and hence more concurrent transmissions.

For given workload $PA[i]$ and $T[i, j]$, call the arc set E and station configuration that correspond to the maximal communication locality the *optimal arc set* $E_{optimal}$ and the *optimal station configuration,* respectively. Similarly, the *worst station configuration* corresponds to the minimal communication locality.

The objective of the optimization method is to find the optimal arc set and node configuration for a given ring. This problem can be expressed mathematically as follows.

When

$PA[i]$ and $T[i, j]$, for $1 \leq i, j \leq N$,

of a ring are given,

$$Min(\sum_{i=1}^{N} PA[i] \sum_{j=1}^{N} T[i,j] * PD(i, j)) \qquad (4\text{-}1)$$

subject to

$$\{PD(i, i + 1), PD(i, i + 2), \ldots, PD(i, i + N - 1) \} = \{ 1, 2, \ldots, N \}$$

where variables are the path length set

$$\{ PD(i,j) \mid i,j \in V \}$$

This is an integer programming problem [7].

This optimization method can be implemented as follows. Workload parameters such as $PA[i]$ and $T[i, j]$ are collected via monitors installed at stations and, then, $E_{optimal}$ is computed according to (4-1). Like other token ring architectures such as the PRONET [8] and the IBM Zurich Ring [9], the PaRing is a star-shaped ring, which contains a number of wire centers. A wire center consists of several relays and each station of the ring is connected to a relay of a wire center. In addition to maintenance, the original reason for installing wire centers, the wire centers also can be used to implement the PaRing optimization method. The main idea of the implementation is based on the fact that there is one-to-one mapping between stations and relays of wire centers and changing the mapping can alter the configuration of stations. Thus after obtaining the $E_{optimal}$, the reconfiguration can be physically implemented by connecting stations to different relays than the previous ones if necessary.

An experiment that uses both the optimization method and simulation was carried out. The experiment involved an asymmetric traffic in which there is a station that only transmits packets to a particular station. In the experiment, packet arrival interval times at all stations are exponentially distributed. The number of stations and the size of packets are fixed and equal to five and 1024 bits, respectively. The packet arrival rates at each station, $PA[i]$, are 100 and 200 packets per second, respectively. Performances of different configurations are characterized by the packet waiting time. The optimal and worst configurations computed according to (4-1) and their corresponding packet waiting times collected from the simulation are in Table 2. The experiments show the advantage of the optimization method.

5. Conclusion

This paper presents the architecture of the PaRing and results of performance modeling and optimization studies. The main advantages of the PaRing are the following.

(1) Good throughput delay characteristic resulting from concurrent multiple transmissions, short interface latency time, and the optimization method, which gives the maximum number of concurrent multiple transmissions and the best performance characteristics of the PaRing.

(2) Access fairness and adaptability to different applications obtained through the medium access priority assignment method.

(3) Automatic acknowledgment feature.

The analytical model developed in this paper allows accurate determination of packet waiting time for symmetric traffic with the assumption that token and packet circulation time is negligible compared with packet transmission time. The key to the modeling is to decompose a multiple server system into a set of independent and identical queueing single server systems.

REFERENCES

[1] Yaoshuang Qu, Lawrence H. Landweber, Miron Livny, "PaRing: A Token Ring Local Area Network with Concurrency", will be presented at the 10th Conference on Local Computer Networks, Oct. 7-9, Minneapolis, MN.

[2] Reames, Cecil C. and Liu, Ming T. "A Loop Network for Simultaneous Transmission of Variable-Length Message", Proc. 2nd Annual Symp., Comp. Arch., Jan. 1975, pp.124-129.

[3] Liu, M. T. and Rouse, D. M., "A Study of Ring Networks", Proc. of the IFIP WG6.4/Univ of Kent Workshop on Ring Technology Local Area Networks, Kent, U. K., Sep. 1983, pp. 1-39.

[4] Bux, Werner and Schlatter, Marcel, "An Approximated Method for the Performance Analysis of Buffer Insertion Rings", 1983, IEEE Trans. on Communications, Vol. Com-31, No. 1, Jan, pp. 50-55.

[5] Kobayashi, Hisashi, "Modeling and Analysis An Introduction to System Performance Evaluation Methodology", Addison-Wesley, 1978.

[6] Melman, Myron and Livny, Miron, "The DISS Methodology of Distributed System Simulation", SIMULATION, April, 1984, pp. 163-176.

[7] Qu, Yaoshuang and Shiau, T-H. "Performance Optimization of Ring Local Area Network Architectures via Minimization Problem by Integer Programming", (underway).

[8] Proteon Associates, Inc., "Operation and Maintenance Manual for the PRONET Local Network Interface", Issue No. 5, April 30, 1982.

[9] Bux, W., Close, F. H., Kuemmerle, K., Keller, H. J., and Mueller, H. R., "Architecture and Design of a Reliable Token-Ring Network", IEEE on Selected Areas in Communications, Vol. SAC-1, No.5, Nov. 1983, pp. 757-765.

Station Number	Packet Size (bits)	Total Arrival Rate (packets/sec)	Waiting Time (ms)		($WT_{simul} - WT_{model}$)
			WT_{simul}	WT_{model}	WT_{model}
05	0128	1250	0.019	0.014	0.35
05	0128	2500	0.036	0.030	0.20
05	0128	5000	0.114	0.080	0.42
05	1024	0125	0.086	0.085	0.01
05	1024	0250	0.170	0.185	-0.08
05	1024	0500	0.403	0.454	-0.11
05	4096	0031	0.032	0.034	-0.06
05	4096	0062	0.790	0.740	0.07
05	4096	0125	1.650	1.820	0.09
05	4096	0250	7.560	6.530	0.16
10	0128	1250	0.016	0.012	0.33
10	0128	2000	0.022	0.021	0.05
10	0128	3330	0.050	0.039	0.28
10	0128	5000	0.090	0.070	0.29
10	1024	0125	0.062	0.078	-0.20
10	1024	0200	0.095	0.130	-0.27
10	1024	0333	0.240	0.237	0.01
10	1024	0500	0.390	0.401	-0.03
10	4096	0050	0.540	0.520	0.04
10	4096	0083	0.940	0.947	-0.01
10	4096	0167	2.470	2.462	0.00
10	4096	0250	5.210	5.281	-0.01
50	0128	5000	0.120	0.062	0.93
50	0128	6250	0.160	0.088	0.82
50	0128	10000	0.540	0.241	1.20
50	1024	0500	0.450	0.362	0.24
50	1024	0625	0.570	0.496	0.15
50	1024	1000	1.390	1.119	0.02
50	4096	0167	2.200	2.188	0.01
50	4096	0200	3.450	2.939	0.17
50	4096	0250	5.330	4.477	0.19
50	4096	0334	12.90	9.392	0.37

Table 1: Packet Waiting Time

Experiment	a	b
Orgn Cfg Waiting Time (ms)	1->2->3->4->5->1 0.43	1->2->3->4->5->1 2.11
Optimal Cfg Waiting Time	1->5->4->3->2->1 0.37	1->5->4->2->3->1 1.21
Worst Cfg Waiting Time	1->2->3->4->5->1 0.43	1->3->2->4->5->1 2.20

Table 2: Station Configuration and Mean Packet Waiting Time