RING SYSTEMS FOR DIGITAL COMMUNICATIONS

NEW CONCEPTS AND RESULTS

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH
for the degree of
Doctor of Technical Sciences

presented by
Marcel Schlatter
dipl. El. Ing. ETH
born July 12, 1949
citizen of Schaffhausen and Büttenthal

accepted on the recommendation of
Prof. Dr. G. S. Moschytz, referee
Prof. Dr. J. L. Massey, co-referee

1981
Meinen Eltern
Preface

I am indebted to my supervisor, Professor Dr. G. S. Moschytz, for providing everything that I needed to perform interesting research work at the institute of telecommunications. The countless interesting and wide-ranging discussions with him and his constant encouragement were very valuable and stimulating.

I am also grateful to my co-supervisor Professor Dr. J. L. Massey who devoted much of his time to go with me into the arduous details of the various analyses. His advice and constructive criticism helped me to turn my ideas into rigorous arguments and clear concepts. Moreover, Prof. Massey’s expertise was an invaluable help in formulating my arguments and concepts concisely.

My thanks also go to Dr. Werner Bux of the IBM Research Laboratory, Ruschlikon, with whom I had a number of interesting discussions on the analytical buffer insertion ring model.

The support by Hasler Ltd., Berne, Switzerland, of the research project reported in this thesis is gratefully acknowledged.
ABSTRACT

The main issue of this thesis is the analysis of the capacity and the delay-throughput characteristics of various ring systems for digital communications. Capacity is discussed for general ring communication systems consisting of single rings or multiple interconnected rings. Mathematical analyses are derived and simulation results are presented for the delay in a buffer insertion ring. A new type of ring communication system, Etherring, is proposed and analytical analyses of its behavior are presented.

Analytical models have been developed to calculate, for various traffic distributions, the capacity of general ring systems consisting either of a single ring or multiple interconnected rings. It is shown that connecting local rings either through a star network with a central switching node or through a global exchange ring is particularly useful when the stations can be segregated into local groups of stations which often communicate with stations from the same local group but only rarely with stations from the other groups.

An exact analysis is given for the average waiting time of packets in the individual stations of a data collection ring with buffer insertion for any traffic distribution. As a special case, it is then shown that, regardless of the priority rule with which packets are serviced at the individual ring stations, the average total delay in a data collection ring with symmetric traffic is the same as in a first-come first-served star network. It is also shown that the local priority rule is fairer than the more common ring priority rule for servicing packets at each node in the sense that the range of the average waiting times seen by the individual stations is smaller with local priority than with ring priority.

Approximate analyses are given for the average waiting time of packets in a general buffer insertion ring. Two cases are considered: 1) all packet arrival processes are Poisson, and 2) all packet arrival processes are periodic (with random phase shifts between the individual packet streams). Both analyses are compared with simulations, and it is shown that their accuracy is excellent over a wide range of parameter values. The simulations confirm that the priority rule with which packets are serviced at the individual ring stations has no measurable effect on the average waiting time, which is the assumption on which the approximate analysis is based.

A new type of ring transport system for local area communication systems is proposed. The system is called Etherring because it combines the features of Ethernet (which is a Carrier Sense Multiple Access (CSMA) bus with collision detection) with those of a unidirectional ring. Etherring has fewer packet collisions than Ethernet and has a maximum stable utilization of almost 1, even for short packets. An
additional desirable feature of Etherring is that it does not require special precautions to detect lost or duplicated tokens and to recover from such error conditions; in a token passing ring (TPR), these are non-trivial problems. Such fault conditions, typically caused by transient bit errors in the leading or trailing packet flags are shown to have the same effect in the Etherring as collisions and are thus dealt with automatically by the Etherring protocol. An analytical model is developed to show that, under the condition that the ratio of the number of bits per packet to the number of stations is not too small (approximately five or more), the average transmission delay in Etherring is less than (when the utilization is light) or approximately equal to the average transmission delay in a TPR. It is also shown that, for uniform traffic, Etherring has smaller average transmission delays than a buffer insertion ring (BIR) when the utilization is light to moderate but considerably longer average delays than the BIR when the utilization is heavy. If, however, the traffic is of the data collection type, Etherring has smaller average transmission delays than the BIR when the utilization is light to moderate and approximately the same average delays as the BIR when the utilization is heavy. Moreover, Etherring is shown to be fair in the sense that all stations see the same average transmission delays, whereas in the BIR some stations see shorter average transmission delays - at the expense of other stations which see longer average transmission delays.
KURZFASSUNG


Für verschiedene Verkehrsverteilungen werden analytische Modelle hergeleitet, um die Kapazität allgemeiner Ringsysteme, die entweder aus einem einzelnen Ring oder aus mehreren miteinander verbundenen Ringen bestehen können, zu berechnen. Es wird gezeigt, dass es besonders dann sinnvoll ist, lokale Ringe über ein sternförmiges Netzwerk oder über einen globalen "Exchange Ring" miteinander zu verbinden, wenn die Stationen so in lokale Gruppen unterteilt werden können, dass die Stationen oft mit anderen Stationen aus der gleichen lokalen Gruppe kommunizieren und nur selten mit Stationen aus anderen lokalen Gruppen.

Es wird eine exakte Analyse für die mittlere Paketwartezzeit in den einzelnen Stationen eines Data Collection Rings mit Buffer Insertion präsentiert; die Analyse ist gültig für beliebige Verkehrsverteilungen. Am Spezialfall mit symmetrischem Verkehr wird dann gezeigt, dass die mittlere totale Wartezeit in einem Data Collection Ring die gleiche ist wie in einem sternförmigen Netzwerk - unabhängig von der Prioritätsregel nach der die Pakete in den einzelnen Ringstationen behandelt werden. Außerdem wird gezeigt, dass die "local priority rule" fairer ist als die üblichere "ring priority rule".


Es wird ein neuartiges Ring-Transportsystem für lokale Kommunikationssysteme vorgeschlagen. Das System wird Ether-ring genannt, weil es die Eigenschaften von Ethernet (einem Carrier Sense Multiple Access (CSMA) Bus mit Kollisions-Detektion) mit denen eines unidirektionalen Rings verbindet.
Etherring hat weniger Kollisionen als Ethernet und ist stabil bis zu fast 100% Auslastung, selbst wenn die Pakete kurz sind. Gegenüber einem "Token Passing Ring" hat Etherring den Vorteil, dass keine besonderen Massnahmen getroffen werden müssen, um zu erkennen, wenn ein Kontroll-Zeichen (Token) verloren gegangen oder plötzlich mehrfach vorhanden ist, und um solche Fehler zu korrigieren. Es wird gezeigt, dass solche Fehler, die durch transiente Bit-Fehler entstehen können, den gleichen Effekt wie Paket-Kollisionen haben und damit vom Etherring-Protokoll automatisch korrigiert werden. Es wird ein analytisches Modell entwickelt, das zeigt, dass die Paketverzögerung im Etherring kleiner (wenn die Verkehrslast klein ist) oder etwa gleich ist wie im Token Passing Ring (TPR). Für uniformen Verkehr wird gezeigt, dass Etherring kleinere mittlere Wartezeiten als der Buffer Insertion Ring (BIR) hat, wenn die Last klein oder mittel ist, jedoch viel größere mittlere Wartezeiten als der BIR wenn die Last gross ist. Im Fall von Data Collection hat Etherring jedoch kleinere mittlere Wartezeiten als der BIR wenn die Last klein bis mittel ist und etwa gleiche mittlere Wartezeiten wie der BIR wenn die Last gross ist. Ausserdem wird gezeigt, dass Etherring in dem Sinn fair ist, dass die Pakete aller Stationen die gleiche mittlere Wartezeit erfahren, während im BIR die Pakete einzelner Stationen kürzere Wartezeiten haben - auf Kosten von Paketen anderer Stationen, die entsprechend längere Wartezeiten haben.
CONTENTS

1. INTRODUCTION
   1.1 Background
   1.2 Previous work
   1.3 Thesis outline
   1.4 References

2. CAPACITY OF INTERCONNECTED RING COMMUNICATION SYSTEMS WITH UNIQUE LOOP-FREE ROUTING
   2.1 Introduction
   2.2 The capacity region
   2.3 Single ring networks
   2.4 Capacity of interconnected ring systems with uniform traffic
   2.5 Capacity of interconnected ring systems with non-uniform traffic
   2.6 Conclusions
   2.7 References

3. AVERAGE WAITING TIME ANALYSIS OF A DATA COLLECTION RING WITH BUFFER INSERTION
   3.1 Introduction
   3.2 Buffer insertion
   3.3 Data collection ring
   3.4 Waiting time as a function of the station number
   3.5 Conclusions
   3.6 References

4. WAITING TIME ANALYSIS OF A BUFFER INSERTION RING
   4.1 Introduction
   4.2 Buffer insertion ring with Poisson arrivals
   4.3 Simulation results and previous analytical models
   4.4 Buffer insertion ring with periodic arrivals
   4.5 Conclusions
   4.6 References

5. ETHERRING - A RING WITH A CSMA PROTOCOL
   5.1 Introduction
   5.2 Operating principles
      5.2.1 Chaining mode
5.2.2 Collision resolution
5.2.3 Transmission errors

5.3 Performance analysis
5.3.1 Average time to resolve a collision
5.3.2 Average sojourns in Etherring states
5.3.3 State transition probabilities
5.3.4 Retransmission overhead
5.3.5 Average transmission delay

5.4 Conclusions
5.5 References

6. SUMMARY

APPENDIX: BUFFER INSERTION RING SIMULATION PROGRAM
1. Introduction
2. How to use the simulation program
3. Sample simulation run
4. Simulation program listing
1. INTRODUCTION

The main issue of this thesis is the analysis of the capacity and the delay-throughput characteristics of various ring systems for digital communications.

Capacity is discussed for general ring communication systems consisting of single rings or multiple connected rings.

Mathematical analyses are derived and simulation results are presented for the delay in a buffer insertion ring, a modified version of a ring system that was originally proposed by Hafner et al. in 1974 (HAF74).

A new type of ring communication system, Etherring, is proposed and analytical analyses of its behavior are presented.

1.1 Background

The ring communication systems considered in this thesis are constructed by systematically connecting the output of one terminal through a unidirectional link to the input of the next terminal and by connecting the output of the last terminal to the input of the first terminal.

Such ring systems can, in principle, be built starting with two terminals and allowed to grow gracefully. Single rings can be connected through exchange nodes or exchange networks (which may themselves be rings) to form more general and more efficient ring communication systems.

In ring communication systems, messages can be exchanged between any two terminals provided that

1) the intermediate terminals relay the messages without alteration,

2) there is some mechanism to resolve the conflicts which result when two or more terminals simultaneously request access to a single shared transmission link.

In the following it is assumed that all signals are transmitted in digital form. Terminals transmit their messages in packets, each of which carries the address of the recipient.
An important advantage of such a digital ring communication system is that it provides an efficient solution to the "multiple-access" problem, i.e., means whereby many senders of information can share a common communications resource. These solutions can be based on distributed control procedures which do not depend upon a distinguished terminal (some of these rings depend, however, upon a centralized terminal for initialization, timing, error recovery, etc.).

Transmission capacity can be dynamically allocated to the individual terminals only when they need it. When a terminal has no messages to send, it does not use any transmission capacity. This is not the case, for example, in a network with private, dedicated links between any two terminals or in a star network. If, for example, only two terminals are active in a star network, there is no way to make use of the links belonging to the idle terminals to increase the maximum throughput between the active terminals. In fact, all communication systems with fixed division and allocation of resources are inefficient when there is a large number of senders, each of which sends only short bursts of data at random time instants.

The price to be paid for the possibility of dynamic resource sharing in ring communication systems is an increase in the average transmission delay compared to communication systems with private, dedicated links between any two terminals. This is a consequence of the necessity to buffer some of the messages in the case of congestion, i.e., when conflicts occur between two or more messages which request exclusive use of a resource. Congestion increases with the utilization of the ring communication system, i.e., with the ratio between offered load and capacity.

The main contributions of this thesis are new results on the capacity and the delay-throughput characteristics of various ring communication systems. One of these systems, Ethernet, is based on new concepts.

Message delay is defined as the time that elapses between the generation of a message at one terminal until the message has been completely received by the destination terminal. This includes the access time, the waiting times due to congestion in the buffers along the transmission path, and the transmission time. When a message is split into shorter packets of fixed length, both the delay per packet and the delay per message are of interest. Congestion delay is defined as that
part of the message delay that is caused by the interference of messages from other terminals.

There is a saying that you do not hear from the average customer - you hear from the unsatisfied customer. A topic that deserves special interest is therefore the question of fairness of a particular communication system, i.e. the distribution of transmission delays. The messages from some terminals may experience unacceptably long delays although the average delay is below the specified maximum. For example, in a carelessly designed ring communication system two particularly busy terminals, each transmitting to the other with excessive data rate, may be able to dominate the ring to such an extent that all other terminals have to wait unacceptably long until they can transmit their messages.

Total domination by one or two terminals is, however, only an extreme case. If a ring communication system is used, for example, to build a computer data base enquiry system it is very likely that the enquiry terminals receive considerably more data than they transmit. The computer data base terminal is predominantly a data source, and all enquiry terminals are predominantly data sinks. Therefore, the link from the computer data base to the first downstream terminal carries large volumes of data and is heavily utilized whereas the links further downstream carry much smaller volumes of data and are thus less heavily utilized.

A similar example is a data collection system where all terminals send their data to a central data collecting station which returns only short acknowledgement messages. The results in this thesis show that only one of the analyzed ring systems (Etherring) is in such cases fair, in the sense that the messages from all terminals experience the same average congestion delay.

1.2 Previous work

One of the earliest ring networks was developed by Farmer and Newhall (FARM69) at Bell Laboratories. The prototype was a token passing ring, running at 3.1 Mbps. Token passing rings are particularly designed for aperiodic, irregular computer traffic without any common control. At all times, only one station is given the right to send. After sending a message, a terminal generates an end-of-message token, thus passing control (i.e. the right to send) to the next terminal. Token passing
rings depend upon a "supervisor" which provides a central clock and is responsible for regenerating the token if it is destroyed.

Some years later, also at Bell Laboratories, Pierce (PIER72) developed a "network for block switching of data", a slotted ring operating at 1.544 Mbps. In a slotted ring, slots are kept circulating around the ring. The slots are marked "full" or "empty". When a terminal wishes to transmit it waits for an empty slot, marks it "full" and begins to transmit. Packets are removed either at the destination or back at the sender, by marking the slot "empty".

More recently, a slotted ring has been implemented at the University of Cambridge, UK, the so-called Cambridge ring (HOP78). It is used to support a University resource sharing network. In order to avoid the problems of loop hogging that can occur when packets are deleted by the destination, in the Cambridge ring the destination only sets an acknowledgement bit and the packet is allowed to continue back to the sender, who deletes it. The sender is not allowed to transmit further packets until it receives back the earlier packet. This prevents loop hogging and provides an acknowledgement scheme. The disadvantage of this scheme is that (for symmetric traffic\(^*\)) half of the ring capacity is lost, assuming the acknowledgements are not really required for every packet.

Again some years later Hafner proposed the buffer insertion ring (HAF74). In some sense, this was a hybrid of the two previous approaches. In the initial form of this system, packets had to be of constant length. The system was proposed to support a voice communication system; one packet had 25 bits and contained a PCM voice sample.

In the buffer insertion ring, packets are prepared in a buffer in the terminal's ring interface. A new slot is then created dynamically by switching this buffer into the ring (hence the name buffer insertion ring), either immediately (when the terminal is not currently relaying another packet) or when the relaying of the current packet is over. Thus, a new packet gets inserted in front of other packets which arrive at the terminal to be relayed. These packets are routed through the inserted buffer, which operates as a shift register. The inserted buffer is

\(^*\) Symmetric traffic is defined as the situation when all senders generate the same total amount of traffic.
removed from the ring when it contains data that does not need to be relayed any further; this is typically the case when the transmitted packet comes around again.

The same basic idea of the buffer insertion ring was later pursued at the Ohio State University by Liu, Reames, Babic and Pardo (BABI, LIU77, LIU77A, LIU77B, REA76). They called it DLCN, for "Distributed Computer Loop Network". They suggested a slightly modified buffer insertion mechanism that provides for variable length packets. Moreover, they suggested that packets should be taken off the ring at the destination, rather than at the sender after a complete round trip. However, they maintained the original concept of inserting newly arriving packets in front of the packets being relayed. They argued that this "virtually immediate access" was responsible for, in the case of symmetric traffic, their version of the buffer insertion ring having smaller average transmission delays than the token passing and the slotted ring.

The same claim is adopted by Penney and Baghdad in PEN80, an interesting survey of computer communications loop networks. Penney and Baghdad write (p. 236):

"The delay buffer insertion technique in DLCN leads to smaller delays, especially with increasing load. This appears to be due to the immediate access to the loop by new messages causing reduction in queueing time and latency which more than outweighs the extra transmission time due to longer delays in transmit at each node".

The results in this thesis show, however, that the shorter average delay is a result of the fact that with the buffer insertion ring, packets can be removed from the ring at the destination, rather than at the sender after a complete round trip, as in a token passing ring. This results, in the case of symmetric traffic, in a reduced link utilization and, hence, reduced congestion and reduced delay. If the traffic characteristics are such that most packets make almost complete round trips before they reach their destinations, the buffer insertion ring has approximately the same average delay as the token passing ring.

Moreover, our analyses show that the average congestion delay is, under very general conditions, independent of the priority rule employed; it does not depend on the "virtually immediate" access with Hafner's and Liu's rule of giving newly arriving packets priority over those being relayed.
1.3 Thesis outline

In chapter 2, capacity is defined for a given distribution of offered traffic as the maximum rate with which packets can be sent with finite delay through the network by appropriate routing. It is shown how capacity depends on the traffic characteristics and on the topology of ring communication systems interconnected so as to provide unique loop-free routing.

First, the traffic conditions are given under which the capacity of a single ring attains its maximum and its minimum. For the case of uniform traffic\(^(*)\) it is shown that the capacity is equal to twice the minimum capacity.

Then, it is shown that, for uniform traffic, the capacity relative to a single ring communication system can be increased by as much as 33.3 or 80 per cent when the stations are split up into two or three separate rings, respectively, interconnected to give unique loop-free routing\((\$)\). Exact formulas are given for the capacity of systems with an arbitrary number of stations split up into an arbitrary number of separate rings interconnected to give unique loop-free routing.

Finally, it is shown that connecting local rings either through a star network with a central switching node or through a global exchange ring is particularly useful when stations can be segregated into local groups of stations which often communicate with stations from the same local group but only rarely with stations from the other groups. Exact formulas are given to calculate the capacity of such interconnected ring communication systems, as well as the traffic handling requirements of the central switching node or the global exchange ring.

\(^(*)\) Uniform traffic is defined as the situation when all senders generate the same total amount of traffic, and when each sender sends with the same rate to each of the receivers at the other ring stations.

\((\$)\) In a network with unique loop-free routing each node has only one possible outgoing link through which the incoming traffic can be routed such that no closed-loop is created in the resulting path.
In chapter 3, an exact analysis is given for the average waiting time of packets in the individual stations of a data collection ring with buffer insertion for any traffic load. As a special case, it is then shown that, regardless of the priority rule with which packets are serviced at the individual ring stations, the average total delay in a data collection ring with symmetric traffic is the same as in a first-come first-served star network. It is also shown that the local priority (LP) rule is fairer than the more common ring priority (RP) rule for servicing packets at each node in the sense that the range of the average waiting times seen by the individual stations is smaller with LP than with RP.

In chapter 4, approximate analyses are given for the average waiting time of packets in a general buffer insertion ring. Two cases are considered: 1) all packet arrival processes are Poisson, and 2) all packet arrival processes are periodic (with random phase shifts between the individual packet streams). Both analyses are compared with simulations, and it is shown that their accuracy is excellent over a wide range of parameter values. Only for very high utilizations are the waiting time estimates for the Poisson arrival case somewhat larger than the average waiting times observed in the simulation; this discrepancy with the approximate analysis is explained with an intuitive argument. The simulations confirm that the priority rule with which packets are serviced at the individual ring stations has no measurable effect on the average waiting time, which is the assumption on which the approximate analysis is based.

In chapter 5, a new type of ring transport system for local area communication systems is proposed. The system is called Etherring because it combines the features of Ethernet (which is a Carrier Sense Multiple Access (CSMA) bus with collision detection) with those of a unidirectional ring.

For low utilization, Etherring behaves much the same as a CSMA bus; but for high utilization it behaves like a token passing ring (TPR). This latter fact is shown to be a consequence of the fact that, with increasing utilization, Etherring spends more time in a collision-free "packet-chaining mode" and correspondingly less time in a "collision mode". As a consequence, Etherring has fewer packet collisions than Ethernet and has a maximum stable utilization of almost 1, even for short packets.
An additional desirable feature of Etherning is that it does not require special precautions to detect lost or duplicated tokens and to recover from such error conditions; in a token passing ring, these are non-trivial problems. Such fault conditions, typically caused by transient bit errors in the leading or trailing packet flags are shown to have the same effect in Etherning as collisions and are thus dealt with automatically by the Etherning protocol.

An analytical model is developed to show that, under the condition that the ratio of the number of bits per packet to the number of stations is not too small (approximately five or more), Etherning spends only a very small fraction of time to resolve collisions and has, therefore, a small retransmission overhead. Moreover, under the same condition, the average transmission delay in Etherning is less than (when the utilization is light) or approximately equal to the average transmission delay in a TPR. It is also shown that, for uniform traffic, Etherning has smaller average transmission delays than a buffer insertion ring (BIR) when the utilization is light to moderate but considerably longer average delays than the BIR when the utilization is heavy. If, however, the traffic is of the data collection type, Etherning has smaller average transmission delays than the BIR when the utilization is light to moderate and approximately the same average delays as the BIR when the utilization is heavy.

Moreover, Etherning is shown to be fair in the sense that all stations see the same average transmission delay, whereas in the BIR some stations see shorter average transmission delays - at the expense of other stations which see longer average transmission delays.

In the appendix, a program to simulate the operation of buffer insertion rings under various conditions is described. The program consists of an interactive initialization and setup phase where the user is prompted to specify the simulation parameters, the performance quantities to be measured during the simulation, and the output and debugging options. The simulation can then be run either off-line as a batch job or interactively from a time sharing terminal. In the latter case, the simulation can be interrupted at arbitrary time instants to perform intermediate or preliminary statistical analyses of the collected measurement data or to change some simulation parameters if this should be desired. The simulation can then either be resumed or brought to an orderly end immediately.
1.4 References


2. CAPACITY OF INTERCONNECTED RING COMMUNICATION SYSTEMS
WITH UNIQUE LOOP-FREE ROUTING

2.1 Introduction

The "capacity" of a network to carry traffic depends on the offered traffic distribution, i.e., on what fraction of the total traffic originates at node i and is destined for node j for all i,j. For a given offered traffic distribution, capacity is the upper limit of the total offered traffic rate such that there exist protocols that deliver this traffic with finite average delay. If the total offered traffic exceeds capacity, then the average packet delay must be infinite either because a non-zero fraction of the packets are lost by the system (because, say, of overflow of finite length buffers) or because the network buffers (when assumed to have infinite length) hold a store of packets that almost surely increases without bound as further traffic arrives.

The capacity of general networks has been discussed, for example, in ELI56, FOR56, FOR62, KLEI64, FRA71, or CANT74, but the interesting class of networks with unique loop-free routing has apparently been overlooked by previous researchers. In section 2.2, we find the capacity region of this class of networks and show how capacity depends on the offered traffic vector, a quantity that describes the offered traffic characteristics.

In section 2.3, we discuss the capacity of the simplest network with unique loop-free routing, namely a single ring with unidirectional links.

In sections 2.4 and 2.5 we discuss the capacity of connected ring systems with exchange nodes for the uniform and certain non-uniform traffic distributions.

2.2 The capacity region

We consider communication systems with m stations (shown as nodes in Fig. 2.2.1) interconnected by identical unidirectional links to give unique loop-free routing; we assume that each link is an error-free delayless binary channel and hence has a capacity of 1 (bit per bit-time). The networks shown in Fig. 2.2.1 are typical examples of such communication systems. The exchange nodes (shown as squares) route the traffic from all incoming links to the appropriate outgoing links; they
neither generate nor absorb any traffic. To illustrate what we mean when we say that the networks in Fig. 2.2.1 have unique loop-free routing we show, in Fig. 2.2.2, two examples of networks with non-unique routing. In these networks, as illustrated by the dashed lines, there is sometimes more than one possible outgoing link through which an exchange node can route the incoming traffic without creating a closed-loop within the resulting path.
Fig. 2.2.2 Examples of communication networks with non-unique routing.

We assume that each station comprises a sender $S$ and a receiver $R$. Under the assumption here and hereafter that the senders do not transmit to receivers at the same station, there are

$$n = m(m-1)$$  \hspace{1cm} (2.2.1)

sender-receiver pairs $(S_i, R_j)$ with $i \neq j$ to be considered.

We define the $n$-dimensional offered traffic vector

$$f = R\mathbf{0}$$  \hspace{1cm} (2.2.2)

where $R$ is the total offered traffic rate (defined below) and where
\[ \Theta = \{ \Theta_1, 2, \ldots, \Theta_1, m, \Theta_2, 3, \ldots, \Theta_2, l, \ldots, \Theta_m, l, \ldots, \Theta_m, m-1 \} \] (2.2.3)

is the traffic distribution vector. The component \( \Theta_{i,j} \) is the fraction of the total offered traffic that originates at station \( i \) and is destined for station \( j \). The quantity

\[ f_{i,j} = R \Theta_{i,j} \] (2.2.4)

is the offered traffic (in bits per bit-time) from sender \( S_i \) to receiver \( R_j \). Obviously, for all sender-receiver pairs \( (S_i, R_j) \),

\[ f_{i,j} \geq 0 \] (2.2.5)

Any vector \( f \) that satisfies (2.2.5) is a possible offered traffic vector. The total offered traffic from sender \( S_i \) is thus

\[ r_i = \sum_{j=i+1}^{i+m-2} f_{i,j} \] (2.2.6)

where here and hereafter all indices are understood to be reduced by \( m \) when they are formally greater than \( m \). The quantity

\[ R = \sum_{i=1}^{m} r_i = \sum_{i=1}^{m} \sum_{j=i+1}^{i+m-2} f_{i,j} \] (2.2.7)

is thus the total offered traffic rate.

Because of the unique loop-free routing, the traffic that must flow through link \( k \), the outgoing link at station \( k \), if all packets are to reach their destination, is given by

\[ F_k = \sum_{(i,j) \in S_k} f_{i,j} \] (2.2.8)

where \( S_k \) is the set of all origin-destination pairs \( (i,j) \) such that the unique loop-free path from station \( i \) to station \( j \) traverses link \( k \). What makes unique routing networks amenable to simple analysis is that the
offered traffic uniquely determines the link flows, as given by (2.2.8), for all protocols that sensibly produce loop-free routes and that almost surely deliver all packets to their destinations.

Any vector \( f \) that satisfies (2.2.5) and

\[
F_k < 1, \quad 1 \leq k \leq m, \tag{2.2.9}
\]

is said to be a feasible offered traffic vector. (In general networks, an offered traffic vector is said to be feasible (see, e.g., CANT74) if there exists a routing for the flows such that the total flow in each link for this routing does not exceed the capacity of that link; our usage of "feasible" is consistent with this general usage).

We now define the region \( \mathcal{C} \) to be the set of all feasible offered traffic vectors, i.e.,

\[
\mathcal{C} = \{ f : f_{ij} > 0 \text{ all } i \neq j \text{ and } F_k \leq 1 \text{ all } k \} \tag{2.2.10}
\]

The following theorem establishes that \( \mathcal{C} \) is the capacity region.

Coding Theorem for Networks of Noiseless Channels with Unique Loop-free Routing

If the flow vector \( f \) lies within the outer boundary of the capacity region \( \mathcal{C} \) (i.e., if \( F_k < 1 \) for all links \( k \)), then there exist protocols that can serve all packets with finite average delay when the packet arrival processes are independent and memoryless and the second moment of packet length is finite.

Conversely, if the flow vector \( f \) lies beyond the outer boundary of the capacity region (i.e., if \( F_k > 1 \) for at least one link \( k \)) then no protocol can serve all packets with finite average delay regardless of the nature of the packet arrival process or the packet length distribution.

Proof

Let the delay for a packet traversing some link be defined as the time difference between its arrival at the station on the input end of the link and its subsequent arrival at the station on the output end of
the link. Because the network has a finite number of links, the average packet delay will be infinite if and only if there is some link such that the average delay for packets traversing this link is infinite.

When the offered traffic is such that the total offered flow in each link is below the capacity of that link, the following obvious protocol will serve all packets with finite average delay. As each packet arrives at a station, it is inserted into a buffer on the outgoing link that forms part of the loop-free route for that packet. The packets in these buffers are transmitted in first-come first-served fashion over the corresponding output links. Suppose, contrary to the theorem, that the average delay on some link is infinite. This means that the number of packets stored in the corresponding buffer almost surely grows without bound. But the total rate of packets destined to traverse that link is less than capacity and the fact that other links must also serve some of these packets cannot increase their rate of arrival into the buffer. Thus, packets are arriving into the buffer at a rate less than the capacity of the server so that the average number of packets in the buffer will be finite in contradistinction to our supposition provided that the second moment of packet length is finite. This proves the direct part of the theorem. (Note: this argument applies for virtually all arrival processes - the "independent and memoryless" arrival assumption of the theorem is a sufficient but not necessary condition).

If the total offered flow in at least one link k is above the capacity of that link then, for any protocol, the average delay for packets traversing that link must be infinite since their arrival rate exceeds the capacity of the server. Hence, the average packet delay must also be infinite.

Note: Those *f* that lie on the outer boundary of *C* (i.e., those *f* with \( F_k = 1 \) for some links k) can be served with finite average packet delay only in special cases such as periodic arrivals of packets of constant length. When packet arrivals are memoryless and independent, however, the average delay for packets traversing the link with \( F_k = 1 \) will be infinite regardless of the serving protocol.

We now introduce the notation

\[
C(\emptyset) = \max\{R: f \in C \text{ and } f=\emptyset\} \tag{2.2.11}
\]
to denote the capacity as a function of the traffic distribution vector $\varphi$. In the case of networks with unique loop-free routing, $C(\varphi)$ is equal to that value of $R$ which solves the equation

$$F_{k^*} = 1$$ \hspace{1cm} (2.2.12)

where $k^*$ is a value of $k$ that maximizes $F_k$. We shall call (2.2.12) the most-congested-link equation.

2.3 Single ring networks

Here we consider the case of single ring networks, an example of which (with $m=3$ stations) is shown in Fig. 2.3.1. In the following we shall assume that $\theta_{i,i+j}$ has the same value for all $i$. Because this implies that

1) the flows between $S_i$ and $R_{i+j}$, $1 \leq i \leq m$, $1 \leq j \leq m-1$, depend only on the distance $j$ between sender and receiver,

and, in particular, that

2) all senders $S_i$, $1 \leq i \leq m$, generate the same total amount of traffic,

we shall call such a traffic distribution symmetric.

To simplify notation in the symmetric case, we let

$$\gamma_j = m \theta_{i,i+j}$$ \hspace{1cm} (2.3.1)

denote the fraction of the total traffic destined $j$ links from its origin. We shall, with a slight abuse of our previous terminology, call

$$\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_{m-1}\}$$ \hspace{1cm} (2.3.2)

the traffic distribution vector in the symmetric traffic case. The traffic that must flow through link $k$ is in the symmetric case given by
Substitution of $h = i + j$ yields

$$F_k = R \sum_{j=1}^{m-1} \sum_{i=0}^{(m-1)-j} \delta_{k-1,k+j}$$

$$= R \sum_{j=1}^{m-1} \sum_{i=0}^{(m-1)-j} \frac{1}{m} \delta_{i+j}$$  \hspace{1cm} (2.3.3)$$

Substitution of $h = i + j$ yields

$$F_k = \frac{R}{m} \sum_{j=1}^{m-1} \sum_{h=j}^{m-1} \delta_h$$

$$= \frac{R}{m} \sum_{i=1}^{m-1} \delta_i$$  \hspace{1cm} (2.3.4)$$

which, not surprisingly, is independent of $k$. Thus, the most-congested-link equation (2.3.4) gives the 

**symmetric capacity** as
From (2.3.5), we see that symmetric capacity \( C(\vec{\gamma}) \) attains its maximum value, \( C_{\text{max}} \), when the offered traffic is such that each sender transmits only to its nearest neighbouring receiver, i.e., when

\[
\vec{\gamma}_1 = 1, \\
\vec{\gamma}_i = 0, \quad 2 \leq i \leq m-1.
\]

Substitution of (2.3.6) into (2.3.5) yields

\[
C_{\text{max}} = \max_{\vec{\gamma}} C(\vec{\gamma}) = m. \tag{2.3.7}
\]

From (2.3.5), we see further that the symmetric traffic achieves its minimum, \( C_{\text{min}} \), when each sender transmits only to its most distant receiver, i.e., when

\[
\vec{\gamma}_1 = 0, \quad 1 \leq i \leq m-2, \\
\vec{\gamma}_{m-1} = 1.
\]

Substitution of (2.3.8) into (2.3.5) yields

\[
C_{\text{min}} = \min_{\vec{\gamma}} C(\vec{\gamma}) = \frac{m}{m-1}. \tag{2.3.9}
\]

In this case, each link carries the flow from \( m-1 \) senders. For \( m \gg 1 \), \( C_{\text{min}} \) is approximately equal to 1. In other words, when all senders transmit only to their most distant destinations, ring capacity is approximately equal to the capacity of a single link between two stations. The maximum rate with which each station can send is approximately equal to \( 1/m \).

In real applications, the flows will in general be such that the symmetric ring capacity is between \( C_{\text{max}} \) and \( C_{\text{min}} \) as given by (2.3.7) and (2.3.9) respectively. Unfortunately, however, it will usually be much
closer to $C_{\text{min}}$ than to $C_{\text{max}}$. Consider, for example, the case of uniform traffic which we define as the situation when the traffic is symmetric and

$$\delta_1 = \delta_2 = \ldots = \delta_{m-1} = \frac{1}{m-1},$$

(2.3.10)

i.e., when each sender sends with the same rate to each of the $m-1$ receivers at the other stations. Substitution of (2.3.10) into (2.3.4) yields

$$F_k = \frac{R}{m(m-1)} \sum_{i=1}^{m-1} i$$

$$= \frac{R}{2},$$

(2.3.11)

independent of the link index $k$. Thus, the most-congested-link equation (2.3.4) shows the capacity, which we denote by $C_{\text{unif}}$, to be

$$C_{\text{unif}} = 2,$$

(2.3.12)

irrespective of the number of stations. This means that, when each station transmits with the same rate to each of the receivers at the

![Fig. 2.3.2 Ring communication system with terminal-to-host and host-to-terminal traffic.](image)
other stations, the ring as a whole can carry precisely twice as much traffic as a single link.

{ A minor modification of the above arguments shows that the maximum achievable rate (i.e., the supremum of the rates $R$ such that the protocol delivers the offered traffic $f=RQ$ with finite delay for a given $Q$) for protocols that allow packets to be removed from the network only by the sender is at most $1$ in a single ring network regardless of the traffic distribution vector $Q$. Thus such end-to-end protocols can operate at best only at 50 per cent of the capacity of a ring with uniform traffic. }

When one tries to analyze the behavior of ring communication systems, it is often convenient to assume that traffic is symmetric and uniform. Before one draws any conclusions from such an analysis one should, however, carefully test whether this assumption is valid. If it is not, the capacity may be as much as 50 per cent less than less than 2 or as much as $m/2$ times greater. The symmetric and uniform analysis can yield grossly incorrect performance estimates for a given offered load. Consider, for example, a ring communication system with terminal-to-host and host-to-terminal traffic as depicted in Fig. 2.3.2. There are $m-1$ terminals $1, 2, \ldots, m-1$ which transmit each with rate

$$f_{1,m} = f_{2,m} = \ldots = f_{m-1,m} = f_T$$  \hspace{1cm} (2.3.13)

to station $m$, the host computer. The host computer sends return messages at some rate $f_H$ to each of the terminals, i.e.,

$$f_{m,1} = f_{m,2} = \ldots = f_{m,m-1} = f_H$$  \hspace{1cm} (2.3.14)

All elements

$$f_{i,j} = 0$$  \hspace{1cm} (2.3.15)

for $i \neq m$ and $j \neq m$, i.e., the terminals do not communicate with each other (at least not directly).

Let $a$ denote the ratio of host to terminal traffic, i.e.,

$$f_H = af_T.$$  \hspace{1cm} (2.3.16)
For α≥1 (e.g., in an inquiry system where a short message from a terminal usually causes the host computer to send a large amount of data such as a full screen of text back to the terminal) the most congested link is that from the host computer to the first downstream station; the flow through this most-congested link is given by

$$F_1 = \frac{R}{m(1+\alpha)} \cdot m^\alpha = R \frac{\alpha}{1+\alpha}.$$  \hspace{1cm} (2.3.17)

Letting $C_\alpha$ denote the capacity when the traffic is distributed according to (2.3.13)-(2.3.16), we have from (2.3.17)

$$C_\alpha = 1 + \frac{1}{\alpha}, \hspace{0.5cm} \alpha \geq 1,$$  \hspace{1cm} (2.3.18)

which is equal to 2 (as in the case of uniform traffic) when $\alpha=1$ but decreases to $C_{\text{min}}=1$ as $\alpha$ increases.

For $\alpha<1$ (e.g., in a data collection system where all terminals send their data to the host which sends only short acknowledgement messages back to the terminals) the flow is maximum in the last link, just before the host computer, which carries the total traffic from all terminals to the host. The traffic that must flow through this most-congested link is given by

$$F_m = \frac{R}{m(1+\alpha)} = \frac{R}{1+\alpha}.$$  \hspace{1cm} (2.3.19)

Thus, the ring has in this case a capacity of

$$C_\alpha = 1+\alpha, \hspace{0.5cm} 0<\alpha<1.$$  \hspace{1cm} (2.3.20)

Again, ring capacity is less than 2; for $\alpha<1$, $C$ is reduced to almost $C_{\text{min}}=1$. Thus, for $\alpha<1$ or $\alpha>1$, the capacity of the ring is only about half its uniform capacity.
2.4 Capacity of interconnected ring systems with uniform traffic

In the previous section we saw that the capacity of a ring system depends strongly on the vector of flows. At one extreme, each link carries only the message stream from one station to its nearest downstream neighbour. At the other extreme, each link carries the message streams from m-1 stations, each transmitting only to its farthest neighbour.

When, in a given application, the ring capacity would not be sufficient for the desired offered load, one must either increase the capacity of the individual links - or choose a network with an alternative topology which reduces the flow through some links. In this section we consider a special case of the latter approach which is particularly useful to increase the capacity of an existing ring communication system when either more stations are to be connected or when, in the course of time, stations have more data to send than when the system was initially planned and built.

We first consider a system of two rings which are connected by an exchange node (Fig. 2.4.1). Of the total of m stations, \( m_1 \) are connected to ring 1 and

![Diagram of interconnected rings](image-url)

Fig. 2.4.1 Communication system consisting of three rings which are interconnected by an exchange node.
are connected to ring 2. We assume uniform traffic, i.e., that each station $i$ sends with the same rate

$$f_{i,j} = \frac{R}{m(m-1)}$$  \hspace{1cm} (2.4.2)

to all other stations $j \neq i$. To find capacity in this case, we note from inspection of Fig. 2.4.1 that the traffic which must flow through any link in ring 1 is the same as the traffic which would flow through that link if all stations had been connected to form one ring less the traffic that both originates in and is destined for stations in ring 2. Because the flow that a link would have to carry if all stations were on one ring is according to (2.3.11) given by $R/2$, the flow through each of the links of ring 1 is given by

$$F^{(1)}_k = \frac{R}{2} - \frac{R_{2,2}}{2}$$  \hspace{1cm} (2.4.3)

where

$$R_{2,2} = m_2(m_2-1)f_{i,j}$$  \hspace{1cm} (2.4.4)

is the total traffic generated by and destined for the stations in ring 2. Substitution of (2.4.2) and (2.4.4) into (2.4.3) yields

$$F^{(1)}_k = \frac{Rm_1}{m(m-1)} \left( \frac{m_1-1}{2} + m_2 \right)$$  \hspace{1cm} (2.4.5)

Similarly, the flow through each of the links in ring 2 is given by

$$F^{(2)}_k = \frac{R}{2} - \frac{R_{1,1}}{2}$$  \hspace{1cm} (2.4.6)

where

$$R_{1,1} = m_1(m_1-1)f_{i,j}$$  \hspace{1cm} (2.4.7)

is the total traffic generated by and destined for the stations in ring 1. Thus,
When

\[ m_1 > m_2 \]  \hspace{1cm} (2.4.9)

the links in ring 1 become completely utilized before the links in ring 2. Therefore, the links in ring 1 are the most-congested links and the capacity \( C_{\text{unif}}^{(2)} \) for this case is obtained by setting the flow (2.4.5) equal to 1 and solving for \( R \). This yields

\[ C_{\text{unif}}^{(2)} = \frac{2m(m-1)}{m(2m-1)} \]  \hspace{1cm} (2.4.10)

where we also include the case \( m_2 \leq m_1 \) by defining

\[ m^* = \max(m_1, m_2) \]  \hspace{1cm} (2.4.11)

The values of \( C_{\text{unif}}^{(2)} \) shown in table 2.4.1 for various values of \( m_1 \) and \( m_2 \) were calculated from (2.4.10). They show that the capacity gain over a single ring (which was in the previous section shown to have capacity 2) is largest when \( m_1 = m_2 = m/2 \) as can be deduced in general from (2.4.10). In this case (2.4.10) reduces to

\[ C_{\text{unif}}^{(2)} = \frac{8(m-1)}{3m-2} \]  \hspace{1cm} (2.4.12)

Table 2.4.1 shows that (2.4.12) quickly approaches

\[ \lim_{m \to \infty} C_{\text{unif}}^{(2)} = \frac{8}{3} \approx 2.667 \]  \hspace{1cm} (2.4.13)

as the number of stations increases. In other words, the best that can be achieved (in the case of uniform traffic) is a 33.3 per cent increase in total capacity when the number of stations in a given ring system is split up into two rings which are connected by an exchange node. If this increase is not sufficient, the stations must be split up further, e.g., into three separate rings. Such a splitting is shown in Fig. 2.4.2 where the exchange nodes are interconnected by a star network. For convenience
we consider only the case when \( m \), the total number of stations, is split up equally among the three rings, i.e., when each of the rings 1, 2, and 3 have

\[
m' = \frac{m}{3}
\]  

(2.4.14)

Table 2.4.1

<table>
<thead>
<tr>
<th>( m_1 )</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.400</td>
<td>2.143</td>
<td>2.074</td>
<td>2.045</td>
<td>2.031</td>
<td>2.009</td>
<td>2.002</td>
<td>2.001</td>
<td>2.001</td>
<td>2.000</td>
</tr>
<tr>
<td>4</td>
<td>2.143</td>
<td>2.545</td>
<td>2.308</td>
<td>2.200</td>
<td>2.141</td>
<td>2.044</td>
<td>2.013</td>
<td>2.006</td>
<td>2.003</td>
<td>2.002</td>
</tr>
<tr>
<td>6</td>
<td>2.074</td>
<td>2.308</td>
<td>2.588</td>
<td>2.395</td>
<td>2.286</td>
<td>2.097</td>
<td>2.029</td>
<td>2.014</td>
<td>2.008</td>
<td>2.005</td>
</tr>
<tr>
<td>8</td>
<td>2.045</td>
<td>2.200</td>
<td>2.395</td>
<td>2.609</td>
<td>2.448</td>
<td>2.160</td>
<td>2.051</td>
<td>2.025</td>
<td>2.015</td>
<td>2.010</td>
</tr>
<tr>
<td>10</td>
<td>2.031</td>
<td>2.141</td>
<td>2.286</td>
<td>2.448</td>
<td>2.621</td>
<td>2.231</td>
<td>2.076</td>
<td>2.038</td>
<td>2.023</td>
<td>2.015</td>
</tr>
<tr>
<td>20</td>
<td>2.009</td>
<td>2.044</td>
<td>2.097</td>
<td>2.160</td>
<td>2.231</td>
<td>2.644</td>
<td>2.241</td>
<td>2.128</td>
<td>2.080</td>
<td>2.055</td>
</tr>
<tr>
<td>40</td>
<td>2.002</td>
<td>2.013</td>
<td>2.029</td>
<td>2.051</td>
<td>2.076</td>
<td>2.241</td>
<td>2.655</td>
<td>2.374</td>
<td>2.245</td>
<td>2.174</td>
</tr>
<tr>
<td>60</td>
<td>2.001</td>
<td>2.006</td>
<td>2.014</td>
<td>2.025</td>
<td>2.038</td>
<td>2.128</td>
<td>2.374</td>
<td>2.659</td>
<td>2.445</td>
<td>2.323</td>
</tr>
<tr>
<td>80</td>
<td>2.001</td>
<td>2.003</td>
<td>2.008</td>
<td>2.015</td>
<td>2.023</td>
<td>2.080</td>
<td>2.245</td>
<td>2.445</td>
<td>2.661</td>
<td>2.488</td>
</tr>
<tr>
<td>100</td>
<td>2.000</td>
<td>2.002</td>
<td>2.005</td>
<td>2.010</td>
<td>2.015</td>
<td>2.055</td>
<td>2.174</td>
<td>2.323</td>
<td>2.488</td>
<td>2.662</td>
</tr>
</tbody>
</table>

we assume that the links between the exchange nodes and the central switching node in the star network have ample capacity to guarantee that they do not become bottlenecks. Since the links in ring 1 do not carry any of the traffic that has both its origin and its destination outside ring 1 (i.e., either in ring 2 or 3), the flows through each of the links in ring 1 is given by

\[
F_k^{(1)} = R - R_{23,23}^{23}
\]  

(2.4.15)

where

\[
R_{23,23} = (m-m')(m-m'-1)f_{i,j}
\]  

(2.4.16)
is the total traffic generated by and destined for the stations in ring 2 and 3. Substitution of (2.4.2), (2.4.14) and (2.4.16) into (2.4.15) yields

\[ F_k^{(1)} = R \frac{5m-3}{18(m-1)} \quad . \]  

(2.4.17)

By symmetry, it follows that the links in all rings carry the same flows, i.e.,

\[ F_k^{(1)} = F_k^{(2)} = F_k^{(3)} \quad . \]  

(2.4.18)

The capacity, \( C_k^{(3)} \), is obtained by setting the most-congested link flow
of (2.4.17) equal to 1 and solving for \( R \). This yields

\[
C_{\text{unif}}^{(3)} = \frac{18(m-1)}{5m-3}.
\]  

(2.4.19)

When \( m \) goes to infinity,

\[
\lim_{m \to \infty} C_{\text{unif}}^{(3)} = \frac{18}{5} = 3.6,
\]  

(2.4.20)

i.e., with three rings, we can increase capacity by 80 per cent, compared to a single ring.

The obvious generalization of the above argument shows that a system where \( m \) stations with uniform traffic are equally split up into \( L \) rings which are connected through a star network has capacity

\[
C_{\text{unif}}^{(L)} = \frac{2L^2(m-1)}{m(2L-1)-L}.
\]  

(2.4.21)

In the case where each of the rings has only one station connected to it, substitution of \( L = m \) into (2.4.21) shows that in this case \( C_{\text{unif}}^{(L)} = m \), which is correctly the capacity of a star network with \( m \) stations and when the number of stations \( m \) becomes large, we have, in the limit,

\[
\lim_{m \to \infty} C_{\text{unif}}^{(L)} = \frac{2L^2}{2L-1}
\]  

(2.4.22)

which is approximately equal to \( L \) for \( L \gg 1 \). Table 2.4.2, which lists values of \( C_{\text{unif}}^{(L)} \) obtained from (2.4.21) for various combinations of \( m \) and \( L \) shows that \( C_{\text{unif}} \) approaches \( L \) quickly, even for relatively small values of \( m \) and \( L \).
<table>
<thead>
<tr>
<th>m</th>
<th>L</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.00</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.40</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>6.00</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8.00</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.57</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.29</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10.00</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.66</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.53</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10.48</td>
</tr>
</tbody>
</table>

*Table 2.4.2*
2.5 Capacity of interconnected ring systems with non-uniform traffic

In the previous section we saw that ring communication systems with uniform traffic have a higher capacity when the stations are split up into a number of local rings which are connected through a star network. In this section we show that, in applications where the stations can be grouped into stations which often communicate with stations from the same group but only rarely with stations from other groups, the capacity advantage of such connected ring systems compared to a single ring is even greater.

For example, we consider again the ring system of Fig. 2.4.1 which has two rings connected through an exchange node. We assume that both rings have the same number

\[ m' = \frac{m}{2} \]  \hspace{1cm} (2.5.1)

of stations. Moreover, we assume that each station \( i \) transmits with rate \( f_{i,j} \) to all other stations \( j \neq i \) in the same local ring and with rate

\[ f_{i,j}^* = e f_{i,j} \]  \hspace{1cm} (2.5.2)

to all stations \( j \) in the other ring, where \( e \) is the exchange ratio. There are \( m'(m'-1) \) sender-receiver pairs in each local ring, and each of the \( m' \) stations in one local ring sends to \( m' \) stations in the other local ring. The total offered traffic is therefore given by

\[ R = 2m'(m'-1)f_{i,j} + 2(m')^2f_{i,j}^* \]

\[ = f_{i,j} \frac{m(m+em-2)}{2} . \]  \hspace{1cm} (2.5.3)

By the same argument as in section 2.4, the flow through all links is the same and is given by

\[ F_k = \frac{R}{Z} - \frac{R_{\text{local}}}{2} \]  \hspace{1cm} (2.5.4)

where

\[ R_{\text{local}} = \frac{m}{2} \left( \frac{m}{2} - 1 \right) f_{i,j} \]  \hspace{1cm} (2.5.5)
is the total external local traffic, i.e., the traffic that has both its origin and its destination in the other ring. Substitution of (2.5.3) and (2.5.5) into (2.5.4) yields

$$F_k = R \cdot \frac{m(l+2e)-2}{4(l+2e)-2}$$

for all $k$. The most-congested-link equation can now be solved to yield the system capacity

$$C_{\text{non-unif}}^{(2)} = 4 \cdot \frac{m(1+e)-2}{m(1+2e)-2}$$

It is interesting to note that for $e=0$ the system capacity is $C_{\text{non-unif}}^{(2)} = 4$, i.e., when there is no exchange traffic the system capacity is equal to the sum of the capacities of the individual rings since then there is no cross traffic between the rings.

For $e=1$, we have the special case of uniform traffic discussed in section 2.4 and (2.5.7) reduces to (2.4.12).

When $e$ increases further we have, in the limit, when all traffic in each subring is destined for the other subring,

$$\lim_{e \to \infty} C_{\text{non-unif}}^{(2)} = 2$$

i.e., the system capacity is equal to the capacity of a single ring with uniform traffic.

In the general case with $m$ stations split equally into $L$ rings where each ring has $m' = m/L$ stations which send with rate $f_{i,j}$ to all other stations in the same ring and with rate $f_{i,j}^e = ef_{i,j}$ to all stations in the other rings, the total offered traffic is given by

$$R = f_{i,j} \{ Lm'(m'-1) + eLm'(m-m') \} = f_{i,j} \frac{m}{L} \left\{ (m-L)+e(mL-m) \right\}$$

Since the links in a local ring do not carry the traffic between the rings,
sender-receiver pairs within the L-1 other local rings and also not the traffic between the

\[(L-1)m'(m-2m')\]  \hspace{1cm} (2.5.11)

sender-receiver pairs which communicate through the central switching node, the flow in the links of an arbitrary local ring is given by

\[F_k = \frac{R}{2} - \frac{L-1}{2} \{m'(m'-1)f_{i,j} + m'(m-2m')e_{i,j}\}
= R \frac{(m-L)+2em(L-1)}{2L((m-L)+e(mL-m))}.\]  \hspace{1cm} (2.5.12)

Solving the most-congested-link equation (2.5.12) yields

\[C(L)_{\text{non-unif}} = \frac{2L(m+e(L)-1)}{m(1+2e(L-1))-L},\]  \hspace{1cm} (2.5.13)

for the system capacity in this general case.

For e=0, i.e., when there is no exchange traffic, we have L separate local rings with uniform traffic. Because each of these local rings has a capacity of 2, the system capacity must be 2L (note that this result is consistent with (2.5.13) when e=0 is substituted). At the other extreme, when e is large we have, in the limit,

\[\lim_{e \to \infty} C(L)_{\text{non-unif}} = L.\]  \hspace{1cm} (2.5.14)

This is illustrated by Fig. 2.5.1 where

\[\lim_{m \to \infty} C(L)_{\text{non-unif}} = 2L \frac{1+e(L-1)}{1+2e(L-1)}\]  \hspace{1cm} (2.5.15)

is plotted vs. e for L=2, 3, 4, and 5 separate local rings.

As an alternative to the star network an exchange ring can be used to connect the separate local rings (see Fig. 2.5.2). We allow the links of the exchange ring to have capacities greater than 1 so that the exchange ring does not become the bottleneck of the network. The total offered traffic is again given by (2.5.9), and the flow in the links of the exchange ring is given by
Fig. 2.5.1  Capacity of a system with $\infty$ stations which are equally split into $L$ separate rings which are interconnected by a star network.
\[ F_E = \frac{R}{2} - L \left( \frac{R_{\text{local}}}{2} \right) \]  \hspace{1cm} (2.5.16)

where

\[ R_{\text{local}} = m'(m'-1)f_{i,j} \] \hspace{1cm} (2.5.17)

is the traffic that is generated by the stations of one local ring and destined to the other stations in the same ring. Substitution of (2.5.9) and (2.5.17) into (2.5.16) yields

\[ F_E = R \frac{e(mL-m)}{2((m-L)+e(mL-m))} \] \hspace{1cm} (2.5.18)

---

**Fig. 2.5.2** Communication system consisting of L local rings which are interconnected by a global exchange ring.
When the links in the exchange ring have a capacity of at least

\[ E_{\text{min}} = \left. E \right|_{R = C(L)}^{\text{non-unif}} = \frac{e L m(L-1)}{m(1+2e(L-1))-L} \tag{2.5.19} \]

where \( C_{\text{non-unif}} \) is the capacity of the connected ring system with a central switching node (eqn. 2.5.13), the system with the exchange ring has exactly the same capacity as the system with the central switching node, namely \( C_{\text{non-unif}} \) as given by (2.5.13). Note that

\[ \lim_{m \to \infty} E_{\text{min}} = \frac{e L (L-1)}{1+2e(L-1)} \approx \frac{L}{2} \tag{2.5.20} \]

for \( 2e(L-1) \gg 1 \). In other words, when the number of stations is large and when there is much exchange traffic between the local rings, the capacity of the exchange ring must, in the limit, be equal to \( L/2 \) times the capacity of the transmission links in the local rings. At the other extreme, when there is no exchange traffic, i.e., when \( e=0 \), \( E_{\text{min}}=0 \). This is illustrated by Fig. 2.5.3 where \( \lim E_{\text{min}} \) obtained from (2.5.20) is plotted vs. \( e \) for \( L=2, 3, 4, \) and 5 local rings.

Now we consider the case when the links in the exchange ring become the bottlenecks, i.e., when the capacity \( E \) of the exchange links is given by

\[ E < E_{\text{min}} \tag{2.5.21} \]

In this case, the system capacity is obtained from (2.5.18) (which gives now the most-congested-link flow) by setting this flow equal to \( E \) and solving for \( R \). This yields

\[ C_{\text{non-unif}} = 2E \frac{m(1+e(L-1))-L}{e m(L-1)} \tag{2.5.22} \]

When \( m \) increases we have, in the limit,

\[ \lim_{m \to \infty} C_{\text{non-unif}} = 2E \frac{1+e(L-1)}{e(L-1)} \tag{2.5.23} \]

The curves in Fig. 2.5.4 show the capacity of the system in Fig. 2.5.2 vs. the exchange ratio \( e \) for \( E=0.5 \) and for \( L=2, 3, 4, \) and 5 local rings. The curves were obtained by taking the minimum of (2.5.15) and (2.5.23).
Fig. 2.5.3 Minimum exchange ring capacity $E_{\text{min}}$ vs. the exchange ratio $e$. 
Fig. 2.5.4 Capacity of the system in Fig. 2.5.2 vs. the exchange ratio \( e \); the exchange links have capacity \( E = 0.5 \).
For small values of $e$, the local rings are the bottlenecks and the capacity is given by (2.5.15). When, however, the exchange ratio increases beyond a critical point (which depends on $L$), the exchange ring becomes the bottleneck and the system capacity is given by (2.5.23).

Fig. 2.5.5 is a plot of the system capacity $C^{\text{non-unif}}(L)$ vs. the exchange ring capacity $E$ for $L=2$ and 4 local rings and for exchange ratios $e=0.1$ (solid curves) and $e=1.0$ (dashed curves). This figure illustrates that it does not pay to increase $E$ beyond some value which depends both on $L$ and $e$. When $E$ is increased beyond this value, the system capacity remains constant because now the local rings are the bottlenecks.

![Fig. 2.5.5](image)

*Fig. 2.5.5* Capacity of the system in Fig. 2.5.2 vs. the capacity $E$ of the exchange links.
2.6 Conclusions

We have shown how capacity, the maximum rate with which packets can be sent through a ring communication system, depends on how the senders and receivers are physically located relative to each other. For the case of uniform traffic, i.e., when each sender sends with the same rate to all receivers at the other stations, the capacity of a single ring is equal to twice the capacity of a transmission link between two stations.

A way to increase the capacity of a single ring without increasing the capacity of the individual transmission links is to split the system up into separate rings. For uniform traffic, the capacity of a ring communication system with two separate rings connected through an exchange node is approximately 2.6 times the capacity of a single transmission link. With three rings connected through a star network with a central switching node, the capacity is approximately 3.6 times the capacity of a single transmission link (exact formulas have been given for the general case with an arbitrary number of stations and arbitrary numbers of individual rings).

Ring communication systems with separate local rings which are connected either through a star network with a central switching node or a global exchange ring are particularly well suited to applications where stations can be segregated into local groups of stations which often communicate with stations from the same local group but only rarely with stations from the other groups. When there is much exchange traffic between the individual rings, system capacity is approximately equal to half the sum of the capacities of the individual rings. When, however, there is little exchange traffic, system capacity approaches the sum of the capacities of the individual rings.
2.7 References


3. AVERAGE WAITING TIME ANALYSIS OF A DATA COLLECTION RING WITH BUFFER INSERTION

3.1 Introduction

Ring communication systems have been proposed for a variety of applications; for a recent survey see PEN79. Previous discussions of rings with the decentralized buffer insertion access mechanism in HAF74, REA75, LEE79, and BRA80 concentrated on the peculiarities of specific priority rules, packet removal techniques, reliability, implementation problems, etc. Analytical models of buffer insertion rings have been discussed by Liu (LIU77, LIU77A), by Kanakia (KAN79) and by Schultze (SCHU79). The models of Liu and Kanakia are essentially identical (and inaccurate for rings with more than approx. 10 stations, as will be shown in chapter 4), and Schultze's model is only an approximation and is restricted to constant length packets and to one particular priority rule.

In this chapter we describe (in section 3.2) a very general buffer insertion mechanism that is suitable for arbitrary priority rules and packet removal techniques. In section 3.3 we give an exact analysis of the average waiting time in a buffer insertion ring that is used for data collection, i.e., a system where stations 1,2,...,m transmit through a single unidirectional link only to station m+1, which operates only as a data collection station, and transmits to no one (Fig. 3.1.1). The main results of that section are

1) a theorem on the average waiting time of packets in station i, 1 ≤ i ≤ m, which shows that this average wait is the same for all "work-conserving" queueing disciplines, and

2) a corollary on the average total delay in the ring.

In section 3.4 we show, again by an exact analysis, how the delay of a packet depends on the position of the sending station relative to the data collecting station. It will be seen that the priority rule used to resolve conflicts between packets that are already on the ring and packets that attempt to access the ring affects the range of average
delays seen by the individual stations. Unfortunately, there is however no (at least no simple) way to guarantee fairness in the sense that the average delay is the same for all stations, but it is shown that giving local packets priority is fairer than the more commonly proposed ring priority rule.

Fig. 3.1.1 Data collection ring.
3.2 Buffer insertion

To describe the method of decentralized access control in a buffer insertion ring it is convenient to focus on a single station $i$ as depicted in Fig. 3.2.1. Each station has two buffers, one (the ring buffer) to store packets coming from the ring and the other (the access buffer) to store local packets until their transmission. These buffers are assumed to be of infinite length. We assume that initially both the ring input and output in Fig. 3.2.1 are idle, that switches A, B, and C are all in position 1, and that the access buffer (AB) and the ring buffer (RB) at this station are empty. Consider first the case when no local traffic arrives at this station to interfere with the incoming ring traffic. Whenever a packet arrives at the ring input, the watchdog (WD) compares the receiver address field (RAF) of the packet header with the local station address (STA). We assume that this requires a delay of $T_a$ seconds, after which correct decisions are made whether the packet has to be passed on to the next station or not, and whether the packet has to be distributed locally. (Naturally, addresses are not needed when only one station receives packets as in a data collection ring, but we suppose here that such a ring will be implemented with more general purpose hardware).

When the packet has to be passed on, switch A is set to position 2 and held there until the entire packet has passed through this switch. Note that, because there are no incoming local packets, the ring packets are delayed only by the constant amount of time $T_a$ required to compare the RAF with the STA; there is no waiting time in a buffer.

(*) The same idea has been proposed in (KER79) for networks with arbitrary topology. When in that system, called virtual cut-through, a message arrives at an intermediate node and its selected outgoing channel is free just after the reception of the header, then, in contrast to store-and-forward message switching, the message is sent out to the adjacent node towards its destination before it is received completely at that node. Only if the message is blocked due to a busy output channel is the message buffered in the intermediate node. The delay analysis of that system required Kleinrock's independence assumption which is clearly not applicable to our ring.
Fig. 3.2.1 Block diagram of station i in a buffer insertion ring.

Now suppose that a local packet requests access to the ring during a ring output idle period. In this case, the request is granted immediately, i.e., switch C is set to position 3 and the packet is transmitted without delay. Simultaneously, switch B is set to position 2 to ensure that all ring packets that may arrive during the transmission of the local packet will enter the RB. Switch B remains in this position until the end of the current ring output busy period. It is this "insertion" of the RB into the transmission path between ring input and output that led to the name "buffer insertion ring". At the end of the local packet transmission, the downstream link arbiter (DLA) checks to see if there are any other packets waiting for transmission, either in the AB or in the RB. We have to distinguish between three cases:
1) If there are no packets waiting in either buffer, the ring output returns to its idle state, and switches A, B and C are set again to position 1,

2) If there is exactly one buffer that has packets waiting, then switch C is set to position 2 or 3 according to whether packets are waiting in the RB or AB, respectively. The most recent packet in this buffer is then transmitted without any further delay,

3) If there are packets waiting in both buffers, the DLA selects the next packet to be transmitted (by setting switch C either to 2 or 3) according to some queueing discipline, such as first-come-first-served (FCFS), ring priority (RP) or local priority (LP).

Notice, that in case 3) above, the next packet is transmitted without further delay in the sense that it immediately follows the preceding packet.

A service algorithm for a queueing system is said to be work-conserving (KLEI76, p. 113) when it meets the following requirements: (a) the server never stands idle when there is work to be done, and (b) no customers depart before they are completely serviced. In other words, within a work-conserving queueing system, no work is created or destroyed. We note now that the above-described access control technique is work-conserving regardless of the queueing discipline, because conditions (a) and (b) are always satisfied.

The local packet arrival rate \( \lambda_i \) at station \( i \) is defined as the almost sure limit

\[
\lambda_i = \lim_{t \to \infty} \frac{n_i(t)}{t}
\]  

(3.2.1)

where \( n_i(t) \) is the number of packets generated at node \( i \) in the time interval \([0,t)\). The ring packet arrival rate at station \( i \), \( \lambda_r(i) \), is similarly defined and

\[
\Lambda(i) = \lambda_a(i) + \lambda_r(i)
\]  

(3.2.2)

is the total arrival rate of packets at node \( i \).
Theorem 3.2.1

The average waiting time of packets at station $i$ is given by

$$ W(i) = \frac{\lambda_a(i)}{\Lambda(i)} \cdot W_a(i) + \frac{\lambda_r(i)}{\Lambda(i)} \cdot W_r(i) \tag{3.2.3} $$

where $W_a(i)$ and $W_r(i)$ are the average waiting times of local packets in the AB and of ring packets in the RB, respectively, at station $i$, and where $\Lambda(i)$ is defined by (3.2.2).

Proof

Let $A(i)$ denote the event that a randomly chosen packet that is serviced at node $i$ is a local packet. The complementary event $A^C(i)$ is then the event that a randomly chosen packet serviced at node $i$ is a ring packet. These events have the probabilities

$$ P\{A(i)\} = \frac{\lambda_a(i)}{\Lambda(i)} \tag{3.2.4} $$

$$ P\{A^C(i)\} = \frac{\lambda_r(i)}{\Lambda(i)} \tag{3.2.5} $$

From the theorem of total expectation we have therefore

$$ W(i) = E\{w(i)\} 
= E\{w(i)\|A(i)\}P\{A(i)\} 
+ E\{w(i)\|A^C(i)\}P\{A^C(i)\} \tag{3.2.6} $$

where $w(i)$ is the waiting time of the randomly chosen packet serviced at station $i$. By definition

$$ E\{w(i)\|A(i)\} = W_a(i) \tag{3.2.7} $$

and

$$ E\{w(i)\|A^C(i)\} = W_r(i) \tag{3.2.8} $$

thus theorem 3.2.1 follows directly from (3.2.4)-(3.2.8).
Kleinrock (KLEI76, p.320) has shown that the average total waiting time in a network with arbitrary topology is equal to the sum over all i of the average waiting times in node i, weighted by the ratio of the rate of the traffic serviced at node i to the total incoming traffic rate. Thus, the total average waiting time at the stations in the ring can be written

\[ W_{\text{tot}} = \sum_{i=1}^{m} \lambda_i \frac{W(i)}{\lambda} \]  

(3.2.9)

for a randomly chosen packet where

\[ \lambda = \sum_{i=1}^{m} \lambda_i \]  

(3.2.10)

is the total arrival rate of packets into the ring.

### 3.3 Data collection ring

In this section we give an exact analysis of a data collection ring with buffer insertion. We assume until further notice that the time, \( T_a \), for address comparison is zero, and, therefore, packet arrival instants at RB(i) in Fig. 3.3.1(a) coincide with the instants when service begins at server i-1. Because this is in contrast to other queueing networks where the buffer arrival instants usually coincide with the instants when service is completed at the preceding server, Fig. 3.3.1(a) is somewhat nonstandard. Whenever there is at least one packet either in AB(i-1) or in RB(i-1) at an instant when server i-1 completes a service, then exactly one packet is immediately selected for the next service by server i-1 according to the queueing rule in force, and a copy of the selected packet is also transferred (conceptually) simultaneously to RB(i). Switch S(i-1) is then opened until the packet has been served; then the next packet is taken from either AB(i-1) or RB(i-1), etc.

**Lemma 3.3.1**

Given the following two conditions:

1) The arrival process into Q0 of Fig. 3.3.1(b) is the superposition of the arrival processes into AB(1), AB(2), ..., AB(i-1) in Fig. 3.3.1(a).
Fig. 3.3.1(a) Queueing network representation of the buffer insertion ring.

Fig. 3.3.1(b) Equivalent queueing network in the sense that the statistics of the waiting time in Q1 and Q2 coincide with the waiting times in RB(i) and AB(i), respectively, in Fig. 3.3.1(a).

Fig. 3.3.1(c) Equivalent queueing system in the sense that the average waiting time in Q3 coincides with the total average waiting time in the system of Fig. 3.3.1(b).
2) Servers B in Fig. 3.3.1(b) and i in Fig. 3.3.1(a) use the same serving discipline,

there exists a genie-aided(*) work-conserving server A for the system of Fig. 3.3.1(b), where the genie informs server A only of the station of origin for each packet entering Q0, such that

1) The statistics of the waiting time in Q1 coincides with that in RB(i) of Fig. 3.3.1(a), in particular,

\[ W_{Q1} = W_r(i) \]  \hspace{1cm} (3.3.1)

and

2) the statistics of the waiting time in Q2 coincides with that in AB(i), in particular,

\[ W_{Q2} = W_a(i) \]  \hspace{1cm} (3.3.2)

Proof

We assume that server A in Fig. 3.3.1(b) knows the queueing disciplines that will be used by servers 1, 2, ..., i-1 in Fig. 3.3.1(a). We show that the following service algorithm for server A is work-conserving and yields the desired statistics for waits in Q1 and Q2: Service a packet at the same instant that same packet is serviced by server i-1 in Fig. 3.3.1(a). This is a possible service rule for server A since, with the genie's information, server A can "simulate" the actions of servers 1, 2, ..., i-1 in Fig. 3.3.1(a) for any realization of the arrival process. This service rule gives the desired statistics in Q1 and Q2 when server B uses the same service rule as server i in Fig. 3.3.1(a). It remains only to show that this service rule for server A is work-conserving, i.e., that server A is never idle when Q0 is not empty. Suppose on the contrary

(*) The genie (or "oracle") of server A is the possessor of information that may be unknown to this server. The genie informs the server of this "extra information", whose specification defines the genie.
there is some first time when server A becomes idle with QO non-empty. This implies that server i-1 also becomes idle. Moreover, at least one packet not yet serviced by server A and hence also not yet serviced by server i-1 is in service by some server j (1 ≤ j < i-1) as follows from the fact that servers 1, 2, ..., i-2 are work-conserving (as was pointed out in section 3.2). But, as we noted previously, when server j begins to service this packet a copy is inserted into RB(j+1) so that server j+1 must also now have a packet in service. Thus, servers j+1, j+2, ..., i-2 must also have packets in service and hence RB(i-1) is not empty. It follows, since RB(i-1) is not empty but server i-1 is idle, that server i-1 is not work-conserving. This is a contradiction; we conclude that QO must be empty whenever server A is idle, as was to be shown.

Lemma 3.3.2

Provided that

1) server A is work-conserving,

2) the local packet arrival processes at station j (1 ≤ j < i) are independent Poisson processes (the rates do not need to be equal),

3) the length of all packets originating at stations 1, 2, ..., i-1 are drawn independently from the same distribution,

the average waiting time in QO in Fig. 3.3.1(b) is given by

\[ W_{Q0} = MGl(\lambda_{i}(i)S) \]  \hspace{1cm} (3.3.3)

where

\[ \lambda_{i}(i) = \sum_{j=1}^{i-1} \lambda_{a}(j) \]  \hspace{1cm} (3.3.4)

and where MGl(xS) is the average waiting time of packets of average length S arriving at rate x for service at a FCFS M/G/1 queueing system, namely

\[ MGl(xS) = \frac{xS^2}{1-xS} \cdot \frac{1+c^2}{2} \]  \hspace{1cm} (3.3.5)
where $C_s$ is the coefficient of variation of the packet lengths and is defined by

$$C_s = \frac{\text{var}(s)}{s^2} \quad (3.3.6)$$

where $\text{var}(s)$ is the variance of the packet length $s$.

Proof

Because the superposition of Poisson arrival processes is again Poisson, it follows that the arrival process into QO of Fig. 3.3.1(b) is Poisson. Kleinrock (KLEI76, p.113) has shown that, for Poisson arrivals in a work-conserving system, the average wait for service is independent of the order of service provided that the order is not service-time dependent. But the genie identifies for server A only the origin of the packets in QO; the genie does not even need to know the length of the packets in QO. Provided that the lengths of all packets arriving at station $j$ are drawn independently from the same distribution for $1 \leq j \leq i$, the average wait in QO for the genie-aided work-conserving rule of lemma 3.3.1 is therefore the same as that when server A uses any other work-conserving rule, even ones that ignore the genie's advice such as FCFS. But this average wait for FCFS service of Poisson arrivals is, by definition, $M_G(\lambda_r(i)S)$; formulas (3.3.5) and (3.3.6) for this wait are taken from Kleinrock (KLEI75 p. 190 and 381).

Remark: Kleinrock's argument (KLEI76, p. 113) actually shows that the average wait is independent of order of service, even if customers are segregated into classes that are serviced differently provided only that, within each class, the order of service is not service-time dependent and that the average service time is the same for all classes. Thus lemma 3.3.2 holds under the weaker condition that the average length of packets arriving at station $j$ is the same for $1 \leq j \leq i$, even if the distribution of packet length depends on $j$. However, we have no interest in this greater generality in this paper.
Lemma 3.3.3

Given that

1) The arrival process into Q3 of Fig. 3.3.1(c) is the superposition of the two stationary and independent Poisson arrival processes into Q0 and Q2 of Fig. 3.3.1(b),

2) the packet lengths of all packets arriving at Q0 and Q2 are drawn independently from the same distribution,

3) servers A, B and C are work-conserving,

the average waiting time in the system of Fig. 3.3.1(b) is equal to the average waiting time \( MG1(\lambda(i)S) \) in the system of Fig. 3.3.1(c), i.e.,

\[
MG1(\lambda(i)S) = \frac{\lambda_r(i)}{\Lambda(i)} \cdot \left[ W_{Q0} + W_{Q1} \right] + \frac{\lambda_a(i)}{\Lambda(i)} \cdot W_{Q2} \quad (3.3.7)
\]

Proof

We first assume that server C has a genie to tell him which packet to serve next such that the output processes of the systems in Figs. 3.3.1(b) and 3.3.1(c) coincide. The same argument as in the proof of lemma 3.3.2 then shows that, with the genie-dictated order of service, the average waiting time in Q3 is the same as for a FCFS server (who ignores the genie's advice), namely \( MG1(\lambda(i)S) \).

Theorem 3.3.1

Regardless of the particular work-conserving queueing discipline by which packets are selected from AB(i) and RB(i) in Fig. 3.3.1(a), the average waiting time of packets in station i in Fig. 3.3.1(a) is given by

\[
W(i) = MG1(\lambda(i)S) - \frac{\lambda_r(i)}{\Lambda(i)} \cdot MG1(\lambda_r(i)S) \quad (3.3.8)
\]

provided that the local packet arrivals at all stations are generated by independent stationary Poisson processes, and that the lengths of all packets are drawn independently from the same distribution.
Using lemma 3.3.2 in lemma 3.3.3, we have
\[ M_{Gl}(\Lambda(i)S) = \frac{\lambda_r(i)}{\Lambda(i)} \cdot \left( M_{Gl}(\lambda_r(i)S) + W_{Q1} \right) + \frac{\lambda_a(i)}{\Lambda(i)} \cdot W_{Q2}. \] (3.3.9)

Now using lemma 3.3.1 gives
\[ M_{Gl}(\Lambda(i)S) = \frac{\lambda_r(i)}{\Lambda(i)} \cdot \left( M_{Gl}(\lambda_r(i)S) + W_r(i) \right) + \frac{\lambda_a(i)}{\Lambda(i)} \cdot W_a(i). \] (3.3.10)

Theorem 3.2.1 now gives
\[ M_{Gl}(\Lambda(i)S) = W(i) + \frac{\lambda_r(i)}{\Lambda(i)} \cdot M_{Gl}(\lambda_r(i)S), \quad \text{a.e.d.} \]

We remark that theorem 3.3.1 remains valid even when our assumption is removed that the address comparison time, \( T_a \), is zero, provided that the total wait of a packet is defined to be the waiting time in the AB and RB buffers only, i.e., if we ignore the waiting time in the address comparison buffer. This follows because the effect of \( T_a > 0 \) is merely to introduce a "time origin shift" to the packet arrival processes of preceding stations as seen by following stations. To account for the additional wait in the address comparison buffer when \( T_a > 0 \), we note simply that every ring packet will wait exactly \( T_a \) seconds in the address comparison buffer at station \( i \). Thus (3.3.8) becomes
\[ W(i) = M_{Gl}(\Lambda(i)S) - \frac{\lambda_r(i)}{\Lambda(i)} \cdot \left( M_{Gl}(\lambda_r(i)S) - T_a \right). \] (3.3.11)

An interesting special case of theorem 3.3.1 is contained in the following corollary.

**Corollary 3.3.1**

Under the same conditions as in theorem 3.3.1, in a data collection ring with symmetric traffic, i.e., where \( \lambda_a(i) = \lambda_a \), \( 1 \leq i \leq m \), the average total wait of a randomly chosen packet is
Proof

The symmetric traffic condition implies

\[ \lambda = m \lambda_a \]  \hspace{1cm} (3.3.13)

and also

\[ \lambda_r(i) = (i-1) \lambda_a \]  \hspace{1cm} (3.3.14)

and

\[ \Lambda(i) = i \lambda_a . \]  \hspace{1cm} (3.3.15)

Substituting the above expressions into (3.3.8) gives

\[ W(i) = MGL(i \lambda_a S) - \frac{i-1}{i} \cdot MGL((i-1) \lambda_a S) . \]  \hspace{1cm} (3.3.16)

Moreover, (3.2.9) now becomes

\[ W_{tot} = \frac{1}{m} \cdot \sum_{i=1}^{m} iW(i) \]

\[ = \frac{1}{m} \cdot \left( \sum_{i=1}^{m} iMGL(i \lambda_a S) - \sum_{i=1}^{m} (i-1)MGL((i-1) \lambda_a S) \right) \]

\[ = \frac{1}{m} \left( \sum_{i=1}^{m} iMGL(i \lambda_a S) - \sum_{i=0}^{m-1} iMGL(i \lambda_a S) \right) . \]  \hspace{1cm} (3.3.17)

All terms cancel except the \( i=m \) term in the first sum so that

\[ W_{tot} = MGL(\lambda S) \]  \hspace{1cm} (3.3.18)

as was to be shown.

At first glance corollary 3.3.1 is certainly surprising: it states that the average total waiting time in the symmetric data collecting ring
is the same as in a FCFS star network (Fig. 3.3.2) with a central node serving capacity equal to the capacity of a single link between ring stations. There is, however, an intuitively pleasing argument to confirm this result: one can imagine that all stations, except station \( m \) just before the data collecting station, do not cause any delay at all. All these stations do is, in some cases (depending on the queueing discipline), rearrange the order of packet arrivals at the last link. This last link then operates as an \( M/G/1 \) serving system with the same average waiting time as the single shared link in the star network.

Remark: Including the effect of \( T_a > 0 \), we get in place of (3.3.12) the following expression for the average delay in a data collection ring with symmetric traffic:

\[
W_{\text{tot}} = MGl(\lambda S) + \frac{m-1}{2}T_a . \tag{3.3.19}
\]
Fig. 3.3.2 Data collection with a FCFS star network.
3.4 Waiting time as a function of the station number

In the previous section we showed that the average waiting time is the same for the data collection ring and the FCFS star. There is however one significant difference: the FCFS star network is fair in the sense that packets from all stations have the same average delay, i.e.

\[ W_\text{tot}^{(i)} = W_\text{tot} = MGI(\lambda S); \quad 1 \leq i \leq m \]  \hspace{1cm} (3.4.1)

whereas in the ring the delay depends on the station number, because

\[ W^{(x)}_\text{tot}^{(i)} = W^{(x)}_a^{(i)} + \sum_{j=i+1}^{m} W^{(x)}_r^{(j)} \]  \hspace{1cm} (3.4.2)

where \( W_a^{(x)}^{(i)} \) and \( W_r^{(x)}^{(i)} \) denote the average waiting times of ring packets in RB(i) and of local packets in AB(i) when the queueing discipline in node i is x. We now proceed to derive explicit expressions for \( W_a^{(RP)}^{(i)} \), \( W_a^{(LP)}^{(i)} \), \( W_r^{(RP)}^{(i)} \), and \( W_r^{(LP)}^{(i)} \).

For RP, the queueing system of Fig. 3.3.1(b) is equivalent to a non-preemptive priority queueing system with Poisson arrivals in which the local packets are the low priority customers. Thus, the average sum of the waiting times in Q0 and Q1 (Fig. 3.3.1(b)) is equal to the waiting time of the high-priority customers, which can be shown (ALL78, p. 370) to be

\[ W_{Q0} + W_{Q1} = \frac{\Lambda(i)E\{s^2\}}{2(1-\lambda_r^{(i)}S)} \]  \hspace{1cm} (3.4.3)

But, using (3.3.1), (3.3.3) and (3.3.5), we obtain from (3.4.3)

\[ W_r^{(RP)}^{(i)} = \frac{\Lambda(i)E\{s^2\}}{2(1-\lambda_r^{(i)}S)} - MGI(\lambda_r^{(i)}S) \]

\[ = \frac{\Lambda(i)E\{s^2\} - \lambda_r^{(i)}S^2(1+C_s^2)}{2(1-\lambda_r^{(i)}S)} \]  \hspace{1cm} (3.4.4)

But (3.2.3) and (3.3.8) give

\[ \frac{\lambda_a^{(i)}}{\Lambda(i)} \cdot W_a^{(RP)}^{(i)} + \frac{\lambda_r^{(i)}}{\Lambda(i)} \cdot W_r^{(RP)}^{(i)} = MGI(\Lambda(i)S) - \frac{\lambda_r^{(i)}}{\Lambda(i)} \cdot MGI(\lambda_r^{(i)}S) \]  \hspace{1cm} (3.4.5)
Substituting (3.4.4) into (3.4.5) gives

\[ W_{a}(RP) (i) = \frac{\Lambda(i)}{\lambda_a(i)} \left( \frac{\Lambda(i)S^2(1+C_s^2)}{2(1-\Lambda(i)S)} - \frac{\lambda_r(i)E[s^2]}{2(1-\lambda_r(i)S)} \right) \]  \hspace{1cm} (3.4.6)

For the special case of an exponential packet length distribution, \( C_s = 1 \) and \( E(s^2) = 2S^2 \) so that (3.4.4) and (3.4.6) become

\[ W_{r}(RP) (i) = \frac{\lambda_a(i)S^2}{1-\lambda_r(i)S} \]  \hspace{1cm} (3.4.7)

\[ W_{a}(RP) (i) = \frac{\Lambda(i)S^2}{\lambda_a(i)} \left( \frac{\Lambda(i)}{1-\Lambda(i)S} - \frac{\lambda_r(i)}{1-\lambda_r(i)S} \right) \]  \hspace{1cm} (3.4.8)

For the special case of constant length packets, \( C_s = 0 \) and \( E(s^2) = S^2 \) so that the values of \( W_{r}(RP) (i) \) and \( W_{a}(RP) (i) \) are equal to half the values given by the right sides of (3.4.7) and (3.4.8), respectively.

When the queueing disciplines for the data collection ring is the LP rule, a similar analysis to that which led to (3.4.4) and (3.4.6) gives

\[ W_{a}(LP) (i) = \frac{\Lambda(i)E[s^2]}{2(1-\lambda_a(i)S)} \]  \hspace{1cm} (3.4.9)

\[ W_{r}(LP) (i) = \frac{\Lambda(i)}{\lambda_r(i)} \left( \frac{\Lambda(i)S^2(1+C_s^2)}{2(1-\Lambda(i)S)} - \frac{\lambda_a(i)E[s^2]}{2(1-\lambda_a(i)S)} \right) \]  \hspace{1cm} (3.4.10)

Fig. 3.4.1 is a plot of \( W_{tot}(LP) (i) \) and \( W_{tot}(RP) (i) \) (eqn. 3.4.2), normalized with respect to the average packet length \( S \), for a data collecting ring with \( m=15 \) stations, constant length packets, symmetric traffic, and various utilizations \( \phi = \lambda S \). We see that

* For low utilization, the range of delays is relatively small. As the utilization increases, however, some stations have considerably smaller delays than other stations,

* RP gives better service to stations far away from the data collecting station - at the expense of the stations near the data collecting
station. With RP, the last station just before the data collecting station has the longest delays although its packets have only one hop to go until they reach their destination! RP is thus seen to penalize downstream stations in a similar way as the data collecting ring with random slot seizure (SPR72) does,
* LP, on the other hand, gives better service to stations near the data collecting station - at the expense of the stations far away from the data collecting station. While this qualitative result is fairly obvious, it is interesting and perhaps counterintuitive to note that LP is fairer than RP, i.e., it has a smaller range of delays:

\[
\max_i W_{\text{RP}}^{(\text{tot})}(i) - \min_i W_{\text{RP}}^{(\text{tot})}(i) > \max_i W_{\text{LP}}^{(\text{tot})}(i) - \min_i W_{\text{LP}}^{(\text{tot})}(i).
\]

LP is therefore superior (at least with respect to fairness) to RP.

The buffer insertion ring would in principle allow for more general service disciplines than either pure LP or RP, e.g., dynamic or time dependent priorities. There is however no (at least no simple) such service discipline that would in a general data collection ring give equally fair service (as defined by eqn. 3.4.1) to all stations in the same way as the FCFS star does. This would in some applications be considered to be a major drawback of a data collection ring.

3.5 Conclusions

We have given an exact waiting time analysis of a data collection ring with a (work-conserving) decentralized buffer insertion access mechanism. It was shown, under the assumptions that

1) packet arrival processes at individual ring stations are independent Poisson processes,

2) all packet lengths are drawn independently from the same distribution,

3) the processing time per station is zero,

that the average delay in the data collection ring with symmetric traffic is the same as in a FCFS star network, regardless of the priority rule
with which packets are served at the individual stations.

The more common rule of giving packets on the ring priority over the packets that attempt to access the ring was shown to penalize downstream terminals in a similar way as in the slotted ring (SPR72), especially when the utilization is high. Surprisingly, the converse rule of giving new packets priority when they attempt to access the ring was found to be fairer in the sense that the range of average delays as seen by the individual stations is less than with the first rule. Unfortunately, there is however no (at least no simple) priority rule that makes the average delay equal for all stations. This is considered to be a major drawback of ring systems in some data collection applications.

3.6 References


SPR72  J. D. Spragins, "Loops used for Data Collection," Symp. Computer-Communications Networks and Teletraffic, Polytechnic Inst. of Brooklyn, April 4-6, 1972, pp. 59-76.

4. WAITING TIME ANALYSIS OF A BUFFER INSERTION RING

4.1 Introduction

In chapter 3, we considered the case of a data collection ring\(^(*)\) with buffer insertion and gave an exact analysis of the average waiting time of packets in the individual stations as well as of the average total waiting time of packets. Here we consider a general buffer insertion ring where stations 1, 2, ..., m (Fig. 4.1.1) can transmit to each of the other stations and give approximate analyses of the average total waiting time for the cases where 1) all packet arrival processes are Poisson, and where 2) all packet arrival processes are periodic (with random phase shifts between the individual packet streams). Both approximations are then compared with simulations.

The local packet arrival rate at node \(i\), \(\lambda_a(i)\), is defined as the almost sure limit

\[
\lambda_a(i) = \lim_{t \to \infty} \frac{n_i(t)}{t} \quad (4.1.1)
\]

where \(n_i(t)\) is the number of packets that enter at node \(i\) in the time interval \([0, t)\). The packet departure rate at node \(i\), \(\lambda_d(i)\), is similarly defined. These packet rates obey the conservation law

\[
\sum_{i=1}^{m} \lambda_d(i) = \sum_{i=1}^{m} \lambda_a(i) \quad (4.1.2)
\]

whenever the average packet delay is finite. Note, however, that the individual arrival rates \(\lambda_a(i)\) need not equal the corresponding departure rates \(\lambda_d(i)\) because packets need not be removed from the ring at the same node where they entered, although this is often done primarily to simplify the mechanization of packet removal but also to provide an implicit acknowledgement of packet reception. One of the advantages of

\(^(*)\) A data collection ring is a system where stations 1, 2, ..., m-1 transmit through a single unidirectional link only to station m, which operates only as a data collecting station, and transmits to no one.
Fig. 4.1.1 General buffer insertion ring where all stations can transmit to each of the other stations.
the buffer insertion mechanism is that packets can be conveniently removed from the ring as soon as they reach their destination. In the case of "uniform" traffic, it will be shown in section 4.2 that this immediate removal of packets results in a doubling of the capacity of the ring compared to systems where packets are removed only after a round trip.

The total rate of packets transmitted from station $i$ to station $i+1$ is given by

$$\Lambda(i) = \Lambda(i-1) - \lambda_d(i) + \lambda_a(i)$$

and the rate of ring packets that station $i$ relays to station $i+1$ is given by

$$\lambda_r(i) = \Lambda(i-1) - \lambda_d(i).$$

Kleinrock (KLEI76, p.320) has shown that the average total waiting time in a network with arbitrary topology is equal to the sum over all $i$ of the average waiting times in node $i$, weighted by the ratio of the rate of the traffic serviced at node $i$ to the total incoming traffic rate in the network. Thus, the total average waiting time at the stations in the ring can be written

$$W_{tot} = \sum_{i=1}^{m} \frac{\Lambda(i)}{\lambda} W(i)$$

where

$$\lambda = \sum \lambda_a(i)$$

is the total arrival rate of packets into the ring, and where $W(i)$ is the average waiting time of packets at station $i$.

For (4.1.5) to be useful, we must somehow determine $W(i)$. In section 4.2, we assume that the local packet arrival processes are Poisson and we approximate $W(i)$ by assuming that a property which we proved in chapter 3 to hold in a data collection buffer insertion ring will also hold (at least approximately) in a general buffer insertion ring where all stations can transmit to each of the other stations. In section 4.3, we shall compare these estimates with simulation results and previously
published analytical models. It will be shown that our estimates agree closely with the simulation results over a wide range of parameter values for low and moderate utilizations, but are somewhat larger than the simulated values for high utilization. In section 4.4, we drop the assumption that the local packet arrival processes are Poisson. Instead we consider the case when the local packet arrival processes are periodic (with random phase shifts between the packet streams generated by the individual stations) and use an approximate analysis and simulations to show that in this case the average total waiting time is considerably less than with Poisson arrivals.

In the following, we assume that ring packets are relayed only until they reach their destination, where they are removed from the ring. The method of decentralized access control in a buffer insertion ring has been described in chapter 3 and is not repeated here. We repeat, however, the following theorem from chapter 3.

**Theorem:**

For any work-conserving queueing discipline by which packets are selected from buffers $AB(i)$ and $RB(i)$ in Fig. 4.1.2, the average waiting time of packets in the buffers of station $i$ is given by

$$W(i) = MGl\{\lambda(i)S\} - \frac{\lambda_r(i)}{\lambda(i)} MGl\{\lambda_r(i)S\}$$

(4.1.7)

provided that

1) the local packet arrivals at all stations are generated by independent Poisson processes,

2) the lengths of all packets are drawn independently from the same distribution with mean $S$,

3) the ring operates as a data collection ring, i.e., stations 1, 2, ..., $m-1$ transmit only to station $m$, which transmits to no one.

In (4.1.7), $MGl(xS)$ denotes the average waiting time of packets of average length $S$ arriving at rate $x$ for service at a first-come-first-served (FCFS) $M/G/1$ queueing system, namely
Fig. 4.1.2 Block diagram of station i in a buffer insertion ring.

\[MG_1(xS) = \frac{xS^2}{1-xS} \frac{1+C^2}{2}\]  \hspace{1cm} (4.1.8)

where \(C_s\) is the coefficient of variation of the packet length \(s\) and is defined by

\[C_s = \frac{\text{var}(s)}{S^2}\]  \hspace{1cm} (4.1.9)

where \(\text{var}(s)\) is the variance of the packet length \(s\).

Remark:

Condition 3) above implies that no station (except the data collecting station) ever removes packets from the ring. In the following section we shall consider a more general case where all stations may
remove packets from the ring so that this theorem is not directly applicable. The theorem will, however, be the basis of an approximate analysis.

4.2 Buffer insertion ring with Poisson arrivals

In this section we consider the case of a buffer insertion ring with uniform traffic, i.e., a ring where each station \( i, 1 \leq i \leq m \) transmits with the same rate

\[
\lambda_a(i) = \lambda_a = \frac{S_a(i)}{m-1} = \frac{\lambda}{m(m-1)} ; 1 \leq i \leq m \tag{4.2.1}
\]

to each other station \( j, 1 \leq j \leq m, j \neq i \). Therefore, we have

\[
\lambda_a(i) = \lambda_a = \frac{\lambda}{m} ; 1 \leq i \leq m. \tag{4.2.2}
\]

Moreover, since station \( i \) relays ring packets from \( m-2 \) nodes destined for node \( i+1 \), from \( m-3 \) nodes destined for node \( i+2 \), etc., we see that

\[
\lambda_r(i) = \sum_{j=2}^{m-1} (m-j) \lambda_a = \frac{(m-2)(m-1)}{2} \lambda_a, \tag{4.2.3}
\]

and thus

\[
\lambda_r(i) = \lambda_r = \lambda_a \left( \frac{m}{2} - 1 \right) ; 1 \leq i \leq m \tag{4.2.4}
\]

and

\[
\Lambda(i) = \Lambda = \frac{\lambda}{2} ; 1 \leq i \leq m. \tag{4.2.5}
\]

Using (4.2.5) in (4.1.5) now gives the following expression for the average delay of a randomly chosen packet:

\[
W_{\text{tot}} = \frac{mW}{2} \tag{4.2.6}
\]

where \( W \) is the average wait in the buffers of an arbitrary station.

It is intuitively clear that there is no fundamental difference between the queueing dynamics of the data collection ring and those of
the more general buffer insertion ring described here. We claim that the characteristics of both systems will be almost identical and use the theorem of the preceding section as the basis for the following conjecture.

Conjecture

Provided that the local packet arrivals at all stations in the buffer insertion ring with uniform traffic are generated by independent Poisson processes and that the lengths of all packets are drawn independently from the same distribution with mean S; then, for any work-conserving queueing discipline (regardless of the priority rule by which packets are selected from buffers AB and RB in Fig. 4.1.2) the average wait of a randomly chosen packet in the buffers at station \( i \) is well-approximated by (4.1.7). This implies that the average total waiting time in the buffers of the buffer insertion ring with uniform traffic is well-approximated by

\[
W_{\text{tot}} = \frac{m}{2} \cdot MGl\left(\frac{\lambda S}{2}\right) - \frac{m-2}{2} \cdot MGl\left(\frac{m-2}{2} \cdot \lambda a S\right). \tag{4.2.7}
\]

Remarks:

1) If one assumes that (4.1.7), which holds exactly for the data collection ring, is approximately satisfied by the buffer insertion ring with symmetric traffic, then (4.2.2)-(4.2.5) can be used on the right side in (4.1.7) to yield

\[
W = W(i) = MGl\left(\frac{\lambda S}{2}\right) - \frac{m-2}{m} \cdot MGl\left(\frac{m-2}{2} \cdot \lambda a S\right) \tag{4.2.8}
\]

as an approximation for the average wait at each station \( i \), \( 1 \leq i \leq m \). Substitution of (4.2.8) into (4.2.6) then gives (4.2.7)

2) The right side of (4.2.7) becomes infinite as the utilization \( \rho = \lambda S \) approaches 2. This value of the maximum stable utilization results from the fact that packets are removed from the ring at the destination and thus (because of the uniform traffic assumption) packets make on the average only half a ring round trip before they reach their destination. In other words, we have twice the "capacity"
of systems in which packets are removed from the ring only after they have returned to the originating station after a complete round trip since \( W_{\text{tot}} \) becomes infinite as the utilization approaches 1 in such systems.

**4.3 Simulation results and previous analytical models**

In the previous section the total average wait for a randomly chosen packet in the buffers of a buffer insertion ring with symmetric traffic has been conjectured to be independent of the priority rule. We know, however, (from chapter 3) that the waiting time distribution depends strongly on the priority rule even in the data collection ring. We should thus expect a strong dependence of the waiting time distribution on the priority rule in the buffer insertion ring with symmetric traffic. This dependence is confirmed in Fig. 4.3.1 which contains histograms of the average total wait \( W_{\text{tot}} \), normalized with respect to the average packet length \( S \), from simulations of a buffer insertion ring with uniform traffic with \( m=5 \) stations, constant length packets and utilization \( \rho = \lambda S = 1.5 \). The average total wait measured in the simulations was 2.605±.122 (95 per cent confidence interval) with local priority (LP) and 2.591±.105 for ring priority (RP), which confirms our conjecture that the average wait is not significantly affected by the priority rule. We see, however, that with LP a relatively large fraction of packets has very short waiting times which is offset by only a few packets with extremely large waiting times; the maximum observed value was 40.6. With RP, for the same realization of the arrival process, the maximum observed value was only 27.2. The probability of both extremely short and extremely long waiting times is seen to be smaller with RP than with LP. An intuitively appealing explanation of this property is that, with RP, most of the total waiting time is local access time, which is independent of the number of hops a packet has to go and is therefore constant for all packets. On the other hand, with LP, a packet spends most of its waiting time while it is on the ring. Therefore, the total waiting time increases with the number of hops to go. With LP, packets with only a few hops to go have extremely short waiting times, but those with many hops to go have extremely long waiting times. Packets with only a few hops to go are therefore better off with LP than with RP, but those with many hops to go are better off with RP.
Fig. 4.3.1 Histogram of the average total waiting time $W_{\text{tot}}$, normalized with respect to the (constant) packet length $S$ (from the simulation of a buffer insertion ring with symmetric traffic with $m=5$ stations and utilization $\rho=\lambda S=1.5$). The solid curve is for RP, the dashed curve is for LP.

Figs. 4.3.2(a), (b), and (c) show 95 per cent confidence intervals of the (normalized) average total waiting time as obtained from simulations of rings with $m=5$, 16, and 100 stations, respectively, and with an exponential packet length distribution. The simulation results in Figs. 4.3.2(a), (b) and (c) are labelled with LP or RP according as to whether the LP priority rule or the RP priority rule, respectively, was in effect. Notice that
95 per cent confidence intervals of the average total waiting time estimates (normalized with respect to the average packet length $S$) obtained from simulations of rings with $m=5$ stations. One half of the simulations used the LP rule, the other half used the RP rule. The solid curves are the estimates obtained from our new model, eqn. 4.2.7. The dashed curves are from Liu's model (eqns. 4.3.2-4.3.6).

1) The experimentally obtained points in Figs. 4.3.2(a), (b) and (c) alternate between RP and LP, yet appear to be following the same smooth curve. This is rather dramatic evidence of the fact that, over the entire range of utilizations, the priority rule had no visible effect on the average waiting time.

2) For low and moderate utilizations, the estimates obtained from (4.2.7) (solid curve) are excellent approximations to the waiting times determined by simulation.
3) For higher utilizations, the estimates from (4.2.7) are somewhat too large. This is probably due to the fact that the "gaps" in the RB input traffic (resulting from removed packets) are often filled by local packets which reduces the average total waiting time.

Table 4.3.1 shows the simulated waiting times in a ring with \( m = 5 \) stations and with exponential packet length distribution (as plotted in Fig. 4.3.2(a)) together with simulated waiting times in the same ring, but with constant length packets. Comparison of the two columns in Table 4.3.1 confirms that, with the exponential packet length distribution, the
average waiting time is approximately twice the average waiting time with constant length packets. Note that conjecture (4.2.7) predicts this doubling of the average delay for exponentially distributed packet length compared to constant packet length.

To our knowledge, there have been previously published only two essentially different analytical models for the waiting times in buffer insertion rings. One of these models is due to Liu (LIU77, LIU77A) who represented the ring stations simply as a head-of-the-line (HOL) priority queueing system with both arrival processes assumed to be Poisson (see Fig. 4.3.3, where Q1 and Q2 correspond to RB and AB respectively). This model ignores the fact that the ring packet arrivals are in reality much
<table>
<thead>
<tr>
<th>$\rho$</th>
<th>priority</th>
<th>$W_{tot}/S$ constant</th>
<th>$W_{tot}/S$ exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>LP</td>
<td>0.12±0.01</td>
<td>0.23±0.02</td>
</tr>
<tr>
<td>0.50</td>
<td>RP</td>
<td>0.27±0.02</td>
<td>0.54±0.03</td>
</tr>
<tr>
<td>0.75</td>
<td>LP</td>
<td>0.49±0.06</td>
<td>1.01±0.09</td>
</tr>
<tr>
<td>1.00</td>
<td>RP</td>
<td>0.91±0.02</td>
<td>1.83±0.02</td>
</tr>
<tr>
<td>1.25</td>
<td>LP</td>
<td>1.47±0.01</td>
<td>3.17±0.04</td>
</tr>
<tr>
<td>1.50</td>
<td>RP</td>
<td>2.60±0.10</td>
<td>5.90±1.10</td>
</tr>
<tr>
<td>1.75</td>
<td>LP</td>
<td>6.10±0.90</td>
<td>13.80±3.50</td>
</tr>
</tbody>
</table>

Table 4.3.1

Fig. 4.3.3 Queueing system representation used in previous analytical waiting time models of buffer insertion rings.
more regular and that, in particular,

\[ \tau(i,j) \geq s(i,j) \]  \hspace{1cm} (4.3.1)

where \( \tau(i,j) \) = interval between the arrivals of packets \( j \) and \( j+1 \) in the RB of station \( i \), and where \( s(i,j) \) = length (in time units) of the \( j \)-th ring packet at station \( i \). Liu considered only the LP case and showed that for his model (in our notation)

\[ W_{\text{LP}}^{(a)} = \frac{\Lambda E\{s^2\}}{2(1-\lambda_{a}S)} \]  \hspace{1cm} (4.3.2)

and

\[ W_{\text{LP}}^{(r)} = \frac{\Lambda E\{s^2\}}{2(1-\lambda_{a}S)(1-\lambda_{S})} \]  \hspace{1cm} (4.3.3)

are the average waiting times in the AB and RB, respectively, for LP. Liu then found the average total waiting time from

\[ W_{\text{tot}} = W_{\text{a}} + W_{\text{r}}(\frac{m}{2} - 1) \]  \hspace{1cm} (4.3.4)

which follows from the fact that on the average a randomly chosen packet will wait in \( (m/2)-1 \) RB's at nodes where it is relayed. Liu's estimate of \( W_{\text{tot}} \), obtained by using (4.3.2) and (4.3.3) in (4.3.4), is plotted in Figs. 4.3.2(a), (b) and (c) and is seen to agree reasonably well with the simulation results for the ring with \( m=5 \) stations (Fig. 4.3.2(a)). For rings with more than approximately \( m=10 \) stations, Liu's estimates are however so pessimistic that they appear to be of little use. (Note that, for example with \( m=100 \) and \( p=1 \), Liu's estimate is approximately 20 times larger than the simulation value).

Kanakia (KAN79) presented essentially the same model as Liu, but extended Liu's analysis to the case of RP (Kanakia called a buffer insertion ring with RP an ATDM ring, for "Asynchronous Time Division Multiplex"). Changing the priority rule by which packets are selected from Q1 and Q2 in Fig. 4.3.3 and neglecting the basic requirement (4.3.1), Kanakia obtained

\[ W_{\text{LP}}^{(a)} = \frac{\Lambda E\{s^2\}}{2(1-\lambda_{a}S)(1-\lambda_{S})} \]  \hspace{1cm} (4.3.5)
Fig. 4.3.4(a) 95 per cent confidence intervals of the average total waiting time estimates (normalized with respect to the average packet length $S$) obtained from simulations of rings with LP, constant length packets, and $m=5$ stations. The solid curves are the estimates obtained from our new model, eqn. 4.2.7. The dashed curves are from Schultze's model (eqn. 4.3.9).

$$W_r^{(RP)} = \frac{\Lambda B(S^2)}{2(1-\lambda_r S)}$$

(4.3.6)

as the average waiting times in the AB and RB, respectively, for RP. Because

$$W_r^{(RP)} < W_r^{(LP)}$$

(4.3.7)

Kanakia concluded from (4.3.4) that "as the number of stations increases, the mean response time for the ATDM loop will be better than that of the DLCN loop" (DLCN stands for Distributed Loop Computer Network which corresponds to our buffer insertion ring with LP). Kanakia's conclusion fails to take into account the fact that
must be the same for both LP and RP in his model. Thus (4.2.6) and (4.3.8) imply that \( W_{tot} \) be the same in Kanakia's model for LP and RP. Thus, Kanakia's estimate of \( W_{tot}^{(RP)} \) if plotted in Figs. 4.3.2(a), (b), and (c) would be exactly the same as Liu's estimate of \( W_{tot}^{(LP)} \).

A different model of a buffer insertion ring has been proposed independently by Schultze (SCHU79). His model applies only to the case of LP and constant length packets. In order to be able to represent the contents of the buffers \( AB \) and \( RB \) as a 2-dimensional Markov chain, Schultze made the following assumptions:
1) packet arrivals and departures from the buffers can be considered to occur only at integer multiples of the packet length $S$,

2) the arrival processes into AB and RB can be considered as independent Bernoulli processes.

Schultz's approximation for the average delay in a buffer insertion ring with symmetric traffic and infinite-length buffers is surprisingly
simple, namely

$$W_{\text{tot}}^{(LP)} = \left(1 - \frac{2}{m}\right) \frac{\rho S}{2 - \rho}$$

(4.3.9)

Figs. 4.3.4(a), (b), and (c) show the confidence intervals from simulations of rings with LP, constant length packets and m=5, 16, and 100 stations. These figures show that the accuracy of Schultze's model is only slightly inferior to our model. Schultze's model, however, (especially in rings with many stations and high utilization) slightly underestimates the waiting time, whereas our model gives estimates that are slightly higher than the true waiting times. For the same accuracy, one would generally prefer the conservative estimate. Moreover, our model covers a broader range of situations than does that of Schultze in that it allows an arbitrary distribution of packet length and an arbitrary queueing discipline.

4.4 Buffer insertion ring with periodic arrivals

Previously, we have considered the local packet arrival processes to be Poisson. In order to gain some insight into the sensitivity of the waiting times with respect to the nature of the local packet arrival processes, we consider in this section the case when the local packet arrival processes are periodic.

We now consider an LP ring wherein all stations periodically transmit packets of constant length

$$S = N_0 T_0$$

(4.4.1)

where $N_0$ is the number of 8-bit bytes per packet and where $T_0$ is the time length of a byte; the interval between two successive local packets of an arbitrary station is $1/\lambda_a$. We further assume a random phase shift between the individual local packet streams. For example, for a packetized voice system where each packet carries the same number of digitized speech samples, these assumptions are much more realistic than the Poisson arrival assumption. We anticipate that, with the periodic arrivals, the average total waiting time will be less than with Poisson arrivals.
We now assume that \( m \) is even and that the \( m \) stations can be partitioned into \( m/2 \) randomly chosen pairs of stations communicating with each other. This would be the case, for instance, when \( m/2 \) telephone conversations were being sent over the ring using packetized voice. Thus, \( m \) can here be interpreted as the number of active stations on the ring rather than the total number of stations.

We now assume that ring and local packets are synchronized with the byte clock and that the bytes of each packet are labelled \( 0,1,2,...,N_o-1 \). Together with the assumption that the LP priority rule is in effect this implies that the access time of a randomly chosen local packet is given by

\[
\omega_a = n_b T_0
\]  

(4.4.2)

where \( n_b \) is an integer-valued random variable taking on values between 0 and \( N_o-1 \) inclusive that denotes the index of the byte that is about to be relayed when the local packet arrives. When the local packet finds the station idle, then we can also consider that \( n_b=0 \). When \( n_b>0 \), the local packet must wait until all bytes of the packet currently being relayed have been transmitted before it is granted access since priority is granted on a packet rather than byte basis. The average access time is thus

\[
\bar{\omega} = \mathbb{E}[\omega_a] = T_0 \left( (N_o-1)P{n_b=1} + (N_o-2)P{n_b=2} + ... \right.
\]

\[
\left. + \frac{P{nb=N_o-1)}}{(N_o-1)} \right)
\]

(4.4.3)

where \( P(n_b=i) \) denotes the probability that the byte about to be relayed when the local packet arrives has the index \( i \). Because it is assumed that the local packet arrives at a randomly chosen byte time instant, all non-zero values of \( i \) are equally likely so that

\[
P{n_b=i} = \frac{P_d}{N_o-1} \quad 0<i<N_o
\]

(4.4.4)

where \( P_d \) is the probability that the local packet has to wait in the AB because it does not arrive either in an idle period or simultaneously with byte 0 of a ring packet (note that in the latter case, because of the LP rule, the ring packet is delayed and the local packet is transmitted immediately).
In an interval of length $1/\lambda_a$ between two successive local packets, an arbitrary station transmits at most

$$N_B = \frac{1}{\lambda_a T_o}$$

(4.4.5)

bytes. But

$$\rho = \lambda S = m\lambda_a N_o T_o$$

(4.4.6)

so that (4.4.5) becomes

$$N_B = \frac{mN_o}{\rho} .$$

(4.4.7)

In such an interval of length $1/\lambda_a$, a station relays on the average $(m/2)-1$ ring packets, because the periodicity assumption implies that in each interval of length $1/\lambda_a$ there will be exactly one packet generated from each of the other $(m/2)-1$ active conversations that must be relayed by this station. Among the $N_B$ byte time instants in an interval of length $1/\lambda_a$ when the local packet could arrive (because of the random phase), there are thus on the average

$$\left(\frac{m}{2} - 1\right)\left(N_o - 1\right)$$

instants that will result in delay of the local packet. The probability $P_d$ of non-zero access delay is therefore given by

$$P_d = \frac{\left(\frac{m}{2} - 1\right)\left(N_o - 1\right)}{N_B}$$

$$= \frac{\left(\frac{m}{2} - 1\right)\left(N_o - 1\right)}{mN_o} .$$

(4.4.8)

Substituting (4.4.1), (4.4.4), and (4.4.8) into (4.4.3) gives

$$W_a = \frac{\rho S(m-2)\left(N_o - 1\right)}{4mN_o} .$$

(4.4.9)
We now derive an approximation to the average waiting time of ring packets in the RB of an arbitrary station where ring packets arrive with the same rate

\[ \lambda_r = \lambda a \left( \frac{m}{2} - 1 \right) \]  \hspace{1cm} (4.4.10)

as in the buffer insertion ring with uniform traffic (eqn. 4.2.4). As long as there is no local packet to interfere with the ring packets, the number of ring packets departing from the station is the same as the number of ring packets arriving at the station; thus, RB remains empty.

We now make the assumption that traffic is flowing smoothly through this station at the average rate. There may be a ring packet in service but the RB is empty. It is now convenient to assume that the time when a local packet arrives is \( t=0 \), and that such an arrival causes the RB to begin to fill at the average rate. The transmission of the local packet begins, on the average, after the average access delay \( W_a \) and ends, on the average, at time \( t=W_a+S \). At this latter time, the RB begins to empty with the maximum possible rate of one packet per \( S \) seconds until the RB is again empty. This happens on the average at the time \( T_q \) (see Fig. 4.4.1) when

\[ D_{RB}(t) = \frac{t-W_a-S}{S} \]  \hspace{1cm} (4.4.11)

the cumulative number of packets departing from RB after \( t=0 \), catches up with

\[ A_{RB}(t) = \lambda_r t = \lambda a t \left( \frac{m}{2} - 1 \right) \]  \hspace{1cm} (4.4.12)

the cumulative number of packets arriving at RB after time \( t=0 \) where we have assumed no other local packet arrives before the RB is empty. Setting (4.4.11) equal to (4.4.12) and solving gives the average time required to empty the queue as

\[ T_q = \frac{W_a + S}{1 - \lambda a \left( \frac{m}{2} - 1 \right)} \]  \hspace{1cm} (4.4.13)

The number of ring packets waiting in the RS is given by
Cumulative number of ring packets arriving at and departing from the ring buffer RB after a local access request has been issued at time $t=0$ which is granted after an average delay $W_a$.

$$n_r(t) = \begin{cases} A_{RB}(t) & ; 0 < t < W_a + S \\ A_{RB}(t) - D_{RB}(t) & ; W_a + S < t < T_q \\ 0 & ; T_q < t < \frac{1}{\lambda_a} \end{cases}$$ (4.4.14)

(note that at time $\frac{1}{\lambda_a}$ the next local packet arrives and initiates the next rush-hour). The average number of ring packets in the RB is therefore given by

$$N_r = \frac{1}{\lambda_a} \int_0^{1/\lambda_a} n_r(t) \, dt$$ (4.4.15)

where the integral is simply equal to the shaded area in Fig. 4.4.1. Thus,

$$N_r = \lambda_a \frac{A_{RB}(T_q)(W_a + S)}{2}$$ (4.4.16)
Substituting (4.4.10), (4.4.12) and (4.4.13) into (4.4.16) and using Little's result (KLEI75) that
\[ W_r = \frac{N_r}{\lambda_r} \quad (4.4.17) \]
we finally obtain the average waiting time of ring packets in the RB from (4.4.16) and (4.4.17) as
\[ W_r = \frac{\lambda_a (W_a + S)^2}{2(1 - \lambda_a S(m^2 - 1))} \quad (4.4.18) \]
where the approximation sign is now used to emphasize that the exactness of (4.4.11)-(4.4.17) depended on the assumption that the RB began to fill at the average rate when the local packet arrived at time t=0. The approximate average total waiting time in all buffers between sender and receiver can now be obtained by substituting (4.4.9) and (4.4.18) into
\[ W_{tot} = W_a + \left( \frac{m}{2} - 1 \right) W_r \quad (4.4.19) \]
as in the buffer insertion ring with Poisson arrivals.

As an example, we consider a packet voice system with voice sampling rate \( r_v \) bit/sec and ring transmission rate \( r_t \) bit/sec. The packet time length and packet arrival rate are therefore given by
\[ S = \frac{8N_0}{r_t} \quad (4.4.20) \]
and
\[ \lambda_a = \frac{r_v}{8N_0} \quad (4.4.21) \]
respectively. When there are \( m/2 \) full-duplex voice conversations active between randomly chosen pairs of stations, the utilization is given by
\[ \rho = \frac{m r_v}{r_t} \quad (4.4.22) \]
Fig. 4.4.2 is a plot of the average total waiting time \( W_{tot} \) as given by (4.4.19) for a packet voice system with \( N_0 = 6 \) bytes per packet, \( r_v = 64 \text{ kbit/sec} \) and \( r_t = 2.13 \text{ Mbit/sec} \) (note that \( W_{tot} \) does not include
Fig. 4.4.2 95 per cent confidence intervals of the average total waiting time from the simulation of a packet voice ring system with $N_0 = 6$ bytes per packet, voice digitization rate $r_v = 64$ kbit/sec and ring transmission rate $r_t = 2.13$ Mbit/sec. The curves are estimates obtained from (4.4.9), (4.4.18), and (4.4.19) for the case of periodic arrivals, and (4.2.7) for Poisson arrivals.
propagation delays and constant processing delays in the address comparison registers, etc.).

In Fig. 4.4.2, the approximation to $W_{tot}$ obtained from (4.4.19) is plotted together with the confidence intervals obtained from the simulation of such a packet voice system with the same parameters. The results in Fig. 4.4.2 confirm that our approximation is very close to reality. The upper curve in Fig. 4.4.2 is that for Poisson arrivals of constant length packets (eqn. 4.2.7). Comparison of this curve with that for periodic arrivals shows that the waiting time with periodic arrivals is considerably less than with Poisson arrivals, as was anticipated.

In practical communication systems, one would often have mixed traffic, i.e., some voice traffic with periodic arrivals interspersed with data traffic having randomly spaced intervals between arrivals. In such a case, the average total waiting time can be anticipated to lie between the approximations for the model for Poisson arrivals (4.2.7) and that for periodic arrivals (4.4.19).

4.5 Conclusions

We have given approximate analyses of the packet waiting times in a general buffer insertion ring where all stations can transmit to each of the other stations, for both the case when the packet arrival processes are Poisson and the case when the packet arrival processes are periodic.

For the case where all packet arrival processes are Poisson and for low and moderate utilizations, the estimates obtained from the approximate analysis were shown to be in excellent agreement with the waiting times as determined by simulation. An intuitive argument was used to explain why the estimates are somewhat too large for higher utilizations. It was further confirmed by the simulations that, as the approximate analysis assumes,

1) the average total waiting time is the same (at least within the observation accuracy) for LP and RP,

and that, as the approximate analysis predicts,
2) the average waiting time for exponentially distributed packet length is approximately twice the average total waiting time for constant packet length.

The approximate analysis for the case when the local packet arrival processes are all periodic with random phase shifts between the individual packet streams was also found to give estimates of delay that are in excellent agreement with the waiting time as measured by simulations. This arrival model, which would be appropriate in packetized voice systems, was shown to result in considerably smaller delays than for the Poisson arrival model for the same utilization.
4.6 References


5. ETHERRING - A RING WITH A CSMA PROTOCOL

5.1 Introduction

We propose a new type of ring transport system for local area communication systems that we shall call Etherring. The name Etherring was chosen because the proposed system combines the features of Ethernet (MET76) with those of a unidirectional ring. Etherring is a random access (see, e.g., SCHW77) ring with listening before and during transmission (similar to that in a carrier sense multiple access (CSMA, see e.g., KLEI75, KLEI76) bus or broadcast channel) and with packet chaining. The concept of packet chaining, which depends on the orderly flow of information around the ring, appears to be new.

In a CSMA bus there is a non-zero probability of collision after each packet transmission because all ready stations transmit with probability p when they see the bus go idle. Etherring uses packet chaining to prevent such collisions. A transmitting station always concludes its transmission with an end-of-transmission (EOT) flag. This EOT flag then propagates around the ring, similar to a token. The first ready station (if any) to receive the EOT flag changes it to an end-of-packet (EOP) flag and immediately thereafter starts to transmit its own packet. This collision-free mode of operation will be called the packet chaining mode. Collisions can occur only when the ring has gone completely idle, i.e., after the chain of successive transmissions has been disrupted because the EOT flag propagated completely around the ring after the last transmission without finding a ready station. In this case, Etherring enters the collision mode.

In the collision mode, each station transmits as soon as it becomes ready. A collision occurs if one or more other stations become ready and start to transmit before they begin to receive the transmission of the first station that became ready. Collisions are detected by the transmitting stations when the header of the first packet received after the station begins transmission from the idle mode does not coincide with the header of its own transmitted packet. The stations involved in a collision then make independent retransmission attempts at randomly chosen time instants until eventually one packet is transmitted successfully, thereby bringing the ring back to the packet chaining mode. All of the remaining packets that were involved in the collision are then
transmitted sequentially without risk of further collisions. In fact, the ring does not return to the collision mode again until there is no ready station when at the end of a packet transmission the EOT flag propagates around the ring.

In a CSMA bus special "time-out" provisions or use of more sophisticated retransmission protocols such as the Capetanakis tree algorithm (CAP79, CAP79A) are required to prevent possible deadlock when more and more stations attempt to transmit during a collision resolution period. Etherring avoids this instability problem in a simple way: Because no new transmission attempts are permitted until at least one of the packets in the initial collision has been transmitted successfully, the pool of retransmitting stations cannot enlarge. There is no possibility of retransmission deadlock and the maximum stable utilization is almost 1, even for short packets.

Another desirable feature of Etherring is that transient bit errors in the packet flags have the same effect as collisions and are thus dealt with automatically. Etherring thus has an inherent reliability advantage over token passing rings (TPR's, see e.g., CLAR78, KAY72, KUE79, PEN79). Under all channel error conditions in the ring, recovery is possible with a simple default feature; detection of lost or duplicated tokens is not required.

In section 5.2 we give a detailed description of Etherring and its behaviour. In section 5.3 we develop an analytical model which will then be used to calculate the average fraction of time that is wasted to resolve collisions, the retransmission overhead, and the average transmission delay. A comparison is given between the average transmission delay in the Etherring, the TPR, and the buffer insertion ring (BIR, see chapters 3 and 4).

5.2 Operating principles

Etherring is a ring transport system (Fig. 5.2.1) for packets with a packet format as depicted in Fig. 5.2.2(a). All packets begin with a start-of-packet (SOP) bit (Fig. 5.2.2(b)), followed by a packet header, and end either with an end-of-packet (EOP) or end-of-transmission (EOT) flag. An EOP indicates that another packet follows immediately behind this one on the ring whereas an EOT indicates that the ring is free thereafter so that a new packet can be transmitted if the receiving
station has one waiting. The EOT and EOP flags must be of the same length and must differ only in the last bit, for reasons that will be shown in section 5.2.1. The packet origin address (OA) and destination address (DA) are coded, preferably in fields of fixed length, immediately following the SOP. After this preamble, the data packet is transmitted together with some form of redundancy check (RC). Bit stuffing (ART72, NIE73) is used to eliminate EOP and EOT bit patterns from the data even when the packet length is fixed (for reasons that will be given later).

In fig. 5.2.2(b) we give an example of possible EOT and EOP flags. The length of the EOT and EOP flags is chosen so as to produce an acceptably small expansion of the data because of bit stuffing.

Etherring uses a single unidirectional ring network (Fig. 5.2.1) to transmit packets between user stations 1, 2, ..., m. We assume that when power is switched on, or after a temporary ring failure, a special start-up protocol takes care that:
<table>
<thead>
<tr>
<th>SOP</th>
<th>OA</th>
<th>DA</th>
<th>DATA</th>
<th>RC</th>
<th>EOP/EOT</th>
</tr>
</thead>
</table>

- **SOP**: start of packet
- **OA**: origin address
- **DA**: destination address
- **RC**: redundancy check
- **EOP**: end of packet
- **EOT**: end of transmission

**Fig. 5.2.2(a)** Ethering packet format.

**Fig. 5.2.2(b)** Ethering packet header format.
Exactly one station is designated monitor. This station then provides the clock for all other stations. In principle, any station could be designated monitor but in Fig. 5.2.1 and in the following discussion we assume, without loss of generality, that station 1 has been designated monitor.

When all stations are synchronized, the designated monitor sends a special message to inform all other stations that they are now allowed to transmit packets according to the normal Ethernning protocol which will be described presently, after which message the monitor transmits all zeroes to maintain bit synchronization on the ring. No station begins to transmit packets before it has received this initial message authorizing such transmissions.

Fig. 5.2.3(a) is a block diagram of an Ethernning station that recovers the clock from the received data. Fig. 5.2.3(b) is an Ethernning station state transition diagram in which the labels on the arrows between the states indicate the events that cause these transitions. After the start-up phase has terminated, all stations are initially in the "relay idle" state and all transmission requests (TR's) are granted immediately when they are issued. Collisions can therefore occur when two or more stations begin to transmit nearly simultaneously; Ethernning is thus said to be in the collision mode. A transit from the "relay idle" state occurs either when the station watchdog (WD, Fig. 5.2.3(a)) detects an SOP bit in the incoming ring data, or when a local TR is submitted to the transmission controller (Fig. 5.2.3(a)). In the latter case, the TR is granted immediately, the station enters the "transmit" state and begins to transmit the packet that caused the TR. Simultaneously, the WD begins to monitor the data that is received from the ring input. The first non-zero bit encountered in the received data is assumed to be the SOP of the packet that returns after a round trip. We have to distinguish between two cases:

1. The DA field in the received packet coincides with the transmitter's own station address (SA). This indicates a successful transmission and causes Ethernning to enter the chaining mode which will be described in section 5.2.1
Fig. 5.2.3(a) Etherring station block diagram.
2) The OA field in the received packet is different from the transmitter's own SA. This indicates either a collision or a transmission error. The collision resolution mechanism will be described in section 5.2.2.

5.2.1 Chaining mode

When the OA field in the received packet coincides with the transmitter's own SA, the transmission attempt has been successful. This occurs when, in the vulnerable period between the start of the initial transmission and the time when each other station sees this transmission, each other station has not made its own transmission attempt. Now all other stations have seen the SOP and have made the transition to the "relay busy" state. They know that they are actively relaying a packet and hence will refrain from granting a local TR until the relaying in progress is terminated with an EOT. The ring is now in the collision-free packet chaining mode. The transmitting station remains in the "transmit" state until it receives either its own EOT or the SOP following an EOP (Note: after the EOT has been transmitted, the station continues to send 0's to keep the other stations synchronized). An EOP indicates that another station had a packet waiting which was then inserted immediately behind the first packet and which now must be relayed by the initial transmitting station. Therefore, this station now also enters the "relay busy" state and refrains from granting further TR's.

When a station in the "relay busy" state has a TR pending when the EOT is received, packet chaining occurs. The last bit of the EOT flag, which we shall call the chaining bit, is changed from 0 to 1, thereby converting the EOT flag to the EOP flag (in Fig. 5.2.3(a) this function is performed by the box labelled EOT=EOP). Then the packet that caused the TR is transmitted without any further delay, immediately behind the preceding packet. Thus, each station induces only a one bit delay on the relayed packet; this is the reason that we demanded that the EOP and EOT flags differ in only the last bit. Note that, as utilization increases, Etherrning spends more and more time in the collision-free packet chaining mode and less time (or, equivalently, channel capacity) is wasted in resolving collisions. When Etherrning is in the packet chaining mode, its performance is very similar to that of a token passing ring (TPR, see e.g., CLAR78, KAY72, KUE79, PEN79).
Fig. 5.2.3(b) Ethernet station state transition diagram.
When a station in the "relay busy" state has no TR pending when the EOT is received (as indicated by TR in Fig. 5.2.3(b)), the station transits to the state "wait m+1 bits". When a station receives an SOP in this state, this indicates that another station farther downstream had a TR pending when the EOT came by so that the waiting packet was chained. Therefore, when this SOP is received, the station returns to the "relay busy" state and further refrains from granting its own TR's. If, however, the station receives no SOP in a time window of length m+1 bit-times (where bit-time is the time required to transmit one bit at a station), this indicates that no other station downstream including the originator of the previous packet had a TR pending when the EOT initially came by. The ring has gone idle, and the station thus transits to the state "relay idle" where its own TR's are granted. Et-herring is again in the collision mode.

In the limit, when the traffic is so heavy that all stations always have pending TR's when they receive an EOT, the only wasted time for each packet is one bit-time used to assign the channel to station i+1 when station i has finished transmitting its packet. Thus, when the packet length is restricted to at most $T_{max}$ bits, the Etherrino protocol is in the case of high utilization fair in the sense that no station can seize the channel permanently for periods longer than $T_{max}$ bit-times. For the case where all packets are of the same fixed length $T$, all stations can transmit with the same (reduced) rate even when the offered load exceeds the ring capacity, although of course the queues at each station then grow without bound.

Ring capacity $C$ is defined as the upper limit of the number of data bits per channel bit time that can be carried such that the average delay is finite. When the rate that data bits are offered to the ring is below capacity, packets are transmitted, on the average, faster than they are generated. Therefore, queue sizes and average delay are finite. $C$ is a function of how traffic is distributed among the stations, i.e., a function of

$$
\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)
$$

$$
= R\Phi
$$

(5.2.1)
where

\[ R = \sum_{i=1}^{m} \lambda_i \]  \hspace{1cm} (5.2.2)

is the total offered traffic; \( \lambda_i \), \( 1 \leq i \leq m \), is the fraction of total traffic originating at station \( i \), and \( \phi \) is the traffic distribution vector. When we neglect the internal overhead due to EOT or EOP flags and address fields, and when all packets are of total fixed length \( T \) bits, the Ethernet capacity is bounded by

\[ \frac{T}{T+m} \leq C(\phi) \leq \frac{T}{T+1} \]  \hspace{1cm} (5.2.3)

Equality on the right is attained when

\[ \phi_i = \frac{1}{m}, \hspace{0.5cm} 1 \leq i \leq m, \]  \hspace{1cm} (5.2.4)

as this condition implies that when the network becomes congested each station will have a packet to send every time it is given permission to send. The essence of the argument is perhaps best seen by considering the specific case of an Ethernet with \( m=3 \) stations and packets of length \( T=3 \) bits. We assume that at time \( t=0 \) within the congested period, station 1 begins to send a packet, denoted \( 1(1) \) in Fig. 5.2.4, over its outgoing link. Station 2 begins to receive this packet starting at \( t=1 \) and immediately begins to relay this packet over its outgoing link. At \( t=4 \), station 2 begins to send its own packet, denoted \( 2(1) \) in Fig. 5.2.4, over its outgoing link; this packet is chained to packet \( 1(1) \). Fig. 5.2.4(a) shows the times during which packets are transmitted over the originating station's outgoing link. Fig. 5.2.4(b) shows which packets each of the stations is transmitting or relaying over its outgoing link at each time instant. The shaded areas in Fig. 5.2.4 are the intervals when the outgoing link is idle -- during these intervals the station is receiving a packet that it had itself originated. Note that whereas the packets are initially transmitted (as shown in Fig. 5.2.4(a)) periodically at the rate of 1 packet every \( T+4 \) time units, the station transmitters (as shown in Fig. 5.2.4(b)) are busy for \( mT=9 \) time units after which they go idle for \( T \) time units. On the average, it takes \( mT+m=12 \) bit-times to transmit \( mT=9 \) bits, i.e., the capacity is given by

\[ C = \frac{mT}{mT+m} = \frac{T}{T+1} \]  \hspace{1cm} (5.2.5)
i.e., \( C = \frac{3}{4} \). In fact, (5.2.5) holds true for all \( m \) and \( T \) as follows from an obvious generalization of the above argument.

Equality on the left in (5.2.3) is attained when for some \( i \)

\[
\Phi_i = 1, \\
\Phi_j = 0, \quad 1 \leq j \leq m, \quad j \neq i, 
\]

(5.2.6)
i.e., when in a busy period there is only one station transmitting and all other stations are idle. This is proved simply by noting that the single transmitting station receives its own EOT back \( m \) bit times after it has been sent. This causes the station to transit to the "relay idle" state where its next TR is granted immediately. Thus, the station wastes \( m \) bit times for every \( T \) bit times of transmission and the left equality of (5.2.3) follows.
5.2.2 Collision resolution

When the OA field in the received packet does not coincide with the station's own SA, a collision has been detected. At least one other station has made a transmission attempt in the vulnerable period so that none of these packets is transmitted successfully. Immediately after the transmission controller detects the collision, the transmitting station aborts the transmission, transmits an EOP and enters the "abort" state (Fig. 5.2.3(b)). The station transmits 0's until the EOP from the upstream colliding station is received. Then the station enters the "wait" state (Note: the time instants when the colliding stations enter the "wait" state are not necessarily the same; the differences are, however, small and will be neglected in the analysis in section 5.3). The whole of its own collided packet is kept in the transmission buffer (Fig. 5.2.3(a)) of each colliding station for later retransmission. If the intended destination lies before the next colliding station downstream, that destination station will receive the SOP, a partial packet addressed to itself and the EOP; the RC (or possibly some form of byte count built into the header) indicates to this station that the received data is not valid and should therefore be discarded.

All stations that are not involved in the collision are now in the "relay busy" state and will therefore refrain from granting TR's until they receive an EOT flag. All collided stations make independent retransmission attempts at randomly selected time instants. Each station waits on the average 1/\lambda_r bit-times between two retransmission attempts either by choosing the next transmission time randomly and uniformly between 1 and 2/\lambda_r-1 or by, independently with probability \lambda_r, choosing to retransmit at each successive time instant. In either case, the total retransmission rate generated by k collided stations is k\lambda_r. The choice of the optimal value \lambda_r for a minimum average collision resolution time is discussed in section 5.3.1.

When a station makes a retransmission attempt it returns to the "transmit" state. Again, the first non-zero bit encountered is assumed to be the SOP of the station's own packet that returns after a round trip. If the OA field in the received packet coincides with the station's own SA, the retransmission attempt has been successful and the collision is resolved. When the transmission is terminated with an EOT, the first downstream station with a pending TR will chain its packet, etc. The ring is now in the packet chaining mode and all waiting packets are transmitted sequentially as in a TPR.
If, however, a retransmission attempt fails because two or more of the collided stations begin to retransmit almost simultaneously so that they collide again, the transmissions are again aborted and the colliding stations return first to the "abort" and then to the "wait" state, as after the initial collision. At some later time they will make a new retransmission attempt.

A very desirable feature of Etherring is that those stations that were involved in the initial collision but not in the collision caused by the failed retransmission attempt were sent to the "relay busy" state when they received the SOP from the failed retransmission attempt. Thus, those stations are now no longer in the pool of stations liable to suffer further collisions. The eliminated stations have still a TR pending but they will wait to grant it until they receive an EOT at the end of a successful transmission when they chain their own packet behind the preceding one. Therefore, the number of stations involved in a collision will, on the average, decrease rapidly and collisions are resolved quickly. A quantitative analysis in section 5.3 will show that the amount of time wasted to resolve collisions is in most cases negligible.

5.2.3 Transmission errors

We conclude section 5.2 with a brief discussion of the effect of random bit errors in the SOP bit or in the packet flags. When an SOP bit is corrupted, the packet will not be transmitted correctly, but a spurious SOP bit will with high probability appear soon. The transmitting station will, however, in any case diagnose this situation as a collision when it fails to receive its own address and will then interrupt transmission and retry at some later time. No special precautions are therefore required to recover from such an error.

Recovery from error conditions caused by corrupted EOT or EOP flags is achieved by a simple default feature. Each user station has a received bit counter (RBC, Fig. 5.2.3(a)) which is reset to zero whenever an SOP or an EOP flag is received and which is stopped in the "relay idle" state. This counter can, in the absence of errors in the EOP or EOT flags, only in the exceptional case described below reach a number $N_{\text{max}}$ where $N_{\text{max}} = \frac{T_{\text{max}}}{0.5}$ and where $T_{\text{max}}$ is the maximum allowed packet length (including the SOP bit and the EOP flag) specified for the system. Thus when this counter reaches $N_{\text{max}}$, excluding the exceptional case described
below, it is certain that an EOP or EOT flag has been corrupted. A transition to the "relay idle" state is enforced to prevent the station from being blocked, i.e., not being allowed to transmit forever because an EOT flag has been missed. Note that this default feature also provides recovery from the error condition where a station erroneously makes a transition from the "relay idle" state to the "relay busy" state because of a fallacious SOP generated by a random bit error.

Very rarely it may be that the interval between two retransmission attempts is so exceptionally long that the RBC reaches $N_{\text{max}}$. Thus, a transition to the "relay idle" state may be enforced (although with very low probability when $N_{\text{max}}$ is chosen sufficiently large), erroneously in the absence of flag errors. This may cause additional collisions and, therefore, increase the time until the packet chaining mode is entered, but otherwise the normal Ethernet operation is not affected.

5.3 Performance analysis

An efficient access protocol should assign the shared communication channel quickly to exactly one ready user whenever the channel is free and there are one or more ready users who have packets to send. The more time wasted for this task, the less time is left for the transmission of useful data (i.e., the useful capacity is reduced), and the longer is the average message delay. In order to show that Ethernet wastes very little time to assign the channel to one particular user, and that the average message delay is in most cases smaller than in a TPR, we first consider the Ethernet state transition diagram, Fig. 5.3.1.

When the two simplifying but conservative assumptions stated below are invoked, Ethernet, considered as a single system, can be viewed as a Markov system which is always in one of the following seven states:

1) Idle

All stations are in the "relay idle" state (Fig. 5.2.3(b)). No transmission takes place and none of the stations has packets ready to transmit.
2) Collision vulnerable (A)

A station in the "relay idle" state had a TR that was granted immediately. With probability $\alpha$ (which we will calculate in section 5.3.3), no other station collides and Etherring transits to state 4,
"send rest of packet". With probability $1-a$, however, at least one other station also grants a TR that causes a collision. The collided stations abort their transmissions and enter the "wait" state (fig. 5.2.3(b)) when they receive the EOP flag from the upstream colliding station. Here we invoke the first of our simplifying assumptions: we assume that all colliding stations enter the "wait" state simultaneously $2m$ bit-times after the initial transmission started, and that this causes Ethernet, considered as a single system, to transit to state 3, "retransmission". With respect to the time wasted to resolve collisions this is a conservative assumption because the actual earlier entry into the "wait" state by some transmitters gives an increased probability of successful retransmission until that point where all colliding transmitters are in the "wait" state.

3) Retransmission

Ethernet was in the "idle" state when two or more stations began to transmit almost simultaneously so that their packets collided. All transmissions were aborted and all collided stations entered the "wait" state. The simplifying assumption stated above implies that all collided stations simultaneously begin to select random time instants for their retransmission attempts. The average period between two retransmission attempts by a single station is given by $1/l_r$. When a retransmission attempt is successful, i.e., when there is no new collision, Ethernet transits from state 3 to state 4, "send rest of packet". When a retransmission attempt ends, however, in a new collision, the transmissions are again aborted and the stations return to the "wait" state. Ethernet as a system remains in the "retransmission" state but only those stations that collided again remain in the pool of stations liable to further collisions. All other stations that were involved in the previous collision but not in the last collision were sent to the "relay busy" state when they received the SOP bit during the failed retransmission attempt. As a consequence, the total rate of retransmission attempts

(*) $2m$ is for all cases of interest an upper bound on the average time before all transmitters enter the "wait" state after a collision.
is reduced with high probability after each retransmission failure although the retransmission rate \( \lambda_r \) of each station in the "wait" state remains constant.

4) Send rest of packet

One station is in the "transmit" state. The channel has been assigned successfully to exactly one station, either after the system has been idle or after a collision has been resolved. The collision vulnerable period of length \( m \) bit-times is over and the station sends the rest of its packet. When the rest of the packet has been sent, Etherring transits to state 5, "assign".

5) Assign

A packet has been transmitted successfully and terminated with an EOT flag. The EOT propagates around the ring until it finds a station with a pending TR or until it completes its round trip back to the originating station. In the former case Etherring transits from state 5 to state 6, "chaining". The probability \( \hat{S} \), which we shall call the chaining probability, of such a transition depends on how we got to the "assign" state; for example, \( \hat{S} \) is larger when the "assign" state is entered after a successful retransmission than when entered after a successful initial transmission. These fluctuations are, however, unimportant so that we shall assume that \( \hat{S} \) is constant. This is our second simplifying assumption that is required to model Etherring as a Markov system. In section 5.3.3 we will calculate a lower bound on the average value of the chaining probability. Using this bound on the value of \( \hat{S} \) gives conservative estimates of the average time wasted to resolve collisions.

With probability \( 1-\hat{S} \) the EOT flag returns to the station where it was generated without finding a station with a pending TR. With probability \( (1-\hat{S})\gamma \) Etherring transits to state 7, "collision vulnerable (B)" where \( \gamma \) is the probability of the event that at least one station had a TR in the interval between the time when the EOT passed by and the time when the EOT returned to the station where it was generated. In all other cases, i.e., with probability \( (1-\hat{S})(1-\gamma) \), Etherring transits to state 1, "idle". Conservative estimates of the transition probabilities \( \hat{S} \) and \( \gamma \) will be calculated in section 5.3.3.
6) Chaining

An EOT flag that terminates a successful transmission has found a station with a pending TR. The EOT flag is converted to an EOP flag and the packet is chained immediately behind the preceding packet. There is no risk of a collision. When the transmission is over, Etherneting returns to state 5, "assign", with a new EOT flag propagating around the ring.

7) Collision vulnerable (B)

When the EOT flag propagated around the ring at the end of a transmission, no station was found with a pending TR. At least one station had, however, a TR in the interval between the time when it let the EOT go by unused and the time when the EOT returned to the station where it was generated. Etherneting transits from state 5 to state 7 when the first of these stations begins to transmit (i.e., m+1 bit-times after the EOT passed by that station. With probability $\delta$ (which we will calculate in section 5.3.3) no other station collides and Etherneting transits to state 4, "send rest of packet". With probability $1-\delta$, however, at least one other station also grants a TR that causes a collision. When they receive the EOP flag from the upstream colliding station, the collided stations abort their transmissions. We make the same simplifying but conservative assumption as in the discussion of the transition from state 2 to state 3: we assume that all collided stations simultaneously enter the "wait" state, and that this causes Ethernet, considered as a single system, to transit from state 7 to state 3.

Etherneting state transitions are governed by the state transition equations (see Fig. 5.3.1)

\[ \pi_1 = (1-\beta)(1-\gamma)^{-5} \quad (5.3.1) \]
\[ \pi_2 = \pi_1 \quad (5.3.2) \]
\[ \pi_3 = (1-\alpha)\pi_2 + (1-\delta)\pi_7 \quad (5.3.3) \]
\[ \pi_4 = \alpha\pi_2 + \pi_3 + \delta\pi_7 \quad (5.3.4) \]
\[ \pi_5 = \pi_4 + \pi_6 \]  
\[ \pi_6 = \beta \pi_5 \]  
\[ \pi_7 = (1-\beta)\gamma \pi_5 . \]

From these equations and the condition that
\[ \sum_{i=1}^{7} \pi_i = 1, \]
the state probabilities are obtained as
\[ \pi_1 = ab \]  
\[ \pi_2 = ab \]  
\[ \pi_3 = \{a(l-a)+\gamma(l-\beta)(1-\delta)\}b \]  
\[ \pi_4 = \{a+\gamma(l-\beta)\}b \]  
\[ \pi_5 = b \]  
\[ \pi_6 = \beta b \]  
\[ \pi_7 = \{\gamma(l-\delta)\}b \]

where \( a \) is defined as
\[ a = (l-\beta)(1-\gamma) \]

and where \( b \) is defined as
\[ b = \frac{1}{a(4-a)+1+\beta+\gamma(l-\beta)(3-\delta)} . \]
In the following sections we present a quantitative analysis of the state transition probabilities, and the average sojourns in states 1 to 7. All calculations are such that they produce worst case estimates of the average time required to resolve collisions and, thus, of the average delay.

Utilization is in the following defined as

$$\rho = \lambda T$$  \hspace{1cm} (5.3.18)

where $T$ is the number of bits per packet (all packets are assumed to be of the same length) and where $\lambda$ is the total packet arrival rate from all stations. It is assumed that each individual station generates traffic with the same rate

$$\lambda' = \frac{\lambda}{m}$$  \hspace{1cm} (5.3.19)

and that the packet arrival processes at the individual stations are all independent Poisson processes.

5.3.1 Average time to resolve a collision

The average time required to resolve a collision is given by

$$m \quad r = \sum_{i=2}^{m} r(i)c(i)$$  \hspace{1cm} (5.3.20)

where $r(i)$ is the average time required to resolve a collision among $i$ stations and where $c(i)$ is the probability that there are $i$ stations involved in the initial collision.

Under worst case assumptions, the number in a collision is equal to the number of packets generated by all stations in an interval of length $2m$ bit-times, irrespective of where the collision occurs (either in state 2 or state 7). The probability distribution of this number is the binomial distribution given by

$$c'(k) = \binom{m}{k} \cdot p_{2m}^k (1-p_{2m})^{m-k}$$  \hspace{1cm} (5.3.21)

for $0 \leq k \leq m$ where
\[ P_{2m} = 1 - \exp(-2\lambda) \]  

(5.3.22)

is the probability that a given station has a packet in an interval of length \(2m\). Provided that there is a collision, the probability distribution of the number involved is therefore given by

\[ c(k) = \frac{c'(k)}{1-c'(0)-c'(1)} \]  

(5.3.23)

for \(2 < k < m\).

Now we proceed to derive \(r(2)\), the average time required to resolve a collision among \(i\) stations. First we consider the case when there are only two stations involved in the collision. By assumption, both stations make their retransmission attempts at randomly chosen time instants with an average time of \(1/\lambda_r\) bit-times between the attempts of one station; both stations have the same time origin. Therefore, Ethernet stays, on the average, in state "two in collision" (Fig. 5.3.2) for a period of length \(1/2\lambda_r\) before one station begins to send. If the second station refrains from making another retransmission attempt in a time window of length \(m\) bit-times the collision is resolved (Note: this is a conservative assumption for the worst case where the second station is the farthest neighbour of the first station). This worst-case event happens with probability \(\exp(-\lambda_r m)\). With probability \(1-\exp(-\lambda_r m)\), the two stations collide again and return to state "two in collision" until the next retransmission attempt. The average number of retransmission attempts required to resolve this worst-case collision is thus equal to the average number of trials until the first success in a series of Bernoulli trials with success probability \(\exp(-\lambda_r m)\) and is thus given by

\[ \frac{1}{\exp(-\lambda_r m)} = \exp(\lambda r m) \]  

(5.3.24)

When a retransmission attempt fails, it takes, in the worst case, \(2m\) bit-times from the time the first station begins to send until both stations are again in the "wait" state. An upper bound on the average time required for each attempt is therefore \((1/2\lambda_r) + 2m\) bit-times. Multiplying by the upper bound of (5.3.24) on the average number of retransmissions, we obtain

\[ r(2) \leq (\frac{1}{2\lambda_r} + 2m)(\exp(\lambda_r m) - 1) + \frac{1}{2\lambda_r} \]  

(5.3.25)
as an upper bound on the average time to resolve a collision between two transmitters.

In the case when there are three stations involved in the initial collision when Etherring enters state 3, it is convenient to describe the worst case operation of the retransmission protocol as follows. When the first collided station begins to send, the other two collided stations make independent Bernoulli trials. Within a vulnerable period of length \( m \) bit-times each begins to retransmit with probability

\[ P_r = 1 - \exp(-\lambda_r m) \]  

(5.3.26)

The probabilities of 0, 1, and 2 further retransmissions are thus given by the binomial distribution

\[ b_3(k) = \binom{2}{k} P_r^k (1-P_r)^{2-k} \]  

(5.3.27)

for \( k = 0, 1, \) and 2, respectively. \( b_3(2) \) is the probability that Etherring returns to the state "three in a collision" (see Fig. 5.3.3). Because, in the worst case, each attempt to leave this state takes on the average \( (1/3\lambda_r + 2m) \) bit-times, it follows that

\[ t_3 \leq \left( \frac{1}{3\lambda_r} + 2m \right) \left( 1 + \frac{b_3(2)}{1-b_3(2)} \right) \]  

(5.3.28)

is an upper bound on the average time in this state. When Etherring leaves state "three in collision" it transits with probability

\[ 1 - \exp(-\lambda_r m) \]

\[ \text{Fig. 5.3.2 Collision resolution between two collided stations: state transition diagram.} \]
\[ p(3,2) = \frac{b_3(1)}{b_3(0) + b_3(1)} \]  
(5.3.29)

to state "two in collision" and with probability
\[ p(3,1) = \frac{b_3(0)}{b_3(0) + b_3(1)} \]  
(5.3.30)

to state "collision resolved" (Fig. 5.3.3). Therefore,
\[ r(3) \leq t_3 + p(3,2)r(2) \]  
(5.3.31)

is an upper bound on the average time to resolve a collision when initially three stations are involved.

Generalization of these arguments shows that, in the rare event of a collision among a large number of stations, collisions thin out quickly.

Fig. 5.3.3 Collision resolution between three collided stations: state transition diagram.
An upper bound on the average time to resolve a collision among \( i \) stations can be obtained recursively from
\[
r(i) \leq t_i + p(i,i-1)r(i-1) + p(i,i-2)r(i-2) + \ldots + p(i,2)r(2),
\]
\[
i \leq 3 \leq m,
\]
\[
t_i \leq \frac{l \lambda}{1-b_i(i-1)}
\]
is an upper bound on the average time in state "i in collision" and where
\[
p(i,k) = \frac{b_i(k-1)}{b_i(0)+b_i(1)+\ldots+b_i(i-2)}
\]
\( 2 \leq k \leq i-1 \), is the probability that Etherring transits from state "i in collision" to state "k in collision". The probabilities \( b_i(j) \) are given by the binomial distribution
\[
b_i(j) = \binom{i-1}{j}p_i^j(1-p_i^{i-1-j})
\]
for \( 0 \leq j \leq i-1 \), where \( p_i \) is given by (5.3.26). The recursion starts with \( r(2) \) obtained from (5.3.25). Substitution of (5.3.23) and (5.3.32) into (5.3.20) finally gives an upper bound on the desired average time to resolve a collision.

The average time \( r \) required to resolve collisions depends on the average interval \( 1/\lambda_r \) between the retransmission attempts of a collided station. If \( 1/\lambda_r \) is too short, the packets collide again with high probability, and the collision resolution time is longer than necessary. When, on the other hand, \( 1/\lambda_r \) is chosen too large, the collision is resolved with high probability after the first retransmission attempt, but the collision resolution time is nevertheless large because no one sends for a long time. Thus, there is some optimal choice of \( 1/\lambda_r \) such that the collision is resolved in the minimum possible time.

The curves depicted in Fig. 5.3.4 were obtained from eqn. (5.3.20). They illustrate that this optimal choice of \( 1/\lambda_r \) depends on the number of stations. As a rule of thumb, inferred from Fig. 5.3.4, we propose
\[
\frac{1}{\lambda_r} = 2m
\]
The curves in Fig. 5.3.4 have been calculated for various values of utilization $p$. For $m=10$ and $m=20$ stations there is, however, no visible difference between the curves. Only for $m=30$ stations and $1/\lambda_r$ extremely small, does the difference between the curves for the utilizations $p=0.1$ and $p=0.99$ become visible. It is perhaps counterintuitive that $r$ is relatively insensitive to the utilization. Further reflection will show, however, that this is a consequence of the fact that, even for utilizations near capacity, there are in the vast majority of cases only two stations involved in a collision.

The average time to resolve a collision is equal to the average sojourn in state 3 (Fig. 5.3.1). In the following section we will calculate the average sojourns in the other states.
5.3.2 Average sojourns in Etherring states

By assumption, packets are generated from \( m \) independent Poisson processes, each with rate \( \lambda' \); the total arrival rate is given by \( \lambda = m \lambda' \). The average residual time to the next packet arrival is therefore \( 1/\lambda \), irrespective of when one begins to measure time. The average sojourn in state 1 (Fig. 5.3.1) is therefore given by

\[
\tau_1 = \frac{1}{\lambda}.
\]

(5.3.37)

To compute the average time Etherring remains in state 2 we have to distinguish between two cases. When there is no collision, Etherring transits to state 4 after \( m \) bit-times but when there is a collision it takes (at most) \( 2m \) bit-times until all collided stations are in the "wait" state (Fig. 5.2.3(b)), which is the time when Etherring transits to state 3. These events occur with probabilities \( \alpha \) and \( 1-\alpha \), respectively. An upper bound on the average sojourn is therefore given by

\[
\tau_2 \leq \alpha m + (1-\alpha)2m \\
\leq m(2-\alpha)
\]

(5.3.38)

The average sojourn in state 3, "retransmission", is the average time required to resolve a collision and is given by

\[
\tau_3 = r
\]

(5.3.39)

where (an upper, conservative bound on) \( r \) has been calculated in section 5.3.1.

The sojourn in state 4, "send rest of packet", is given by

\[
\tau_4 = T-m.
\]

(5.3.40)

To calculate \( \tau_5 \), the average sojourn in the "assign" state, we again invoke the simplifying, conservative assumption that we can approximate the slightly fluctuating chaining probability \( \beta \) by a constant (which we will calculate in section 5.3.3). With probability \( 1-\beta \), Etherring transits from state 5 either to state 1 (after a sojourn of \( m \) bit-times) or to state 7 (after an average sojourn of \( 1.5m \) bit-times). With
probability 6, packet chaining occurs and Etherring transits from state 5 to state 6. Unfortunately, the average time \( \tau_5' \) it takes the EOT flag to propagate from the station where it is generated to the first ready station cannot be computed exactly. This latter time is, however, bounded by

\[ 1 \leq \tau_5' \leq \frac{m}{2} . \]  

(5.3.41)

The lower bound is approached when the traffic is very heavy so that the station next to the one transmitting with high probability has a packet ready to send. The upper bound, on the other hand, is approached when the traffic is very light so that there is with high probability only one station ready to send when Etherring enters the "assign" state. This station is, on the average, close to \( m/2 \) stations away from the station that has just finished transmitting when \( m \), the time required for the EOT flag to traverse the ring, is small compared to the average interval \( 1/\lambda_r \) between packets. The average sojourn in state 5 is therefore approximately given by

\[ \tau_5 = 3\tau_5' + (1-\beta)(1-\gamma)m + 1.5(1-\beta)\gamma m . \]  

(5.3.42)

The average sojourn in state 6 is simply the time to transmit a packet, i.e.

\[ \tau_6 = T. \]  

(5.3.43)

Finally, an argument similar to that used in the calculation of \( \tau_2 \) shows that

\[ \tau_7 \leq \delta m + (1-\delta)2m \]

\[ \leq m(2-\delta) . \]  

(5.3.44)

5.3.3 State transition probabilities

The probability \( \alpha \) that there is no collision when Etherring is in state 2 is the probability of the union of \( m-1 \) independent events that the user \( i \) (\( i=1,2,\ldots,m-1 \)) stations away from the first station to have a
packet has no packet in a window of i bit-times, i.e.

\[ \alpha = \prod_{i=1}^{m-1} \exp\left( - \frac{i\lambda}{m} \right) \quad (5.3.45) \]

(Note that the signal propagation time on the links has been neglected because it is generally much smaller than the delay that is caused by the 1-bit processing delay per station). Evaluation of (5.3.45) yields

\[ \alpha = \exp\left( - \frac{\lambda (m-1)}{2m} \right) \quad (5.3.46) \]

and hence

\[ \alpha = \exp\left( - \frac{\lambda (m-1)}{2} \right) \quad (5.3.47) \]

To calculate a conservative lower bound on the Ethernet average chaining probability \( \beta \), we consider the situation of an M/G/1 single server queueing system with the same arrivals as those generated by the union of all Ethernet stations. If we let \( \beta' \) be the probability that an M/G/1 queue is nonempty when the service of a randomly selected customer is completed, then \( 1-\beta' \) is the probability that the randomly selected packet is the last one in a busy period so the average number of packets in a busy period must be \( 1/(1-\beta') \). But this average number of customers serviced in a busy period is also given for an M/G/1 queue by \( 1/(1-p) \) where \( p = \lambda \tau \) (see KLEI75, p.217). It therefore follows that

\[ \beta' = \rho \quad (5.3.48) \]

In the M/G/1 single server queueing system just considered, service of the next customer (if there is one) begins immediately after the completion of the previous customer's service. In the Ethernet, on the other hand, there is an idle period of at least one bit-time between two successive transmissions. Because new packets could arrive during this idle period, the average Ethernet chaining probability \( \beta \) must therefore be at least as large as \( \beta' \), the probability that the M/G/1 queue is nonempty when the service of a randomly selected customer is completed. From (5.3.48) it therefore follows that

\[ \beta \geq \rho \quad (5.3.49) \]
The probability \( \gamma \) that at least one station has a TR in the interval between the time when the EOT passes by and the time when the EOT returns to the station where it was generated is equal to one minus the probability that none of these stations has a packet in the same interval. The probability of the latter event is, however, equal to \( \alpha \), so that

\[
\gamma = 1 - \alpha \quad (5.3.50)
\]

A conservative estimate of \( \delta \) is given by the probability that no other station has a packet in a time window of length \( 2m \) bit-times. This probability is given by

\[
\delta = \frac{m-1}{m-1} \cdot p^m_s \cdot (1-p_s)^0 \quad (5.3.51)
\]

where

\[
p_s = \exp(-2\lambda) \quad (5.3.52)
\]

is the probability that one station has no packet in \( 2m \) bit-times. Thus,

\[
\delta = \exp(-2\lambda(m-1)) \quad (5.3.53)
\]

5.3.4 Retransmission overhead

For the case of very low utilization, the transition probabilities become in the limit (see Fig. 5.3.1)

\[
\alpha = \delta = 1 \quad , \quad (5.3.54)
\]

and

\[
\beta = \gamma = 0 \quad . \quad (5.3.55)
\]

Substitution of (5.3.54) and (5.3.55) into the state equations (5.3.9)-(5.3.15) shows that for this case

\[
\pi_3 = 0 \quad , \quad (5.3.56)
\]
i.e., there are no collisions (and, therefore, no retransmission overhead), and

$$\pi_6 = \pi_7 = 0$$ \hspace{1cm} (5.3.57)

i.e., Ethernet returns after each transmission to the "idle" state. All TR's are granted immediately, much the same as in a CSMA bus or in Ethernet.

On the other hand, when the traffic is very heavy, the state transition probabilities become

$$\alpha = \delta = 0$$ \hspace{1cm} (5.3.58)

and

$$\beta = \gamma = 1$$ \hspace{1cm} (5.3.59)

and the time to assign the channel to the next ready station is given by

$$\tau_5 = 1$$ \hspace{1cm} (5.3.60)

bit-time. Substitution of (5.3.58)-(5.3.60) into the state equations (5.3.9)-(5.3.15) shows that for this case

$$\pi_1 = \pi_2 = \pi_3 = \pi_4 = \pi_7 = 0$$ \hspace{1cm} (5.3.61)

$$\pi_5 = \pi_6 = \frac{1}{2}$$ \hspace{1cm} (5.3.62)

There is again no retransmission overhead and the average fraction of time Ethernet is transmitting useful information,

$$\pi_2 \gamma_m + \pi_3 \gamma_m + \pi_4 (T-m) + \pi_6 T + \pi_7 \delta_m$$

is found (by substituting (5.3.58)-(5.3.62) into (5.3.63)) to be equal to the upper bound on the Ethernet capacity

$$C = \frac{T}{T+1}$$ \hspace{1cm} (5.3.64)
as was derived in section 5.2 from an analysis of the link busy and idle times.

Etherrina is thus seen to exhibit optimal behaviour both for extremely light and for extremely heavy traffic conditions in the sense that

1) when the utilization is light, all TR's are granted immediately,

2) capacity is close to 1, even for relatively short packet lengths.

Alternative conventional communication systems are optimal either for light traffic (Ethernet) or for heavy traffic (TPR), but it is unusual that a system should be optimal at both extremes of traffic. It now remains to show that Etherrina is also a good choice for moderate utilizations, i.e., when Etherrina occasionally returns to the "idle" state, and when there is a non-zero probability of collisions, 1-\(\alpha\).

The retransmission overhead \(h\) is defined as the ratio between the average wasted time and the average time Etherrina is transmitting successfully, namely

\[
h = \frac{2\pi_m(1-\alpha) + \pi_3(T-m) + 2\pi_7m(1-\delta)}{\pi_2m + \pi_3m + \pi_4(T-m) + \pi_6T + \pi_7m}.
\]

(5.3.65)

Note that all quantities in (5.3.65) were bound such that (5.3.65) is a worst-case estimate of \(h\). Fig. 5.3.5 illustrates that in an Etherrina with \(m=100\) stations and packets of length \(T=1000\) bit-times, \(h\) is less than one per cent over the whole range of utilization. As expected, \(h\) is maximum for moderate utilizations, i.e., when traffic is light enough that Etherrina returns relatively often to the "idle" state while traffic is at the same time heavy enough for collisions to occur relatively frequently.

Fig. 5.3.5 also illustrates that the retransmission overhead depends on the ratio between packet length and the number of stations. For the same utilization and the same number of stations, the packet arrival rate increases when the packet length decreases. As a consequence, there are more collisions, and the retransmission overhead increases; the efficiency is reduced. For example, when the packet length is reduced to 500 bits, the maximum value of \(h\) is approximately 3.5 per cent.
Fig. 5.3.5 Retransmission overhead $h$ vs the utilization $\rho$ (eqn. 5.3.65) for $m=100$ stations, and for packet lengths $T=500$ bits and $T=1000$ bits.

In Fig. 5.3.6, $h$ is plotted vs. $T$ for $\rho=0.5$ and for $m=20, 50$, and 100 stations. This figure illustrates that, in rings with few stations, packets can be shorter without increase in retransmission overhead. When the number of stations is small, the higher packet arrival rate is compensated by a shorter collision vulnerable period. If, however, very short packets are to be transmitted in an Ethernet with a large number of stations, $h$ becomes excessive. For example, Fig. 5.3.6 shows that for $\rho=0.5$, $m=100$, and $T=300$, $h$ is in the order of 10 per cent, which might be considered to be unacceptable. In such a case, a TPR would be better suited.
Fig. 5.3.6 Retransmission overhead h vs the packet length T for m=20, 50 and 100 stations; the utilization is ρ=0.5.

5.3.5 Average transmission delay

When traffic is very light, the average Ethernet packet delay is in the limit for ρ→0 m/2 bit-times shorter than the average packet delay in a TPR with the same number of stations and the same traffic. The reason for this is that the Ethernet transmission starts immediately when the TR is issued whereas in a TPR transmission starts on the average only m/2 bit-times after the TR has been issued, i.e., when the token is received.

When the utilization increases, this Ethernet advantage is, however, more and more negated by the time that is wasted to resolve
collisions. In the following, we assume that the average number of packets serviced in an Ethernet chain of successive transmissions is approximately equal to the average number of packets in a TPR busy period, irrespective of whether the chain starts with a collision resolution period or not. This is a reasonable assumption when the retransmission overhead is small, which has in section 5.3.4 been shown to be the case for typical parameter values.

The probability of the event that Ethernet begins a chain without first having to go through a collision resolution period is given by

\[ p_1 = \frac{\pi_2 \alpha + \pi_7 \delta}{\pi_2 + \pi_7} \]  \hspace{1cm} (5.3.66)

Accordingly, the probability of the complementary event that a chain begins with a collision resolution period is given by

\[ p_2 = \frac{\pi_2 (1-\alpha) + \pi_7 (1-\delta)}{\pi_2 + \pi_7} \]  \hspace{1cm} (5.3.67)

Because all chains are assumed to consist, on the average, of the same number of packets, \( p_1 \) is also approximately equal to the probability that a randomly chosen packet belongs to a chain that begins without collision; \( p_2 \) is approximately equal to the probability that a randomly chosen packet belongs to a chain that begins with a collision.

All packets in a chain that begins without collision have, on the average, a delay that is approximately

\[ d_1 = \frac{m}{2} \]  \hspace{1cm} (5.3.68)

bit-times shorter than the delay of the packets in the same busy period in the TPR. On the other hand, all packets in a chain that began with a collision have, on the average, a delay that is approximately

\[ d_2 = \tau_3 \]  \hspace{1cm} (5.3.69)

bit-times longer than the delay of the packets in the same busy period in the TPR. The average delay of a randomly chosen packet in an Ethernet is therefore approximately given by

\[ D_E = D_T - p_1 d_1 + p_2 d_2 \]  \hspace{1cm} (5.3.70)
where $p_1$, $p_2$, $d_1$, and $d_2$ are as defined in (5.3.66)-(5.3.69).

The average delay from the time a packet is generated at a station in a TPR until the packet is completely received at the destination, $D_T$, has been derived in BUX80 by adapting an earlier discrete-time analysis of a polling system in K0N74. In our own notation,

$$D_T = \frac{\rho T + m(1 - \frac{2}{m'})}{2(1-\rho)} + \frac{m}{2} + T \quad (5.3.71)$$

(recall that we assume packets of constant length). However, it must be emphasized that (5.3.71) is exact only under the assumption that, once a station has the token, it transmits all its packets; only when the station has no more packets waiting does it pass the token to the next station. Thus (5.3.71) is only a lower bound on $D_T$ when the station must relinquish the token after a packet is transmitted as we required in the Etherring chaining mode. (Note: the only reason for this Etherring rule is to prevent the unfairness that arises when a station seizes the channel for long periods at the expense of the other stations with waiting packets). When all stations have the same traffic and when the number of stations is not too small this difference is, however, not significant because then very rarely is more than one packet waiting at a station.

In Fig. 5.3.7(a) the normalized average delays $D_T/T$ and $D_T/T$ obtained from (5.3.70) and (5.3.71), respectively, are plotted for $T=100$ bit/packet and for $m=20$ and $m=100$ stations. This figure illustrates that the normalized average delay in both the Etherring and the TPR are approximately equal over the whole range of utilization for $m=20$, i.e., when the ratio between the number of stations and the packet length is small. When, however, more stations are connected to the ring (e.g., $m=100$ stations) while the packet size is kept constant, Etherring has a smaller average delay than the TPR when the utilization is light but a higher normalized average delay than the TPR when the utilization is heavy.

Fig. 5.3.7(b) shows, however, that it is sufficient to increase the packet length to $T=500$ bits/packet to guarantee that, for $m=100$ stations, the average Etherring delay is again approximately equal to the average TPR delay when the utilization is high. This moderate packet size, which is typical for many applications, is sufficient to guarantee that, even in the case of a large ring with $m=100$ stations and high utilization,
Fig. 5.3.7(a) Normalized average delay $D/T$ vs. the utilization $\rho$ in the Etherring and in the Token Passing Ring for packets of length $T=100$ bits, $m=20$ and $m=100$ stations.

Fig. 5.3.7(b) Same as Fig. 5.3.7(a) but for packets of length $T=500$ bits and $m=100$ stations.
Etherring has so few collisions that the retransmissions do not (at least not significantly) increase the average delay.

A comparison of the average normalized transmission delays in a CSMA collision-detection bus (Ethernet, MET76) and in the TPR has been given in BUXT80 (the Ethernet analysis was adopted from LAM80). It has been shown that Ethernet and the TPR delays are approximately equivalent when the ratio between the packet (time) length and the signal propagation time in the bus is at least on the order of 100. The Ethernet protocol leads, however, to a reduced maximum stable utilization and longer transmission delays if this ratio is smaller. For example, on a bus of length 2km and 5 microseconds propagation delay per kilometer, transmission rate 10Mbit/sec, 50 stations and packets of average length 1000 bit (exponential distribution), Ethernet has a maximum stable utilization of only approximately 0.6; above $\rho=0.2$, the TPR has considerably shorter transmission delays than the Ethernet. On the other hand, our analysis shows that for these parameters the TPR and the Etherring have approximately the same average delay*). It is therefore clear that Etherring has also considerably shorter transmission delays than Ethernet.

We conclude this section with a comparison of the average transmission delay in a buffer insertion ring (BIR) described in chapter 3 to the average transmission delay in the Etherring. First we consider the case of a ring with uniform traffic, i.e., a ring where each station $i$, $1 \leq i \leq m$, transmits with the same rate to each other station $j$, $1 \leq j \leq m$, $j \neq i$. For this case the average total waiting time in the buffers of the BIR has in chapter 4 been shown to be well-approximated by

$$W_{B,s} \approx \frac{mcT}{4(2-\rho)} - \frac{(m-2)^2\rho T}{4(2m-(m-2)\rho)}.$$  \hspace{1cm} (5.3.72)

(*) Note that we have ignored the signal propagation time in the derivation of the Etherring formulas - for convenience, and because it is much smaller than the one-bit processing delay per station when the transmission rate (in bit/sec) is low or moderate. However, when the transmission rate is so high that the time to transmit one bit is in the same order of magnitude as the signal propagation time between two stations, one has to add the signal propagation times to the collision vulnerable periods.
On the average, the distance between sender and receiver is half a ring revolution. Therefore, the average transmission time from the generation of a packet until the packet has been completely received at the destination is approximately given by

\[ D_{B,s} = W_{B,s} + T + \frac{m\Delta_B}{2} \]  

(5.3.73)

where \( \Delta_B \) is the processing delay per BIR station (Note: \( \Delta_B \) includes the time a BIR station requires to decide whether a packet has to be relayed to the next downstream station or not, and is therefore at least equal to the address length).

Fig. 5.3.8(a) is a plot of the normalized average transmission times \( D_E/T \) and \( D_{B,s}/T \) vs. the utilization \( p \) for rings with \( m=20 \) and \( m=100 \) stations, and with \( T=100 \) bit/packet; the processing delay in the BIR is \( \Delta_B=8 \) bit/station. Fig. 5.3.8(a) illustrates that, for \( m=20 \) stations and light utilization, Etherneta transmits packets with almost the same average delay as would a fully connected network with private, dedicated links between every pair of stations. On the other hand, in the BIR, the average normalized delay is almost twice as long. Moreover, the figure illustrates that the average normalized BIR delay is much more affected than the average normalized Etherneta delay when the number of stations is increased to \( m=100 \). This is a consequence of the fact that, in the BIR, a relatively large fraction of the transmission delay is processing delay that increases with the number of stations. Fig. 5.3.8(b) shows, however, that it is sufficient to increase the number of bits per packet to \( T=500 \) to make the normalized average BIR delay for \( m=100 \) stations approximately equal to that for \( m=20 \) stations and \( T=100 \) bit/packet. In general, when the utilization is light, the difference between the average normalized transmission delays in Etherneta and in the BIR is significant only when \( T \) and \( m \) are approximately equal.

Figs. 5.3.8(a) and 5.3.8(b) also illustrate that Etherneta gets saturated and its transmission delay goes to infinity when the utilization increases and approaches \( T/T+1 \). The BIR, however, has a capacity of 2 and is thus still far from saturation and has finite average delay when Etherneta gets saturated. The latter advantage of the BIR depends, however, on
Fig. 5.3.8(a) Normalized average delay $D/T$ vs. the utilization $\rho$ in the Etherring and in the Buffer Insertion Ring for packets of length $T=100$ bits.
Fig. 5.3.8(b) Same as Fig. 5.3.8(a) but for packets of length $T=500$ bits.

$m = 100$ stations
$T = 500$ bit-packet
$\Delta_B = 8$ bit/BIR station

normalized average transmission times

utilization
1) the BIR's ability to remove packets from the ring as soon as they reach their destination (in Etherring, packets return to the sender where they are removed from the ring),

2) the 50 per cent reduction of link utilization which is a consequence of the symmetric traffic assumption (Note: because of the uniform traffic assumption, packets make on the average only half a ring revolution between sender and receiver).

When in an Etherring a packet returns to the sender, this is an implicit acknowledgement of a successful transmission. If the sender fails to receive this acknowledgement, a transmission error has occurred, and the packet has to be retransmitted. Because the BIR does not have this implicit acknowledgement feature, the destination is required to transmit an explicit acknowledgement to the sender to indicate whether the transmission has been successful or not. This negates, however, the 50 per cent reduction of the link utilization, at least partially (the acknowledgement may be shorter than the message itself).

To illustrate that the described BIR advantage critically depends on the uniform traffic assumption we consider the case of a data collection ring where stations 1,2,...,m transmit only to station m+1, which operates only as a data collection station and transmits to no one. For this case the average total wait in the buffers of a BIR was in chapter 3 shown to be given exactly by

\[ W_{B,c} = \frac{\rho T}{2(1-\rho)} \]  

On the average, the distance between sender and receiver is half a ring revolution. The average transmission delay is therefore given by

\[ D_{B,c} = W_{B,c} + T + \frac{m \Delta B}{2} \]  

Fig. 5.3.9(a) illustrates that the normalized average delay in an Etherring used as a data collection system with m=20 transmitting stations and T=100 bit/packet over the whole range of utilization is less than or equal to the average delay in a corresponding BIR. When the packet length is kept constant while the number of stations is increased to m=100, the Etherring is considerably faster for low and moderate
Fig. 5.3.9(a) Normalized average delay $D/T$ vs. the utilization $\phi$ in the Etherring and in the Buffer Insertion Ring when the traffic distribution is of the data collection type; the packet length is $T=100$ bits and the number of stations is $m=20$ and $m=100$. 

$T = 100$ bit-packet

$I_B = 8$ bit/BIR station
Fig. 5.3.9(b) Same as Fig. 5.3.9(a) but for packets of length $T=500$ bits and $m=100$ stations.
utilization; only when the utilization is high, is Ethernet slightly slower than the BIR. As in the case of symmetric traffic it is, however, again sufficient to increase the packet size when more stations are to be connected to the ring to reduce the normalized average delay in the BIR to almost the same normalized average delay as in the Ethernet; this is illustrated in Fig. 5.3.9(b) for a data collection ring with m=100 stations and T=500 bit/packet.

Moreover, Figs. 5.3.9(a) and 5.3.9(b) illustrate that both the BIR and the Ethernet get saturated at approximately the same utilization. The reason for this is that in a data collection ring, the link utilization increases from station to station and the last link just before the data collecting station carries the total traffic from all transmitting stations. When the BIR is used in a data collection application, its capacity is therefore approximately equal to the capacity of the Ethernet.

In a data collection application, it is an essential advantage that Ethernet is fair in the sense that all packets suffer the same average queueing delay, irrespective of their station of origin (of course, the nearer stations benefit from the fact that they have to go through fewer stations each of which gives one bit of delay). In the BIR no (at least no simple) priority strategy can be specified such that all stations see the same average delay. It has, in chapter 3, been shown that local access priority gives preferential treatment to the stations near the data collecting station (at the expense of the stations far away from the data collecting station) whereas ring priority gives preferential treatment to the stations far away from the data collecting station (at the expense of the stations near the data collecting station).

In reality, the traffic in a ring communication system is usually not strictly uniform. Before making a design choice on the basis of the results of a comparative analysis under the assumption of uniform traffic (which predicts that the BIR has twice the capacity of Ethernet and, under certain circumstances, smaller transmission delays), one should therefore carefully ascertain to what extent this uniform traffic assumption is valid. If the traffic tends to be more of the data collection type, the BIR will probably offer no major benefit, i.e., both the capacity and the transmission delay will be approximately the same as in Ethernet.
5.4 Conclusions

We have proposed Etherring, a new type of ring transport system for local area communication systems. We have discussed how one can take advantage of the ring topology to arrive at a system that combines the desirable features of a system with a random access protocol (Ethernet) with those of a system with a deterministic access protocol.

We have shown that the proposed Etherring protocol is very simple and insensitive to random bit errors in the packet flags. Unlike the TPR, Etherring does not require special precautions to detect and recover from error situations where the token has been lost or duplicated.

Moreover, Etherring has a maximum stable utilization of almost 1 (much the same as a TPR) whereas Ethernet has a considerably lower maximum stable utilization unless the ratio between the packet (time) length and the signal propagation time is at least in the order of 100.

When the utilization is light, Etherring has smaller transmission delays than the TPR because in the Etherring a station can begin to transmit immediately when it has a packet to send, whereas in the TPR a station has to wait until the token arrives before it can begin to transmit. When the utilization increases, this Etherring advantage is, however, reduced by the time that is wasted to resolve collisions. If, however, the number of bits per packet is larger than approximately five times the number of stations, so little time is wasted that, over the whole range of utilization, the average Etherring transmission delay never exceeds that of the TPR. The TPR is superior only when very short packets are to be transmitted in a ring with many stations.

Compared to the BIR, Etherring is inferior in the case of uniform traffic. In this case the BIR has a 50 per cent reduced link utilization advantage because packets make, on the average, only half a round trip between sender and receiver, before they are removed from the ring. Therefore, the BIR has approximately twice the capacity of Etherring (where packets return to the sender where they are removed from the ring) and, for higher utilizations, considerably smaller transmission delays than Etherring. The BIR however, suffers the disadvantage that the received messages are not acknowledged. On the other hand, in a data collection application where all packets go to a single station, Etherring has smaller average transmission delay than the BIR when the utilization is light, and approximately the same average transmission
delay when the utilization is heavy; the BIR cannot exploit the fact that it removes packets from the ring at the receiver. Moreover, Etherring has the advantage that it is fair in the sense that all packets suffer the same average queueing delay, irrespective of their station of origin (of course, the nearer stations benefit from the fact that they have to go through fewer stations each of which gives one bit of delay). In the BIR, there appears to be no practicable (and possibly no) priority strategy such that all stations see the same average queueing delay.

5.5 References

---


6. SUMMARY

The main contributions of this thesis are new results on the capacity and the delay-throughput characteristics of various ring communication systems. One of these systems, Etherring, is based on new concepts.

In chapter 2, we showed how capacity, the maximum rate with which packets can be sent through a ring communication system, depends on how sender and receiver are physically located relative to each other. For the case of uniform traffic, i.e. when each sender sends with the same rate to all receivers at the other stations, the capacity of a single ring is equal to twice the capacity of a transmission link between two stations.

An alternative way to increase the capacity of a single ring by increasing the capacity of the individual transmission links is to split the system up into separate rings. Ring communication systems with separate local rings which are connected either through a star network with a central switching node or through a global exchange ring are particularly well suited to applications where the stations can be segregated into local groups of stations which often communicate with stations from the same local group but only rarely with stations from the other groups. We derived exact formulas to calculate the capacity of connected ring systems with an arbitrary number of stations and arbitrary numbers of individual rings.

In chapter 3, we gave an exact waiting time analysis of a data collection ring with a decentralized buffer insertion access mechanism (a data collection ring is a system where stations 1, 2, ..., m-1 transmit through a single unidirectional link only to station m, which operates only as a data collecting station, and transmits to no one). It was shown that, for the conditions specified, the average delay in the data collection ring with symmetric traffic is the same as in a first-come first-served star network, regardless of the priority rule with which packets are served at the individual stations. The more common rule of giving packets on the ring priority over the packets that attempt to access the ring was shown to penalize downstream terminals. Surprisingly, the converse rule of giving new packets priority when they attempt to access the ring was found to be fairer in the sense that the range of average delays as seen by the individual stations is less than with the first rule.
In chapter 4, we gave approximate analyses of the packet waiting times in a general buffer insertion ring where all stations can transmit to each of the other stations, for both the case when the packet arrival processes are Poisson and when the packet arrival processes are periodic.

For the case where all packet arrival processes are Poisson and for low and moderate utilizations, the estimates obtained from the approximate analysis were shown to be in excellent agreement with the waiting times as determined by simulation. An intuitive argument was used to explain why the estimates are somewhat too large for higher utilizations.

The approximate analysis for the case when the local packet arrival processes are all periodic with random phase shifts between the individual packet streams was also found to give estimates of delay that are in excellent agreement with the waiting times as measured by simulations. This arrival model, which would be appropriate in packetized voice systems, was shown to result in considerably smaller delays than for the Poisson arrival model for the same utilization.

In chapter 5, we proposed Etherring, a new type of ring transport system for local area communication systems. We discussed how one can take advantage of the ring topology to arrive at a system that combines the desirable features of a system with a random access protocol (Ethernet) with those of a system with a deterministic access protocol. The proposed Etherring protocol is very simple and insensitive to random bit errors in the packet flags. Etherring has the disadvantage that occasionally two or more packets collide. This happens, however, with such a low probability, and collisions can be resolved so quickly, that this has, in all cases of interest, only a negligible effect on the average packet delay and on the maximum achievable rate with which packets can be transmitted with finite average delay.

In the case of uniform traffic, Etherring is inferior to the buffer insertion ring. In this case, the buffer insertion ring has a 50 per cent reduced link utilization advantage because packets make, on the average, only half a round trip between sender and receiver before they are removed from the ring. Therefore, the buffer insertion ring has approximately twice the capacity of Etherring (where packets return to the sender where they are removed from the ring) and, for higher utilizations, considerably smaller transmission delays than Etherring. On the other hand, in a data collection application where all packets go to
a single station, Etherring has smaller average transmission delay than the buffer insertion ring when the utilization is light, and approximately the same average transmission delay when the utilization is heavy; the buffer insertion ring cannot exploit the fact that it removes packets from the ring at the receiver. Moreover, Etherring has the advantage that it is fair in the sense that all packets suffer the same average queueing delay, irrespective of their station of origin.
APPENDIX

BUFFER INSERTION RING SIMULATION PROGRAM

1. Introduction

From the very beginning of the analysis of the buffer insertion ring it was clear that it would not be possible to derive exact models, except probably in some very special cases. It was therefore decided at a very early stage that a general purpose simulation program should be available in order to obtain performance estimates in those cases where no analytical tools are available at all, and in order to quantitatively judge the errors involved in using an approximate model. Moreover, the simulation program was expected to provide valuable insight into the peculiarities of the buffer insertion mechanism and thereby help to find either exact or approximate analytical models. This, in fact, happened and led to the genie argument to prove theorem 3.3.1 in chapter 3. At the beginning, there had been only the conjecture in chapter 4, section 4.4.2, but the discovery of the extraordinarily good agreement between the simulation results and the analytical model led to an additional effort to prove that the model was exact in the case of a data collection ring.

In section 2 of this appendix we describe the program structure and show how to use the program. Section 3 contains a sample simulation run. The computer program listing is contained in section 4 of this appendix.

2. How to use the simulation program

The simulation program is written in APL and can, in principle, be used on an arbitrary computer system with an APL interpreter. Although APL is not one of the most popular programming languages, it was found to be the one that was best suited to the problem. It should in general be easy to get access to such a computer system (APL is supported by IBM, DEC, HP, UNIVAC and other computer manufacturers). The program has been
developed and tested on an IBM 5100 desktop computer and has then been used on a big IBM 3033 where small rings can easily be simulated on-line; for larger rings with up to 100 stations it is advantageous to run the simulations off-line as batch jobs.

Knowledge of APL is not required to run the program. When the APL workspace that contains the simulation program is loaded, the initialization and simulation setup phase is started automatically and the user is prompted to specify the various simulation parameters. There are three different types of prompts:

1) The user is asked to specify a simulation parameter numerically, e.g., the ring length in km, the transmission rate in bit/sec, or the number of stations in the ring, etc.

2) The user is asked to make his selection from a menu of options. This is done by typing the number of the desired option. In the example in section 3, the user chose to tell the simulation program that the local access priority scheme should be used, and that the sender-receiver pairs should be determined randomly.

3) The user is asked whether he wants to activate or reject a given option. The option is activated when Y or YES is entered and rejected when N, NO or an empty line is entered. In the example in section 3, the user decided that he didn't want to have separate histograms for each station, but he wanted to have access and ring delay buffer histograms in the output listing.

When the current seed of the random number generator is printed, the user is given the options of either accepting the current value (simply by pressing the enter key before any other key is pressed) or to specify an arbitrary new value. The seed is then used as the starting value of the random number generator, e.g., to determine random sender-receiver pairs, random packet arrival instants or random packet lengths. This solution provides an easy means to reproduce simulation runs, or intentionally produce new random configurations, whichever is appropriate.

Finally, the user is asked whether the built-in debug option is to be turned on. This option provides a print-out of the system state after each state transition and was implemented mainly to assist in the program.
development. It is, however, also a valuable aid to those users who are interested in the internal characteristics of the program. The debug option can be turned off at arbitrary time instants during the simulation simply by setting the variable DEBUG to 0.

The user is also given the option to specify a time instant (measured in bit times) when the simulation is to be interrupted. If the user responds with an empty line, no interruption occurs but when a positive integer is typed, the simulation is interrupted at the specified time instant. The keyboard becomes unlocked and the user can execute arbitrary display or statistics functions or even ordinary APL expressions, he can change parameters, etc. When an empty line is entered, the user is asked whether and when the simulation is to be interrupted again, and the simulation is resumed from the point where it left off before the interruption.

An alternative way to interrupt the simulation is to press the attention or break key. The program responds with the number of the line of code to be executed next when the simulation is resumed. The keyboard unlocks so that the user can retrieve the desired information or change parameters, before the simulation is eventually resumed by typing -GO.

When the setup phase ends, the user is given precise instructions what he has to do next to run the simulation as a batch job. The user is asked to specify in which system queue the job is to be submitted (if, e.g., on our IBM 3033, the E-queue is specified, the job is executed during the night hours, i.e., between 10 p.m. and 8 a.m. - at only 50 percent of the nominal cost), and where the output is to be routed (we have two local and two remote printers). In any case, if the user responds with an empty line, typical default values are used.

When the job is selected for execution, the simulation starts and runs until either

1) The confidence intervals of the estimated performance quantities are sufficiently small,

2) The specified maximum number of simulation runs has been executed,

3) The specified computer time has been used up. Before the job is aborted, there is then in any case enough time left to make a statistical analysis of the measurement data collected up to that
time, to print the results, and to store the current state of the model together with the collected measurement data on a disk file for possible resumption of the simulation at a later time.

Most computing centers try to discourage their users from submitting jobs which require large amounts of computing time. In order to be able to offer a fast job turn-around time to the majority of users with low and moderate requirements, jobs which require larger amounts of computer time are given such a low priority that their job turn-around time becomes excessive. There are however certain moments when the results of a large job are required quickly. Although it should certainly not be practiced generally, one would therefore sometimes like to split a large job into a number of small jobs. If the user responds with Y or YES to the question SUBMIT SAME JOB ONCE MORE another job is submitted which will automatically resume the simulation from the point where it left off at the end of the first job. The same question is then asked again, so that the simulation can be split up into as many jobs as is required.

When the simulation is run as a batch job it is, of course, not possible to interrupt the simulation at intermediate time instants. If this is desired, the simulation has to be run on-line at the time sharing terminal where direct interaction with the program is possible. To start the interactive execution of the program, the user enters SIBIR (for Simulation of a Buffer Insertion Ring) and can then monitor the simulation, interrupt it, display or change parameters, and then resume the simulation, as described above, etc.

When the simulation ends, a time stamp and a summary of the current parameters are printed for later reference (see section 3). Then the simulated time is printed and the number of packets that have been generated and delivered. In the example in section 3, the simulated transmission of 7000 packets corresponds to 1.18 seconds in real time. Probably the most important simulation results are the total average packet waiting times and the total average packet transmission times which are printed together with their 95 per cent confidence intervals. In the example in section 3 the estimated average transmission time is 2.216±.119 (normalized with respect to the packet length). Also included in the simulation output are histograms of

- the normalized transmission time (as defined above),
- the ring buffer delay (i.e., the packet waiting times in the ring delay buffers),

- the access delay (i.e., the packet waiting times in the access buffers),

- the packet length.

All histograms are accompanied by a table that contains the number of observations (NO OBS), the maximum and minimum observed values, the range of observed values, the average and the standard deviation (STD DEV).

3. Sample simulation run

RING LENGTH (KILOMETRES): 1
TRANSMISSION RATE (BIT/SEC): 1E6
PRIORITY SCHEME:
  1-LOCAL ACCESS PRIORITY     2-RING PRIORITY
[]:
  1
RECEIVER RELATIVE TO SENDER
  1-RANDOM
  2-NEAR NEIGHBOUR
  3-FAR NEIGHBOUR
  4-ANTIPODES
  5-P[DISTANCE] = GEOMETRIC PROGRESSION
  6-ALL TRANSMISSIONS GO TO THE SAME RECEIVER
[]:
  1
NUMBER OF STATIONS: 5
MESSAGE LENGTH DISTRIBUTION
  1-EXPONENTIAL   2-CONSTANT
[]:
  1
AVERAGE MESSAGE LENGTH (BIT): 100
HEADER LENGTH (BIT): 16
PROCESSING DELAY PER STATION (BIT): 8
NORMALIZED THROUGHPUT: .6
NUMBER OF PACKETS PER SIMULATION RUN: 1000
MIN. NUMBER OF RUNS: 5
MAX. NUMBER OF RUNS (±30): 10
HISTOGRAM FOR:
1-TRANSMISSION TIME   2-WAITING TIME

1

SEPARATE HISTOGRAMS FOR EACH STATION ?
MAX. CPU TIME (SEC): 600
ESTIMATED CPU TIME TO ANALYZE THE COLLECTED DATA (SEC): 30
RANDOM LINKS IS 16807. ENTER CR OR NEW VALUE:
ACCESS AND RING DELAY BUFFER HISTOGRAMS ? Y
DEBUG?
INTERRUPT SIMULATION AT TIME :
ENTER

)WSID JOB-NAME
)SAVE
)BATCH

WHEN THE MESSAGE 'ENTER APL BATCH INPUT' IS DISPLAYED, ENTER
)LOAD JOB-NAME
)DROP JOB-NAME
)OFF HOLD
)CR

)WSID JOB
WAS SIB2
)SAVE
14:21:13 09/18/81 JOB
)BATCH
PRINTER = :
JOB ID = : 0
REGION = : 3000
JOBCLASS = :
SYSOUT CLASS = : T
ENTER APL BATCH INPUT
)LOAD JOB
)DROP JOB
)OFF HOLD

JOB K7021100(JOB08804) SUBMITTED
0
SUBMIT SAME JOB ONCE MORE?

14:38:54 JOB 8804 *HASP165 K7021100 ENDED AT MVS CN(00)
RING LENGTH = 1 KILOMETRES
TRANSMISSION RATE = 1000000 BIT/SEC
NUMBER OF STATIONS = 5
LOCAL ACCESS PRIORITY
RECEIVER RELATIVE TO SENDER: RANDOM
MESSAGE LENGTH DISTRIBUTION: EXPONENTIAL
AVERAGE MESSAGE LENGTH = 100 BIT
HEADER LENGTH = 16 BIT
ADDRESS RECOGNITION TIME = 0 BIT
PROCESSING DELAY PER STATION = 8 BIT
HISTOGRAM FOR: TRANSMISSION TIME
NORMALIZED THROUGHPUT = .60
NUMBER OF PACKETS PER SIMULATION RUN: 1000
MIN. NUMBER OF RUNS = 5
MAX. NUMBER OF RUNS = 10
MAX. CPU TIME = 600 SEC
CPU TIME FOR STATISTICS = 30 SEC
RANDOM LINK = 16807
ACCESS AND RING BUFFER DELAY HISTOGRAMS: YES

SIMULATION HALTED - REQUESTED NO. OF PACKETS DELIVERED

TIME = 1.184736 SEC
PACKETS GENERATED = 7000
PACKETS DELIVERED = 7000
PACKETS UNDERWAY = 0

ESTIMATED MEAN +/- 95 PERCENT CONFIDENCE INTERVAL:
2.216 +/- .119

RING UTILIZATION = .205 .205 .205 .205 .207
AVERAGE = .205

ACCESS UTILIZATION = .135 .138 .136 .134 .141
AVERAGE = .137

THROUGHPUT = 0.5893152567

| *************************************************** |
| 18-SEP-81 14.22.40 |
| *************************************************** |
**NORMALIZED TRANSMISSION TIME**

<table>
<thead>
<tr>
<th>LOWER</th>
<th>UPPER</th>
<th>NO</th>
<th>CDF</th>
<th>REL. FREQUENCY IN PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>.2</td>
<td>1.8</td>
<td>3668</td>
<td>.553</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>3.4</td>
<td>1771</td>
<td>.806</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>5.0</td>
<td>765</td>
<td>.915</td>
<td></td>
</tr>
<tr>
<td>5.0</td>
<td>6.6</td>
<td>308</td>
<td>.959</td>
<td></td>
</tr>
<tr>
<td>6.6</td>
<td>8.2</td>
<td>134</td>
<td>.978</td>
<td></td>
</tr>
<tr>
<td>8.2</td>
<td>9.8</td>
<td>78</td>
<td>.989</td>
<td></td>
</tr>
<tr>
<td>9.8</td>
<td>11.4</td>
<td>25</td>
<td>.993</td>
<td></td>
</tr>
<tr>
<td>11.4</td>
<td>13.0</td>
<td>19</td>
<td>.995</td>
<td></td>
</tr>
<tr>
<td>13.0</td>
<td>14.6</td>
<td>10</td>
<td>.997</td>
<td></td>
</tr>
<tr>
<td>14.6</td>
<td>16.2</td>
<td>8</td>
<td>.998</td>
<td></td>
</tr>
<tr>
<td>16.2</td>
<td>17.8</td>
<td>2</td>
<td>.998</td>
<td></td>
</tr>
<tr>
<td>17.8</td>
<td>19.4</td>
<td>4</td>
<td>.999</td>
<td></td>
</tr>
<tr>
<td>19.4</td>
<td>21.0</td>
<td>2</td>
<td>.999</td>
<td></td>
</tr>
<tr>
<td>21.0</td>
<td>22.6</td>
<td>2</td>
<td>.999</td>
<td></td>
</tr>
<tr>
<td>22.6</td>
<td>24.2</td>
<td>1</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>24.2</td>
<td>25.8</td>
<td>0</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>25.8</td>
<td>27.4</td>
<td>1</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>27.4</td>
<td>29.0</td>
<td>1</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>29.0</td>
<td>30.6</td>
<td>0</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>30.6</td>
<td>32.2</td>
<td>0</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>32.2</td>
<td>33.8</td>
<td>1</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

MAXIMUM 32.87
MINIMUM 0.18
AVERAGE 2.2162 +/- .119
STD DEV 2.164
RANGE 32.69
NO OBS 7000
**ACCESS DELAY**

MAXIMUM 842
MINIMUM 0
AVERAGE 39.952 +/- 3.550
STD DEV 88.832
RANGE 842
NO OBS 7000

**RING BUFFER DELAY**

MAXIMUM 1618
MINIMUM 0
AVERAGE 34.508 +/- 5.468
STD DEV 108.93
RANGE 1618
NO OBS 10446
4. Simulation program listing

ABORT           ABORT TEXT
               [1] ''
               [2] TEXT
               [3] +
               [v

ABUFINCR

               [1] ''
               [2] ACCBUF+(S,S,ABUFSIZE+ABUFSIZE+1)+ACCBUF
               [3] A+(S,ABUFSIZE)+A
               [4] 'TIME = .((\vTIME),': ACCESS BUFFER SIZE = \vABUFSIZE
               [5] ''

ANALYZE

               [v ANALYZE;N0
               [1] -(2>N0+1+pTTR)/0
               [2] N+2
               [4] MEAN+(+/N)*NOP
               [5] MEAN+(+/MEAN)*N
               [6] SY+(N-1)*+(MEAN-MEAN)*2
               [7] SY+SY*0.5
               [8] E+SYxTALPHA95[30\N-1]*N+0.5
               [9] ''
               [10] 'LAST ,\(\vN), RUNS ONLY:
               [12] (11 3 \vMEAN), +/- ', 11 3 \vE
               [13] -(N0\N+N+1)/AG
               [v
APL

\[ APL \]

[1] 'SIMULATION INTERRUPTED AT TIME ',TIME
[2] LL:'ENTER APL COMMAND OR CR TO CONTINUE'
[3] INPUT←
[4] +(0=INPUT)/LAB
[5] INPUT
[6] →LL
[7] LAB:CHK+(\text{ENDTIME}) PRIOD 'NEXT INTERRUPT '

\[ \text{ANALYZE} \]

\[ APL \]

\[ \text{NOP ANALYZE X;NO;X} \]
[1] N←2
[2] NO←(pX)*NOP
[3] X+(NO,NOP)*p,X
[4] +(2>N0)/0
[5] AG←N+(\text{-N})*NOP+X
[6] MEAN←+/N)*NOP
[7] MEAN←+/MEAN)*N
[8] SY←(N-1)*(MEAN-MEAN)*2
[9] SY←SY*0.5
[10] E←SY*TALPHA95(30\text{|N-1})*N*0.5
[12] 'LAST',(\text{-N})*NOP,' PACKETS ONLY:'
[13] PRT:'ESTIMATED MEAN +/- 95 PERCENT CONFIDENCE INTERVAL: '
[14] (11 3 \text{MEAN}),' +/- ', 11 3 \text{E}
[15] +(NO=N+N+1)/AG

\[ \text{BATCH} \]

\[ APL \]

\[ \text{BATCH;Y} \]
[1] RT←'RPUBPRT' PRIOD 'PRINTER = '
[2] JID←'A' PRIOD 'JOB ID = '
[3] CPU←[0.001*MAXCPU*60]
[4] REG←'500' PRIOD 'REGION = '
[5] JC←'D' PRIOD 'JOBCLASS = '
[6] SC←'A' PRIOD 'SYSOUT CLASS = '
[7] DIM←pX+BATCHJCL
[8] X←1? PVM X
[9] X+X RS 'TI',CPU
[10] X+X RS 'JC'
[12] X+X RS 'RPUBPRT',RT
[13] X+X RS 'R',REG
[14] X←1? FNW X
[15] X←DIM+X
[16] X(1:10)+JID
[17] 'ENTER APL BATCH INPUT'
[18] LP:+(0=pY+DTB Y)/SUB
[19] X+X,[□IO] 1 80 p80+y
[20] →LP
[21] SUB:'VSAPL.APLBATCH' WRITECARDS X
TSO 'FREE ALL'

AG: TSO 'SUB VSAPL.APLBATCH'
+(YES 'SUBMIT SAME JOB ONCE MORE ?')/AG

TSO 'DELETE VSAPL.APLBATCH'

BIRTH

V BIRTH:AI;L;DMY
NEXT[1;STATION]+ENDTIME
A+(5,ABUFSIZE) ACCBUF[;STATION;]
+(PDISTR=1)'A[;1]+TIME,DEST,(L+H+1)[1 EXPLENGTH
LM),STATION,TIME'
+(PDISTR=2)'A[;1]+TIME,DEST,(L+H+LM),STATION,TIME'
ACCBUF[;STATION;]+(5,1,ABUFSIZE)pA
ACCESS[STATION]+ACCESS[STATION]+L
NEXT[1;S-1*RECLOC=6]+TIME+1 EXPLENGTH PACKETPERIOD
DMY+TIME
*((NEXT[2;STATION]<ENDTIME)APRIO=2)/'DMY+DMY[NEXT[2;STAT
TION]'
+(NEXT[3;STATION]<=TIME)/'NEXT[2;STATION]+DMY'
LEN+LEN,L
DEBUG/'DISPLAY'

CCCT
R+X CCCT Y
COLUMN CONCATENATION
X+(2t(px), 1 1)pX
Y+(2t(py), 1 1)pY
R+(((1 0 ×pY)[pX]×X),[1+IO])(1 0 ×pX)[pY]+Y

DBUFINCR

V DBUFINCR
DELETE+(5,S,ABUFSIZE+ABUFSIZE+1)+DELBUFF
D+(5,ABUFSIZE)+D
'TIME = ',(\TIME),': DELAY BUFFER SIZE = ',\DBUFSIZE

DEST
D+DEST
D+S
D[STATION]+0
D+D/:S
D+D[S-1]
DEST1

  \[ D + \text{DEST1} \]
  \[ [1] D + S p_1 \]
  \[ [2] D [\text{STATION}] + 0 \]
  \[ [3] D + D / S \]

DEST2

  \[ D + \text{DEST2} \]
  \[ [1] D + \text{STATION} + 1 \]
  \[ [2] D + D - S \times D > S \]

DEST3

  \[ D + \text{DEST3} \]
  \[ [1] D + \text{STATION} - 1 \]
  \[ [2] D + D + S \times D < 1 \]

DEST4

  \[ D + \text{DEST4} \]
  \[ [1] D + \text{STATION} + 0.5 \times S \]
  \[ [2] D + D - S \times D > S \]

DEST5

  \[ D + \text{DEST5} \]
  \[ [1] P \times (1 - Q) + 1 - Q \times S - 1 \]
  \[ [2] Q \times \backslash (S - 2) p Q \]
  \[ [3] P \times p_{1.0} \]
  \[ [4] D_{1+1} \text{ RANDINT} P \]
  \[ [5] D + \text{STATION} + D \]
  \[ [6] D + D - S \times D > S \]

DEST6

  \[ D + \text{DEST6} \]
  \[ [1] D + S \]

DISPLAY

  \[ \text{DISPLAY} \]
  \[ [1] \textbf{''} \]
  \[ [2] \textbf{TEXT-DB}, \textbf{(VECMAI 'BIRTH,TRANSMISSION')}, \textbf{WHAT; \}} \]
  \[ [3] \textbf{''SITUATION AFTER '}, \textbf{TEXT,' AT TIME = '}, \textbf{''TIME \}
  \[ [4] \textbf{''} \]
  \[ [5] \textbf{''STATION = '}, \textbf{''STATION \}
  \[ [6] \textbf{''} \]
[7] 'ACCBUF'
[8] QACCBUF
[9] ''
[10] 'DELBUF'
[12] ''
[13] 'NEXT'
[14] QNEXT

DTB

\[ R+DTB \downarrow \]
[1] \[ R+(-+/A\phi' \ 'y)\downarrow \]

DMB

\[ R+DMB \downarrow \]
[1] \[ m \text{ DELETE MULTIPLE BLANKS} \]
[2] \[ R+((RV1+R\cdot 0)A\phi'R-y)'y)\downarrow \]

DRAWHL

\[ R+X \text{DRAWHL} \downarrow \]
[1] \[ m \text{ DRAW HORIZONTAL LINES AT ROWS} \ X \ IN \ Y \]
[2] \[ y+(\text{2}+1+ \rho \text{y})\downarrow \]
[3] \[ R+(y.[\pi\text{IO}] \ '-'[(\pi\text{IO}+1+\text{y})\downarrow (1+\text{y}),\text{x};] \]

DTB

\[ R+DTB \downarrow \]
[1] \[ m \text{ DELETE TRAILING BLANKS} \]
[2] \[ R+(-+/A\phi' \ 'y)\downarrow \]

ENDDDD

\[ \text{ENDDD} ;A;B \]
[1] \[ \text{TC}[3\rho2] \]
[2] \[ A+28+ \text{ HRS MIN SEC MIL'} \]
[3] \[ B+28 \rho' \text{ SIGN ON CPU} \]
[4] \[ A, [1], B, 50 v 0 60 60 1000 \text{TAI}[3, 2]-\text{AYEL} \]

EXPLENGTH

\[ T+N \text{EXPLENGTH MEAN};\pi\text{IO} \]
[1] \[ \pi\text{IO}=0 \]
[2] \[ T+0.5+(\text{-MEAN})\times(1-1E-10)\pi\text{Np100000000000} \]

\[ \]
\[ F_{MV} \]

\[ R+x \ F_{MV} \ y; s; t; u; v; \]

[1] FORM MATRIX FROM VECTOR Y WITH DELIMITERS X
[2] \[ F_{MV} \]

\[ F_{VM} \]

\[ R+x \ F_{VM} \ y \]

[1] FORM VECTOR WITH DELIMITERS X FROM MATRIX Y
[2] \[ F_{VM} \]

GO

\[ R+go \]

[1] \[ F_{VM} \]

HIST

\[ TEXT \ HIST \ x; r; i; z; size; sum; freq; b; p; max; min; xbar; sdev; n; \]

[1] \[ F_{VM} \]

\[ CUM \]

[1] \[ F_{VM} \]

\[ TC[2] \]

\[ OBOX TEXT \]

[1] \[ F_{VM} \]

\[ NCELLS+1+x \]

[1] \[ F_{VM} \]

\[ STATISTICS x+1+x \]

[1] \[ F_{VM} \]

\[ CUM=0 \]

[1] \[ F_{VM} \]

\[ START=R+1+max-min \]

[1] \[ F_{VM} \]

\[ SIZE+=1+i \]

[1] \[ F_{VM} \]

\[ NCELLS+=R+size \]

[1] \[ F_{VM} \]

\[ 1 \text{ DRAWHL 'FREQUENCY HISTOGRAM (EACH STAR = 1 PER CENT)'} \]

[1] \[ F_{VM} \]

\[ TC[2] \]

[1] \[ F_{VM} \]

\[ 6p ' ' , 'LOWER', (6p ' '), 'UPPER', (9p ' '), 'NO', (6p ' '), 'CDF', (5p ' '), 'REL. FREQUENCY IN PERCENT' \]

[1] \[ F_{VM} \]

\[ I+1 \]

[1] \[ F_{VM} \]

\[ B+MIN+SIZEx0, \]

[1] \[ F_{VM} \]

\[ B[1+NCELLS]+B[1+NCELLS]+0.00001 \]

[1] \[ F_{VM} \]

\[ UP:=((1+NCELLS)<I-I+1)/0 \]

[1] \[ F_{VM} \]

\[ SUM+=/(B[I]-1)x<\]

[1] \[ F_{VM} \]

\[ CUM+CUM+SUM+N \]

[1] \[ F_{VM} \]

\[ 110 | B[I]-1, B[I]-1, (110 | SUM), (93 | SUM), (5p ' '), (70 | 0.5+SUMxP+100+N | P(69p '*')); ' +' \]

[1] \[ F_{VM} \]

\[ UP \]

\[ F_{VM} \]
HISTOGRAM

\[ \text{TEXT HISTOGRAM} \]
\[ X;Z;R;I;SIZE;SUM;FREQ;B;P;MAX;MIN;XBAR;SDEV;N;CUM \]

[1] \( sp[TC[2]] \)
[2] ' ' 
[3] \( OBOX \) \( TEXT \)
[4] \( sp[TC[2]] \)
[5] \( NCELLS+1+X \)
[6] \( STATISTICS \) \( X+1+X \)
[7] \( +N=0)/0 \)
[8] \( sp[TC[2]] \)
[9] \( CUM+0 \)
[10] \( \text{START}\) \( SIZE+0.2\times1[5\times R+NCELLS \]
[11] \( NCELLS=1[R+SIZE \]
[12] \( \text{1 DRAW IH 'FREQUENCY HISTOGRAM (EACH STAR = 1 PER} \)
\[ \text{CENT)'} \]
[13] \( sp[TC[2]] \)
[14] \( \{6p', 'LOWER', (6p', 'UPPER', (9p', 'NO', (6p' 
\]
[15] \( I+1 \)
[16] \( B+MIN+SIZE\times 0, NCELLS \)
[17] \( B[1+NCELLS]+B[1+NCELLS]+0.00001 \)
[18] \( UP+:((1+NCELLS)<I+I+1)/0 \)
[19] \( SUM++/B[I-1][I] \times X B[I] \)
[20] \( CUM++SUM+N \)
[21] \( (11 1 \times B[I-1], B[I]), (11 0 \times SYN), (9 3 \times CUM), (5p' 
\]
[22] \( +UP \)

\[ \text{INTERRUPT} \]

\[ \text{INTERRUPT} \]

[1] \( NOP+NRUN=0 \)

\[ \text{IPM} \]

\[ \text{R+IPM Y;Z} \]

[1] \( sp[TO] \)
[2] \( R+0 0 p' \)
[3] \( sp[Y] \)
[4] \( L1:+(0=pZ+, 0)/0 \)
[5] \( R+((1+pR), S)+R),[DIO][(S+(pZ)|1+pR)+Z \]
[6] \( =L1 \)

\[ \text{IPMENU} \]

\[ \text{R+X IPMENU Y;Z;S;T} \]

[1] \( sp[(T),(1)p;Z+1+pX), '-', X \)
[2] \( S+1[I][PW]+1+pX \)
[3] \( T+((T+(P), S)+p(T+(S, 1))*((pX)*(S, 1)+X) \)
[4] \( sp[Y] \)
[5] \( sp[T] \)
[6] \( L1:+(R\times2, A1=p, R+0)/0 \)
INVALID SELECTION - TRY AGAIN

JUSTR

JUSTR Y

R+JUSTR Y

JUSTIFY RIGHT

R+(-+/\phi', '=#Y)\phi Y

MATVEC

MATVEC Y

MATRIX-TO-VECTOR CONVERSION (CR APPENDED)

R+\phi\phi', '=#Y), 1

\phi \phi. 1+R/Y, Y, CR

OBNE

OBNE Y; SP

R+(-,X/((((1+pX),SP)+X)\phi =X((1+pY),SP)+Y))\phi X

QBOX

QBOX Y

FRAME Y

R+''', '[IO]((-2+ 1 1 ,pY)pY), [IO] '"

QIBR

QIBR Y

INSERT BLANK ROWS INTO Y AFTER X

R+(-RcX)/R+\phi(1+pY), X

PREPARE

PREPARE

PW+70

RESUME+AREC+THRUPT+RUN+MEAN+0

ABFSIZE=DBFSIZE+STATION+TIME+\phi IO+1

LENGTH+PRI 'RING LENGTH (KILOMETERS);'

CL-PRI 'TRANSMISSION RATE (BIT/SEC);'

PRIo=(VECMAI 'LOCAL ACCESS PRIORITY ,RING PRIORITY '

IPMENU 'PRIORITY SCHEM: '

\phi(PRIo=1)/'D+OCR 'TRANSMITLP''

\phi(PRIo=2)/'D+OCR 'TRANSMITDSP''

D[1;9,10,11]+ ''

D+D'FX D
D+VECMAT 'RANDOM ,NEAR NEIGHBOUR ,FAR NEIGHBOUR ,ANTIPODES '

D+D RCCT VECMAT 'P[DISTANCE] = GEOMETRIC PROGRESSION
,ALL TRANSMISSIONS GO TO THE SAME RECEIVER'

RECLOC+D+D IPMENU 'RECEIVER RELATIVE TO SENDER'

LO:TEXT+VECMAT 'ENTER PARAMETER:,<1: NEAR DESTINATIONS
MORE LIKELY'

TEXT+TEXT RCCT '->1: FAR DESTINATIONS MORE LIKELY'

*(RECLOC=5) '/' TEXT'

*(RECLOC=5) '/' Q+Q'

*(RECLOC=5)AQ=1) '/' A' '->L0', 16+Q=' Q=1 NOT
ALLOWED',QTC[2]'

L2:S+PRI 'NUMBER OF STATIONS: '

*(RECLOC=4)AQ=2|S) '/' A' '->L2', 13+Q=' MUST BE
EVEN',QTC[2]'

ACCESS+UTIL+S+0

D+QFX D

PDISTR+(VECMAT 'EXPONENTIAL ,CONSTANT') IPMENU
'MESSAGE LENGTH DISTRIBUTION'

LM+PRI (0 8)(1+PDISTR=2) 'AVERAGE MESSAGE LENGTH
(BIT):'

H+PRI 'HEADER LENGTH (BIT): '

TAU+PRI 'PROCESSING DELAY PER STATION (BIT): '

NPP+_1+(VECMAT 'YES ,NO') IPMENU 'SURVIVING PACKET
FRAGMENTS'

*(NPP=0) '/' AREC+PRI 'ADDRESS RECOGNITION DELAY PER
STATION (BIT): '

PACKETPERIOD+10.5+LM+PRI 'NORMALIZED THROUGHPUT: '

NOP+PRI 'NUMBER OF PACKETS PER SIMULATION RUN: '

MINRUN+PRI 'MIN. NUMBER OF RUNS: '

NRUN+30+PRI 'MAX. NUMBER OF RUNS (≤30): '

TSW+(VECMAT 'TRANSMISSION TIME ,WAITING TIME') IPMENU
'HISTOGRAM FOR: '

SST+YES 'SEPARATE HISTOGRAMS FOR EACH STATION ? '

MAXCPU+1000+PRI 'MAX. CPU TIME (SEC): '

CPU+MAXCPU-1000+PRI 'ESTIMATED CPU TIME TO ANALYZE THE
COLLECTED DATA (SEC): '

ENDTIME=99999999

RL+RL+*(4||RL) PRIOR 'RANDOM LINKS IS ',(4||RL)',, ENTER
CR OR NEW VALUE'

ACCBUF+(5,S,ABUFSIZE)p0

DELBUF+(5,S,DEBUFSIZE)p0

NEXT+(3,S)pENDTIME

NEXT[3;1]--ENDTIME

SLIST+LEN+WAB+WAB+TTR+10

RUN+1

TTR+(0,NOP)p0

MAT+VECMAT 'BIRTH ,TRANSMIT'

SW1+YES 'ACCESS AND RING DELAY BUFFER HISTOGRAMS? ' 
[55] DEBUG=YES 'DEBUG?'
[56] CHK+2(VENDTIME) PRIOD 'INTERRUPT SIMULATION AT TIME ' TEXT+'ENTER "DEBUG+O" TO TERMINATE DEBUGGING'
[57] *DEBUG/'TEXT'
[58] WHAT=1
[59] 'ENTER'
[60] 'WSID JOB-NAME'
[61] 'SAVE'
[62] 'BATCH'
[63] 'WHEN THE MESSAGE "ENTER APL BATCH INPUT" IS DISPLAYED, ENTER'
[64] 'LOAD JOB-NAME'
[65] 'DROP JOB-NAME'
[66] 'OFF HOLD'
[67] 'C-R'
[68] 'LX+' 'SIBIR'

PRI

\[ R+PRI Y \]
[1] a PROMPT AND REQUEST INPUT
[2] \( R+2((V'='R)\&R=(PR)+Y)/R+0,0+Y \)

PRIM

\[ R+X PRIM Y ; Z \]
[1] a PROMPT AND REQUEST INPUT FROM MENU
[2] \[ L1=+((i=pR+((Z,PR)+X)\&.=-R+PRI Y)/Z+1+pX)/0 \]
[3] \( X+\) 'SELECT FROM:
[4] \( X+\) '
[5] =L1

PRIOD

\[ R+D PRIOD P \]
[1] a PROMPT (P) AND REQUEST INPUT OR SET Default (D)
[2] \( D+P,' : \)
[3] \(+0<PR+(/A',='R)+R+DTB D)/0 \)
[4] \( R+D \)

RANDINT

\[ R+N RANDINT PDENS; PD \]

REMOVE

\[ REMOVE \]
[1] *DEBUG//"PACKET REMOVED\"
RECEIVER=RECEIVER+S*RECEIVER<SENDER

*TSW=1)*TTR-TTR,((5*6*CL*LENGTH*(RECEIVER-SENDER)*S)+
(TIME*A[3;1]-A[1;1]))+LM'

*TSW=2)*TTR+TTR,TIME-A[1;1]

SLIST+SLIST,SENDER

THRU+THRU+H

+(NOP=0)/STOP

+(NOP+P+TTR)/0

TTR+TTR RCCT TTR

END\n
"'

'RUN ',(\'RUN\'),' COMPLETED AT TIME ',(\'TIME\'),'.

',('RUN\x26;NOP\'),' PACKETS.'

RUN\+RUN+1

TTR+0

MEAN+(+/TTR)*NOP

MEAN+(+/MEAN)*N+P

*(N<2)/'E+0'

+(N<2)/PRT

SY+(n-1)*+/MEAN-MEAN)+2

SY\+SY+0.5

E+SY\+TALPHA95[N-1]*N+0.5

PRR:'ESTIMATED MEAN +/- 95 PERCENT CONFIDENCE INTERVAL: '

(11.3 *MEAN),'+/- ',11.3 \E

UTILIZATION

MEAN+3*MEAN,MEAN

+(3<</MEAN>0)/0

+(RUN\+MINRUN)/0

+(0.05+/|MEAN-MEAN|)/STOP

+(NRUN\+RUN)/0

STOP:UX+'RESUME'

STOP 'SIMULATION HALTED - REQUESTED NO. OF PACKETS DELIVERED'

V

RESTART

V RESTART

PACKETPERIOD+0.5+LM+PRI 'NORMALIZED THROUGHPUT: '

ACCESS+UTIL+UTIL+0

RESUM+THRU+THRU+MEAN+0

RUN+ABFSIZE+DBFSIZE+STATION+TIME+IO+1

RL+RL

ACCBUF+(5,S,ABFSIZE)0

DELBUF+(5,S,ABFSIZE)0

NEXT+(3,S)ENDTIME

NEXT[3;]+ENDTIME

SLIST+LEN+WAB+WBB+TTR+10

NOP\+PRI 'NUMBER OF PACKETS PER SIMULATION RUN: '

MINRUN\+PRI 'MIN. NUMBER OF RUNS: '

NRUN\+30\+PRI 'MAX. NUMBER OF RUNS (530): '

MAXCPU+1000\+PRI 'MAX. CPU TIME (SEC): '

CPU+MAXCPU+1000\+PRI 'ESTIMATED CPU TIME TO ANALYZE THE COLLECTED DATA (SEC): '

TTR\+(0,NOP)p0

DEBUG\+YES 'DEBUG ?'

CHK+2(\'ENDTIME\') PRIOR 'INTERRUPT SIMULATION AT TIME '
[19] TEXT+"ENTER "'DEBUG+0" TO TERMINATE DEBUGGING"

[20] "%DEBUG/"TEXT"

[21] WHAT+1

[22] 'ENTER'

[23] ' )WSID JOB-NAME'

[24] ' )SAVE'

[25] ' )BATCH'

[26] 'WHEN THE MESSAGE "ENTER APL BATCH INPUT" IS
DISPLAYED, ENTER'

[27] ' )LOAD JOB-NAME'

[28] ' )DROP JOB-NAME'

[29] ' )OFF HOLD'

[30] ' )C-R'

[31] LX+"SIBIR"

\V RESUME

\V RESUME

SPEC

[1] ✱((1xpixDiC)vffOP=0)/'ABORT "'CANNOT RESUME"'"

[2] MINRUN+NRUN+NRUN+1

[3] RESUME+1

[4] WHAT+2

[5] SIBIR

\V RCCT

\V R+A RCCT B

[1] ✱ ROW CONCATENATION

[2] A+(-2+ 1 1 ,pA)pA

[3] B+(-2+ 1 1 ,pB)pB

[4] R+((0 1 ×pB)(pA)+A),[[IO][(0 1 ×pA)(pB)+B

\V RS

\V R+X RS Y;0;N;Z;IO

[1] ✱ REPLACE SUBSTRINGS OF X

[2] L OLD AND NEW SUBSTRINGS IN Y, SEPARATED BY ' 

[3] ✱IO+0

[4] O+(Z+Y+"."")+Y+Y

[5] N+(Z+1)+Y

[6] Z+(X+,X) ISNO 0

[7] Z+(Z,px)−5,Z+pO

[8] R+" 


[10] X+(Z[0]+pO)+X

[11] +(0=pZ+1+Z)/0

[12] R+R,N

[13] +L1

\V
SIBIR

[SIBIR]

[1] PW+120
[2] START
[3] +RESUME/RESUME
[4] DLX+
[5] TEXT+'RUN ''PREPARE' AND THEN 'SIBIR''
[6] +(TIME=+1)'/ABORT TEXT'
[7] SPEC
[8] BIRTH
[9] LOOP:-(ENDTIME<=TIME+1/X+|NEXT)/TOOSMALL
[10] WHAT+(1,2)[1+(N+X:TIME)>S]
[12] STATION=N-SxN>S
[13] STATION+STATION-SxSTATION>S
[14] +(TIME<=CHK)'/APL'
[15] +(DAI(2)=CPU)'/STOP 'SIMULATION HALTED - SPECIFIED CPU TIME USED UP''
[16] RESUME:±,MAT[WHAT;]
[17]IF
[18] TOOSMALL:STOP 'SIMULATION HALTED AFTER ',(ENDTIME),'
BIT TIMES.','TC[2].'INCREASE 'ENDTIME' IF NECESSARY.'

SINGLESTAT

[SINGLESTAT;TEMP;I]

[1] I+1
[2] +(TSW=2)/LL
[3] L:('PACKETS FROM STATION ',(I),' ') HISTOGRAM
20,TEMP+(SLIST=I)/TTR,TTR
[4] (10|LNP=S) ANALYZE TEMP
[5] +(>I+I+1)/L
[6] SI
[7] LL:('PACKETS FROM STATION ',(I),' ') HIST
20,TEMP+(SLIST=I)/TTR,TTR
[8] (10|LNP=S) ANALYZE TEMP
[9] +(>I+I+1)/LL

SPEC

[SPEC]

[1] OBOX DATS
[3] 'RING LENGTH = ',(LENGTH),' KILOMETRES'
[4] 'TRANSMISSION RATE = ',(1CL),' BIT/SEC'
[5] 'NUMBER OF STATIONS = ',S
[6] (VECMAT 'LOCAL ACCESS PRIORITY,RING PRIORITY')P[PRIO;]
[7] TEXT+VECMAT 'RANDOM,NEAR NEIGHBOUR,FAR
NEIGHBOUR,ANTIPODES,PDISTANCE] = GEOMETRIC PROGRESSION'
[8] TEXT+TEXT RCCT 'ALL TRANSMISSIONS GO TO THE SAME
RECEIVER'
[9] TEXT+.TEXT[RECOLC;]
[10] TEXT+(1,0TEXT)pTEXT
[11] +(RECOLC=5)'/TEXT+TEXT_RCCT 'PARAMETER = ',Q'}
CCCT TEXT

MESSAGE LENGTH DISTRIBUTION: 

' \text{EXPONENTIAL, CONSTANT} \text{PDISTR} = 2 \text{AVERAGE MESSAGE LENGTH = } \text{(VECMAT 'VALUE') BIT} \\
\text{HEADER LENGTH = ', ' (VECMAT 'VALUE') BIT} \\
\text{ADDRESS RECOGNITION TIME = ', ' (VECMAT 'VALUE') BIT} \\
\text{PROCESSING DELAY PER STATION = ', ' (VECMAT 'VALUE') BIT} \\
\text{SURVIVING PACKET FRAGMENTS: ', ' (VECMAT 'VALUE')}

'HISTOGRAM FOR: ', (VECMAT 'VALUE')

'NORMALIZED THROUGHPUT = ', 6 2 \text{LM: PACKETPERIOD} \\
\text{NUMBER OF PACKETS PER SIMULATION RUN = }, \text{NWP} \\
\text{MIN. NUMBER OF RUNS = }, \text{MINRUN} \\
\text{MAX. NUMBER OF RUNS = }, \text{NRUN} \\
\text{MAX. CPU TIME = }, (\text{MAXCPU-CPU}) \text{ SEC} \\
\text{CPU TIME FOR STATISTICS = }, (\text{MAXCPU-CPU}) \text{ SEC} \\
\text{RANDOM LINK = '}, \text{RL} \\
\text{ACCESS AND RING BUFFER DELAY HISTOGRAMS: '}, (\text{VECMAT 'VALUE'})

3pITC[2]

START

\text{STOP T}

M = \text{VECMAT TIME, PACKETS GENERATED PACKETS DELIVERED, PACKETS UNDERWAY} \\
M = \text{VECMAT 'VALUE'} \\
m = \text{VECMAT DMB 12 6 9 0 9 0 0} \\
\text{TIME*CL, (pLEN), ((pTTR)+p,TTR), (+/+/0*ACCBUF[1;;])} \\
+/+/0*DELBUP[1;;]

\text{ENDAAA}

ANALYZE

3pITC[2]

STRING+''NORMALIZED TRANSMISSION TIME' HISTOGRAM 20,TTR,TTR' 
\text{TSW=1}/\text{STRING} \\
\text{TOTAL WAITING TIME' HISTOGRAM 20,TTR,TTR'} 
\text{TSW=2}/\text{STRING} \\
\text{'SINGLESTAT'} \\
\text{ACCESS DELAY' HIST 20, WAB} \\
\text{ANALYZE WAB} \\
\text{'RING BUFFER DELAY' HIST 20, WBR} \\
\text{ANALYZE WBR}
'DELAY PER WAIT' HIST 20,WAB,WRB
NOP ANALYZE WAB,WRB
* (PDISTR=1)/'"PACKET LENGTH"' HIST 20,LEN'
4p[rTC[2]
UTILIZATION
EXIT:ENDAAA

STATISTICS

\[ \text{STATISTICS X;\text{DPPL}} \]
\[ \text{UPP+5} \]
\[ +0=N+pX/0 \]
\[ \text{MAXIMUM } x:MAX+/X \]
\[ \text{MINIMUM } x:MIN-/X \]
\[ \text{AVERAGE } x:XBAR+(+/X)*N+pX \]
\[ +20000<x>/G0+2 \]
\[ \text{STD DEV } x:SDEV+((+/(X-XBAR)*2)/(N-1))^{0.5} \]
\[ \text{RANGE } x:R-MAX-MIN \]
\[ \text{NO OBS } x:N \]

TAPRTV

\[ \text{TAPRTV;TAPRD;LAB;WS;JOB} \]
\[ \text{JOB+ 1 60 +TAPRD} \]
\[ \text{DIM+TAPRD} \]
\[ X+? \"\text{EVN TAPRD} \]
\[ SC+'A' PRIOD 'SYSOUT CLASS = ' \]
\[ PR+'RPUBPRT' PRIOD 'PRINTER = ' \]
\[ X+X RS 'RPUBPRT',PR \]
\[ VS+'SCRATCH' PRIOD 'VOL=SER = ' \]
\[ X+X RS 'T',='SC \]
\[ JOB+JOB RS 'T',='SC \]
\[ X+X RS 'VOLSER',VS \]
\[ X+? \"\text{FMV X} \]
\[ X+DIM+X \]
\[ JOB+JOB RCCT X[2;] \]
\[ AG:LAB+'1' PRIOD 'LABEL ?' \]
\[ +(1=1LAB)/WR \]
\[ $(0=(1LAB)(199))/+(1=1'0','',+1'LABEL ERROR'') \]
\[ WS+', PRIOD 'WSID ?' \]
\[ TSO 'ATTRIB ALST RECFM(F B S) LRECL(80) BLKSIZE(4240) DSORG(PS)' \]
\[ TSO 'ALLOC DA(APL.';WS,'') NEW SP(1 2) TRACKS USING(ALST)' \]
\[ TSO 'FREE ATTRLIST (ALST)' \]
\[ TAPRD=x \]
\[ TAPRD[4;]+80+(DTB TAPRD[4;]),WS,' \]
\[ TAPRD[9;]+80+(DTB TAPRD[9;]),LAB \]
\[ JOB+JOB,[1] 2 0 +TAPRD \]
\[ AG \]
\[ WR:'JOB.CNTL' WRITECARDS JOB \]
\[ TSO 'FREE ALL' \]
\[ TSO 'SUB JOB.CNTL' \]
\[ TSO 'DEL JOB.CNTL' \]
TAPSAV

V TAPSAV;WS;LAB;TAPSAV;JOB;X
[1] LST*"'
[2] DIM=D TAPSAV
[3] JOB=1 80 +TAPSAV
[4] X*'?') EVN TAPSAV
[5] SC*+'A' PRIOD 'SYSOUT CLASS ='
[6] PR*+'RPUBPRT' PRIOD 'PRINTER ='
[7] X*X RS 'RPUBPRT".PR
[8] VS*+'SCRATCH' PRIOD 'VOL=SER ='
[9] X*X RS '=T"'".SC
[10] JOB+JOB RS '*T..',".SC
[12] X*'?') PMV X
[13] X=DIe+X
[14] JOB+JOB RCCT X[2;]
[16] +(0=WS)/WR
[17] LST=LST RCCT WS
[18] LAB*1+1*oLST
[19] 'LABEL '.xLAB
[20] *(LAB=99)/+*1+"'0'"."#"'LABEL ERROR'
[21] TAPSAV+X
[22] TAPSAV[4;]+80+(DTB TAPSAV[4;])WS,''
[23] TAPSAV[9;]+80+(DTB TAPSAV[9;])xLAB
[24] JOB+JOB,[1] 2 0 +TAPSAV
[25] A=AG
[26] WR:'JOB.CNTL' WRITECARDS JOB
[27] TSO 'FREE ALL'
[28] TSO 'SUB JOB.CNTL'
[29] TSO 'DEL JOB.CNTL'
[30] '"

THROUGHPUT

V THROUGHPUT
[1] PACKETPERIOD*0.5+LM+PRI 'NORMALIZED THROUGHPUT: '
[2] SPEC

V

TRANSMIT

V TRANSMIT
[1] ADB=ADD=0
[2] +RESUME/RESUME
[5] +NA=0)ABUFEMPTY
[6] ADB=1
[7] ACCBUF[;STATION;]+(5,1,ABUFSIZE)P(5,ABUFSIZE)+ 0 1 +A
[10] STATION+STATION+1

THROUGHPUT

V THROUGHPUT
[1] PACKETPERIOD+0.5+LM+PRI 'NORMALIZED THROUGHPUT: '
[2] SPEC

V

TRANSMIT

V TRANSMIT
[1] ADB=ADD=0
[2] +RESUME/RESUME
[5] +NA=0)ABUFEMPTY
[6] ADB=1
[7] ACCBUF[;STATION;]+(5,1,ABUFSIZE)P(5,ABUFSIZE)+ 0 1 +A
[10] STATION+STATION+1
[12] +SW/L1
[16] +(STATION=A[2;1])/GOAHEAD
[17] REMOVE
[18] RESUME:RESUME+0
[19] +NP/L1
[20] D+(5.,DBUFSIZE)pDELBUF[;STATION;]
[21] *(DBUFSIZE-I+1++/O=D[1;])/"DBUFINCR" 
[22] A[2,3;1]+(S+1),H
[23] D[;I]+A[;1]
[24] DELBUF[;STATION;]+(5,1,EBFSIZE)pD
[26] +L1
[27] GOAHEAD:D+(5,EBFSIZE)pDELBUF[;STATION;]
[29] *(DBUFSIZE-I+1++/O=D[1;])/"DBUFINCR" 
[31] NEXT[2;STATION]+NEXT[3;STATION]TIME+TAU
[32] DELBUF[;STATION;]+(5,1,EBFSIZE)pD
[33] +L1
[34] ABUFEMPTJ:A+(5,EBFSIZE)pDELBUF[;STATION;]
[35] RDB+1
[36] NA++/O=A[1;]
[37] +(NA=O)/EMPTY
[38] +(WHAT=NEXT[2;STATION]>TIME)/L2
[39] DELBUF[;STATION;]+(5,1,EBFSIZE)p(5,EBFSIZE)+ 0 1 +A
[40] +TRANSPORT
[41] EMPTY:NEXT[2 3 ;STATION]+ENDTIME,-ENDTIME
[42] L1:*DEBUG/'DISPLAY'
[43] +0
[44] L2:NEXT[3;STATION]+ENDTIME
[45] *DEBUG/'DISPLAY'

TRANSMITLP

\[\begin{aligned}
\text{TRANSMITLP} \\
\text{ADB=RDB=0} \\
\text{+RESUME/RESUME} \\
\text{A+(5,ABFSIZE)pACCBUF[;STATION;]} \\
\text{NA++/O=A[1;]} \\
\text{+(NA=0)/ABUFEMPTJ} \\
\text{RDB+1} \\
\text{ACCBUF[;STATION;]+(5,1,ABFSIZE)p(5,ABFSIZE)+ 0 1 +A} \\
\text{TRANSPORT:SW+*(A[2;1]=S+1)*(TIME*A[5;1]+AREC)*STATION=S} \\
\text{NEXT[2 3 ;STATION]+TIME+A[3;1]*SW=0} \\
\text{STATION+STATION+1} \\
\text{STATION=STATION-S*STATION>S} \\
\text{+SW/L1} \\
\text{*(ADB$SW1)'/WAB+WAB,TIME=A[5;1]'} \\
\text{*(RDB$SW1)'/WRB+WRB,TIME=A[5;1]+TAU'} \\
\text{A[5;1]+TIME} \\
\text{+(STATION=A[2;1])/GOAHEAD} \\
\text{REMOVE} \\
\text{RESUME:RESUME+0}
\end{aligned}\]
TRANSMITARP

TRANSMITARP

ADB=RDB=0

+RESUMEB/RESUMEB

A+(5, DBUFSIZE) pDELBUF[; STATION ;]

NA++/0xA[1;]

+(NA=0)/DELBUFEMPTY

+(WHAT NEXT[2; STATION]>TIME)/L2

RDB=1

DELBUF[; STATION ;]+(5, DBUFSIZE) p(5, DBUFSIZE)+ 0 1 +A


NEXT[2 3 ; STATION ;]+TIME+A[3; 1] x SW=0

STATION+STATION+1

STATION+STATION-S x STATION=S

SW/L1

+ (ADB x SW) /'WAB+WAB,TIME-A[5; 1] '

+ (RDB x SW) /'WRB+WRB,TIME-A[5; 1]+TAU'

A[5; 1]+TIME

-(STATION A[2; 1])/GOAHEAD

REMOVE

RESUMEB : RESUMEB = 0

-NPF/L1

D+(5, DBUFSIZE) pDELBUF[; STATION ;]

+(DBUFSIZE<1+1+/0xD[1;]) /'DBUFINCR'

A[2; 3; 1]=+ (S+1), H

D[; I]=A[; 1]

DELBUF[; STATION ;]+(5, 1, DBUFSIZE) pD

NEXT[2 ; STATION ;]+NEXT[3; STATION ;] TIME+TAU

+L1

GOAHEAD: D+(5, DBUFSIZE) pDELBUF[; STATION ;]

UTIL[STATION ;]+UTIL[STATION ;]+A[3; 1]

+(DBUFSIZE<1+1+/0xD[1;]) /'DBUFINCR'

D[; I]=A[; 1]

NEXT[2; STATION ;]+NEXT[3; STATION ;] TIME+TAU

DELBUF[; STATION ;]+(5, 1, DBUFSIZE) pD

+L1

ABUFEMPTY: A+(5, DBUFSIZE) pDELBUF[; STATION ;]

RDB+1

NA++/0x A[1;]

+(NA=0)/EMPTY

+(WHAT NEXT[2; STATION]>TIME)/L2

DELBUF[; STATION ;]+(5, DBUFSIZE) p(5, DBUFSIZE)+ 0 1 +A

TRANSPORT

EMPTY : NEXT[2 3 ; STATION ;]+ENDTIME,-ENDTIME

L1 : DEBUG /'DISPLAY'

+0

L2 : NEXT[3; STATION ;]+ENDTIME

* DEBUG /'DISPLAY'
[26] NEXT[2;STATION]+NEXT[3;STATION][TIME+TAU
[27] -L1
[28] GOAHEAD:D+(5, DBUFSIZE) p DELBUP[; STATION;]
[29] UTIL[STATION;]+UTIL[STATION;]+A[3;1]
[30] *(DBUFSIZE<1+1+/0×D[1;])/′DBUFSIZE′
[31] D[;1]+A[;1]
[32] NEXT[2;STATION]+NEXT[3;STATION][TIME+TAU
[33] DELBUP[; STATION;]+(5, DBUFSIZE) p D
[34] →L1
[35] DELBUF EMPTY: A+(5, ABUFSIZE) p ACCBUF[; STATION;]
[36] ADB+1
[37] NA++/0×A[1;]
[38] →(NA=0)/EMPTY
[39] ACCBUF[; STATION;]+(5, ABUFSIZE) p (5, ABUFSIZE)+ 0 1 +A
[40] →TRANSPORT
[41] EMPTY: NEXT[2 3 ; STATION]+ENDTIME, -ENDTIME
[42] L1:*DEBUG/′DISPLAY′
[43] →0
[44] L2:NEXT[3; STATION]+-ENDTIME
[45] *DEBUG/′DISPLAY′

TSO

\[ z+TSA X; Y \]
[1] \( z+100 \); SVO ′Y′
[2] \( →(0×1+p X)/C M D \)
[3] LOOP:+(0=p X+DTB \( \backslash o) /0
[5] Z+Y
[6] ′RC = ′, \( ∪ Y \)
[7] →LOOP
[8] CMD:Y+X
[9] Z+Y

\[ z=M \] UNIFORM \( R; MAX \)
[1] \( z+(R/R)+>((I/R)=L/R)×(Z+?R×MAX)+MAX+2×30 \)

\[ z=USERID; X \]
[1] \( z=((7-p X)+′K00′) , X+4AI[; IO] \)

\[ UTILIZATION \]

\[ Utilization \]
[1] ′′
[2] ((1,19) p′RING UTILIZATION = ′) CCCT 8 3
\( ∀(N R, 5) p(S×NR+[S+5]×UTIL)×TIME \)
[3] ′AVERAGE = ′, 8 3 ∀(+/UTIL)×S×TIME
[4] ′′
-168-

\[
((1, 21) p'\text{ACCESS UTILIZATION = ') } CCCT 8 3
\]

\[
\n((\text{NR}, 5) p(5 x \text{WR} + [S + 5] + \text{ACCESS}) + \text{TIME})
\]

\[
'\text{AVERAGE = '), 8 3 (+/\text{ACCESS}) + S x \text{TIME}'
\]

\[
'} \text{THROUGHPUT = '), \text{THURPUT} + \text{TIME}
\]

**VECMAT**

\[
R + \text{VECMAT } X; A; B; C; D; z I O; CR
\]

1. VECTOR-TO-MATRIX CONVERSION (DELMITERS \(\to\) CR)
2. A FAST BUT SPACE CONSUMING VERSION OF VECMAT
3. z I O + 1
4. CR = '
5. *('!; 'y)/'CR+'!; '''
6. R + ((pA).C) r((B - 2; C + [(B + A - 1 + (D + pY) + 1 + pY) / (D + Y = C - R)]) / Y = , Y

**WRITECARDS**

\[
Z + D S N \text{ WRITECARDS } M; T S O; R E C; C T L; X; I
\]

1. X = 100 D S V O 'TSO'
2. + (0 x + T SO) / NOSHARE
3. T S O + 'FREE FI(SYSOUTB) ATTRLIST(FB)'
4. T S O + 'ATTR FB B L K S I Z E (800) L R E C L (800) REC FM(F B S)'
5. T S O + 'ALLOC FI(SYSOUTB) NEW SP(1 1) TRACKS USING(FB)
   DA(''DSN', '')'
6. + (0 x + T S O) / T SO ERR
7. R E C + 'SYSOUTB (256'
8. C T L + R B C , 'CTL'
9. X = 111 D S V O 2 3 p'RECCTL'
10. + (0 x + C T L) / NOSHARE
11. L: R B C + 8 0 + M[I O;]
12. + (O X + C T L) / I O ERROR
13. + (O X + p M + 1 O + M) / L
14. 'DONE... '
15. Z + S V R 2 3 p'RECCTL'
16. T S O + 'FREE FI(SYSOUTB)'
17. + 0
18. I O ERROR: (R C 11) + D S N[1 1 2 1 1 1 7 4 4 0 4 4 1 4 4 2 4 4 3 4 4 4 4 5 9 1 3
   9 3 7 x ; I ), ' RC = 'X
19. H O L D E R R: S A WRITECARDS + RESTART
20. R E S T A R T: + L
22. X HOLD E R R
23. T S O ERR: 'TSO ERROR ', 'X
24. X HOLD E R R

\[
XX
\]

1. FIND INDICES OF SUBSTRING Y IN VECTOR X
2. R + 0
3. + ((O = pY) v (pX + X) < pY + Y) / 0
\[ R^* = (J+X)/1(\text{pX}) + J_1 - pY \]
\[ + (0 = pJ + 1 + Y)/0 \]
\[ R^* = (X[Ro. + (\text{pIO}) + 1pJ] + j = J)/R \]

\[ \text{XSNO} \]

\[ R^* X \text{XSNO } Y; Z \]
[1] a FIND INDICES OF SUBSTRING Y IN VECTOR X WITHOUT OVERLAP
[2] Z + X \text{IS } Y;, Y
[3] R^* = (0 = pZ) + Z
[4] L1: = (0 = pZ + 1 + Z)/0
[5] R^* = R^* = ((1 + Z) - 1 + R)_pY + Z
[6] \text{L1}

\[ \text{YES} \]

\[ R^* \text{YES Q} \]
[1] \text{Q} + Q = Q
[2] R^* = Y^ = 1 + R^* = (\text{Z} = \text{R}) \text{AV} \text{R} = \text{Z} + (\text{pR}) + \text{Q})/R^*, t

\[ \text{TAPSAV} \]

// K702110 JOB SCHL, MSGLEVEL=(1,1), MSGCLASS=T 
// ROUTE PRINT RPUBPRT 
// EXEC PGM=IEBGENER 
// SYSUT1 DD DSN=K702110.APL. 
// DISP=OLD 
// SYSUT2 DD UNIT=TAPE, VOL=SER=VOLSBR, DISP=(OLD, PASS), 
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=4240), 
// DSN=APLSAV, 
// LABEL= 
// SYSPRINT DD SYSOUT=T 
// SYSIN DD DUMMY

\[ \text{TAPRD} \]

// K702110 JOB SCHL, MSGLEVEL=(1,1), MSGCLASS=T 
// ROUTE PRINT RPUBPRT 
// EXEC PGM=IEBGENER 
// SYSUT2 DD DSN=K702110.APL. 
// DISP=OLD 
// SYSUT1 DD UNIT=TAPE, VOL=SER=VOLSBR, DISP=(OLD, PASS), 
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=4240), 
// DSN=APLSAV, 
// LABEL= 
// SYSPRINT DD SYSOUT=T 
// SYSIN DD DUMMY
BATCHJCL
//K702110+ JOB APL,CLASS=J,MSGCLASS=A,TIME=TI,MSGLEVEL=(1,1),
// REGION=RK
/*ROUTE PRINT RPUBPRNT
//TMP EXEC PGM=IKJEFT01,DYNAMNBR=5,REGION=RK
//SYSTSPRT DD SYSOUT=A
//APLPRINT DD SYSOUT=A
//SYSTSIN DD *
PROFILE PREFIX(K702110)
VSAPL SH(20) AUE( )
//APLIN DD *

TALPHA95
12.706 4.303 3.182 2.776 2.571 2.447 2.365 2.306 2.262 2.228 2.201 2.179
2.145 2.131 2.12 2.11 2.101 2.093 2.086 2.08 2.074 2.069 2.064 2.06
2.052 2.048 2.045 2.042
LEBENSLAUF


Herbst 1981.