

# Design and Analysis of a Pipeline Ring Protocol

P.C. Wong, *Senior Member, IEEE*, and Tak-Shing Peter Yum, *Senior Member, IEEE*

**Abstract** — A new distributed protocol which supports concurrent message transmissions on different ring segments in a ring network is proposed. This protocol allows the destination station to remove the message body from the ring and to issue a new token for the succeeding stations to establish another transmission in the remaining ring segment. This protocol requires only one-bit latency at each station and supports variable size messages. We derived the maximum throughput of the pipeline ring and found it to be heavily dependent on the message size distribution. The maximum throughput of a single ring for exponential messages and fixed size messages are 1.23 and 1.68 respectively; while for the double ring, the per-ring throughput is 1.7 and 3.25 respectively. Due to analytical complexity, the delay performance is obtained by simulation. Two service disciplines are compared. It is found that the Furthest Within Segment (FWS) discipline always performs better than the First Come First Serve (FCFS) discipline. The Short Message Transmission (SMT) scheme is introduced. Under bimodal traffic, it can significantly increase the ring efficiency.

## I. INTRODUCTION

Ring networks are very common in local and metropolitan area networks. In particular the Token Ring and the FDDI were developed into the IEEE 802.5 standard [1] and the ANSI X3 standard [2] respectively. The node-to-node connective nature of the ring is particularly suitable for using optical fibres and for supporting concurrent message transmissions on different ring segments. This spatial reuse ability of the ring is used in a number of single and double ring protocols [3-12]. Examples include:

1. *Slotted Rings* (e.g. Cambridge Ring [3], Orwell Ring [4])  
By allowing the destination to remove the message, the emptied slots can be immediately used by the succeeding stations.
2. *Buffer Insertion Rings* (e.g. DLCN [5], DDLCN [6])  
The ring interface contains a variable length shift register which can buffer incoming messages during the transmission of locally generated messages.
3. *Segmented Rings* (e.g. Jafari Loop [7], Leventis Double Loop [8], Circuit-switched Playthrough Rings [9,10], T-S Ring [11], Concurrent Transmission Ring (CTRing) [12])  
The ring is logically partitioned into a number of segments, each supports a message transmission. In Jafari loop, when a node wants to communicate with another node, it sends a request message to the controller via a separate control ring. The controller then sends out messages to create a communication path between the two nodes on the data ring. Leventis loop is a double ring operating in cycles which are initiated by controller stations. Forward and backward transmissions are limited by the positions of these controller

Paper approved by Inder Gopal, the Editor for Network Protocols of the IEEE Communications Society. Manuscript received May 8, 1988; revised July 31, 1991. This paper was presented at the IEEE International Conference on Communications, Philadelphia, PA, June 1988.

The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong.

IEEE Log Number 9401014.

stations. In the playthrough ring, a higher priority preemptive token *GO* circulates continuously in the ring to collect information and set up new message transmissions on non-overlapping idle ring segments. All message transmissions are continually interrupted by *GO* and so the effective transmission time is increased. In the T-S protocol, destination stations are responsible for removing messages from the ring and for initiating new message transmissions by issuing tokens downstream. In order to prevent a new message from colliding with the current transmissions, all current transmissions must finish before the arrival of that new message. This requires long messages to be cut into small units before transmission and the protocol works like a slotted ring. If the message is too small, the destination will not issue any token. Instead, a new token will be released by the source station after the current transmission just like a token ring.

These protocols can all achieve a maximum throughput greater than one but on the other hand, they have one or more of the following limitations:

1. Messages have to be cut into smaller units for transmission. This adds overheads in transmission and processing (buffer insertion ring, slotted ring and T-S ring).
2. Centralized ring access control by means of a separate ring is needed (Jafari loop).
3. Each station has to buffer a sufficient number of inbound characters to check the address to determine if the passing message needs to be removed or not. Extra delay (say, 30 bits) is introduced at each station which might affect the ring efficiency if the number of stations is large (buffer insert rings, playthrough ring, Jafari loop and T-S ring).
4. Message transmission is periodically interrupted by a higher priority control token. The ring interface design is complicated and the effective message transmission time is prolonged (playthrough ring).
5. Only the double ring structure is supported (Leventis double loop and Jafari loop).

In the following, we describe the pipeline ring protocol [13] which attempts to avoid the above limitations. This protocol performs like an ordinary token ring at light load but gives significantly higher throughput at heavy load by supporting concurrent transmissions on different ring segments. It operates in cycles and provides a fair access to all stations by rotating the *start-up* station status among the stations on a round-robin basis. The CTRing is similar to the pipeline ring with the absence of the reset phase. It will be discussed in Section V.

## II. PROTOCOL DESCRIPTION

Consider a ring network connecting  $N$  stations as shown in figure 1. When the ring is operating, each station can either be in the *bypass* mode or in the *transmit/receive* mode (figure 2).

In the *bypass* mode, each bit arriving at the station is simply copied out onto the ring. One-bit latency is introduced to the messages passing by. A station in the *transmit/receive* mode can be transmitting messages, receiving messages, or both.

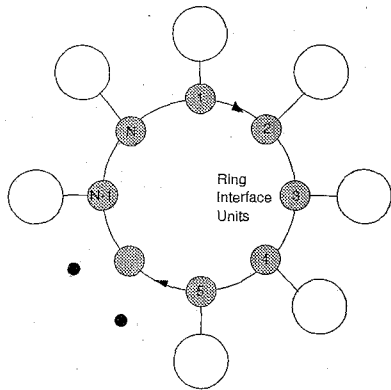


Fig. 1. Physical ring topology.

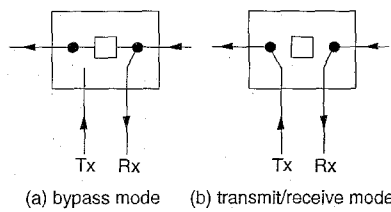


Fig. 2. Ring interface operating modes.

Token Types	Token Fields						
	S D	T	S	Limit	F	C	ED
START-UP		1	1	NIL	-	-	
SEGMENT		1	0	XX..XX	1	1	
RESET		0	0	XX..XX	-	-	

**Legend:**

- a. *SD, ED* : token start-delimiter and end-delimiter (1 byte each)
- b. *Limit* : station address marking the end of segment (1 byte)
- c. *F* : token status  
*F=1* - a free token  
*F=0* - a busy token (or indication of a message header)
- d. *C* : class of messages  
*C=1* - single destination messages or broadcast messages (to be determined by the forth coming address field)  
*C=0* - messages with multiple destinations

Fig. 3a. Tokens formats

Message Header								Message Body		
S D	T	S	Limit	F	C	Destination Address	Source Address	Message Data	CRC	ED
1	0		XX..XX	0	C	DD..DD	SS..SS	AA..AA	XX	

Fig. 3b. Format of messages (header plus body)

Message Header							
S D	T	S	Limit	F	C	Destination Address	E D
1	0		XX..XX	0	C	DD..DD	

Fig. 3c. Format of dummy message header

There are three types of control tokens: the *START-UP* token, the *SEGMENT* token and the *RESET* token. Their formats are shown in figure 3a. The *T* and the *S* bits are used to identify the tokens and the messages respectively. The *Limit* field stores the station address which marks the end of a free segment in the ring. The *F* bit indicates whether a *SEGMENT* token is free or not. The *C* bit is used to identify messages with multiple destinations.

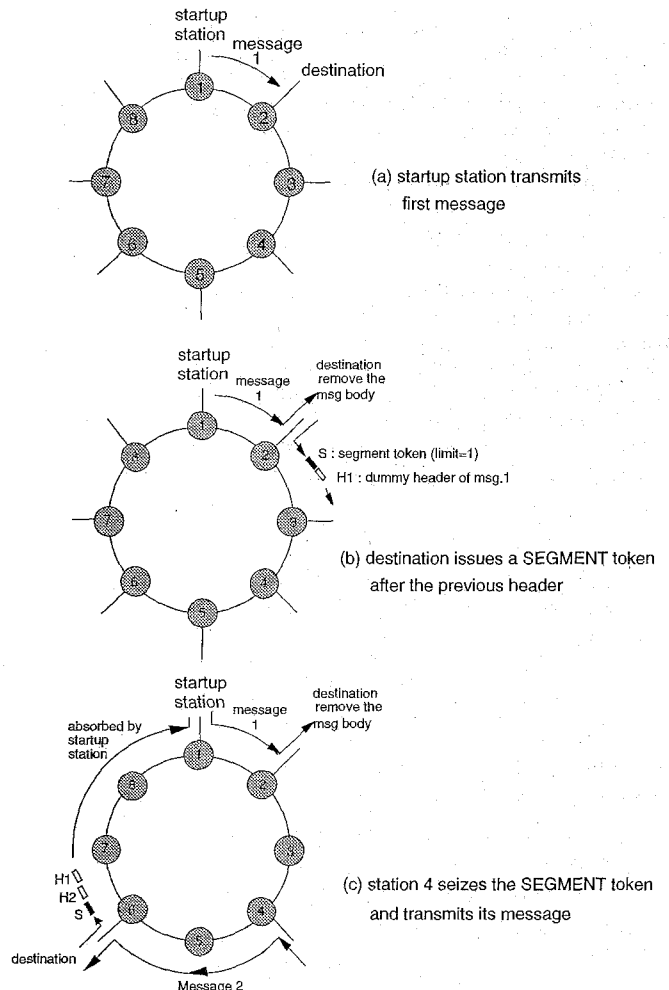


Fig. 4. Step-by-step pipeline ring operation.

To illustrate the ring operation, we use an example shown in figure 4. When the ring is idle, a *START-UP* token circulates in the ring and all stations are in the *bypass* mode, i.e., monitoring the ring channel bit by bit. Let station 1 has a message to transmit to station 2. When station 1 detects the *START-UP* token (i.e. *S=1*), it changes to the *transmit/receive* mode and converts the token into a message header (figure 3b) by setting the *S*-bit to 0, *Limit* field to 1 (its own address), *F*-bit to 0, *C*-bit to 0 or 1 depending on the message class and appends the destination address and the source address fields. The message body is then followed. Station 1 assumes the status of the *start-up* station and starts a busy cycle in the ring.

Succeeding stations relay the message on a bit-by-bit basis if they are not the destination and introduces one bit latency per station. When station 2 (the destination of the message) detects its own address, it changes to the *transmit/receive* mode and removes the message body from the ring. At the same time, it appends an end-delimiter (*ED*) to terminate the message header.

This dummy message header will be ignored by all the downstream stations until it reaches the *start-up* station and gets absorbed.

The *Limit* field in the message header is recorded by the destination station. If that station has a message with a destination that is within the free segment limit, it can transmit the message with a new message header attached, following the previous dummy message header. Otherwise, the destination station issues a *SEGMENT* token downstream to create more transmission segment. In figure 4b, station 2 has no message to transmit, so it issues a *SEGMENT* token downstream.

Let station 4 has a message to transmit to station 6. When station 4 detects the *SEGMENT* token and finds that its message falls within the *Limit* (i.e. station 6 is between station 4 and the *start-up* station), it converts the token to a message header and follows it with the message body. Again, when this message reaches its destination (station 6), another *SEGMENT* token will be generated and forwarded. This process continues until the *SEGMENT* token reaches the *start-up* station and gets absorbed together with all the dummy message headers. In figure 4c we show two concurrent message transmissions (1 to 2 and 4 to 6) in this busy cycle.

When the *start-up* station (station 1) finishes its transmission, it sends out a *RESET* token (with the *Limit* field set to its own address). This *RESET* token is stored and forwarded by the "still transmitting" stations without preemption until it gets back to the *start-up* station. After the *start-up* station absorbs back the *RESET* token, the ring is idle again. The *start-up* station then issues a new *START-UP* token for the succeeding stations to start a new cycle.

#### A. Broadcast and Multicast Messages Transmission

When a station wants to broadcast a message, it waits for a *START-UP* token, seizes the token and then transmits the message with the destination address set to 0. All stations will then read the message in the *bypass* mode without removing the message body. When a station wants to send a message to a number of destinations, it sets the *C*-bit to zero and writes all the destination addresses in the message header according to their order in the ring. When a *START-UP* token or a *SEGMENT* token with sufficient segment size arrives, it seizes the token and transmits the multicast message. The last destination is responsible for removing the message body.

#### B. Short Messages Transmission (SMT) Scheme

Traffic patterns in LANs are often bimodal, i.e. a mixture of short and long packets [14]. When a station has a very short message (say, a few characters) to transmit, the advantage of concurrent message transmissions is less pronounced. An optional *SMT* scheme can be used to increase the ring efficiency in this case. When a station has a short message, say the transmission time is less than the round-trip propagation delay on the ring, it waits for a *START-UP* token, seizes the token and transmits the packet with the *Limit* field set to zero. This setting prohibits the downstream stations from issuing *SEGMENT* tokens. After the message transmission, a new *START-UP* token is released immediately to start a new busy cycle. Short message

are hence transmitted in cycles separated from long messages. This scheme provides a fair round-robin assignment of the *start-up* station status. Its performance is given in section V.

### III. ERROR RECOVERY PROCEDURES

In the IEEE 802.5 token ring, the destination station can send an implicit acknowledgement back to the source station by setting a bit at the end of the message. The message will return to the source station after a round-trip delay and be removed from the ring. In the pipeline ring, errors are handled by the link-level protocols using explicit acknowledgement frames similar to those used in Ethernet. We now describe the recovery procedures for losing tokens and detecting unwarrented tokens:

#### A. Loss of *START-UP* token

When a station detects that there is no passage of tokens or messages for a specific time interval, it may conclude that the *START-UP* token is lost. It then switches into the *transmit/receive* mode and sends out a *RESET* token with its own address stored in the *Limit* field. When the *RESET* token returns, the station can issue a new *START-UP* token downstream. Since a number of stations may issue *RESET* tokens independently, the *Limit* field in each of the received *RESET* tokens is checked. If the *Limit* field is larger than the station's own address, the *RESET* token is discarded. Otherwise, the *RESET* token is forwarded. In this way, only the station with the smallest address will retrieve its *RESET* token.

#### B. Loss of *RESET* token

After a station issues a *RESET* token, it starts a timer. If it does not retrieve the issued *RESET* token within the time-out period, it reissues the *RESET* token.

#### C. Loss of *SEGMENT* token

The loss of *SEGMENT* token only prohibits possible concurrent transmission in a particular cycle. Since the ring is reset after each cycle, no particular recovery procedure is needed.

#### D. Detection of unwarrented tokens

If there are multiple *START-UP* and/or multiple *SEGMENT* tokens with overlapping segment limits in the ring, message transmissions may overlap. To cope with this situation, when a transmitting station receives a message which is not for itself, it terminates its own transmission immediately by appending an End-Delimiter (*ED*) and issues a *RESET* token to the ring. If multiple *RESET* tokens are generated, they are handled as in Case A. If a *START-UP* or a *SEGMENT* token is received while a station is transmitting, the token is simply discarded.

### IV. THROUGHPUT ANALYSIS

Consider an  $N$ -node single ring. To find the maximum throughput, we assume that the ring is under heavy traffic and each station always has a message ready for transmission. We assume that the probability that the message in station  $i$  ( $i=1,2,\dots,N$ ) destined for station  $j$  ( $j \neq i$ ) is  $1/(N-1)$ , independent of  $i$  and  $j$ . Let  $K$  be a random variable representing the total number

of message transmissions in a cycle and  $T(K)$  be the length of a  $K$ -transmission cycle (see below). In the following, we shall use a 3-node ring to illustrate the analysis.

A. 3-node Single Ring

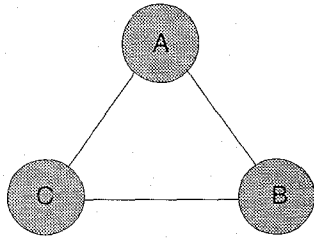


Fig. 5. A 3-node ring network.

Let stations A, B and C in a 3-node ring be arranged as in figure 5 and let the direction of transmission be clockwise. Let  $h_0$  be the number of hops required for the message at the *start-up* station to reach its destination. Similarly, let  $h_1$  and  $h_2$  be the number of hops required for the messages in stations one and two hops away from the *start-up* station to reach their destinations respectively. We denote  $(h_0, h_1, h_2) = (1, 1, 1)$  as state 0, or  $A_0$ . At  $A_0$ , since the hop counts for the messages in all three stations are one, three simultaneous transmissions are possible in the cycle. After the cycle, all three nodes generate new messages with hop counts equal to 1 or 2 with equal probability. Therefore, there are eight possible state transitions from  $A_0$ , indicated by  $\{(h_0, h_1, h_2) | h_0, h_1, h_2 \in \{1, 2\}\}$ , each with probability  $1/8$ .

To further illustrate the state transition process, consider  $A_7$  with  $(h_0, h_1, h_2) = (2, 2, 2)$ . Let station A be the *start-up* station. In this state, only the transmission by station A is possible. After the transmission is over, station B becomes the new *start-up* station. But station B has a message with hop count equal to 2 still not transmitted. Therefore  $h_0$  for the new cycle is 2. Similarly, station C has a message with hop count 2; but since station C is one hop away from the *start-up* station,  $h_1 = 2$ . After the message transmission, station A generates a new message with hop counts 1 or 2 with equal probability and in the new cycle, station A is 2 hops away from the *start-up* station. Hence  $h_2 = X$ , where X is equal to 1 or 2 with equal probability. The transitions of the other states are similar.

TABLE 1 : SYSTEM STATES OF THE 3-NODE PIPELINE RING

Current State	Old Hop Count			Number of transmissions	New Hop Count			Next State	Transition probability
	$h_0$	$h_1$	$h_2$		$h_0$	$h_1$	$h_2$		
$A_0$	1	1	1	3	X	X	X	$A_0$ to $A_7$	1/8
$A_1$	1	1	2	2	X	2	X	$A_2, A_3, A_6, A_7$	1/4
$A_2$	1	2	1	2	X	1	X	$A_0, A_7, A_4, A_5$	1/4
$A_3$	1	2	2	2	X	2	X	$A_2, A_3, A_6, A_7$	1/4
$A_4$	2	1	1	2	1	X	X	$A_0$ to $A_3$	1/4
$A_5$	2	1	2	1	1	2	X	$A_2, A_3$	1/2
$A_6$	2	2	1	2	2	X	X	$A_4$ to $A_7$	1/4
$A_7$	2	2	2	2	2	2	X	$A_6, A_7$	1/2

Table 1 shows the eight possible states, their associated hop counts in the present cycle, the number of simultaneous message transmissions in each state, the possible hop counts of the messages in the next cycle (an X indicates that the entry could be either 1 or 2), the possible states in the next cycle and the transition probability to each of the next states (all equal).

Figure 6 shows all the possible state transitions from cycle to cycle in the 3-node single pipeline ring. Let  $P_i$  be the probability that the system is in state  $A_i$ . The state equations are:

$$P_0 = P_1 = (P_0 + 2P_2 + 2P_4)/8$$

$$P_2 = P_3 = (3P_0 + 2P_2 + 6P_4)/8$$

$$P_4 = P_5 = (P_0 + 2P_2 + 2P_6)/8$$

$$P_6 = P_7 = (3P_0 + 2P_3 + 6P_6)/8$$

Together with  $P_0 + P_1 + \dots + P_7 = 1$ , the  $P_i$ 's are solved to be  $P_0 = P_1 = 1/16$ ,  $P_2 = P_3 = 1/8$ ,  $P_4 = P_5 = 3/32$  and  $P_6 = P_7 = 7/32$ . The probabilities that there are 1, 2 and 3 messages in transmission in a cycle are

$$P[K=1] = P_5 + P_7 = 5/16$$

$$P[K=2] = P_1 + P_2 + P_3 + P_4 + P_6 = 10/16$$

$$P[K=3] = P_0 = 1/16$$

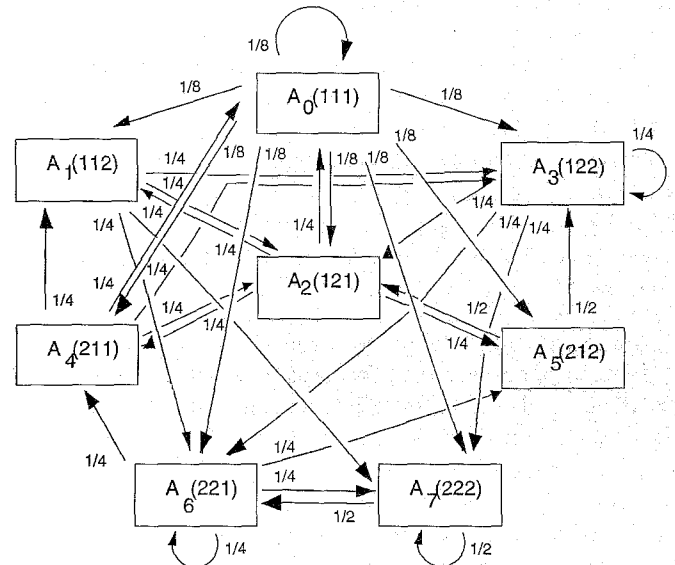


Fig. 6. State transition diagram.

In general, for an  $N$ -node ring, the number of states in the Markov chain is  $(N-1)^N$ . Due to symmetry, only  $(N-1)^{(N-1)} - 1$  simultaneous equations need to be solved. Even so, analytical solution beyond  $N=5$  is difficult. We have solved the 3 and 4 node cases and resort to simulation for larger values of  $N$ . As will be shown in section V, the performance of the pipeline ring is relatively independent of the number of stations in the ring. So the analysis for small number of stations is still useful.

B. 3-node Double Ring

In a bidirectional double ring, each station selects the ring with the smaller hop count to transmit its message. The maximum size of each transmission segment is hence reduced from  $N-1$  hops to  $L_1 = \lfloor N/2 \rfloor$  and  $L_2 = N - \lfloor N/2 \rfloor - 1$  hops for the clockwise and the anti-clockwise sub-rings respectively. As in the single ring case, we assume that the ring is under heavy traffic and each station always has a message ready for transmission at

each sub-ring. Let the throughput of the two sub-rings be denoted as  $S_1$  and  $S_2$ . Then for the two sub-rings having the same bit rate, the average throughput per ring is  $(S_1+S_2)/2$ .

Consider first the clockwise sub-ring for the 3-node double ring. Here  $L_1 = \lfloor 3/2 \rfloor = 1$  and  $L_2=1$ . Each message therefore requires only one hop to reach its destination. Hence three transmission segments can always be set up. Similarly, the number of simultaneous message transmissions in the anti-clockwise sub-ring is also 3. The average throughput per sub-ring is therefore equal to 3. To solve for the distribution of simultaneous message transmissions per cycle for double rings with four or more nodes, the same technique of enumerating all possible states used in the single ring case can be employed. Again, we limit the analytical computation to  $N \leq 4$  and resort to simulation for larger values of  $N$ .

C. Cycle Length

Since messages can be of variable sizes, short messages have to wait for long messages to complete before a cycle can end (figure 7). Let  $r$  be the ring propagation delay and  $b$  be the bit transmission time. Since the protocol introduces one-bit delay at each station, the ring walk time  $W$ , defined as the ring propagation delay plus the  $N$ -bit station delay, is  $W=r+Nb$ . Assuming the  $N$  stations are uniformly located on the ring, the walk time from one station to the next is  $W/N$ . Let the transmission time of the *START-UP* and *RESET* tokens be  $t_1$  and the transmission time of the *SEGMENT* tokens be  $t_2$ . Let the message transmission time  $X$  has a distribution function  $F_X(x)$ .

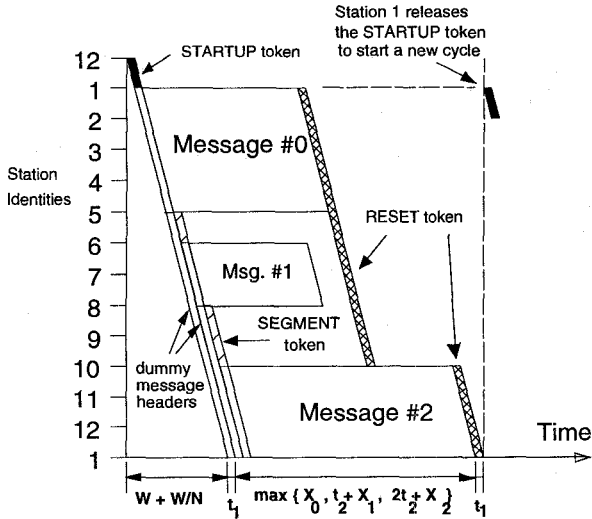


Fig. 7. A typical cycle showing the tokens and messages transmission.

It can be seen from figure 7 that the length of a cycle with  $K$  message transmissions, denoted as  $T(K)$ , consists of:

1. The walk time of a token from the last *start-up* station to the present *start-up* station,  $W/N$ .
2. The walk time of the token around the ring,  $W$ .
3. The total transmission time of the *START-UP* token and the *RESET* token,  $2t_1$ .
4. The duration of the  $K$  simultaneous message transmissions including the staggered *SEGMENT* token transmission delays.

Adding up, we have

$$T(K) = W/N + W + 2t_1 + \max\{X_0, t_2 + X_1, \dots, (K-1)t_2 + X_{K-1}\} \quad (1)$$

where  $X_0, X_1, \dots, X_{K-1}$  denote the  $K$  message transmission times with common distribution  $F_X(x)$ . Let  $Y(K) = \max\{\cdot\}$  in (1), then from [15] it can be shown that

$$F_{Y(K)}(x) = \prod_{j=0}^{K-1} F_X(x - jt_2) \quad (2)$$

and

$$E[Y(K)] = \int_0^\infty 1 - F_{Y(K)}(x) dx \quad (3)$$

D. Maximum throughput  $S_{max}$

The maximum throughput of the pipeline ring  $S_{max}$  is equal to the ratio of expected amount of message transmissions per cycle to the expected cycle length or

$$S_{max} = \frac{\sum_{k=1}^N k E[X] P[K=k]}{\sum_{k=1}^N \{W/N + W + 2t_1 + E[Y(k)]\} P[K=k]} \quad (4)$$

For fixed size messages of length  $B$ ,

$$E[X] = B$$

$$E[Y(k)] = (k-1)t_2 + B.$$

Hence

$$S_{max} = \frac{B \sum_{k=1}^N k P[K=k]}{\sum_{k=1}^N \{W/N + W + 2t_1 + B + (k-1)t_2\} P[K=k]} \quad (5)$$

For exponentially distributed messages with mean transmission time  $1/\mu$ , we have

$$E[Y(k)] = \int_0^\infty 1 - F_{Y(k)}(x) dx$$

$$= \int_0^\infty 1 - F_{Y(k-1)}(x) dx + \int_0^\infty F_{Y(k-1)}(x) - F_{Y(k)}(x) dx \quad (6)$$

Denote the second integrand as  $D_{k-1}$ , we have

$$E[Y(k)] = E[Y(k-1)] + \int_0^\infty D_{k-1} dx \quad (7)$$

Repeated use of (7) and note that  $E[Y(1)] = 1/\mu$ , we have eventually

$$E[Y(k)] = \frac{1}{\mu} + \sum_{i=1}^{k-1} \int_0^\infty D_i dx \quad (8)$$

We now expand the  $D_i$ 's into a series:

$$D_i = F_{Y(i)}(x) - F_{Y(i+1)}(x)$$

$$= \prod_{j=0}^{i-1} (1 - e^{-\mu x} e^{j\mu t_2}) - \prod_{j=0}^i (1 - e^{-\mu x} e^{j\mu t_2})$$

$$= e^{-\mu x} e^{i\mu t_2} \prod_{j=0}^{i-1} (1 - e^{-\mu x} e^{j\mu t_2})$$

$$= e^{-\mu x} e^{i\mu t_2} \sum_{j=0}^i (-1)^j A_i(j) e^{-j\mu x} \quad (9)$$

The coefficients  $A_i(j)$  are evaluated as follows. Let  $\Phi_i(j) = \{\Omega_1, \Omega_2, \dots\}$  be the set of all  $j$ -element subsets of  $\{0, 1, 2, \dots, i-1\}$  where  $\Omega_1, \Omega_2, \dots, \binom{i}{j}$  (in total) are the individual subsets and  $g_n$  be

the sum of all elements in  $\Omega_n$ . (Note that  $\Omega_i$ 's and  $g_n$ 's are all functions of  $j$ .) Then it can be shown that

$$A_i(j) = \sum_{n=1}^{\binom{i}{j}} e^{i\mu_2 g_n} \quad (10)$$

Integrating (9), we have

$$\int_0^\infty D_i dx = e^{i\mu_2} \int_0^\infty \sum_{j=0}^i (-1)^j A_i(j) e^{-(j+1)\mu x} dx$$

$$= e^{i\mu_2} \sum_{j=0}^i (-1)^j \frac{A_i(j)}{(j+1)\mu} \quad (11)$$

Substituting (11) back to (8), we have

$$E[Y(k)] = \frac{1}{\mu} + \sum_{i=1}^{k-1} e^{i\mu_2} \sum_{j=0}^i (-1)^j \frac{A_i(j)}{(j+1)\mu} \quad (12)$$

Substituting (12) into (4),  $S_{max}$  can be explicitly evaluated.

V. RESULTS AND DISCUSSIONS

A. Analytical Results for 4-node Rings

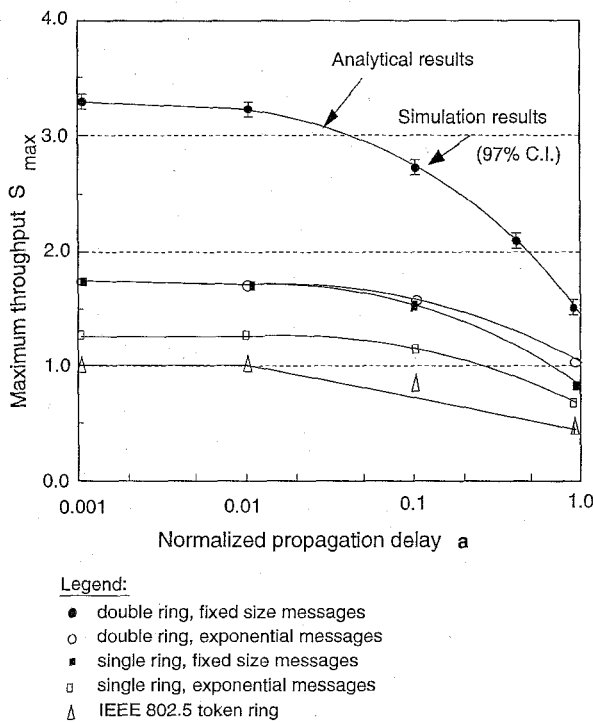


Fig. 8. Maximum throughput vs. normalization delay  $a$  on a 4-node ring, FCFS discipline.

In the following discussions, we assume that the expected message size is 16 kbits for both exponential and fixed size messages. The channel data rate is 5 Mbps for each ring and the propagation delay in the cable is 5  $\mu$ s/km. Figure 8 shows the maximum throughput  $S_{max}$  of 4-node single and double pipeline rings with different values of normalized propagation delay  $a$  where  $a=r/E[X]$  is defined as the ratio of the round-trip ring delay to the expected message transmission time. The maximum

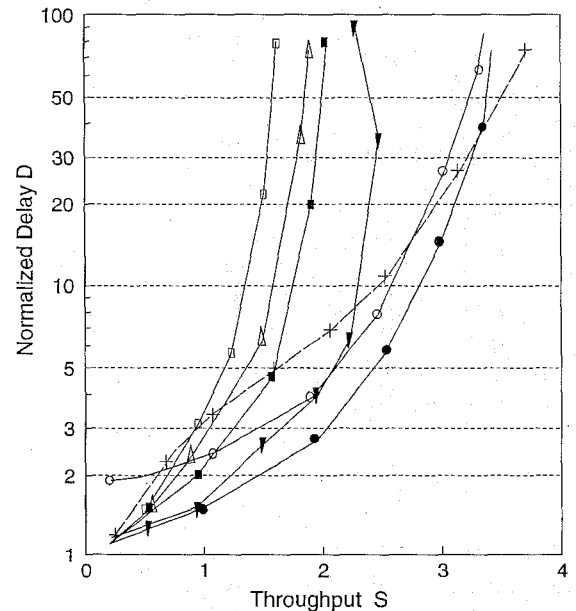
throughput of a single ring for exponential messages and fixed size messages are 1.23 and 1.68 respectively. In the double ring, each station selects the ring with the smaller hop count to transmit

its message. The possibility of concurrent transmissions is therefore increased. The per-ring throughput for exponential and fixed size messages are 1.7 and 3.25 respectively. The total throughput is twice the per-ring throughput, i.e., 3.4 and 6.5 respectively. The maximum throughput of the pipeline rings therefore is always larger than that of the IEEE 802.5 token ring, which is 1. The throughput decreases as  $a$  (or the ring size) increases. The ring throughput with fixed size messages only is larger than that with exponential messages because there is no need to wait for the longest message to complete transmission before a new cycle can be started. As shown, the analytical results agree very closely with the simulation results.

B. Simulation Results

In the simulation model, we assume that the mean message size is 16 kbits for both exponential and fixed size messages. The message arrivals are assumed to be Poisson. The number of stations is 15. The data rate on each ring is 5 Mbps and the propagation delay in the cable is 5  $\mu$ s/km. The expected message transmission time  $E[X]$  therefore is 16 kbits/5Mbps, or 3.2 ms. With an assumed 6.4 km cable length, the normalized propagation delay  $a$  is 0.01.

1) Comparison with Other Schemes:



Legend:  
 ■ IEEE 802.5 double token ring  
 □ Jafari loop  
 △ Leventis double loop  
 ○ double playthrough ring  
 + double register buffer insertion ring  
 ▼ double CTRing  
 ● double pipeline ring

Fig. 9. Delay comparisons, double rings,  $N = 15$ ,  $a = 0.01$ , exponentially distributed messages, FCFS discipline.

Figure 9 compares the delay performance of pipeline ring, IEEE 802.5 token ring, CTRing, buffer insertion ring, slotted ring, Leventis loop, Jafari loop and playthrough ring. Slotted Ring and T-S ring are not shown for comparison as these rings require data messages to be cut into small units. The message size is exponentially distributed and the destination of a message is uniformly distributed over the ring. The service discipline is First-Come-First-Serve (*FCFS*). A double ring structure is assumed since it is supported by all protocols. Each station selects the ring with the smaller hop count to transmit its message. The *total ring throughput* is used for comparison. The results of the Jafari Loop, the playthrough ring and the buffer insertion ring are quoted from [[7], [10] and [16] respectively. Since other protocols are similar in operation with the pipeline ring, we modified our simulation program and obtained their simulation results under the same set of ring parameters.

As shown, the pipeline ring gives the best overall performance. It has a maximum throughput of about 3.5 (2 times the 1.75 per-ring throughput) and the lowest delay among all the protocols shown up to a throughput of 3.4.

Since the Jafari loop uses only one ring for data transmission, the throughput is definitely lower than that of the other double ring schemes. The maximum throughput is about 1.5 because the number of simultaneous transmissions is limited by the uniform message destination distribution assumed and the *FCFS* discipline. If we allow the Jafari loop to configure transmissions in different directions, the maximum throughput can be slightly increased to 1.75.

Although the Leventis loop can set up multiple messages transmissions in both rings, the throughput is only around 2, which is similar to that of a double token ring. The throughput is not as high as the double pipeline ring because it needs to wait for the longest message in both "loops" to complete transmission before a new cycle can start. It is therefore better to have independent cycles for each ring (as in the pipeline ring case) so as to reduce the cycle length.

Since there is a continuous interruption of control messages, the delay performance of the playthrough ring at light load is slightly worse than that of the other ring protocols. However, it can achieve a higher maximum throughput of 3.3.

The buffer insertion ring has significantly higher delay than the other rings in light to medium traffic conditions. This delay increases with the number of stations in the ring. It can however achieve a maximum throughput close to 4 at very heavy traffic conditions.

The working principle of the CTRing is similar to that of the pipeline ring and was proposed roughly at the same time. Like the pipeline ring, only the message body is removed and the truncated message header is removed by the original source station. Instead of using special segment tokens, it uses the source and destination addresses in the truncated header to determine the transmission limit and hence can use standard IEEE 802.5 token and message formats. The main difference from the pipeline ring is that there is no reset cycle. When a station finishes its transmission, it will not notify other stations. Instead, it waits in *transmit/receive* mode until it retrieves its own message header. Therefore if a station receives the dummy headers of

other stations and gets a token, it can only initiate new transmissions within the calculated transmission limit even if other stations have already finished their transmissions and the entire ring is available for a new cycle of concurrent transmissions. This reduces the possibility of concurrent transmissions and we found from simulation that the total maximum throughput of a double CTRing is only 2.5. As the load increases further beyond the limit, more stations have backlogged messages and the possibility that a transmission can be initiated within a limit is increased. But on the otherhand, the possibility that a new cycle can be restarted for more concurrent transmissions is also reduced. Simulation results in figure 9 show that the throughput is slightly reduced at heavy loads.

## 2) Number of Stations:

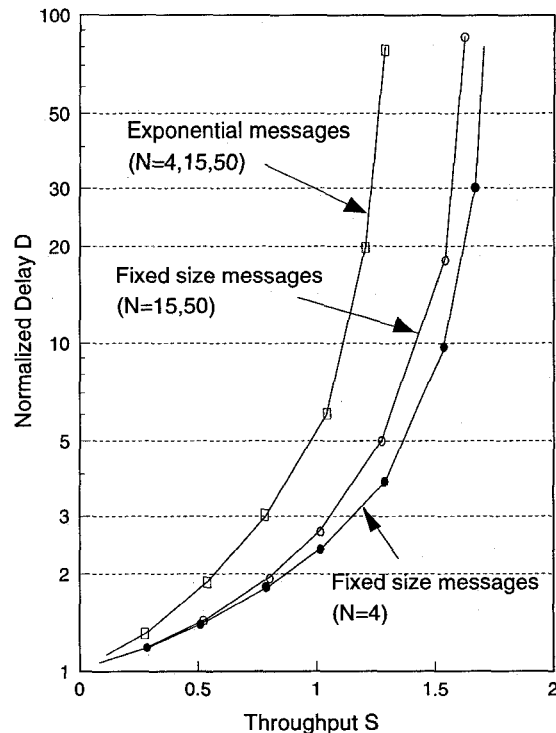


Fig. 10. Delay-throughput characteristics of single pipeline rings,  $a = 0.01$  ( $N$  is the number of stations).

Figure 10 shows the simulation results of the normalized average delay  $D$  for three single pipeline rings with 4, 15 and 50 nodes respectively. It is seen that the delay is essentially independent of the number of stations for  $N \geq 4$ .

## 3) Different Service Disciplines:

Figure 11 shows the average delay of a 15-node single ring and a 15-node double ring, both with fixed size messages. The per-ring throughput is shown. Two service disciplines are compared. The First Come First Serve (*FCFS*) discipline is self-explanatory. The Furthest Within Segment (*FWS*) discipline selects the message in the station output queue that has the furthest destination within the transmission segment. It is shown that *FWS* gives higher throughput and lower delay in comparison to *FCFS*. The maximum throughput values are 1.68 and 1.87 for the single pipeline ring with *FCFS* and *FWS* disciplines respectively. The maximum per-ring throughput for the double

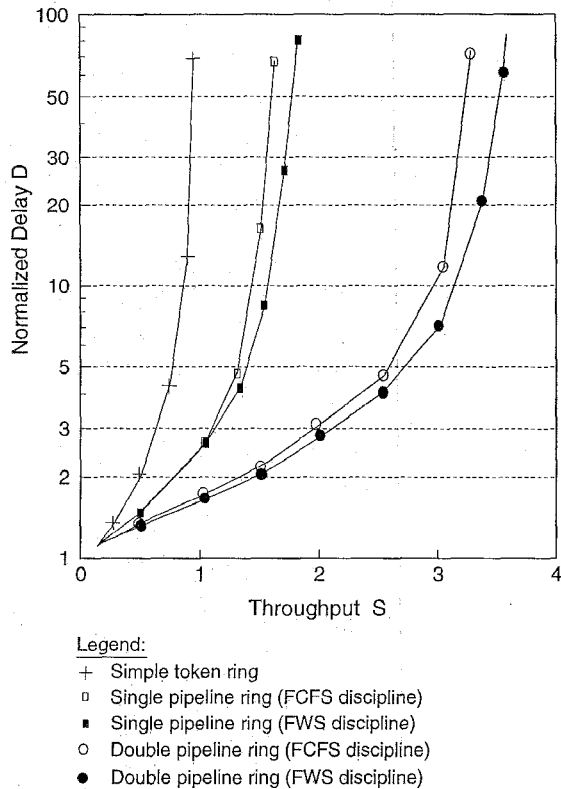


Fig. 11. Delay comparisons,  $N = 15$ ,  $a = 0.01$ , fixed size messages.

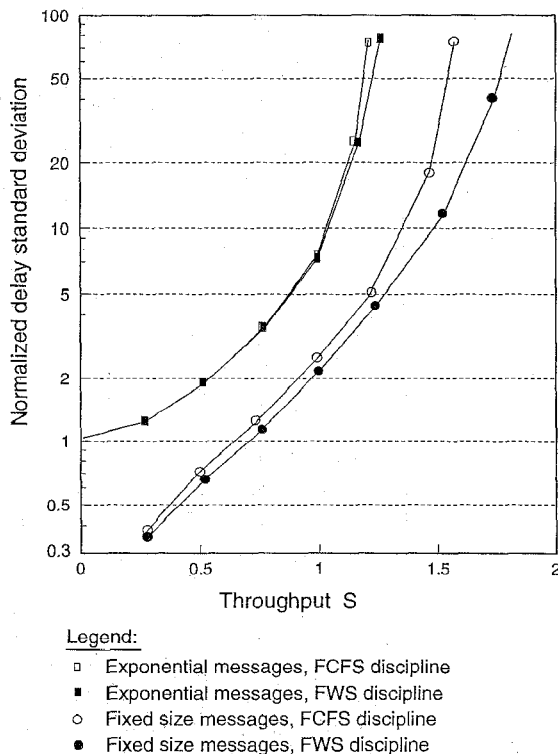


Fig. 12. Comparison of delay standard deviations, single ring,  $N = 15$ ,  $a = 0.01$ .

pipeline rings is 3.26 and 3.51 for *FCFS* and *FWS* respectively. Figure 12 compares the delay standard deviations of single pipeline rings using the two service disciplines. We found that for both exponentially distributed and fixed size messages, there is no increase of delay variance using the *FWS* discipline. In

fact, the delay variance of *FWS* becomes increasingly smaller when compared to *FCFS* as the traffic load increases. Therefore, the overall performance of the *FWS* discipline is always superior than the *FCFS* discipline in the pipeline ring. From simulation, we found that with the use of the *FWS* discipline, messages to a closer destination are still more readily transmitted than messages to a further destination. There is no forever blocking of certain messages because the status of *start-up* station is rotated on a round-robin basis.

4) Bimodal Traffic:

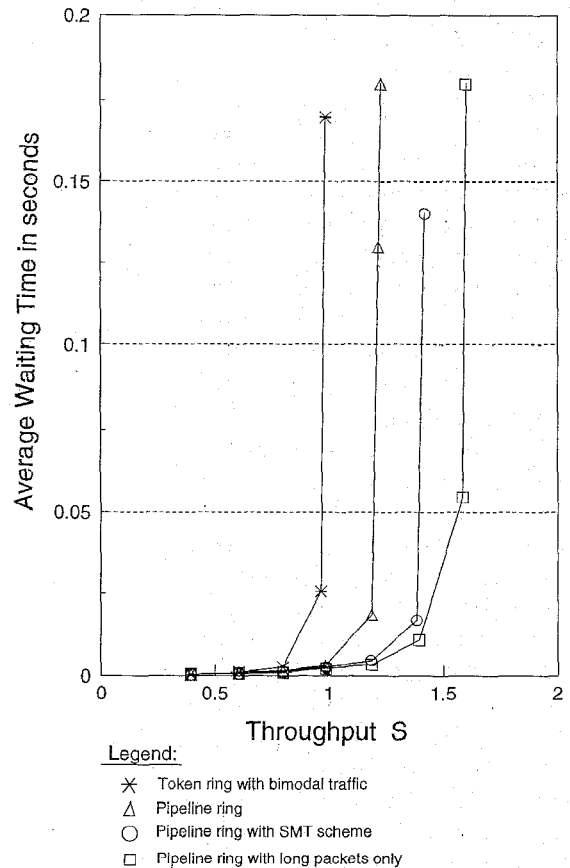


Fig. 13. Ring performance under bimodal traffic (50% long packets, 50% short packets).

Figure 13 shows the delay-throughput performance of the pipeline ring with half of the packets 16 kbits long and half of the packets 160 bits long. Other ring parameters are the same as before. The size of the short packets is comparable to the ring round-trip delay. Instead of the normalized delay which depends on the message transmission time, we showed the average waiting time for comparison. We found that the average waiting time of long and short packets are the same under all traffic conditions. The maximum throughput of the pipeline ring under bimodal traffic is 1.25, which is significantly less than the maximum throughput of a pipeline ring with long packets only, which is 1.63. Using the *SMT* scheme described in section II, the maximum throughput can be increased to 1.46. The delay-throughput performance of the simple token ring is also shown for comparison.



## VI. CONCLUSION

In this paper the pipeline ring protocol was proposed and analyzed. This protocol is distributed and can provide a fair access to all stations. Both single and double rings are supported. We found that the ring efficiency is almost independent of the number of stations under symmetrical traffic.

The Furthest Within Segment (*FWS*) service discipline was proposed for use in the pipeline ring. Its performance was shown to be always better than the traditionally used First Come First Serve (*FCFS*) discipline. A Short Message Transmission (*SMT*) scheme was proposed which can significantly increase the ring efficiency under bimodal traffic conditions. A performance comparison with the IEEE 802.5 token ring, buffer insertion ring, Leventis loop, Jafari loop, CTRing and the playthrough ring was also made.

## REFERENCES

- [1] Local Area Network Standards: Token-passing ring access methods and physical layer specifications, ANSI/IEEE 802.5, 1985.
- [2] American National Standard, "FDDI Token Ring Media Access Control (MAC)," ANSI X3.139-1987.
- [3] A. Hopper and R.M.Needham, "The Cambridge Fast Ring Networking System," *IEEE Trans. on Comp.*, vol.37, no.10, Oct.1988, pp.1214-1223.
- [4] R.M.Falconer and J.L.Adams, "Orwell: A Protocol for an Integrated Services Local Area Network," *British Telecom Technology*, October 1985, pp.27-35.
- [5] M.T.Liu and C.C.Reames, "Message communication protocol and operating system design for the distributed loop computer network (DLCN)," *Proceedings 4th Annual Symp. Computer Architecture*, March 1977, pp.193-200.
- [6] J.J.Wolf and M.T.Liu, "A Distributed Double-Loop Computer Network (DDLNCN)," *Proceedings of 7th Texas Conf.Comp.Syst.*, Nov.1978, pp.6.19-6.34.
- [7] H.Jafari, T.G.Lewis and J.D.Spragins, "Simulation of a class of ring structured network," *IEEE Trans. on Computers*, Vol.C-29, May 1980, pp.385-392.
- [8] S.Leventis, G.Papadopoulos and S.Koubias, "A new experimental computer network and its simulated performance," *Proceedings of INFOCOM'82*, Las Vegas, NV, March 1982, pp.113-121.
- [9] C.B.Silio, Jr., *Performance of Multi-Message, Single-Token Rings*, Technical Report, ENEE85-05-15, Electrical Engineering Department, University of Maryland, 1985.
- [10] C.B.Silio, Jr., "Performance Approximations for multmessage circuit-switched rings," *Proceedings of INFOCOM'86*, Washington D.C., U.S.A., March 1986, pp.85-93.
- [11] G.Pacifici, A.Pattavina, "T-S Protocol: An Access Protocol for Ring Local Area Networks," *Proceedings of IEEE GLOBECOM'86*, U.S.A., 1986, pp.25-28.
- [12] M.Xu, J.H.Herzog, "Concurrent Token Ring Protocol," *Proceedings of IEEE INFOCOM'88*, U.S.A., 1988, pp.145-154.
- [13] P.C.Wong, T.S.Yum, "Design and Analysis of a Pipeline Ring Protocol," *Proceedings of ICC'88*, June 1988, pp.1268-1274.
- [14] P.C.Wong, T.S.Yum, "An Integrated Services Token-Controlled Ring Network," *IEEE Journal on Selected Areas in Communications*, vo.7, no.5, June 1989, pp.670-679.
- [15] A.Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1985.
- [16] J.L.Hammand, P.J.P.O'Reily, *Performance Analysis of Local Computer Networks*, Addison Wesley, 1986.

**P.C. Wong** (S'82-M'89-SM'93) received his B.Sc, M.Phil. and Ph.D. degrees all from the Chinese University of Hong Kong (CUHK). He returned to CUHK in the Department of Information Engineering in 1990. He was a visiting scientist in IBM T.J.Watson Center, NY in the summer of 1992 and a visiting research professor in the Department of Communication Engineering of the National Chiao Tung University in 1993. His research interests include fast packet switching, optical networking, network services and applications, and network software engineering.

**Tak-Shing Peter Yum** (S'76-M'78-SM'86) received his BSc, MSc, MPh and PhD from Columbia University in 1974, 75, 77 and 78 respectively. He worked in Bell Telephone Laboratories, USA for two and a half years and taught in National Chiao Tung University, Taiwan, for 2 years before joining The Chinese University of Hong Kong in 1982. He has published original research on packet switched networks with contributions in routing algorithms, buffer management, deadlock detection algorithms, message resequencing analysis and multiaccess protocols. In recent years, he branched out to work on the design and analysis of cellular network, lightwave networks and video distribution networks. He believes that the next challenge is designing an intelligent network that can accommodate the needs of individual customers. Peter claims himself to be an existentialist. He enjoys playing bridge and badminton.