(4)

WSRL-TR-45/89

AR-005-937

SIMULATING LOCAL AREA NETWORK PROTOCOLS WITH THE GENERAL PURPOSE SIMULATION SYSTEM (GPSS)

M.L. SCHOLZ

COMBAT SYSTEMS DIVISION
WEAPONS SYSTEMS RESEARCH LABORATORY

Approved for Public Release.

MARCH 1990

DSTO AUSTRALIA
DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

CONDITIONS OF RELEASE AND DISPOSAL

This document is the property of the Australian Government. The information it contains is released for defence purposes only and must not be disseminated beyond the stated distribution without prior approval.

Delimitation is only with the specific approval of the Releasing Authority as given in the Secondary Distribution statement.

This information may be subject to privately owned rights.

The officer in possession of this document is responsible for its safe custody. When no longer required the document should NOT BE DESTROYED but returned to the Main Library, DSTO, Salisbury, South Australia.

DSTO AUSTRALIA

TECHNICAL REPORT
WSRL-TR-45/89

SIMULATING LOCAL AREA NETWORK PROTOCOLS WITH THE GENERAL PURPOSE SIMULATION SYSTEM (GPSS)

M.L. Scholz

SUMMARY (U)

Simulation models of ANSI 802.3 (Ethernet) and ANSI 802.5 (Token Ring) transmission control protocols have been constructed to assess the usefulness of the General Purpose Simulation System (GPSS) simulation language for modelling the performance of local area networks. The models are described in detail and the data obtained from them are compared with analytic results. Model verification and other problems associated with analysing simulation output are discussed.

Keywords: Australia, token ring protocols, (KR)

Author's address:
Combat Systems Division
Weapons Systems Research Laboratory
PO Box 1700, Salisbury
South Australia

Requests to: Chief, Combat Systems Division

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. ETHERNET SIMULATION MODEL	1
2.1 Model structure	2
2.1.1 Frame generation	3
2.1.2 Frame transmission	4
2.1.3 Jam and collision recovery	5
2.1.4 Collision sensing	6
2.1.5 Frame reception	7
2.2 Model artifices	8
2.3 Model variables	9
2.4 Simulation results	11
3. TOKEN RING SIMULATION MODEL	14
3.1 Model structure	15
3.1.1 Frame generation	15
3.1.2 Frame delivery	15
3.2 Model artifices	16
3.3 Model variables	17
3.4 Simulation results	18
4. EXTERNAL PROCEDURES USED IN SIMULATION	19
5. MODEL VERIFICATION	20
6. SIMULATION OUTPUT ANALYSIS	21
6.1 Bias and variance reduction	21
6.2 Model execution times	22
7. DISCUSSION AND CONCLUSIONS	24
8. ACKNOWLEDGEMENTS	25
ABBREVIATIONS	26
REFERENCES	27

Page

LIST OF FIGURES

1. Topology of a single segment Ethernet LAN 31

2. Relationship of Ethernet LAN to OSI Reference Model 32

3. Architecture of the Ethernet model 33

4. Ethernet: Frame generation process 34

5. Ethernet: Frame transmission process 35

6. Ethernet: Jam and collision recovery process 36

7. Ethernet: Collision sensing process 39

8. Ethernet: Frame reception process 40

9. Ethernet: Mean transfer delay vs mean throughput 41

10. Ethernet: Probability of zero channel assignment delay vs mean throughput 42

11. Ethernet: Probability of buffer overflow vs mean throughput 43

12. Ethernet: Transmitter and receiver utilisations vs mean throughput 44

13. Topology of a Token Ring LAN 45

14. Relationship of Token Ring LAN to OSI Reference Model 46

15. Token Ring: Frame generation process 47

16. Token Ring: Frame delivery process 48

17. Token Ring: Mean transfer delay vs mean throughput 49

LIST OF APPENDICES

I THE ETHERNET PROTOCOL 51

Figure I.1 State diagram of Ethernet MAC transmit protocol 53

Figure I.2 State diagram of Ethernet MAC receive protocol 54

II ANALYTIC RESULTS FOR ETHERNET PERFORMANCE 55

III THE TOKEN RING PROTOCOL 57

Figure III.1 State diagram of Token Ring MAC protocol 58

IV ANALYTIC RESULTS FOR TOKEN RING PERFORMANCE 59

1. INTRODUCTION

The suitability of the General Purpose Simulation (GPSS) language as a tool for modelling transmission control protocols in complex local area networks (LAN) is examined. GPSS is one of several tools under consideration for evaluating the performance of high-speed LANs that are likely to be offered in the next generation of distributed naval combat systems. This investigation was prompted because GPSS appears to be an ideal language for LAN protocol modelling and because the usage of GPSS in this field has seldom been reported in the literature.

In order to assess the potential of GPSS, two LAN protocol simulation models were developed with the intention of exposing the strengths and weaknesses of the language in this particular application. The protocols chosen were ANSI 802.3(ref.14), which is very similar to the commercial "Ethernet", and ANSI 802.5(ref.13), or "Token Ring" protocol. They were chosen for three reasons. Firstly, they are international (ISO and ANSI) standards and are arguably the most popular asynchronous time-division multiplexed schemes in use today for transmitting high speed serial data in geographically distributed computer systems. Secondly, they are highly relevant to military applications. Ethernet has been adopted by many navies around the world for use in ship-board combat systems (eg Canadian Patrol Frigate, Dutch M-Frigate, West German Meko 200) and will be fitted to the Australian Anzac Ship. The Token Ring is important because it is the design upon which the current development of FDDI (Fiber Distributed Data Interface)(ref.29), a very high speed LAN, is based. FDDI will assume major importance in future military systems. Thirdly, the Ethernet and Token Ring protocols have been extensively studied and the published performance data are adequate for the purpose of validating simulation models.

The GPSS simulation language has been widely employed in modelling computer system performance(refs.10,36,37) and in manufacturing. The process interaction approach of GPSS facilitates the efficient modelling of complex systems that can be represented by queuing networks. This contrasts with the event-scheduling approach of some other simulation languages, such as the popular SIMSCRIPT II.5(ref.21), which involve much greater programming effort. Several dialects have been developed from the original IBM GPSS/V product(ref.15). GPSS/H(ref.12) was chosen on the strength of its extended programming features, computational efficiency, ease of understanding, and popularity. The reader is assumed to have a basic knowledge of the language.

The remainder of this report is organised as follows. An exact GPSS/H model of the Ethernet protocol and an approximate model of the Token Ring protocol are described in Sections 2 and 3. In Section 4, the external routines used in the simulations are described. The facilities provided by GPSS/H for verifying the operation of a model are described in Section 5. The major problems associated with simulation output analysis are addressed in Section 6. The principal conclusions regarding the suitability of GPSS/H for LAN protocol modelling are then summarised in Section 7.

This research was undertaken with the sponsorship of the Director, Naval Combat Systems Engineering, Royal Australian Navy (Task No. NAV 87/226).

2. ETHERNET SIMULATION MODEL

Ethernet (ANSI 802.3) is a CSMA/CD (carrier sense multiple access, collision detection) protocol designed to operate in a LAN bus topology(ref.14). The model described below assumes a simple single segment topology as depicted in figure 1. More complex multiple segment topologies may also be modelled by including repeater delays in the inter-station propagation delays. The results however will be approximate at high loading levels because collisions at the repeaters are ignored; an exact model requires structural extensions to the existing model. Ethernet comprises a number of protocols at the Data Link and Physical Layers of the Open Systems Interconnection (OSI) Reference Model(ref.16) as shown in figure 2.

The simulation model was designed to measure the exact performance of the protocols within the Media Access Control (MAC) sublayer of the Data Link Layer and therefore discussion will be confined to the MAC protocol. The principles of the operation of the protocol upon which the simulation model is based, are expounded in Appendix I.

In Section 2.1 the structure of the simulation model is described with the aid of GPSS block diagrams. Although the model is very detailed, it does not take account of network initialisation, error recovery and other procedures unrelated to frame transmission, and assumes that frames are not corrupted during transmission on the medium. Special techniques, described in Section 2.2, are employed in modelling backoff and delay timeouts, and in handling simultaneous frame reception. These techniques involve the use of programming artifices which have no analogues in the protocol. The variables of the model are described in Section 2.3.

The model is extremely flexible. Any size network (with more than one station) may be specified, together with the physical positions of each station on the medium, different frame generation rate and data length distributions at each station, and routing probabilities for frames transmitted at each station. In addition, the protocol constants, propagation delay constants, frame characteristics, backoff algorithm parameters, and network parameters can be adjusted to reflect specific implementations.

Performance measures may be obtained from the default statistical output and by inserting appropriate diagnostic blocks into the model. In order to simplify the structure and description of the model however, none of the additional blocks used to obtain the results mentioned in this report is shown.

The lack of a suitable hardware test bed facility prevented access to the experimental data necessary to fully validate the model. Instead, published analytical models were used as sources of data. In contrast to the simulation model, analytical models only take into account the major parameters that determine performance and cannot guarantee accurate results over the full operating region of the protocol(ref.9). Standard queuing models cannot be employed, for example, in examining performance in the region where the frame generation rates exceed the throughput capacity of the transmission medium. The simulation model produces credible results against these analytic models when the parameter assumptions are matched.

The results obtained during model validation are presented in Section 2.4.

2.1 Model structure

For descriptive purposes the model is divided into modules according to the major processes performed, as illustrated in Figure 3. The frame generation process mimics the generation of frames by the LLC sublayer. The other processes model the behaviour of the transmit and receive components of the MAC State Machine. The transmit component includes the frame transmission process, which models the transmission and re-transmission of frames, and the jam and collision recovery process which models the events that occur subsequent to collision detection but prior to frame retransmission. The receive component includes the frame reception process, which models the reception of frames and collision fragments and the filtering of complete and correctly addressed frames, and the collision sensing process which models the detection of collisions on the medium and the setting of internal flags that control the behaviour of the transmitter.

Because the above processes must be replicated for each station in the network, structural "folding" is applied to reduce the number of blocks in the model. A penalty must be paid however in the additional complexity of the logic. To facilitate folding, GPSS transaction parameters are used to associate transactions controlling transitions in the MAC State Machine with a particular station, and to store the MAC state variables of that station. In addition, transaction parameters are used to store frame-specific information, such as source and destination addresses, event times (eg frame generation time), and message length in transactions which simulate frames. The reference numbers of GPSS blocks (eg SEIZE, LINK, LOGIC, GATE blocks etc) which model station queues, flags, decision gates, and other station entities, are either indexed by GPSS variables (eg V$FAC) or by transaction parameters.

2.1.1 Frame generation

The frame generation process (figure 4) simulates the generation of fixed length frames with Poisson distributions by the LLC sublayer. The Poisson distribution was chosen because it frequently fits observed arrival rate statistics in operational systems and because it is almost universally employed in analytic LAN models. Frames arriving at the MAC sublayer in conformance with this distribution are uniformly distributed throughout a finite time interval (viz simulation run) and are independent of previous arrivals.

The flexibility of the model permits other frame length or generation rate distributions to be employed by calling external routines to produce random numbers at the appropriate ASSIGN blocks. Frames are represented by transactions in the model.

One transaction per station is generated by the first SPLIT block from a single initial transaction arriving at zero time from the GENERATE block. Parameter PH1 associated with each of these transactions is assigned a unique value $<PH1> \in \{1,2,....,\&N\}$ corresponding to a station address, where the ampervariable &N is the total number of stations in the network. (Note the symbol <x> represents the numerical value assigned to parameter "x"). Each transaction then enters an ADVANCE block and experiences an exponentially-distributed random delay (computed by the external routine &REXPON) with a mean value equal to the reciprocal of the station's mean frame generation rate, &MAR(PH1), before entering the second SPLIT block. Daughter transactions from the second SPLIT block are transferred to the previous ADVANCE block to enable the generation of subsequent frames.

Three items of frame-specific information are attached to the parent transactions after they leave the second SPLIT block and before they enter the frame transmission process. The random frame destination address (computed by the external routine &IDEST), which is uniformly distributed over the range of (integer) station addresses is assigned to PH2. The length of the data field contained in the frame, which is specified as a constant, is assigned to PH4. The random frame transmission delay assigned to PL1 is given by the sum of the data length and the data length dependent frame overhead, V$OVERHEAD (multiplied by eight to convert octets to bits), divided by the data transmission rate, &DATARATE.

2.1.2 Frame transmission

The frame transmission process (figure 5), corresponding to MAC transmit states 1, 2 and 7(ref.14), is responsible for transmitting frames, monitoring the receiver status for collisions over the duration of the transmission interval, and initiating action to reschedule a transmission in the event of a collision.

At each station, frame transactions from the frame generation process enter a buffer (user-chain <PH1>) which has unit capacity. Each station is allowed to store and attempt to transmit at most one frame at a time so that transactions with <PH1>=k, corresponding to frames generated at station k, are discarded on arrival if user-chain k is occupied. The arrival time of each frame is assigned to parameter PL2 to facilitate the subsequent computation of the frame transfer delay. The frame transfer delay is an important measure of performance in local area networks. It is the time between the generation of a frame and its arrival at its destination and includes delays in queuing, access, transmission, and propagation.

A frame is removed from the buffer when a station senses that the medium is available and following a time interval corresponding to the inter-frame gap. In the model, each station's "clear to send" status is denoted by the reset state of a logic switch V$CTS. Switch V$CTS=k+2*&N is set by the receive process when a carrier is detected at station k, and is reset by the transmission medium sensing process. The maximum backoff delay, stored in ML$MAXBKDEL(PH1,1), is initialised to the slot time, &SLOTDEL, the transmitter facility <PH1> is seized, and transmission commences.

The transmission of a frame to each station in the network (except to the transmitting station, <PH1>) is simulated by &N separate transactions which are obtained by replicating the frame transaction in a SPLIT block. Every daughter transaction from the SPLIT block has a value of zero assigned (at birth) to parameter PB1 to denote the head of the frame, and is assigned a unique station address to parameter PH3. The daughter transactions are delayed in an ADVANCE block for a period equal to the DTE (Data Terminal Equipment) output-MAU (Medium Attachment Unit) assert delay, &SGOUTDEL, corresponding to the delay between a frame being sent by the LLC sublayer and the time its first bit is impressed on the medium. These "head-of-frame" transactions subsequently enter the reception process at block address RX0, except for those that have the same source and destination addresses (viz <PH3>=<PH1>) which are deleted.

The frame transaction (parent transaction) from the SPLIT block enters a second SPLIT block thereby spawning a daughter transaction which is used to monitor the medium for collisions during the transmission of the frame. The parent transaction from the second SPLIT block is held at an ADVANCE block for the transmission delay, <PL1>.

Provided no collision occurs during transmission, the frame transaction leaves the ADVANCE block, releases the transmitter facility, enters a third SPLIT block, and is delayed for a further time equal to the inter-frame delay (or gap) before causing the next transaction to be dequeued from user-chain <PH1>. The third SPLIT block generates a single daughter transaction and a value of one is assigned to PB1 to denote the end of a frame. This transaction is then replicated by a fourth SPLIT block into &N transactions for transmission to each station in the network. As before, transactions with <PH3>=<PH1> are deleted since a station is not permitted to receive its own frames. These "end-of-frame" transactions then enter the reception process at block address RX0.

A collision occuring during or after transmission of the preamble affects the transmission of the frame in the following manner. The daughter transaction from the second SPLIT block is "marked" (viz the current simulator absolute clock time, AC1, is stored in the transaction's transit delay attribute, M1, by a MARK block) at the commencement of transmission. It then waits at a conditional TEST block until either a collision occurs (logic switch <PH1> set) or frame transmission is finished (transmitter facility <PH1> not in use). When transmission is complete, the transaction is deleted. If a collision occurs, the transaction is delayed until the preamble is finished. The maximum delay is equal to the preamble transmission delay, &PREAMDEL. The transmit facility <PH1> is then pre-empted resulting in the transaction located in the ADVANCE block at address TX3+4 being sent to the jam and collision recovery process (block address TX12).

2.1.3 Jam and collision recovery

The jam and collision recovery process (figure 6), corresponding to MAC transmit states 3 to 6, and 8 to 10(ref.14), is responsible for terminating a frame transmission in the event of a collision, transmitting a jam, and rescheduling or aborting a transmission.

A transaction enters this process from the frame transmission process following a collision. It seizes the transmitter facility <PH1> and is delayed in an ADVANCE block for a period equal to &JAMDEL whilst the jam is being transmitted, before it releases the transmitter facility. It then enters a SPLIT block, increments the attempt count in PH5, and tests whether the attempt count exceeds the attempt limit, &MAXATTMP. If the limit has been exceeded, the frame is aborted by returning the transaction to the transmit frame process (at block address TX4) where it dequeues the next frame from the buffer (viz transaction on user-chain <PH1>) before being deleted. Otherwise the attempt count is compared with the backoff threshold, &THRESCNT. If the attempt count is less than the backoff threshold, the mean backoff delay stored in ML$MAXBKDEL(PH1,1) is doubled at each attempt in accordance with the binary exponential backoff algorithm.

The initial and subsequent backoff delays are randomly sampled (by the external procedure &RUNIF) from a continuous range of values in the closed interval [0,ML$MAXBAKDEL(PH1,1)] and assigned to parameter PL3. The inter-frame delay is initialised to its maximum value, &IFDEL, and assigned to parameter PL4. The operation of the backoff timer and the delay timer, which rely upon these parameters, is described in Section 2.2. The transaction then enters the *backoff-delay* state. The time of entry to this state is marked before the transaction enters a SPLIT block at address TX14+2. If the inter-frame delay remaining exceeds the backoff delay remaining, the transaction is dispatched to the ADVANCE block at address TX17+1 where it remains until the delay timer expires. Otherwise, the transaction waits in the ADVANCE block at address TX14+5 until the backoff timer expires, after which the remaining inter-frame delay is adjusted and the *delay-wait* state is entered. The station remains in the *delay-wait* state until the transaction leaves the ADVANCE block at address TX15 upon expiry of the delay timer. The transaction re-enters the transmit process (at block address TX3) to simulate the re-transmission of the frame. Upon leaving the ADVANCE block at address TX17+1, the backoff delay remaining is adjusted and the transaction is marked before it enters the *backoff* state.

The daughter transaction leaving the SPLIT block at address TX14+2 waits at a conditional TEST block at address TX21 until either a carrier is detected (logic switch <PH1> set) or the backoff timer expires (transaction leaves the *ADVANCE block at address TX14+5). The first* condition is tested in a GATE block. If the backoff delay timer has expired, the transaction is deleted. If a carrier is present, the parent transaction presently delayed in the ADVANCE block at address TX14+5 is pre-empted and deleted. Thus, only one transaction, namely the parent or the daughter of the SPLIT block at address TX14+2, survives to enter the next state.

Two artifices are employed in conjunction with transaction parameters PL3 and PL4, to model the operation of the backoff and delay timers at each station: an ADVANCE block at address TX17+4, and a facility referred to as the "dummy" facility (V$FAC). Their usage is described in Section 2.2.

In the *backoff* state, a transaction enters the SPLIT block at address TX17+6 and is delayed in an ADVANCE block pending the expiry of the backoff timer. The daughter transaction incurs identical processing to its counterpart in the *backoff-delay* state, as previously described, except that after it has adjusted the backoff delay it enters the *backoff-defer* state at block address TX19. After leaving the ADVANCE block at address TX17+8, the parent transaction releases the dummy facility and then enters the transmit process at block address TX3 to simulate the re-transmission of a frame.

Upon entering the *backoff-defer* state MARK block at address TX19, a transaction enters a SPLIT block *and is delayed in an ADVANCE block* pending the expiry of the backoff timer. The daughter transaction incurs identical processing to its counterpart in the *backoff* state, except that the conditional TEST block at address TX20 causes the transaction to wait until either the end of carrier is sensed or the backoff timer expires. After adjusting the backoff delay remaining, the transaction enters the *backoff-delay* state at block address TX14. After leaving the ADVANCE block at address TX19+3, the parent transaction releases the dummy facility and waits at a conditional GATE block, pending entry to the *defer-wait* state. When the end of the carrier is sensed (viz carrier sense logic switch <PH1> is reset), this transaction reinitialises the delay timer and enters the *delay-wait* state at block address TX15.

2.1.4 Collision sensing

The collision sensing process (figure 7) models the detection of collisions and the setting of a status flag to interrupt a transmission in progress.

A single control transaction is generated at zero time and replicated by a SPLIT block into &N transactions. Each transaction has a parameter, PH1, which is assigned a unique station address and waits at the first GATE block pending the detection of a carrier by the reception process (logic switch 1,2,...,&N set). If a carrier is detected at station k, only the control transaction with <PH1>=k proceeds through the GATE block. It subsequently sets the clear-to-send logic switch (V$CTS) to inhibit frame transmission at station k and waits at the second GATE block pending the end of the carrier. When the carrier sense logic switch k is reset by the reception process at station k the transaction proceeds to an ADVANCE block where it is delayed by the inter-frame gap (&IFDEL) before resetting the clear-to-send logic switch. Permission is thereby given for station k to transmit frames. The control transaction finally returns to the first GATE block and the collision sensing cycle re-commences at station k.

2.1.5 Frame reception

The frame reception process (figure 8), corresponding to MAC receive states 1 and 2(ref.14), receives frames from the medium, discriminates between collision fragments and complete frames, discards incorrectly addressed frames, and passes the correctly addressed frames to the LLC sublayer. In addition, it monitors the medium for collisions and initiates action to cease the transmission of a frame in progress.

Head-of-frame and end-of-frame transactions enter the reception process directly from the transmit frame process. They are delayed in an ADVANCE block for a time equal to the propagation delay, ML$PROPDEL(PH1,PH3), between the transmitting station <PH1> and the receiving station <PH3>. A TEST block sends an end-of-frame transaction to block address RX4. A start-of-frame transaction that arrives at station k=<PH3> whose receiver is currently processing a frame (receiver facility V$RX=k+2*&N busy) is sent by the first GATE block to an ASSIGN block at address RX5 where it increments the concurrency count, MX$SIMR(k,1), and is deleted. The concurrency count is employed in an artifice to prevent the resetting of the carrier detect logic switch k at station k (k=1,2,...,&N) and is described in Section 2.2.

If a frame is received whilst station k is transmitting, a collision is detected. A start-of-frame transaction is diverted by the second GATE block to block address RX2 if the transmitter facility k is in use. Otherwise, the transaction sets the carrier sense logic switch k after a delay &SGDETDEL, corresponding to the delay in detecting data input at the LLC sublayer. The transaction then marks the time at the start of reception, seizes the receiver facility (V$RX), and waits at the third GATE block until the carrier sense logic switch k is reset.

The carrier sense logic switch k is reset as follows when the end of a carrier is sensed by station k. An end-of-frame transaction arriving at block address RX4 is delayed in an ADVANCE block for a period equal to &SGENDDEL, the delay in the LLC sublayer in detecting the trailing edge of the signal. If there are no simultaneous receptions at station k (viz the concurrency count MX$SIMR(k,1)=0), the transaction resets logic switch k, otherwise it decrements the concurrency count. Finally, the transaction is deleted.

When a carrier sense logic switch is reset, a start-of-frame transaction leaves the third GATE block, releases the receiver facility and enters a TEST block. The mark time of the transaction is deducted from its time of arrival at this TEST block to determine the length of time that a carrier is detected (viz the duration of frame reception). If this time is less than the time of transmission of the minimum length frame, &MINFMDEL, the received frame is a collision fragment and is terminated. Otherwise, the transaction enters the next TEST block where it is diverted to the TERMINATE block at address RX1 if the receiving station is not the correct destination station (ie <PH2>≠<PH3>). The transfer delay time of a correctly received frame can be computed by subtracting the transmit buffer entry time, saved in transaction parameter PL2, from the simulator clock time, AC1. Additional blocks were inserted between the TEST block and the TERMINATE block at address RX1A to compute transfer delay statistics. However, the blocks will not be described because they are not strictly relevant to the operation of the Ethernet protocol.

When a collision is detected at station k, the station ceases transmission immediately after it has issued a jam. The transaction which enters block address RX2 is delayed in an ADVANCE block for a period &COLSNDEL equal to the delay in the LLC sublayer in detecting a collision. It then sets carrier sense logic switch k, seizes the receiver facility (V$RX) and then waits at a GATE block until the carrier sense logic switch is reset. When the carrier sense logic switch is reset, the receiver facility is released and the transaction is terminated.

2.2 Model artifices

Special programming techniques are employed in modelling timeout and state transition mechanisms, and simultaneous frame reception within each station. The GPSS blocks and variables associated with these mechanisms have no equivalents in the protocol itself.

(1) Backoff and delay timers

The operation of the backoff and delay timers is simulated by reducing the values of the parameters PL3 and PL4 associated with the transactions that enter the jam and collision recovery process. The values of these parameters are adjusted at the times of particular state transitions and represent the amounts of time remaining before the timers expire (ie when the parameter values are zero).

(2) State transitions

In the current version of GPSS/H there is no mechanism for directly removing a transaction from an ADVANCE block (viz future-events chain); therefore a "dummy" facility (V$FAC) is used. One dummy facility is allocated per station. The dummy facility at station k (viz V$FAC=k+&N) is seized by those transactions entering the *backoff-delay*, *backoff* and *backoff-defer* states with <PH1>=k. It is released when transactions leave these states prior to entering the *delay-wait*, *transmit* and *delay-wait* states respectively. It is pre-empted and released to enable transactions to subsequently enter the *backoff-defer*, *backoff-defer* and *backoff-delay* states respectively.

An ADVANCE block with an insignificant delay (1 ns) is inserted at address TX17+4 to prevent two transactions from leaving the *backoff-delay* state in the event that the delay timer expires before a carrier is detected (viz logic switch <PH1> set).

This artifice is necessary because of the way that GPSS moves transactions through a model. During a GPSS scan, an "active" transaction is always moved through as many blocks as possible until one is encountered that impedes forward progress. Therefore, unless this ADVANCE block is provided, the transaction leaving the ADVANCE block at address TX17+1 would enter the *backoff* state and immediately re-capture the dummy facility (V$FAC). The transaction poised at the TEST block at address TX21 would then be moved when either the transaction in the *backoff-delay* state releases the dummy facility upon the expiry of the backoff timer or the carrier is detected. In the latter case, the transaction would not be deleted as required.

The same scheduling situation arises when transactions exit the *backoff-defer* and *backoff-delay* states. However, in each case an additional ADVANCE block is not inserted because an ADVANCE block is already present (for the purpose of delaying a transaction until the backoff timer expires).

(3) Simultaneous frame reception

An artifice is used to prevent the resetting of the carrier detect logic switch k at station k (k=1,2,...,&N) in the reception process when frames are simultaneously received from different stations. For example, if station k is currently receiving a frame from station i when it receives a frame from station j (where i≠j≠k; i,j,k=1,2,...,&N), station k continues to detect the presence of a carrier until both stations i and j detect the collision and end their transmissions. This scenario can be extended to cover more than two simultaneous receptions. In the model, this corresponds to the consecutive receipt of two (or more) start-of-frame transaction before an end-of-frame transaction. The number of simultaneous receptions, or concurrency count, at station k is stored in MX$SIMR(k,1). The concurrency count is incremented whenever a start-of-frame transaction is received by station k provided the carrier sense logic switch k is set, and it is decremented whenever an end-of-frame transaction is received provided the concurrency count is non-zero. Note that there is no concurrency count variable in the actual protocol.

2.3 Model variables

There are seven categories of variables in the model: implementation-dependent, network-dependent, station-dependent, derived, system state, simulation control, and diagnostic variables. The variables are listed and described in the following tables along with their values.

Implementation-dependent variables describe the transmission properties of the medium and various fixed delays in the operation of the protocol.

GPSS VARIABLE	IMPLEMENTATION-DEPENDENT VARIABLE
&DATARATE	Medium transmission rate (bit/s)
&IFDEL	Inter-frame delay (s)
&AUILEN	Length of AUI cables (m)
&M1	MAU data-in assert → input delay (bit)
&M2	MAU output → data-out assert delay (bit)
&M3	MAU data in collision → SQE assert delay (bit)
&D1	Input → input unit delay (bit)
&D2	Output unit → output delay (bit)
&D4	Input idle → carrier status=off delay (bit)
&D7	SQE assert → signal status=error delay (bit)
&D12	Jam output duration (bit)
&FACTCOAX	Coax propagation speed factor
&FACTAUI	AUI cable propagation speed factor
&PKTLEN1	Minimum frame length (oct)
&PKTLEN2	Maximum frame length (oct)
&PKTPRE	Preamble length (oct)
&FADDRLEN	Length of source and dest address fields (oct)
&SLOTIME	Retransmission slot time (bit)
&MAXATTMP	Maximum number of retransmission attempts
&THRESCNT	Backoff attempt threshold

Network-dependent variables describe the topology of the network.

GPSS VARIABLE	NETWORK-DEPENDENT VARIABLE
&N	Number of stations in the network
ML$PROPDEL(j,k)	Propagation delay (s) from station j to k
&POSN(k)	Distance of station k=1,2,...,&N from end of coax

Station-dependent variables describe the properties of the traffic generated at each station.

GPSS VARIABLE	STATION-DEPENDENT VARIABLE
&MAR(k)	Mean frame generation rate at station
ML$PROB(j,k)	Probability station j sends frame to station k

Derived variables are functions of protocol, network and station-dependent variables.

GPSS VARIABLE	DERIVED VARIABLE
&A1= &AUILEN/(3.0E8*&FACTAUI)	AUI propagation delay (s)
&SLOTDEL= &SLOTIME/&DATARATE	Retransmission slot delay (s)
&MINFMDEL= 8*&PKTLEN1/&DATARATE	Minimum frame transmission delay (s)
&JAMDEL= &D12/&DATARATE	Jam transmission delay (s)
&PREAMDEL= 8*&PKTPRE/&DATARATE	Preamble transmission delay (s)
&MAXDATA= &PKTLEN2-2*&FADDRLEN-6	Maximum data field length (oct)
&DATT= &PKTLEN1-2*&FADDRLEN-6	Minimum data field length without pad (oct)
&MAXOVRHD= 8+&PKTLEN1	Maximum overhead (oct)
&MINOVRHD= 14+2*&FADDRLEN	Minimum overhead (oct)
&SGOUTDEL= &A1+(&D2+&DM2/&DATARATE	DTE output → MAU data assert delay (s)
&SGDETDEL= &A1+(&M1+&D1)/&DATARATE	Data input → LLC detect delay (s)
&COLSNDEL= &A1+(&M3+&D7)/&DATARATE	Collision → LLC detect delay (s)
&SGENDDEL= &A1+&D4/&DATARATE	End signal→ LLC detect delay (s)
BV$PAD= 0 if <PH4>≤&DATT, else=1	Zero if pad not required
V$FAC= <PH1>+&N	Index of dummy facility
V$RX= <PH3>+2*&N	Index of receiver facility
V$CTS= <PH1>+2*&N	Index of CTS logic switch
V$OVERHEAD= (&MAXOVRHD-PH4)* BV$PAD+&MINOVRHD*(1-BV$PAD)	Frame overhead (oct)
ML$CUMP(j,k)	Cumulative probability of transmission from station j to k

System state variables describe the dynamic state of the model.

GPSS VARIABLE	SYSTEM STATE VARIABLE
PB1	Frame status (0=head-of-frame, 1=end-of-frame)
PH1	Address of transmitting station
PH2	Address of destination station
PH3	Address of receiving station
PH4	Length of data field in frame (oct)
PL1	Frame transmission delay (incl preamble) (s)
PL2	Frame transfer delay (s)
PL3	Backoff delay remaining (s)
PL4	Inter-frame delay remaining (s)
XF$DEV,..,XF$DEV+2	Savevalues used by external random no. generators
MX$UNOD	Array of size &Nx&N used by ext routine &SRANDINI
ML$MAXBKDEL(k,1)	Maximum backoff delay (s) at station k
AC1	Simulator absolute clock time (s)
M1	Transaction transit delay (s)
CH(i)	Current contents of user-chain i
LR(i)	=1 if logic switch i reset, else=0
LS(i)	=1 if logic switch i set, else=0
FNU(i)	=1 if facility i not in use, else=0

In order to simplify the description of the model, variables that are not strictly relevant to the protocol will not be described. These variables are used in controlling the simulation (eg model warm-up time, run time, sample intervals, random number streams etc) and in the measurement of performance.

2.4 Simulation results

Two simulation experiments were conducted. The first experiment was designed to compare the output from the simulation model, referred to as Model "A", with data from an analytic model. This activity validated the operation of the model. In the second experiment, the protocol delays that were assumed to be zero in the analysis of Model "A" (viz &M1, &M2, &M3, &D1, &D2, &D4, &D7, and &D12) were assigned finite values to assess their effect upon the performance of the protocol. This simulation model is referred to as Model "B".

Several analytical models of Ethernet have been reported in the literature(refs.5,23,28,35). Lam's semi-Markov model is considered to be one of the most accurate analytic representations of the Ethernet MAC protocol(ref.9). The analytic formulae are expressed in convenient form in Appendix II.

Listed below are the parameter values assumed in Model "A".

(1) Single segment LAN topology

(2) Network of 5 stations equally spaced over a distance of 2 km

(3) Unit capacity buffer at each station

(4) 10 M bit/s data transmission rate

(5) Identical, independent Poisson-distributed frame generation rates

(6) 300 oct fixed length frames (including 26 oct overhead)

(7) Propagation delay linearly proportional to distance (5 μs/km)

(8) Inter-frame gap equal to the maximum propagation delay (10 μs)

(9) Standard backoff attempt threshold (10)

(10) Standard re-transmission attempt limit (16)

(11) Other delays neglected (including jam)

Law and Carson's sequential analysis method(ref.25) was employed to estimate the steady-state value of the transfer delay. For each run of the model several hundred initial observations were discarded to avoid biasing the results. Kimbler's algorithm(ref.20) was used to determine the number of observations to be discarded (viz the model "warm-up" period). The results are depicted in figure 9 by a graph of the 95% confidence interval of the normalised mean transfer delay verses normalised mean throughput. The value of the normalised maximum mean throughput asymptote shown is derived in Appendix II.

The simulated and analytic mean delays are generally in excellent agreement since most of the confidence intervals include the analytic values. This result confirms that the error in approximating normal (un-slotted) operation by a discrete imbedded Markov chain model is minimal.

In figure 10, the probability of zero channel assignment delay is plotted against normalised mean throughput. The agreement with analysis is again excellent despite the fact that the simulation model, which replicates the operation of the backoff algorithm in the protocol, does not satisfy the analytic assumption that the probability of a successful transmission within any one of the slots subsequent to a collision is constant (equal to $1/e \approx 0.368$). Discrepancies are attributed mainly to sampling error but this could not be verified because interval estimates were not available from the simulation.

Other results obtained from the simulation model include the (unit) buffer overflow probability (viz probability that a frame is rejected because of insufficient buffer capacity), and the mean transmitter and receiver utilisations. Figure 11 shows that the buffer overflow probability is approximately logarithmically dependent on the normalised mean throughput over the range of throughputs considered. This is emphasised by a line of regression which is fitted to the data and constrained to pass through the coordinate whose normalised mean throughput is the theoretical maximum and whose probability value is unity.

The transmitter and receiver utilisations represent the fractions of time that the transmitter and receiver facilities (1,...,&N and 1+2*&N,...,3*&N respectively) are busy transmitting (complete and partial) frames onto, and receiving those frames from, the transmission medium. Since the specified frame arrival rates at the MAC sublayer of each station are identically distributed, the utilisations are taken as weighted averages of the individual station utilisations in order to improve statistical accuracy. The linear dependence of the utilisations on mean throughput is illustrated in Figure 12. The lines of regression fitted to the data are constrained to pass through zero utilisation at zero throughput. A useful check to confirm correct model operation is to compare the slopes of the lines of regression with the theoretical values. The weighted transmitter utilisation should be equal to E[X]/100, the ratio of the aggregate mean frame arrival rate (λ) to the maximum throughput rate ($1/(E[T_p]+T_g)$) (see Appendix II for symbol definitions). The slope of the regression line for the transmitter should therefore be 0.008214. The measured slope is 0.00803. The ratio of the slopes of the lines of regression should be

(&N-1)=4, the theoretical ratio of the transmitter utilisation to the receiver utilisation for a network of &N=5 stations. The reason for the ratio being one less than the number of stations is that transmitting stations are inhibited from receiving their own frames. The measured ratio of the slopes of the regression lines is 3.91. Both measurements are sufficiently close to the corresponding theoretical values to confirm proper model operation.

In the second part of the experiment, the protocol delays previously assumed to be zero were replaced by the maximum values specified in the ANSI 802.3 standard (viz &M1=6, &M2=3, &M3=17, &D1=18, &D2=3, &D4=4, &D7=3, and &D12=32 bits). In addition, the inter-frame delay, &IFDEL, was slightly reduced from the previous value of 10 µs to the "10BASE5" implementation value of 9.6 µs. Pilot runs were conducted for Model "B" and also analysed using Kimbler's algorithm and Law and Carson's sequential procedure. The difference in performance between the two protocols, one with zero delays and the other with maximal delays, was then measured at several throughput values by comparing the outputs from Model "A" and Model "B". Since at each throughput level the point estimator of the mean transfer delay was less for Model "A" than Model "B", the paired sample, one-tailed Student "t" test was employed to ascertain the statistical level of significance of the differences.

Pairs of fixed-length runs were made using common random variates(ref.26, pp 319 to 322) to reduce the variance of the population difference estimator. The number of observations discarded in the warm-up period and the number of observations required for steady-state analysis were chosen at each throughput level to be the maximum of the corresponding values obtained from Model "A" validation and the pilot run of Model "B". One pair of runs was made at each throughput level. In each pair of runs, the first run was of Model "A", and the second was of Model "B". The pairing of observations was accomplished by assigning a serial number to a transaction parameter in each frame transaction as it entered the model. Upon leaving the model at the point of successful reception at the destination station, the calculated transfer delay was stored in an element of a 2-row matrix indexed by the transaction serial number. One row of the matrix contained observations from the first run and the other row contained observations from the second run. The averages of the paired differences required for statistical analysis were obtained by subtracting the paired observations (viz the elements in each column of the matrix), grouping them serially into 40 equal-sized batches (neglecting any observations left over), and computing the batch means. Note that only paired observations were used. Unpaired observations, indicated by zero-valued elements in the observation matrix, were ignored. Batching was employed to minimise the effect of serial correlation inherent in the observed differences.

The results of the "t" test are tabled below. The estimated values of the normalised mean transfer delay, $E[D_A]$ for Model "A" and $E[D_B]$ for Model "B", are shown as functions of both the mean aggregate frame generation rate (λ), and the normalised mean throughputs, $E[X_A]$ and $E[X_B]$. The latter are dependent on $P_{bA}(\lambda)$ and $P_{bB}(\lambda)$, the load-dependent buffer overflow probabilities for the respective models. In addition, the statistical confidence levels are shown which represent the probability of $E[D_A]<E[D_B]$.

Mean frame generation rate	Normalised Mean Throughput (%)		Normalised Mean Transfer Delay		Prob $\{E[D_A)]<E[D_B]\}$
$\lambda(s^{-1})$	$E[X_A]$	$E[X_B]$	$E[D_A]$	$E[D_B]$	
500	12.0	12.0	1.096	1.100	>0.995
875	21.0	21.0	1.183	1.191	0.95
1500	35.8	35.8	1.404	1.417	0.95
2000	47.2	47.3	1.674	1.781	>0.995
2500	56.9	57.3	2.242	2.455	>0.995
3000	63.5	64.9	3.086	3.534	>0.95
3500	67.4	69.1	4.164	4.535	>0.95

The results indicate that the mean transfer delay for Model "B" has a very high probability of being greater than that of Model "A". The relative difference between the normalised transfer delays increases monotonically with respect to Model "A" delays, from about 0.3% at 12.0% normalised mean throughput to 14.5% at 63.5% normalised mean throughput, but drops to 8.9% at the highest throughput level. Whether or not the inconsistency in this trend is due to a statistical fluctuation or is real cannot be ascertained without significantly increasing the sample size. This action was not undertaken due to the considerable expense involved.

The 95% confidence intervals for the normalised mean transfer delay are depicted in figure 9 as a function of the normalised mean throughput to facilitate the visual comparison of the performance predicted by Models "A" and "B" and the analytic model.

3. TOKEN RING SIMULATION MODEL

ANSI 802.5 is a token-passing protocol designed to transmit data in a closed ring network as shown in figure 13. The relationship of the Token Ring protocols to the OSI Reference Model is depicted in figure 14. The principles of operation of the MAC protocol relevant to the simulation model are described in Appendix III.

In Section 3.1 the structure of the simulation model is described with the aid of GPSS block diagrams. As in the case of the Ethernet simulation model, the Token Ring simulation model does not take account of network initialisation, error recovery and other procedures not directly involved in frame transmission. Neither does it allow for the corruption of frames during transmission over the medium. The structure of the model is much simpler than the Ethernet model, reflecting the fact that it is only an approximation of the protocol. Special techniques are described in Section 3.2 to model the token holding timer and to improve the run-time efficiency of the model. The variables of the simulation model are described in Section 3.3.

Despite the approximations, the model is very flexible. Any number of stations greater than one may be specified, the stations may be positioned anywhere on the ring, and different frame generation rate and frame length distributions may be specified. The major limitation is that only one level of frame priority is allowed because the states and transitions in the MAC protocol responsible for servicing multiple priorities are not modelled. Modifications can be easily made to include priorities, but this was not done since, unlike Ethernet, there was no customer requirement for an exact model of the Token Ring protocol.

Methods similar to those described in Section 2 for Ethernet were employed in analysing simulation output. Analytic results were also used to validate the simulation model since a Token Ring test bed facility was unavailable. The results obtained during model validation are presented in Section 3.4.

3.1 Model structure

The simulation model mimics the frame generation process in the LLC sublayer and the frame delivery process in the MAC sublayer. Only MAC states 0 to 3(ref.13) are modelled. To avoid the repication of GPSS blocks when modelling the processes in each station, structural folding is employed as described in Section 2.1.

3.1.1 Frame generation

The first four blocks in the frame generation process, depicted in figure 15, are identical to those employed in the Ethernet model (see Section 2.1.1). The mean frame generation rate at station k is specified in ML$STAT(k,3). The blocks generate the sequences of frame transactions and assign a transmitting station address to each transaction parameter PH1.

A random transmission duration is assigned to PL1. It is computed by summing an exponentially-distributed data field length whose mean is specified in ML$STAT(PH1,4) to the frame overhead, &OVRHEAD, and dividing the resultant frame length (multiplied by eight to convert octets to bits) by the data transmission rate, &DRATE. The value of PL1 is then tested to see whether it is less than the maximum token holding time, &THTMAX. If it is not, the transaction returns to the ASSIGN block at address GEN1 to enable a new value to be assigned to PL1. If this is not done, a long frame would never be serviced since there would be insufficient time available for a station to transmit it. Furthermore, because frames are buffered in first-in first-out (FIFO) order, stations would be prevented from retaining the token which would eventually cause the model to shut down.

The transaction is then marked with its generation time, which is subsequently used to compute the frame transfer delay, and it is deposited in the station's buffer (user-chain <PH1>). Notice that prior to entering the user-chain, frame transactions pass through a QUEUE block and a LOGIC block which are part of a special mechanism, described in Section 3.2, designed to improve the run-time efficiency of the model.

3.1.2 Frame delivery

The frame delivery process, illustrated in figure 16, is responsible for transmitting frames from station to station and controlling the order in which stations transmit. Control is exerted by passing a (single) permission token around the ring. A station holding frames in its buffer is only given access to the transmission medium when it captures the token. The frames and the token are represented in the model by transactions.

A token transaction is generated at time zero and waits at a GATE block pending the arrival of a frame. This block is used in an artifice to improve the run-time efficiency of the model and is described in Section 3.2. Ignoring the effect of the GATE block for the moment, the address of the next station to be visited by the token is assigned to parameter PH1. The address is efficiently computed by modulo division (using the GPSS arithmetic operator "@"). Initially <PH1> is zero. The next station address is equal to <PH1>+1 if <PH1> < &N, or 1 if <PH1>=&N (viz (<PH1> mod &N)+1). The token transaction then experiences a delay in an ADVANCE block equal to the station latency, ML$STAT(PH1,2), which is the sum of the minimum token holding time, &THTMIN, and the propagation delay between the stations (equal to the distance between the stations, ML$STAT(PH1,1), divided by the propagation speed).

Upon arrival of the token at a station, the station's token holding timer is reset. In the model the reset time is stored in XL$THT and initialised to the simulator clock time, C1. A TEST block at address TOK1 checks whether the station's buffer (viz user-chain <PH1>) is empty. If it is, the token transaction returns to the GATE block at address TOKO, where it is passed onto the following station. Otherwise, the frame transaction is removed from the user-chain by an UNLINK block and sent to block address TOK2. A frame transaction is uncoupled from the user-chain in FIFO order by the token transaction, and the token transaction is deleted.

Before the uncoupled frame transaction is processed, the amount of time available to the station to transmit the frame is compared at a TEST block with the time required to transmit the frame (which is stored in transaction parameter PL1). If there is insufficient time available to transmit, the token holding flag is set (viz the value of the boolean variable BV$THTXPRD is set to unity), so that the transaction is routed to a SPLIT block and returned to user-chain <PH1> where it is positioned in front of the other transactions present. This is achieved by specifying last-in first-out (LIFO) order in the LINK block and guarantees that the transaction will the first one uncoupled when the token transaction subsequently returns to the station. The daughter transaction from the SPLIT block then becomes the new token transaction and is sent to block address TOKO so that it can be passed to the next station.

If there is sufficient time to transmit the uncoupled frame transaction, the frame transaction is delayed in an ADVANCE block for the full transmission period. It then passes through a series of blocks which form part of a mechanism for reducing the run time of the model before being routed to the TEST block at address TOK1 (see Section 3.2). If there is another frame transaction in user-chain <PH1> awaiting transmission, it then proceeds to uncouple the frame transaction which has been waiting the longest and the transmission cycle, previously described, recommences. Otherwise, the transaction becomes a new token transaction and is sent to the GATE block at address TOKO to be passed onto the next station.

3.2 Model artifices

Special techniques are employed to model the token holding timer at each station and to improve the run-time efficiency of the model. The GPSS blocks and variables associated with these mechanisms have no equivalents in the protocol itself.

(1) Token holding timer

The token holding timer in each station is implemented by a savevalue, XL$THT. Only one savevalue is required because only one station at a time ever needs to interrogate the timer. The savevalue is initialised to the simulator clock time, C1, when the timer is reset. The token holding time remaining is computed by the variable V$THTREM=XL$THT+&THTMAX-C1, where &THTMAX is the maximum token holding time.

(2) Token activity monitor

The token activity monitor is a mechanism used to stop the continuous circulation of the token transaction in the model when the ring is idle (viz no frames on the ring and no station with a frame to transmit). The monitor greatly improves the run-time efficiency of the model at low frame generation rates.

Several blocks are included in the model to sense ring idleness and to stop and restart the motion of the token transaction. In the frame generation process, a QUEUE block is used to measure the aggregate queue length thereby avoiding the complexity of having to sum the individual station buffer sizes, viz CH(1)+CH(2)+...+CH(&N). Whenever a new frame is generated at a station the aggregate queue length, Q(QAGG), is automatically incremented, and a logic switch (LOOP) is set to denote the presence of a new frame.

In the frame delivery process, the GATE block at address TOK0 prevents the circulation of the token transaction on the simulated ring if the LOOP logic switch is reset. Four other blocks are also included in this process. On leaving the ADVANCE block responsible for imparting the transmission delay, a frame transaction enters a DEPART block which decrements the aggregate queue length Q(QAGG). If the queue length is then zero, logic switch LOOP is reset and the transaction is routed to the GATE block at address TOK0, via the TEST block at address TOK1, and becomes the new token transaction. This transaction is then stopped at the GATE block pending the generation of a frame. However, if the queue length is non-zero, the frame transaction returns to the TEST block at address TOK1 without resetting logic switch LOOP and uncouples the next frame transaction from user-chain <PH1>.

3.3 Model variables

The seven categories of model variables defined in Section 2.3, exist in the Token Ring model. For the same reasons stated in that section, the simulation control and performance variables will not be described.

GPSS VARIABLE	IMPLEMENTATION-DEPENDENT VARIABLE
&DRATE	Medium transmission rate (bit/s)
&OVRHD	Frame overhead (oct)
&THTMIN	Minimum token holding time (s)
&THTMAX	Maximum token holding time (s)
&FACTOR	Propagation speed factor

GPSS VARIABLE	NETWORK-DEPENDENT VARIABLE
&N	Number of stations in the network
ML$STAT(k,1)	Distance (m) between stations k and (k mod &N)+1

GPSS VARIABLE	STATION-DEPENDENT VARIABLE
ML$STAT(k,3)	Mean frame generation rate at station k (s^{-1})
ML$STAT(k,4)	Mean length of frames (excluding overheads) transmitted by station k (oct)

GPSS VARIABLE	DERIVED VARIABLE
ML$STAT(k,2)=(&THTMIN+ML$STAT(k,1)/ (&FACTOR*3.0E8)	Station latency (s)
V$THTREM=XL$THT+&THTMAX-C1	Token holding time remaining(s)
BV$THTXPRD=0 if V$THTREM<<PL1>, else=1	Token holding flag

GPSS VARIABLE	SYSTEM STATE VARIABLE
PH1	Address of transmitting station
PL1	Frame transmission delay (incl preamble) (s)
XL$THT	Time token holding timer is reset (s)
C1	Simulator relative clock time (s)
CH(i)	Current contents of user-chain i
Q(QAGG)	Current length of aggregate queue, QAGG
M1	Transaction mark-time (s)
XF$DEV,XF$DEV+1	Savevalues used by external random no. generators

3.4 Simulation results

The correct operation of the simulation model was confirmed by comparing the mean frame transfer delay derived from the simulation with the mean transfer delay obtained from Bux's analytic formula in Appendix IV.

In order to effect a proper comparison, the following assumptions were made.

(1) Infinite buffer capacity at each station

(2) 1 M bit/s data rate

(3) Network of 50 stations equally spaced over a 2000 m length ring

(4) Identical, independent Poisson-distributed frame generation rates

(5) Length of data field in frames exponentially-distributed with mean of 125 oct

(6) 3 oct fixed frame overhead

(7) Medium propagation speed factor=0.66

(8) 1 μs minimum token holding time

(9) 1 s maximum token holding time (chosen to ensure that at the highest throughput levels a station does not release the token before its buffer is fully depleted)

The steady-state mean frame transfer delay was computed in identical fashion to the Ethernet model, using Law and Carson's sequential method of Batch Means following a warm-up period determined by Kimbler's algorithm.

Variance reduction was effected using the method of Antithetic Variates(ref.1, chap.2). Two runs were employed and the paired observations averaged to produce a single sequence of data for analysis. Various treatments were examined to determine which was the most effective in reducing the variance of the data. Greater variance reduction was achieved using antithetic variates for both frame inter-arrival times and frame lengths (the only exogenous random variables in the model) compared to using antithetic variates for one variable and independent variates for the other.

To obtain a 95% confidence interval with 5% relative precision at throughput levels less than 50%, about the same amount of computational effort was expended on the two antithetic runs as on a single run. However at higher throughputs, antithetic variates were superior. For example, at the 80% throughput level, a total of 102400 observations (viz 51200 observations in each run) were needed using antithetic variates compared to 204800 observations for a single run.

Figure 17 illustrates the dependence of the normalised mean frame transfer delay on the normalised mean throughput. The 95% confidence intervals of the delay obtained by simulation overlap the data values generated by the analytic model. The mean delay was found to be insensitive to the maximum token holding time provided it was greater than the nominal 10 ms value.

4. EXTERNAL PROCEDURES USED IN SIMULATION

GPSS/H is limited in its ability to perform mathematical operations (only addition, subtraction, multiplication, division and modulo division are possible) and to generate random numbers from continuous distributions. However, it does have a very efficient and easy to use high-level language interface which permits Fortran, C, and Pascal procedures to be called to do these tasks.

Several Fortran procedures were called by the Ethernet and Token Ring simulation models to perform mathematical operations and to generate random numbers. The mathematical routines are not described because they were only used in post-run data analysis. The procedures called to generate random numbers are described below and are based on an external uniform (0,1) random number generator(ref.30). Note the GPSS/H convention of referring to an external procedure named "XXXX" by an external ampervariable "&XXXX". In addition, if the procedure is a Fortran function, the first letter of its name identifies the type of function (viz I-N integer; A-H and O-Z real).

&SRANDEF(M,IV)

This subroutine must be called to initialise the integer vector IV prior to calling any of the external random number generators (viz &REXPON, &RUNIF and &IDEST). M>0 (integer) specifies the number of random number streams to be initialised. It initialises each stream with a common seed value of 266301881 (similar to the GPSS/H built-in generators). The offset of each stream j (j=1,2,...,M) is internally generated by the routine according to the formula 1000000*j. (The same formula is used by GPSS/H for default streams).

&REXPON(N,IV,A)

Generates an exponentially distributed real number with mean and standard deviation A>0 (real). N is the (integer) stream number and IV is an integer vector (see routine &SRANDEF).

&RUNIF(N,IV,A,B)

Generates a uniformly distributed real number in the open interval (A,B) where A and B are real. The mean and variance are (A+B)/2 and (B-A)*(B-A)/12 respectively. N is the (integer) stream number and IV is an integer vector (see routine &SRANDEF).

&SDESTINI(M,RP,RC,ID)

This routine is used to initialise the real array RC and the integer array ID prior to calling the routine &IDEST. It computes the cumulative routing probabilities (viz the elements of RC) from the probability matrix RP. Each (real) element of RP must be specified. RP(i,j) represents the probability of a message being sent by node i (i=1,2,...,M) to node j (j=1,2,...,M) where M is the (integer) number of network nodes. The row elements of RP must lie in the closed interval [0,1] and sum to 1.0 $\pm 10^{-8}$. (It is permissible for diagonal elements of RP to be non-zero.) The routine also computes the elements of ID, a matrix of node addresses ordered such that if i>j, the probability of node ID(k,i) being the recipient of a message from node k≠i,j is greater than, or equal to, the probability of node ID(k,j) being the recipient.

&IDEST(N,IV,L,M,RC,ID)

Generates the (integer) number of the node to which a message will be randomly directed in a network of M (integer) nodes from the sending node L (integer) where L=1,2,...,M. RC is a real matrix and ID is an integer matrix (see routine &SDESTINI). N is the (integer) stream number and IV is an integer vector (see routine &SRANDEF).

5. MODEL VERIFICATION

Confidence in a model is ultimately established by comparing model predictions with the actual data obtained from the system under investigation. But first, the operation of the model must be scrutinised to eliminate programming and conceptualisation errors. All of the possible events, logical conditions and transaction paths in the model, corresponding to the true states and transitions in system, should be exercised to ascertain correct operation.

The interactive debugging facility of GPSS/H is particularly useful for verifying the operation of a model. Model execution can be interrupted at any stage to enable the current values of state variables (eg transaction parameters, logical conditions, statistics associated with static entities, etc) to be observed. Single stepping, and stopping at user-specified break-points (viz blocks) and pre-defined trap conditions (viz model states) are supported. In addition, the current state of the model may be checkpointed (saved) and restored at a later time after further execution of the model. Unless a trap or break-point is encountered, the model executes until completion or until a fatal run-time error occurs. When tracing a fatal run-time error, the state of the model must be saved periodically to avoid having to re-run it from the start. The checkpoint/restore facility can save considerable time and effort when run lengths are large.

Most features of the interactive debugger were utilised in verifying the operation of the Ethernet and Token Ring models. The debugger is efficient and the output is presented in a format similar to the standard printed output and is easy to read provided it is directed to a 132 column visual display.

6. SIMULATION OUTPUT ANALYSIS

In Section 6.1 the problems associated with analysing simulation output data are briefly discussed and several methods of achieving variance reduction in LAN protocol models are described. The measured execution times of the Ethernet and Token Ring models are presented in Section 6.2.

6.1 Bias and variance reduction

The accuracy of the performance estimates derived from a simulation model is critically dependent upon the length of the simulation run. Practical considerations of cost and time often prevail so that runs are shorter than desired. The challenge in analysing simulation output is to concurrently minimise the statistical bias due to model start up, and the variance of mean estimators, given a limited number of samples and a significant degree of serial correlation in the samples.

Provided the underlying distribution of the output random variables is normal and stationary, and the variables are independent, the statistical accuracy of an estimator of a mean value, measured by the relative precision of its confidence interval, is a simple function of sample variance. A sample variance can be easily obtained from GPSS/H default statistics or by using a TABULATE block and an associated TAB'E statement. However, simulation output variables seldom satisfy these stipulations and standard GPSS/H statistics, particularly sample variances, consequently provide little assistance in estimating the accuracy of mean estimators.

The problem of obtaining steady-state measures of performance is compounded by the need to ascertain how long a simulation model must be warmed up to achieve steady-state conditions. The duration of the warm-up period, or the "initial transient" as it is sometimes called, depends upon the choice of initial conditions. Unless the samples acquired during this period are discarded, the steady-state statistics will be biased. Several heuristic procedures are available to estimate the number of samples that should be discarded(refs.7,11,18,20,25,30,32). Most of the procedures have been tested on models of M/M/1 queues which are amenable to mathematical analysis and considered difficult to analyse since they exhibit a very gradual transition from transient to steady-state behaviour. Whether these procedures are as effective in more complex simulation models is a topic of current research.

In the analysis of the Ethernet and Token Ring models, Kimbler's double exponential tracking filter(ref.20) was employed because it has been shown to discard fewer samples and is computationally more efficient compared to other methods. It also performs as well as the more complex methods. A trend toward shorter warm-up periods at higher throughput levels was observed in all of the models. Warm-up periods varied between 0.05 s and 1.76 s for Ethernet Model "A", and between 0.16 s and 5.4 s for the Token Ring model.

An effective method of reducing the sample variance, and reducing the run lengths required for a given statistical accuracy, is to use a variance reduction technique. Several techniques(refs.1,17,26) are available for analysing single systems and for comparing several systems. Amongst the easiest methods to use for single systems are Antithetic Variates and Concomitant Variates. Both methods work best if the input random variables are highly correlated with the variables under analysis. Selecting input random variables is usually very difficult without first conducting pilot runs, especially when there are many from which to choose. Poor choices of variables will produce little variance reduction or may indeed lead to an increase in variance. The key to success in using both of these techniques is to properly synchronise the input variates in repeated runs. This is achieved by generating identical random number streams(ref.30). Despite that fact that in some models only partial synchronisation can be achieved, the amount of variance reduction may still justify the programming effort.

The method of Common Random Numbers is employed when comparing the performance of several similar systems. In the case of only two systems, runs are made in pairs using identical streams of random numbers. The two-sample Student "t" test is then applied on the paired output variates. The correlation between the variates is usually very effective in reducing the variance of the estimator of the mean difference between performance measures compared to using independent random number streams. Additional variance reduction techniques are rarely used concurrently with Common Random Numbers due to programming difficulties and because the combination of techniques is frequently ineffectual.

The method of Common Random Numbers was used to investigate the effect of varying Ethernet parameters, whilst Antithetic Variates was used in analysing the performance of the Token Ring model.

Other variance reduction methods assessed to be of potential in LAN protocol modelling include External Control Variates(ref.34), Conditional Expectations(ref.19) and Indirect Estimation(ref.26, pp 361-363 and ref.8). Whether these methods are employable depends entirely upon the individual character of the model. They are generally limited to Markovian queuing models.

An alternative, albeit indirect, approach to achieving variance reduction, is to replace computationally intensive portions of a simulation model by analytic procedures. For example, instead of simulating a process delay by passing a GPSS transaction through a sub-model (viz a series of GPSS blocks), a transaction may be delayed by a single ADVANCE block whose (deterministic or random) delay value is computed numerically by an external procedure. The values of the independent variables in the formula are copied from the global and local variables in the model when the transaction enters the ADVANCE block. Such hybrid simulation techniques are analogous to analytic network "decomposition" techniques (eg Norton's theorem in electrical circuit theory) and are only applicable if the substituted portions of a model are "weakly" interactive with each other and the remaining (viz simulated) portion of the model(ref.4). Hybrid models of general queuing networks(ref.19) and computer communication networks (refs.3,33) are in wide use and significant improvements in computational efficiency have been claimed for them. Hybrid models hold considerable promise for LAN protocol modelling(ref.24).

The serial correlation present in most simulation output variables leads to the under-estimation of variances and confidence interval widths when classical statistical methods are used. Better methods are based on the analysis of a standardised time series(ref.31), the spectral estimation of of a time series(ref.6) and modified classical methods(ref.27). The simplest modified classical method, that of Batch Means, seeks to reduce the effects of serial correlation by grouping, or "batching", the data so that the batch means are less correlated than the individual sample values. Law and Carson's sequential batch method(ref.25) was used in computing confidence intervals for the mean frame transfer delay times derived from the Ethernet and Token Ring models.

6.2 Model execution times

The Ethernet and Token Ring simulation models are expensive to run. Average execution times per GPSS/H block, measured on a dedicated (viz single user) DEC-VAX 8200 computer running the VMS 5.1 operating system, varied between 100 μs and 114 μs for the Token Ring model, and between 149 μs and 192 μs for the Ethernet model. The differences are primarily due to the different types of blocks used and amount of processing involved in the management of transactions. The Ethernet model, for example, has many more transactions and transaction parameters which impose greater overhead.

More CPU time was needed on the host computer to execute the same interval of simulated time at higher throughputs because the number of active transactions in the models increases with the simulated load level. The CPU times to execute each model were measured. The results, tabled below, include a slight amount of overhead for the collection and analysis of observations and apply to simulation without the use of variance reduction techniques.

ETHERNET MODEL

Normalised mean throughput (%)	Simulated time (s)	CPU time (s)	Execution time ratio
12	3.3	44	13
21	1.4	34	24
36	2.7	120	44
47	3.6	250	69
57	10.7	1095	102
64	9.7	1403	145
67	18.0	3323	185

TOKEN RING MODEL

Normalised mean throughput (%)	Simulated time (s)	CPU time (s)	Execution time ratio
10	261	440	1.68
20	65	210	3.23
30	43	182	4.23
40	32	180	5.63
50	51	334	6.55
60	43	321	7.47
70	293	2146	7.32
80	256	2121	8.29
90	455	3104	6.82

Note the difference in the execution time ratios between the two models. The execution time ratio is the ratio of the CPU time to the simulated time interval. The figures indicate that the Token Ring model is considerably more efficient than the Ethernet model and reflect the additional complexity and transaction activity in the Ethernet model.

The execution time ratio for the Ethernet model increases very rapidly, due to the increased frequency of collisions, as the normalised mean throughput approaches the theoretical saturation limit of 82.14%. This contrasts with the Token Ring model which demonstrates more linear behaviour up to the saturation limit of 100%. The execution efficiency of the Token Ring model was greatly enhanced at low throughputs by the token activity monitor mechanism described in Section 3.2.

7. DISCUSSION AND CONCLUSIONS

Two LAN protocol simulation models, an Ethernet (ANSI 002.3) and a Token Ring (ANSI 802.5) model, were built to assess the potential of the GPSS/H simulation language for modelling the performance of LAN protocols. The Ethernet and Token Ring protocols were selected because they are relevant to military LAN applications and because well-documented analytic models exist which can be used to validate simulation models.

It was found that the process interaction approach of GPSS/H leads to a natural representation of frames and tokens as dynamic entities (transactions) which incur delays as they move through static elements (blocks) in the model. The GPSS blocks represent incremental units of processing associated with the transmission and reception of the frames and tokens. For example, the commonly used SEIZE and ADVANCE blocks, and conditional GATE and TEST blocks impose transaction delays; PREEMPT blocks and unconditional GATE and TEST blocks modify processing sequences; LOGIC, SAVEVALUE, MSAVEVELAUE and BLET blocks alter the global state of the model; ASSIGN blocks change the values of parameters associated with individual transactions; GENERATE and SPLIT blocks inject transactions into a model; and TERMINATE blocks remove transactions from a model. There are many additional block types that have not been mentioned above.

In the Ethernet model, separate transactions are used to denote the head and the end of each frame sent to each station, whereas in the Token Ring model a single transaction is used to model a frame and a token circulating around the ring. Some of the parameters associated with these transactions include frame length, and source and destination station addresses.

State diagrams of the LAN protocols were found to be very useful aids in designing simulation models. The states and transitions associated with the error-free operation of the protocols were mapped into GPSS block diagrams, representing the logical conditions, transactions paths, and synchronisation of events in the models, prior to program coding.

Programming artifices were employed to model timers. In the Ethernet model a dummy facility is used to permit timer transactions to be removed from the GPSS future events chain prior to time-out because no direct method exists in the language for removing a transaction from this chain. In the Token Ring model, a separate timer transaction is not employed to model the token holding timer because no change of state results from the expiry of the timer. Instead, the timer is implemented by a global variable (a savevalue) which is initialised to the simulated time at the receipt of a token and subsequently adjusted at the start of each frame transmission when the amount of time remaining prior to timer expiry is computed.

The only criticisms of GPSS/H are its inability to accurately sample random numbers having continuous distributions and its inability to perform mathematical operations other than basic arithmetic. These problems are being addressed by the manufacturer. Fortunately, GPSS/H includes a simple and efficient mechanism for calling external routines to accomplish these tasks so that these omissions only cause inconvenience and are not major deficiencies in the language.

The GPSS/H interactive debugging facility was found to be an excellent tool for verifying correct model operation, and locating and identifying the causes of run-time errors.

Difficulties associated with the statistical analysis of non-normal and serially correlated simulation output data were outlined. The implications are that standard GPSS/H statistics, particularly sample variances, are of little assistance in estimating the accuracy of steady-state measures of performance. In addition, bias may be introduced unless samples acquired during a model's *initial transient*, or "warm-up" period, are excluded from the analysis. Methods of improving the efficiency of statistical estimators, including variance reduction techniques and hybrid simulation, were described.

In both the Ethernet and Token Ring models, Kimbler's algorithm was used to estimate the number of samples to be discarded during the warm-up period, and Law and Carson's sequential method of Batch Means was employed to determine confidence intervals for steady-state mean values. Two variance reduction techniques were employed. The Common Random Variates technique was used for comparing two models of Ethernet, and Antithetic Variates for analysing the Token Ring model.

Experience gained from simulating the steady-state operation of the Ethernet and Token Ring MAC protocols suggests that GPSS/H is a very flexible, user-friendly and efficient language. GPSS/H should be equally suited to modelling other LAN protocols.

8. ACKNOWLEDGEMENTS

The author wishes to acknowledge with gratitude the encouragement and scholarly advice provided by Mr J.G. Schapel (WSRL) and Dr D.M. Panton (Maths Dept, SA Institute of Technology), and the assistance of Messrs M.J. Coleman and R. Currie (WSRL) in the preparation of the diagrams for this report.

ABBREVIATIONS

ANSI	American National Standards Institute
AUI	Access Unit Interface
c.i.	confidence interval
CSMA/CD	Collision Sense Multiple Access (with) Collision Detection
CTS	clear to send
DEC	Digital Equipment Corporation
DTE	Data Terminal Equipment
FDDI	Fiber Distributed Data Interface
FIFO	first in, first out
GPSS	General Purpose Simulation System
IBM	International Business Machines Corporation
km	kilometre
LAN	Local Area Network
LIFO	last in, first out
LLC	Logical Link Control
m	metre
mod	modulo
ms	milli-second
ns	nano-second
MAC	Media Access Control
MAU	Medium Attachment Unit
M bit/s	mega-bits/ second
oct	octet
OSI	Open Systems Interconnection
SQE	Signal Quality Error
THT	Token Holding Time(r)
TRR	Return to *repeat* (state) Timer
XAC	transaction
μs	micro-second

REFERENCES

No.	Author	Title
1	Bratley, P. Fox, B.L. and Schrage, L.E.	"A Guide to Simulation". 2nd Ed. Springer-Verlag New York NY, 1987
2	Bux, W.	"Local Area Subnetworks: A Performance Comparison". Advances in Local Area Networks. Ed. Kuemmerle, K., Limb, J.O., and Tobagi, F.A. pp 363-382. IEEE Press New York NY., 1987
3	Chiarowongse, J., Park, J.T., Srinivasan, M.M., and Teorey, J.J.	"A Simulation Model for Large Interconnected Local Area Networks by Decomposition Techniques". IEEE Proc. Simulation of Computer Networks, pp 64-72, 1987
4	Courtois, P.	"Decomposability, Instabilities, and Saturation in Multiprogramming Systems". Comms. ACM. Vol. 18, No. 7, pp 371-377, July 1975
5	Fapojuwo, A.O., Chan, W.C. and Irvine-Halliday, D.	"An Iterative Approximation for Modelling Buffered CSMA-CD LANs". J. Electrical and Electronic Engineering IE Aust & IREE Aust. Vol 8, No. 2, pp 102-111, June 1988
6	Fishman, G.S. and Kiviat, P.J.	"The Analysis of Simulation-Generated Time Series". Management Science Vol. 13, No. 7, pp 525-557, March 1967
7	Fishman, G.S.	"Bias Considerations in Simulation Experiments". Ops. Res. Vol. 20 pp 785-790, 1972
8	Glynn, P.W. and Whitt, W.	"Indirect Estimation via $L=\lambda W$". Oper. Res. Vol. 37 No. 1, January/February 1989
9	Gonsalves, T.A.	"Measured Performance of Ethernet". Advances in Local Area Networks Ed. Kuemmerle, K., Limb, J.O., and Tobagi, F.A. pp 383-410. IEEE Press New York NY., 1987
10	Haigh, P.L.	"A Methodology for Simulating Computer Systems". IEEE Ann. Simulation Symposium pp 277-313, 1982

No.	Author	Title
11	Heidelberger, P. and Welch, P.D.	"Simulation Run Length Control in the Presence of an Initial Transient". Ops. Res. Vol. 31, No. 6 pp 1109-1144, November/December 1983
12	Hendriksen, J.O. and Crain, R.C.	"GPSS/H User's Manual". 2nd Edition. Wolverine Software Corp., Annandale VA., 1983
13	Institute of Electrical and Electronic Engineers Inc.	"Token Ring Access Method and Physical Layer Specifications". ANSI/IEEE Standard 802.5, March 1985
14	Institute of Electrical and Electronic Engineers Inc.	"Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications". ANSI/IEEE Standard 802.3, June 1985
15	International Business Machines Corp.	"General Purpose Simulation System V User's Manual". 3rd Edition. Technical Publications, IBM Corp., White Plains NY, 1977
16	International Organization for Standardization	"Data Processing - Open Systems Interconnection - Basic Reference Model". Draft Proposal ISO/DP 7498, 1982
17	James, B.A.P.	"Variance Reduction Techniques". J. Opl. Res. Soc., Vol. 36, No. 6 pp 525-530, 1985
18	Kelton, W.D.	"The Startup Problem in Discrete-Event Simulation". PhD Thesis. Univ. of Wisconsin-Madison, 1980
19	Keuhn P.J.	"Approximate Analysis of General Queuing Networks by Decomposition". IEEE Trans. Comput. Vol. 27 No. 1 pp 113-126, January 1979
20	Kimbler, D.L.	"A Survey of Current Methods for the Elimination of Initialization Bias in Digital Simulation". IEEE Ann. Simulation Symposium pp 133-152, 1987
21	Kiviat, P., Villaneua, R. and Markowitz, H.	"The SIMSCRIPT II Programming Language". Prentice-Hall, Englewood Cliffs, N.J., 1969.

No.	Author	Title
22	Konheim, A.G. and Meister, B.	"Waiting Lines and Times in a System with Polling". J. Ass Comput. Mach., Vol. 21, pp 470-490, 1974
23	Lam, S.S.	"A Carrier Sense Multiple Access Protocol for Local Area Networks". Comput. Networks, Vol. 4 pp 21-32, 1980
24	LaRue, W.W., Frost, V.S., and Shanmugan, K.S.	"Some New Efficient Techniques for the Simulation of Computer Communications Networks". IEEE Proc. Simulation and Computer Networks, pp 152-158, 1987
25	Law, A.M. and Carson, J.S.	"A Sequential Procedure for Determining the Length of a Steady-State Simulation". Ops. Res. Vol. 27., No. 5 pp 1011-1025, September/October 1979
26	Law, A.M. and Kelton, W.D.	"Simulation Modeling and Analysis". Mc Graw-Hill Inc., 1982
27	Law, A.M.	"Statistical Analysis of Simulation Output Data". Ops. Res. Vol. 31, No. 6 pp 983-1029, November/December 1983
28	Metcalfe, R.M. and Boggs, D.R.	"Ethernet: Distributed packet switching for Local Computer Networks". Comm Ass. Comput. Mach. Vol. 19, pp 395-404, July 1976
29	Ross, F.E.	"FDDI- A Tutorial". IEEE Comms. Vol. 24, No. 5 pp 10-17, May 1986
30	Scholz, M.L.	"An Algorithm for Generating Disjoint Streams of Pseudo-random Numbers for Use in Simulation". WSRL-TN-46/89, 1989
31	Schruben, L.	"Confidence Interval Estimation Using Standardized Time Series". Ops Res. Vol. 31, No. 6, pp 1090-1108, November/December 1983
32	Schruben, L. and Singh, H.	"Optimal Tests for Initialization Bias in Simulation Output". Ops. Res. Vol. 31, No. 6. pp 1167-1178, November/December 1983
33	Schwetman, H.D.	"Hybrid Simulation Models of Computer Systems". Comms. ACM Vol. 21, No. 9 pp 718-723, September 1987.

No.	Author	Title
34	Sharon, A.P. and Nelson, B.L.	"Analytic and External Control Variates for Queueing Network Simulation". J. Opl. Res. Soc., Vol. 39, No. 6, pp 595-602, 1988
35	Tobagi, F.A. and Hunt, V.B.	"Performance Analysis of Carrier Sense Multiple Access with Collision Detection". Comput. Networks Vol. 4, pp 245-259, October/November 1980
36	Traub, A.C. Jr and Zachmann, W.F.	"A GPSS Model of a Complex On-line Computer System". SIGSIM/ACM pp 17-37, 1973
37	Wnek, R.M.	"Simulation of a Distributed Database System incorporating a Routing Optimizer". IEEE Ann. Simulation Symposium pp 139-169, 1985

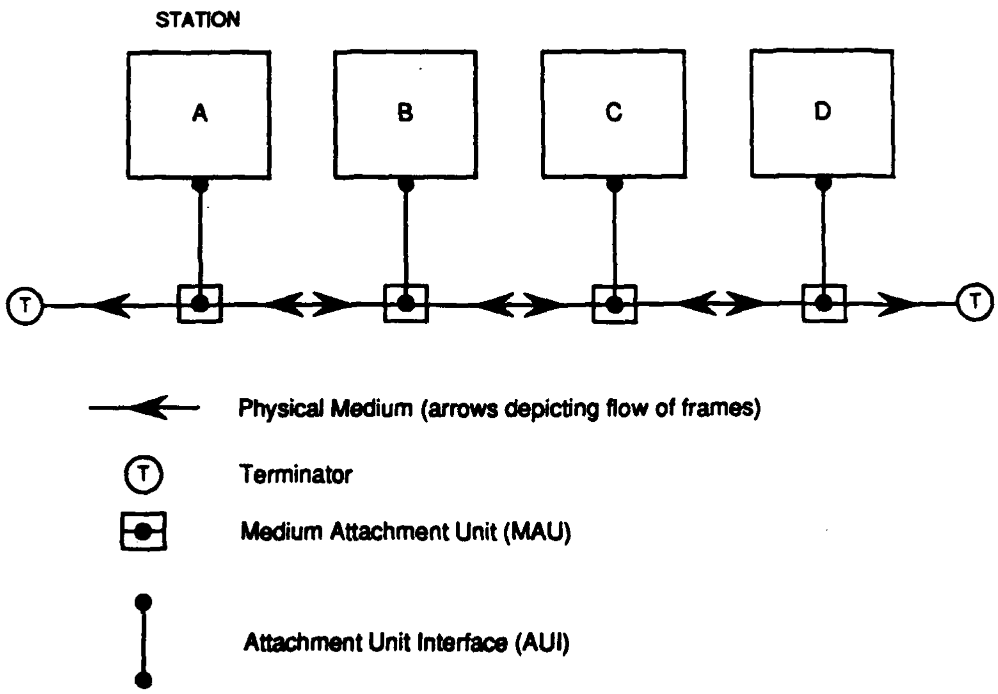

Figure 1. Topology of a single segment Ethernet LAN

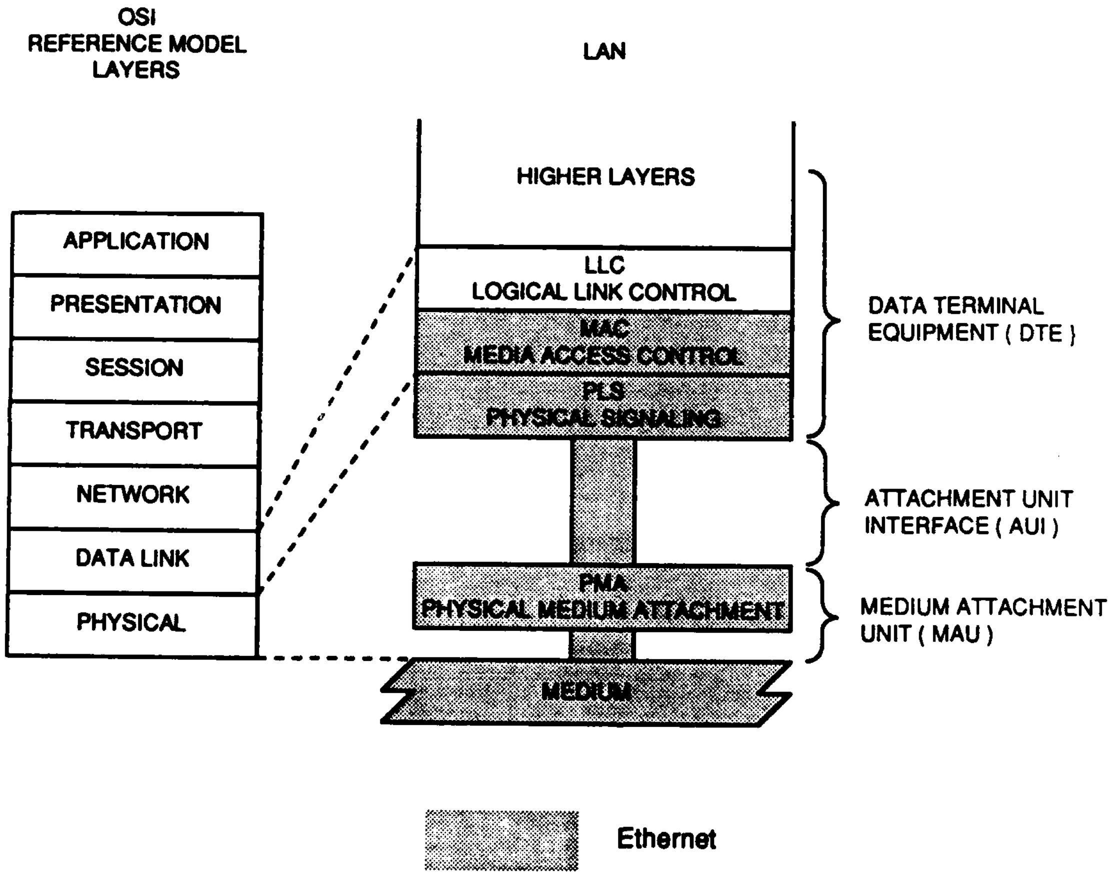

Figure 2. Relationship of Ethernet LAN to OSI Reference Model

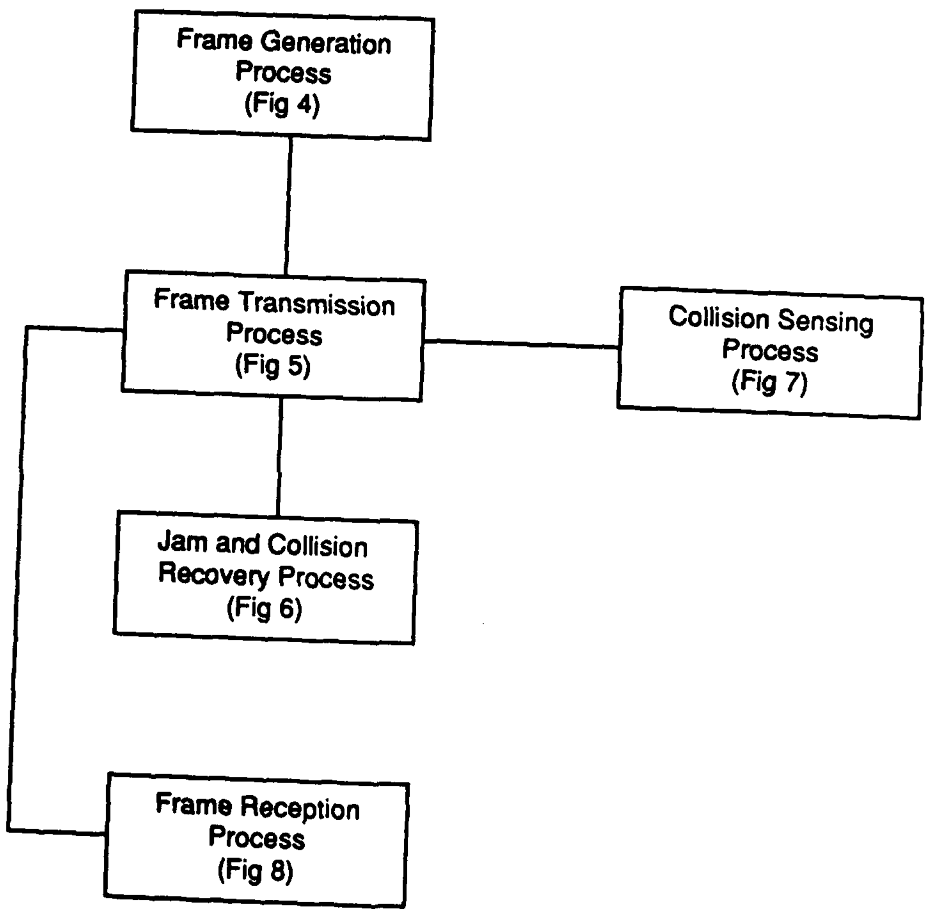

Figure 3. Architecture of the Ethernet model

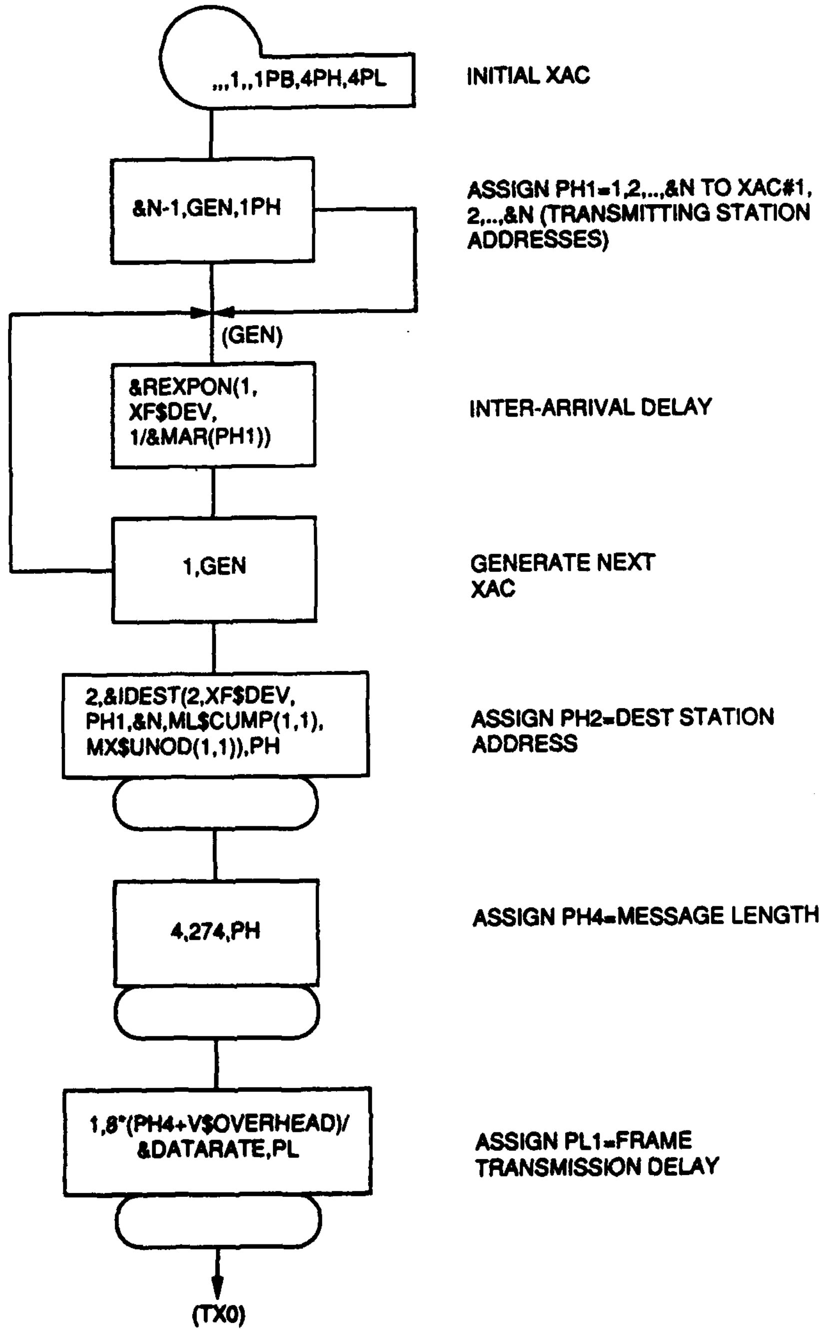

Figure 4. Ethernet: Frame generation process

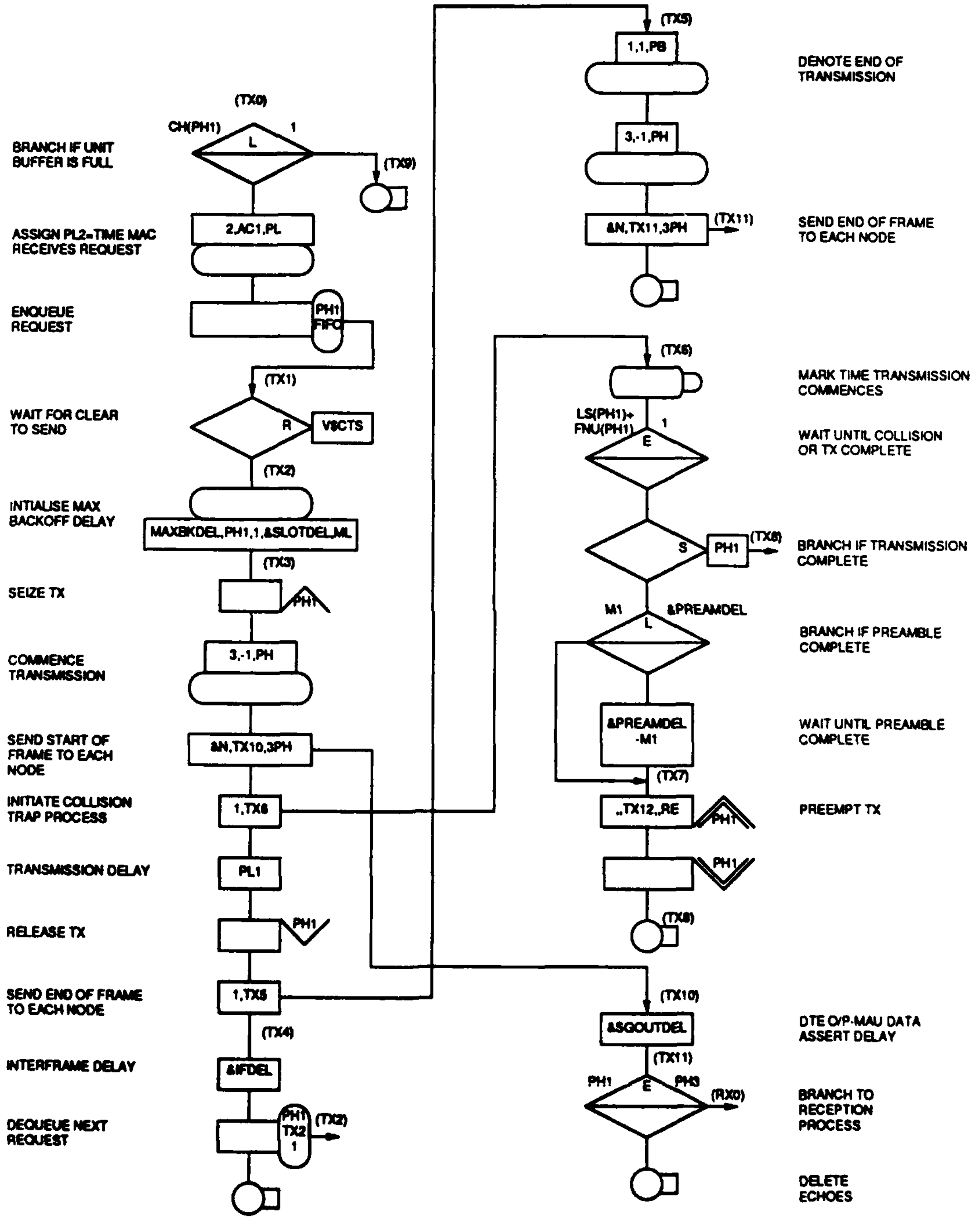

Figure 5. Ethernet: Frame transmission process

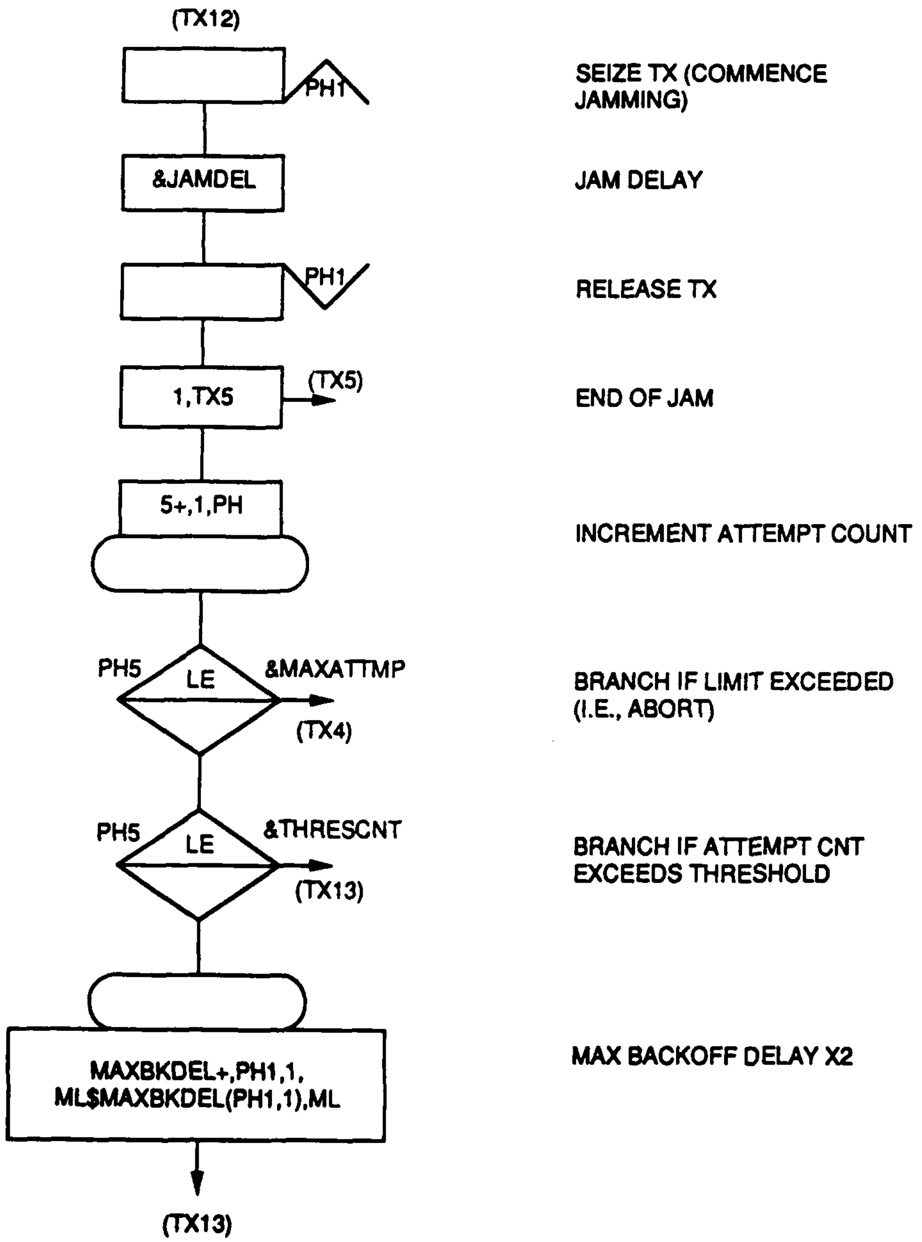

Figure 6. Ethernet: Jam and collision recovery process

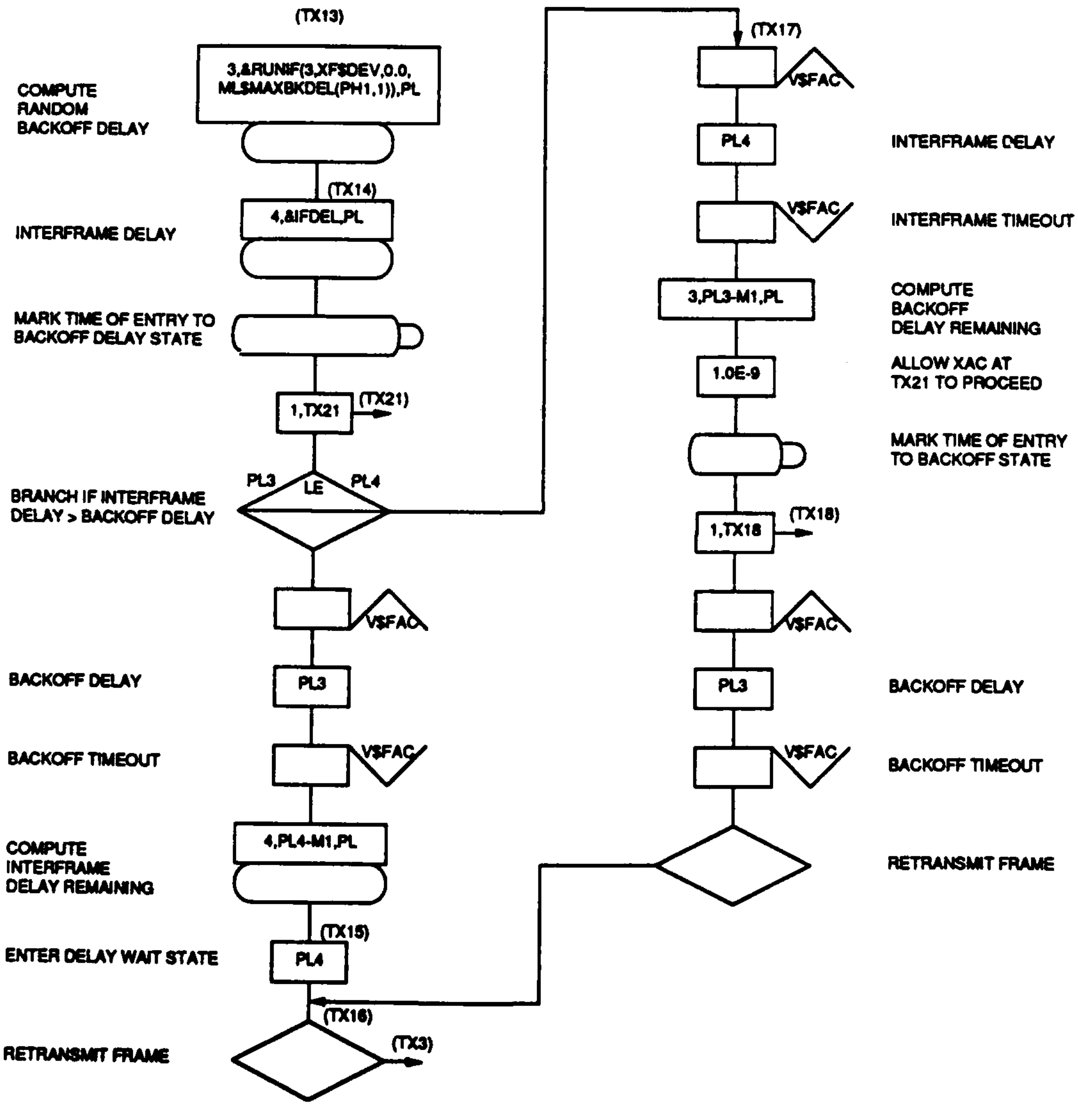

Figure 6 (Contd.)

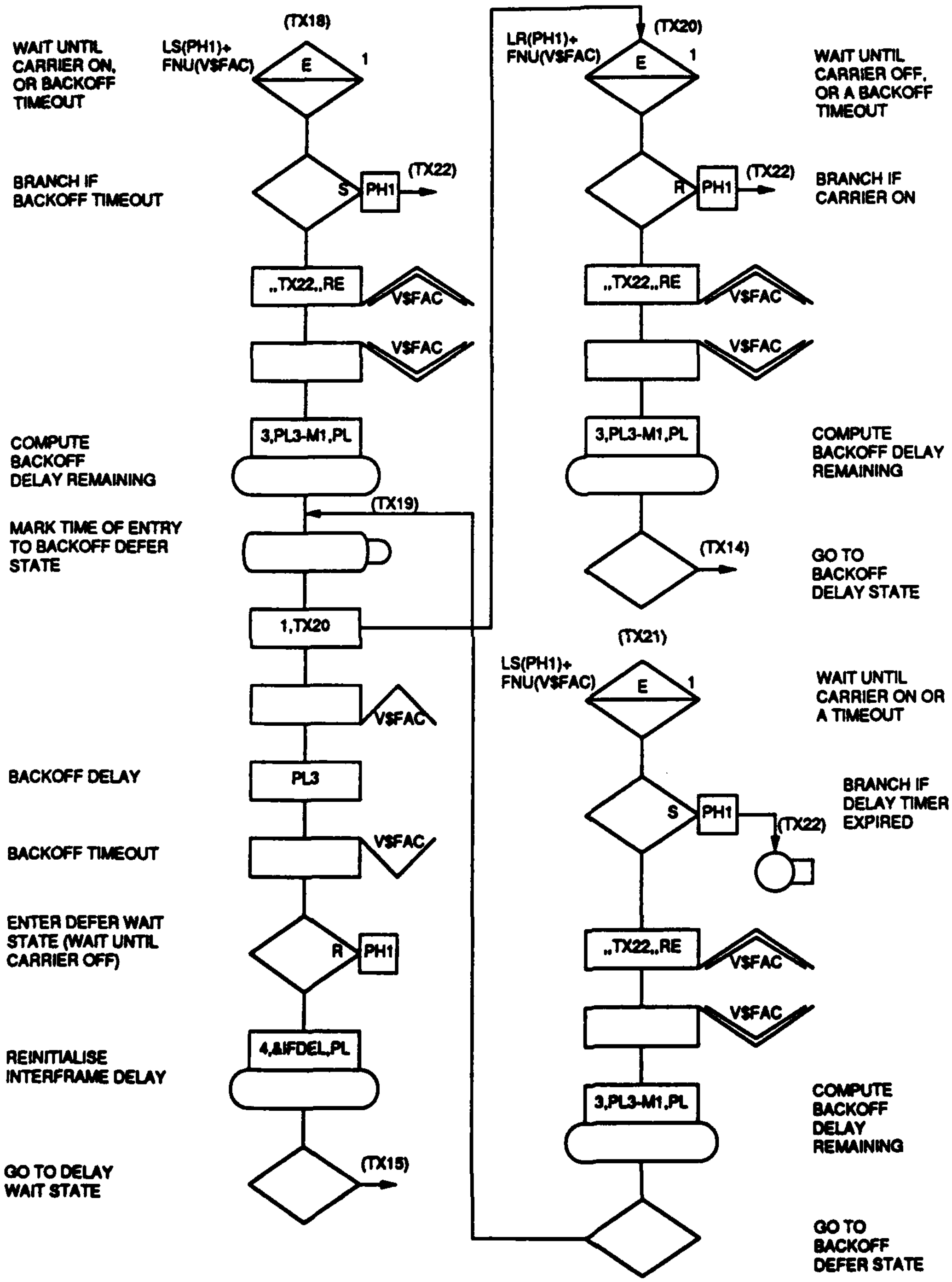

Figure 6 (Contd.)

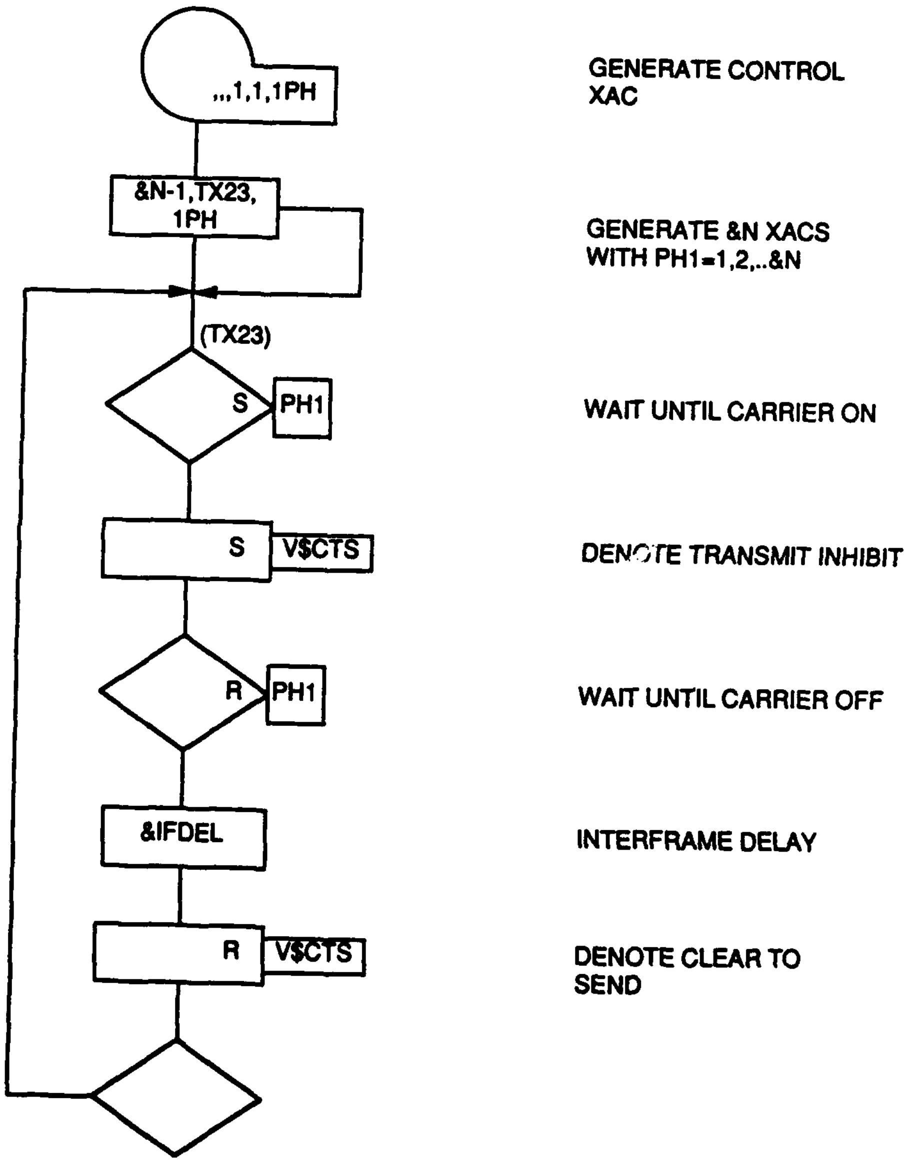

Figure 7. Ethernet: Collision sensing process

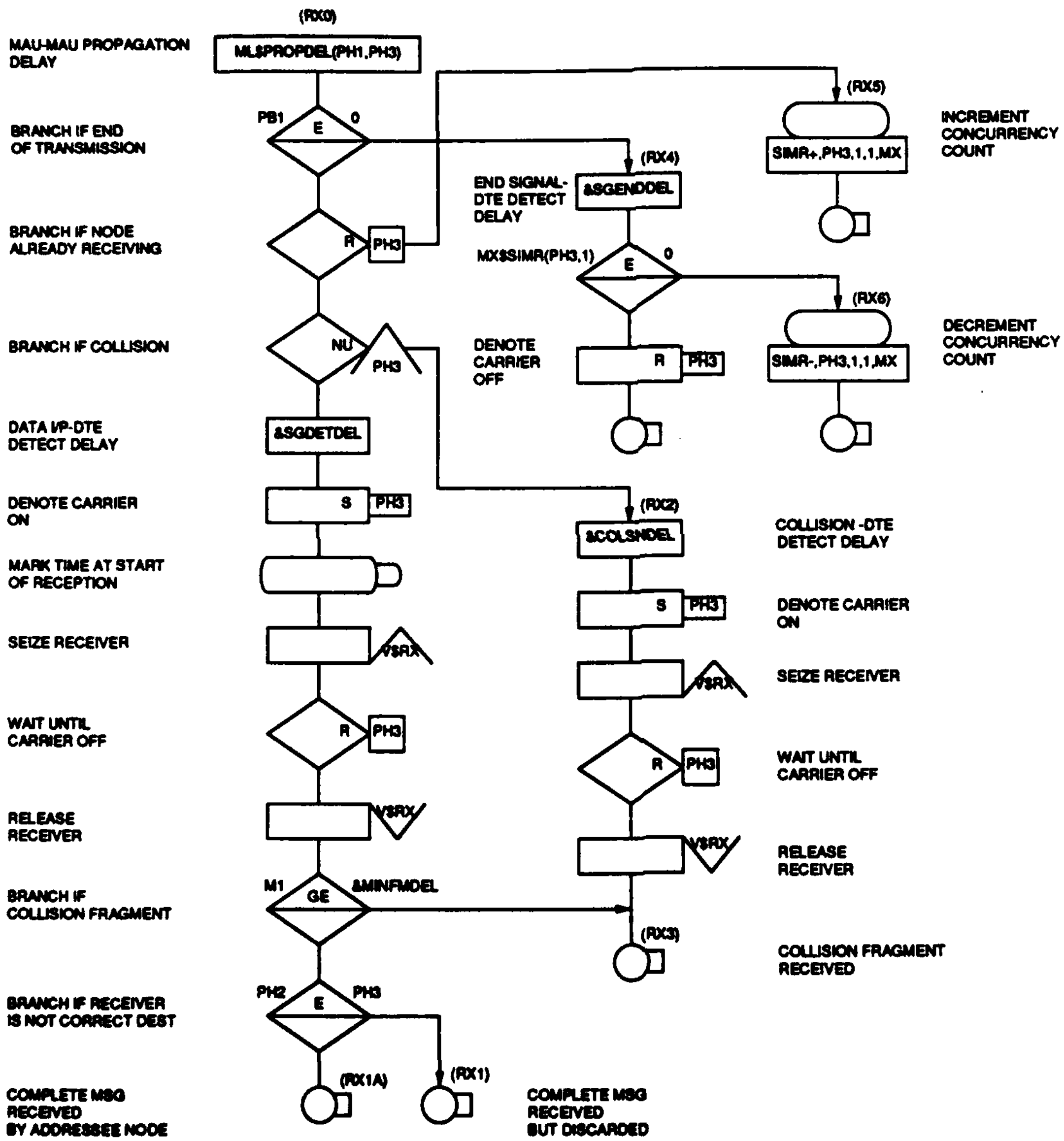

Figure 8. Ethernet: Frame reception process

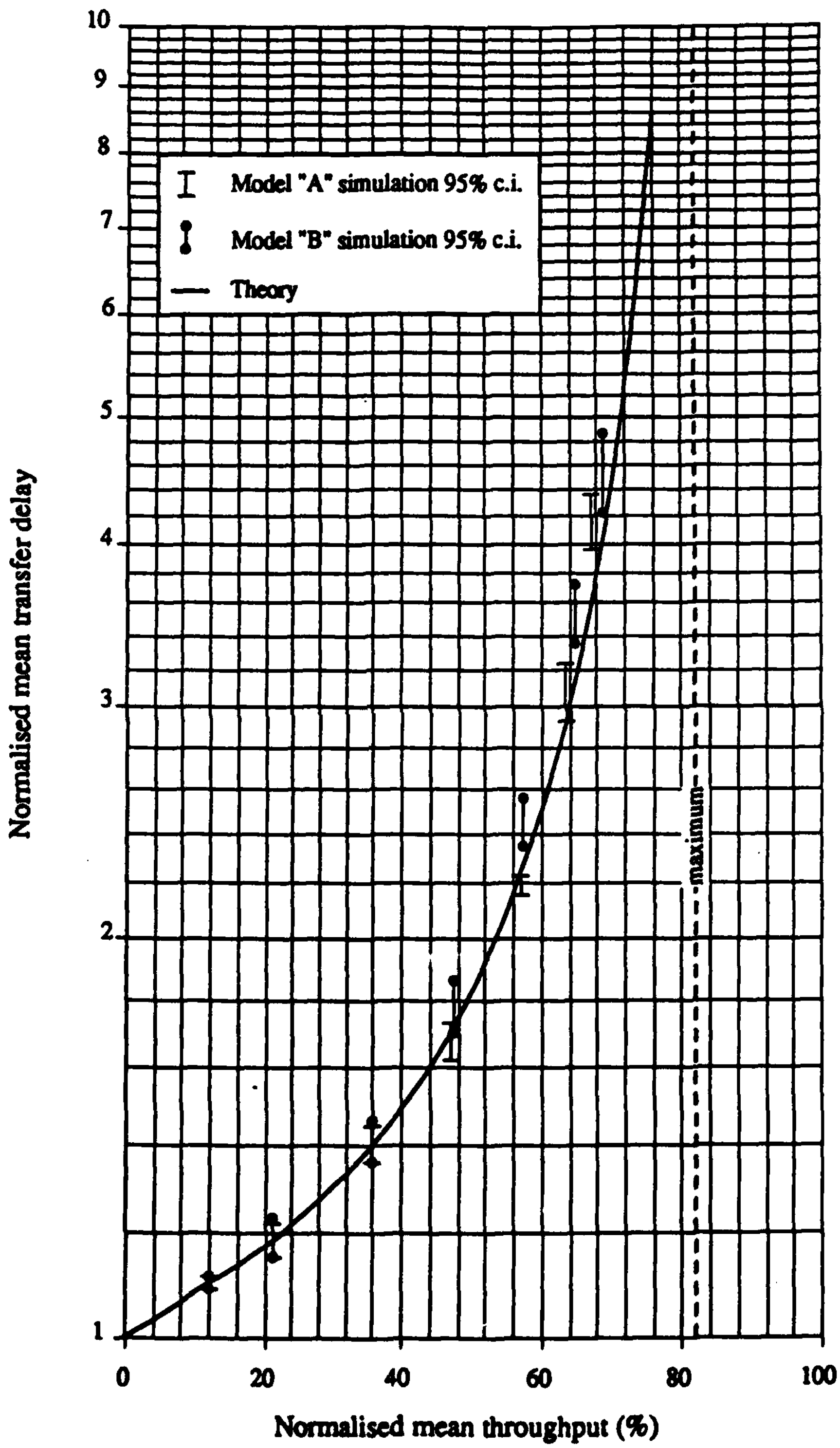

Figure 9. Ethernet: Mean transfer delay vs mean throughput

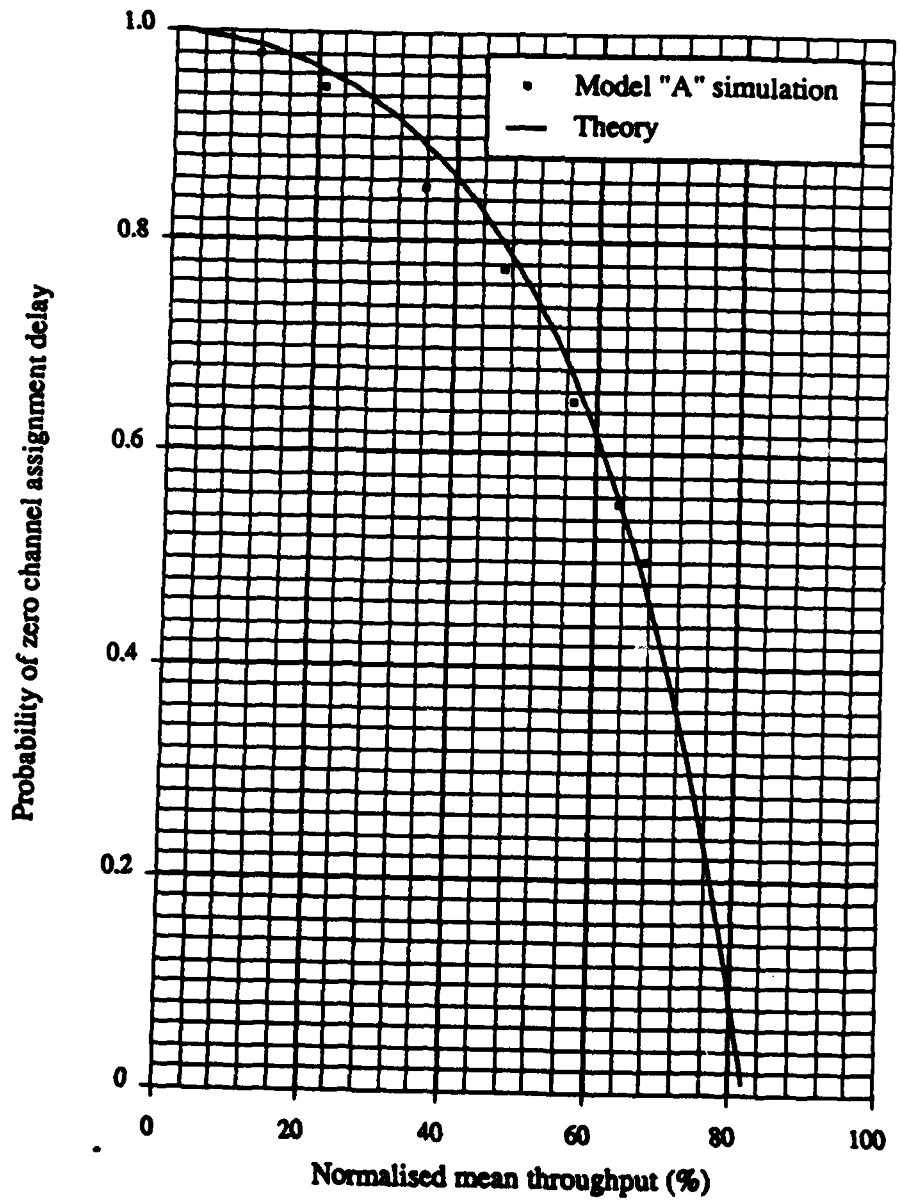

Figure 10. Ethernet: Probability of zero channel assignment delay vs mean throughput

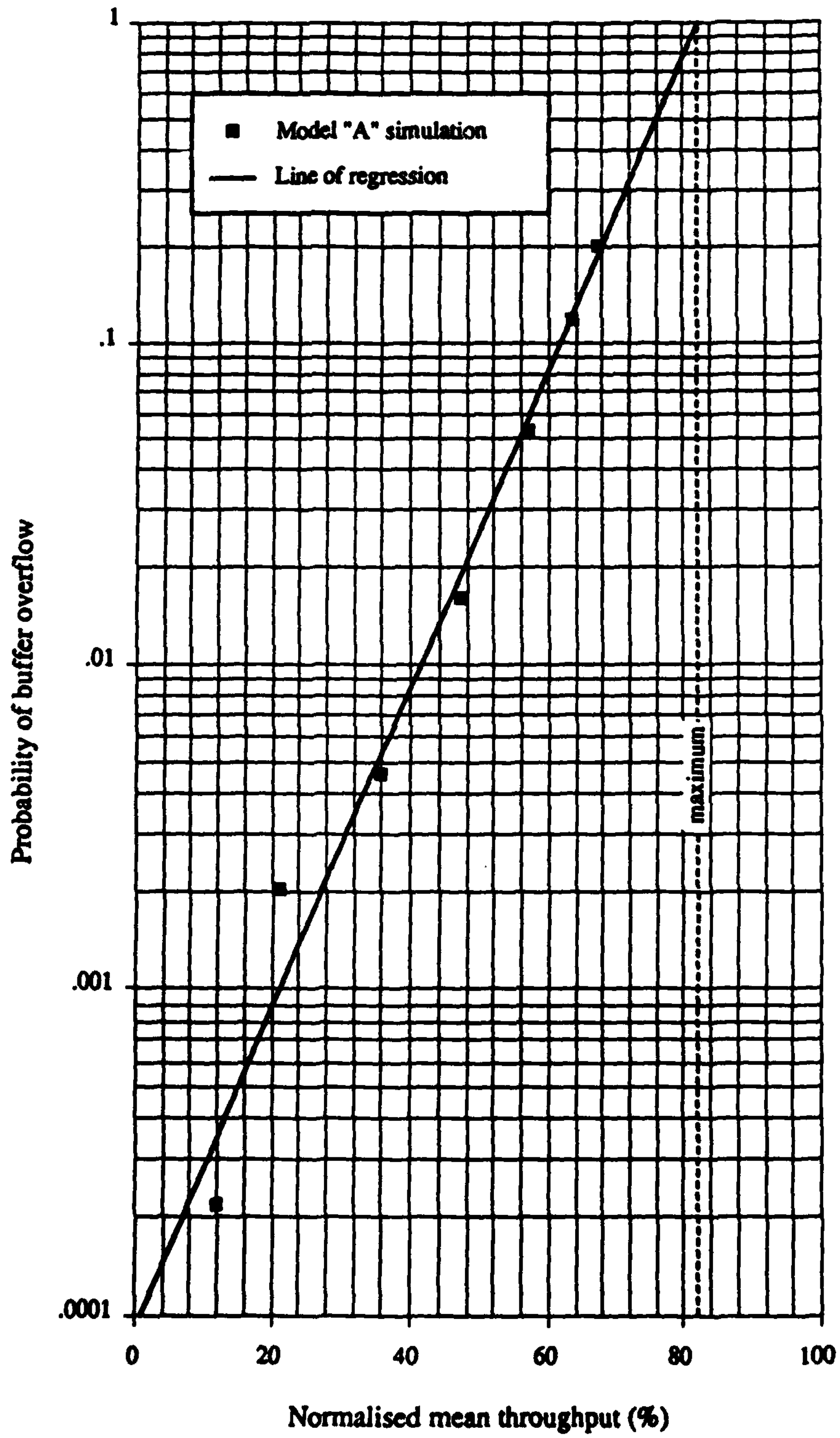

Figure 11. Ethernet: Probability of buffer overflow vs mean throughput

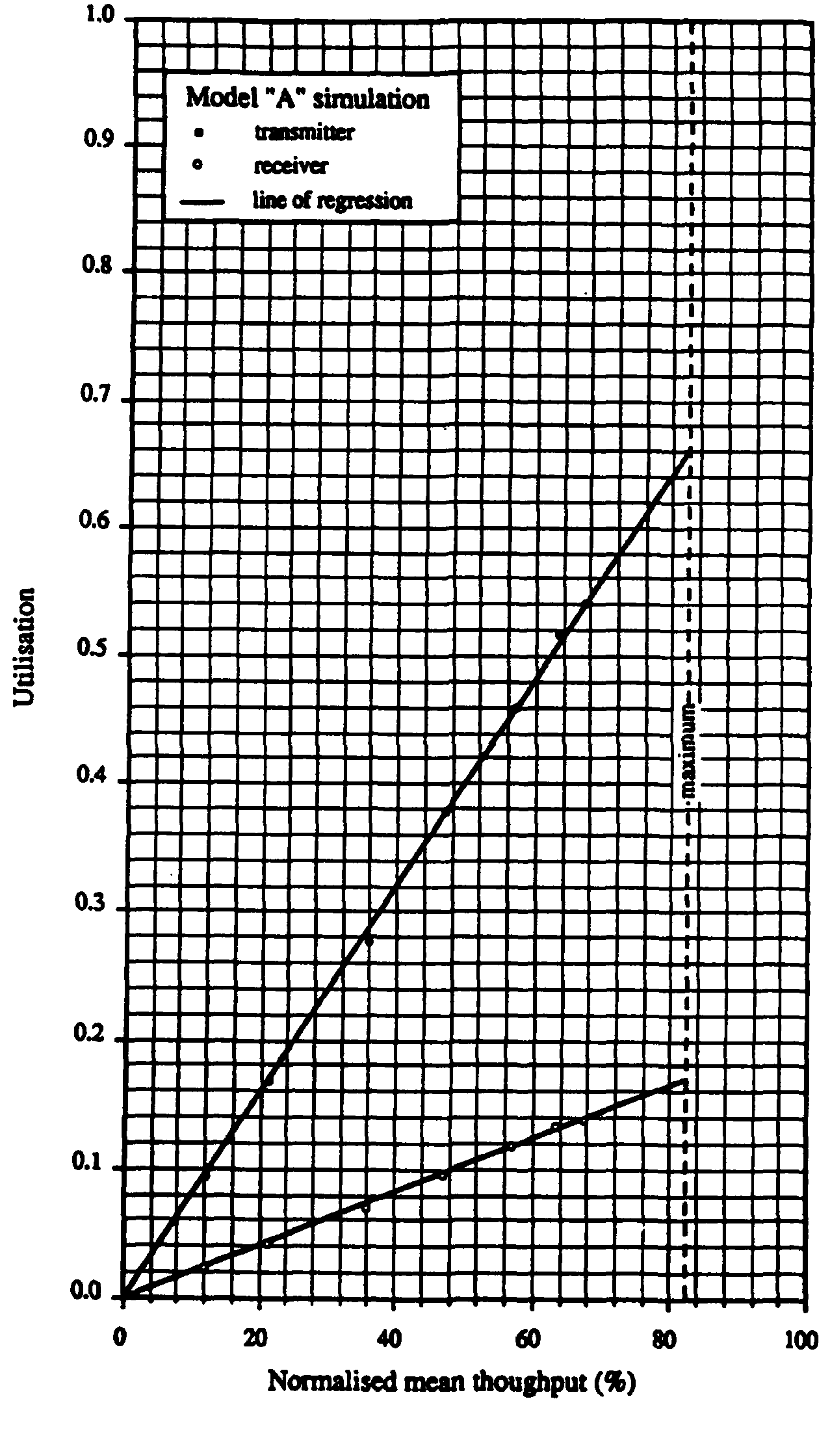

Figure 12. Ethernet: Transmitter and receiver utilisations vs mean throughput

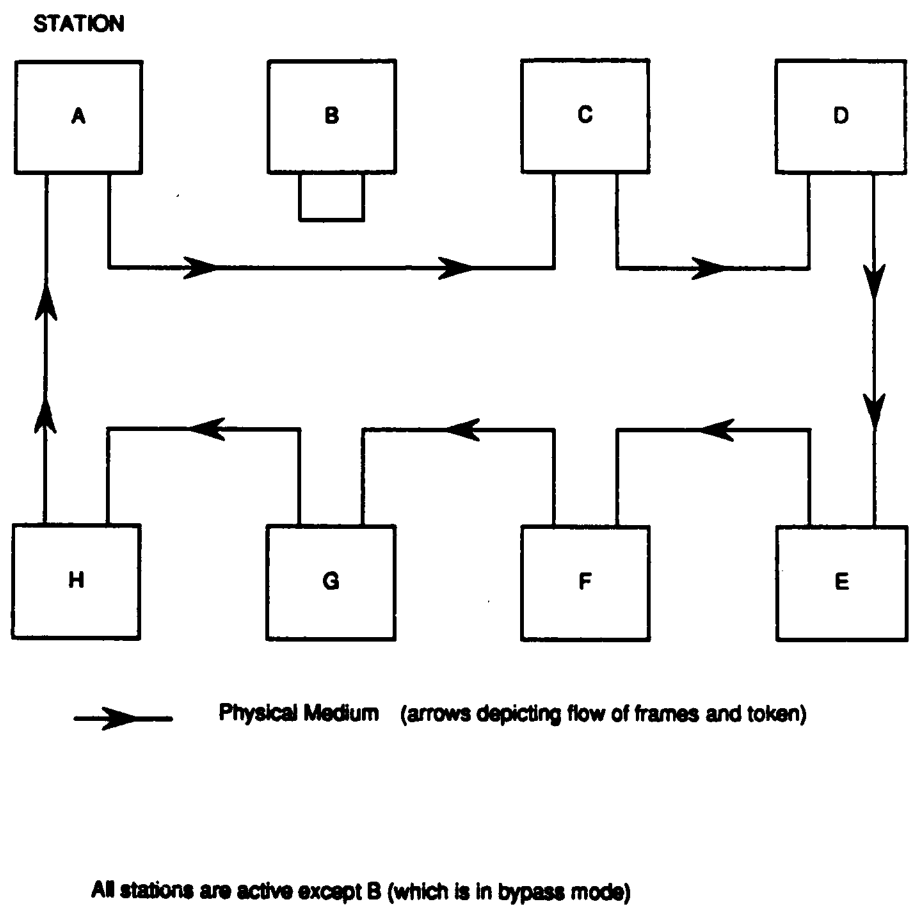

Figure 13. Topology of a Token Ring LAN

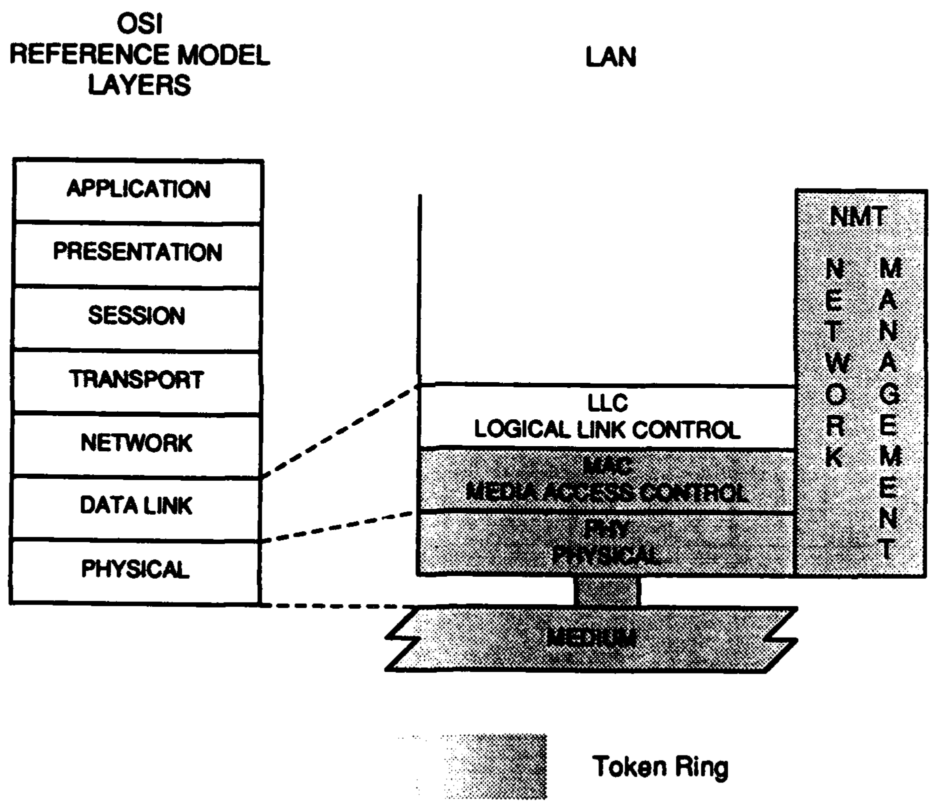

Figure 14. Relationship of Token Ring LAN to OSI Reference Model

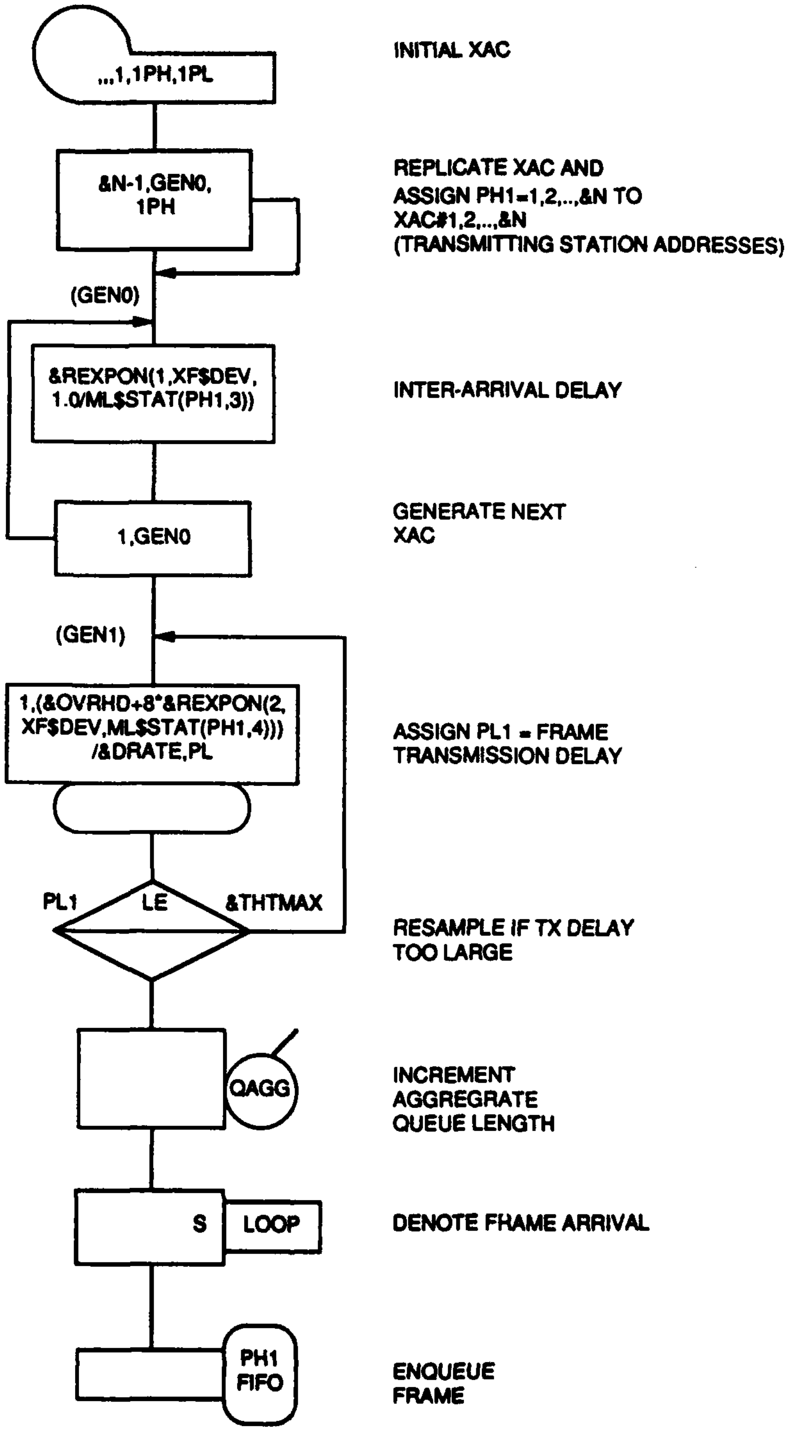

Figure 15. Token Ring: Frame generation process

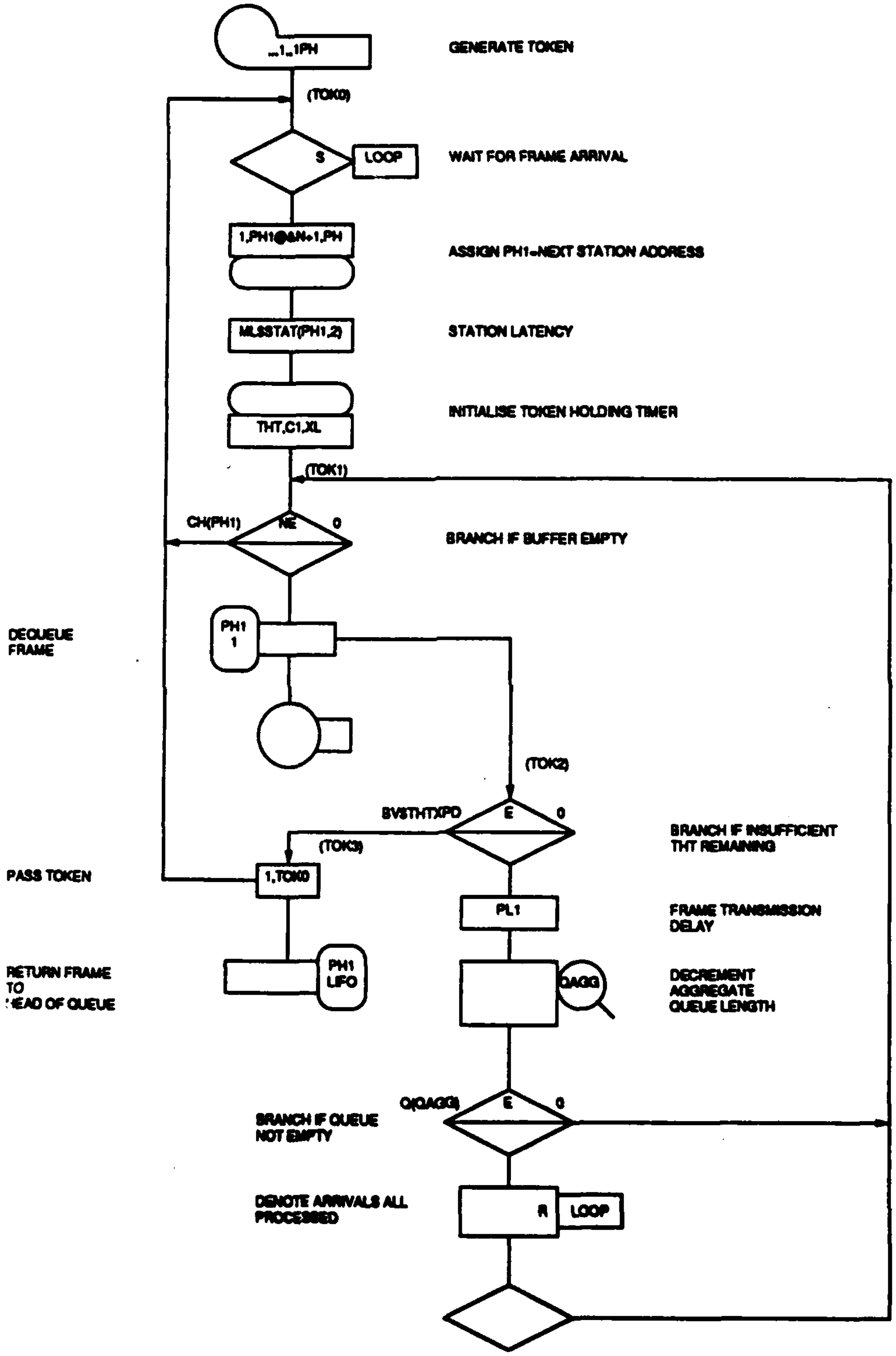

Figure 16. Token Ring: Frame delivery process

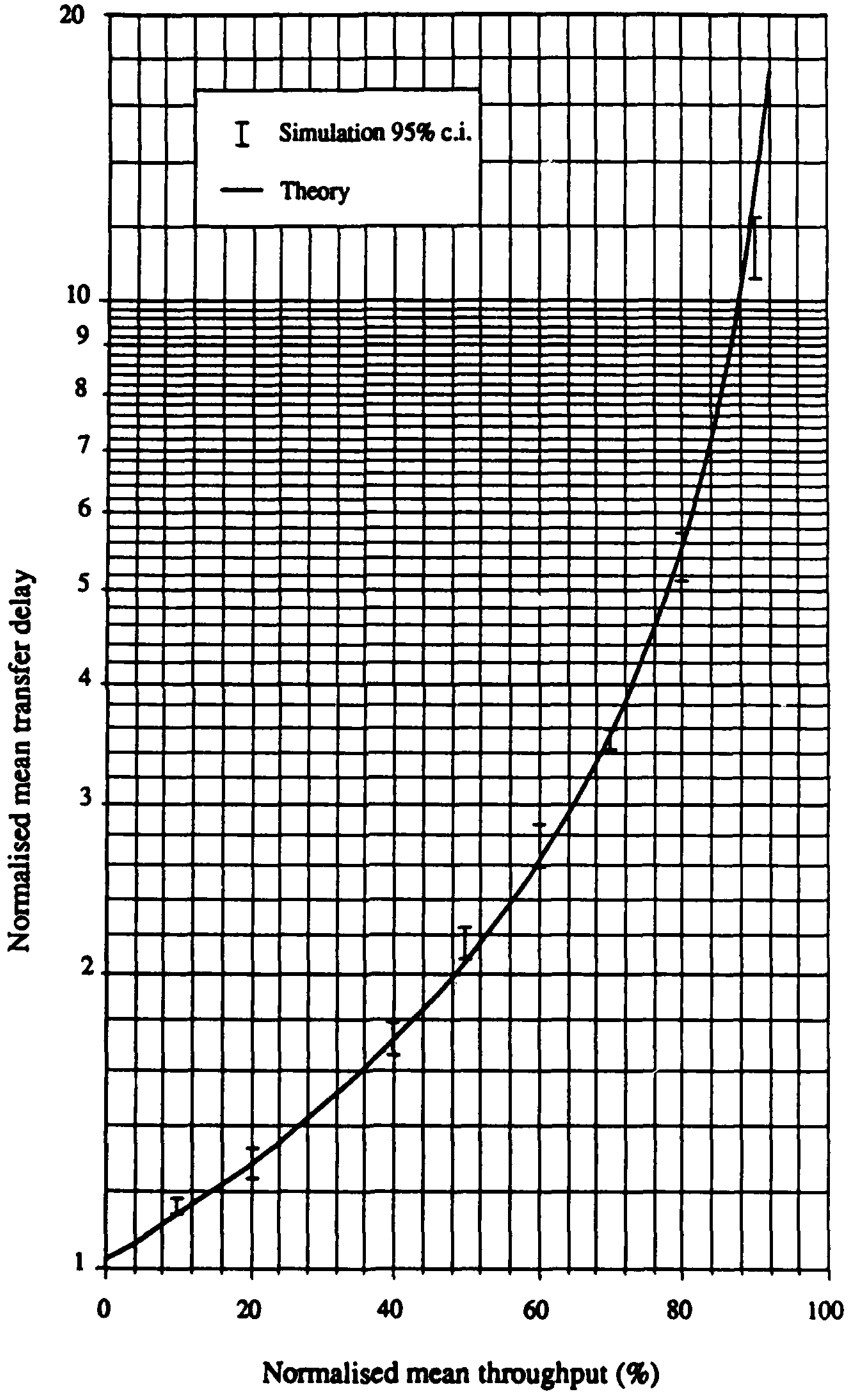

Figure 17. Token Ring: Mean transfer delay vs mean throughput

THIS IS A BLANK PAGE

APPENDIX I

THE ETHERNET PROTOCOL

The state diagram in figure I.1 illustrates the eleven operational states in the MAC transmission protocol described in the ANSI standard(ref.14).

The protocol starts up in state 0 (*start*) and the (retransmission) attempt counter is reset before state 1 (*idle*) is entered. A station remains in the *idle* state until either a frame arrives from the higher-level Logical Link Control (LLC) sublayer of the Data Link Layer for transmission on the medium, or a carrier is detected. A station that has a frame to transmit waits in the *idle* state until the medium is free before entering state 2 (*transmit*). Stations sense activity on the medium by listening for the presence of a carrier signal.

Upon entering the *transmit* state, a station transmits a fixed pattern of bits, called the preamble, at the start of each frame in order to provide time for the local oscillators in the receiver of each station in the network to lock onto the transmitted waveform. During the *transmit* state, the transmitting station continues to listen to the medium and compares bit for bit the data received with the data it is attempting to transmit. Discrepancies that occur as a result of two or more stations attempting to transmit simultaneously are termed "collisions".

Following the collision-free transmission of a frame, the attempt counter is reset, the delay timer is started, and the station enters state 8 (*delay-no-wait*). If a new frame is ready for transmission before the delay timer expires, an immediate transition occurs from the *delay-no-wait* state to state 10 (*delay-wait*) and subsequently to the *transmit* state upon expiry of the delay timer (enabling transmission of a new frame to commence). Otherwise the station returns to the *idle* state upon expiry of the delay timer. The purpose of the delay timer is to force a minimum delay (the inter-frame gap) between the transmission of consecutive frames on the medium.

The occurrence of a collision after the preamble has been transmitted causes the attempt count to be incremented and entry to state 3 (*jam*). The *jam* state is the first of several states (states 4-6, and 9) that are entered prior to a subsequent attempt to retransmit the frame (and its preamble).

Upon entering the *jam* state, a station ceases transmission of its frame and commences to transmit a random pattern of bits, called the jam, to assist other transmitting stations in the network in sensing the collision. There is no delay between transmission of the last frame bit and the first jam bit. At the end of the jamming period, the station's delay timer is started and commences counting down. Provided the number of previous retransmission attempts does not exceed the attempt limit, the backoff timer is initialised by the backoff algorithm, state 6 (*backoff-delay*) is entered, and the backoff timer commences counting down. If the attempt limit is exceeded, the *delay-no-wait* state is entered immediately, the frame is aborted (viz the responsibility for frame retransmission is passed back up to the LLC sublayer), and the station returns to the *idle* state upon expiry of the delay timer.

Three conditions can cause exit from the *backoff-delay* state. Firstly, if the backoff timer expires before the delay timer, the station immediately enters the *delay-wait* state and, upon expiry of the delay timer, enters the *transmit* state and frame retransmission commences. Secondly, if the delay timer expires before the backoff timer, state 4 (*backoff*) is entered. Thirdly, if the carrier is detected, the delay timer is stopped and state 5 (*backoff-defer*) is entered.

In the *backoff* state, the reception of a carrier causes entry to the *backoff-defer* state. Expiry of the backoff timer, on the other hand, causes transition to the *transmit* state and frame retransmission commences.

In the *backoff-defer* state, the sensing of the end of the carrier starts the delay timer and causes entry to the *backoff-delay* state. Expiry of the backoff timer, on the other hand, causes transition to state 9 (*defer-wait*). Subsequent sensing of the end of the carrier whilst a station is in the *defer-wait* state starts the delay timer and causes entry to the *delay-wait* state.

State 7 (*defer-no-wait*) is entered from the *idle* state if a station with no frame to transmit detects a carrier. The station remains in this state until either the end of carrier is sensed or a frame becomes ready to transmit. If the end of carrier is sensed, the delay timer is started and the *delay-no-wait* state is entered. If a frame is ready for transmission, the *defer-wait* state is entered.

The state diagram of the MAC operational receiving protocol is illustrated in figure I.2. It consists of only three states. The protocol starts up in state 0 (*start*) and immediately enters state 1 (*idle*). Upon detection of a carrier, a station enters state 2 (*receive*) and returns to the *idle* state at the end of reception of a frame (ie., when the end of carrier is sensed). Received frames which are correctly addressed are passed up to the LLC sublayer. On the other hand, collision fragments (viz partially transmitted frames with jams appended) are ignored. Collision fragments are recognised by the fact that their lengths are always shorter than the minimum frame length.

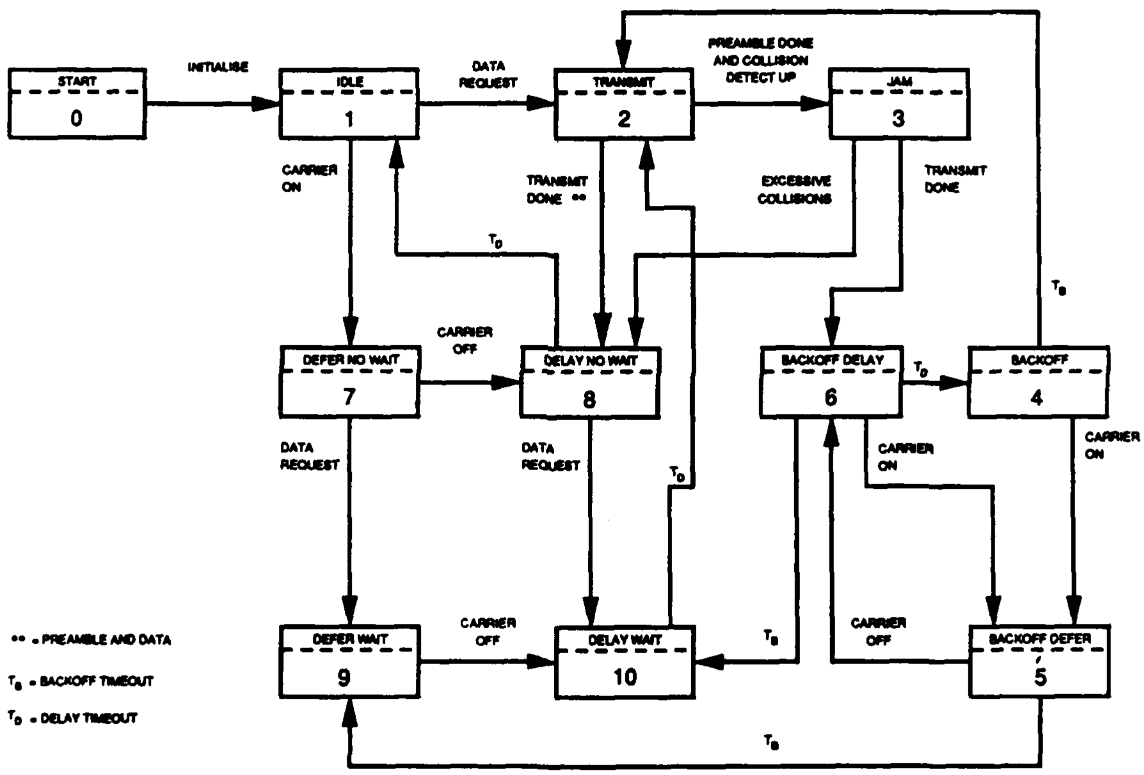

Figure I.1 State diagram of Ethernet MAC transmit protocol

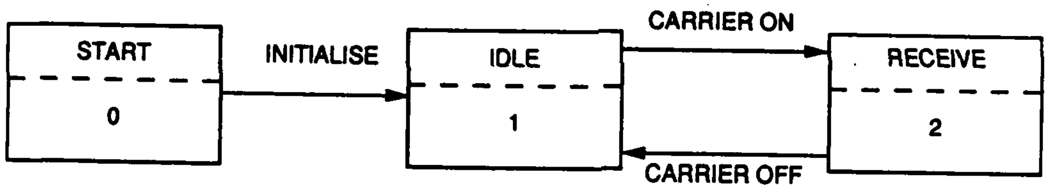

Figure I.2 State diagram of Ethernet MAC receive protocol

APPENDIX II

ANALYTIC RESULTS FOR ETHERNET PERFORMANCE

A general analytic formula for the mean frame transfer delay is given by Bux(ref.2) who modified Lam's formula(ref.23) slightly to extend its application to unslotted operation.

The normalised mean transfer delay, E[D], is the mean ratio of the frame transfer delay (defined in Section 2.1.1) to the frame transmission time in the absence of collisions. The transfer delay is dependent on the throughput, the rate at which complete frames are conveyed over the transmission medium. The normalised mean throughput, E[X], is the ratio of the mean throughput to the maximum mean throughput, expressed as a percentage. Under steady-state conditions, the mean throughput is equal to the mean aggregate frame generation rate $\lambda=\lambda_1+\lambda_2+\ldots$, which is the sum of the mean generation rates of frames in the LLC sublayer of each station, provided all of the frames are transmitted.

E[D] and E[X] are given by:

$$E[D] = t_f/E[T_p]$$

$$E[X] = 100\lambda(E[T_p] + T_g)$$

where t_f is the mean frame transfer delay, $E[T_p]$ is the mean frame transmission delay, and T_g is the inter-frame delay. If the frame generation rates at each station are Poisson distributed, the aggregate (or compound) generation rate is also Poisson distributed and t_f is given by Bux's formula:

$$t_f = \frac{\lambda\{E[T_p^2] + (4e+2)\tau E[T_p] + 5\tau^2 + 4e(2e-1)\tau^2\}}{2\{1-\lambda(E[T_p] + \tau + 2e\tau)\}} + E[T_p]$$

$$+ 2\tau e - \frac{\{1-\exp(-2\lambda\tau)\}\{2/\lambda + 2\tau/e - 6\tau\}}{2\{F_p^*(\lambda)\exp(-\lambda\tau-1) -1 + \exp(-2\lambda\tau)\}} + \tau/2$$

where τ is the maximum transmission propagation delay between any two stations in the network, $F_p^*(\lambda)$ is the Laplace transform of the probability density function of the frame transmission delay $T_p=(L_p+L_h)/\omega$, L_p is the frame length, L_h is the (fixed) frame overhead, ω is the transmission data rate, and $e \simeq 2.718$.

The general expresssion for the probability of zero delay in gaining access to the medium (viz the "zero channel assignment delay" defined by Lam) is:

$$P_a(\lambda) = \frac{1-\lambda(E[T_p] + \tau + 2\tau e)}{2\lambda\tau\{F_p^*(\lambda)\exp(-\lambda\tau)/\{1 - \exp(-2\lambda\tau)\} - e\}}$$

The expressions above only apply over the (equilibrium) range of mean aggregate frame generation rates $0\le\lambda<\Lambda$, where $\Lambda=1/(E[T_p]+\tau+2e\tau)$. If this condition is satisfied, individual frame transfer delays are bounded. The maximum mean throughput depends upon three factors: the mean frame length, $(E[L_p]+L_h)$, the physical dimensions of the network (which govern τ), and the bandwidth of the transmission medium (ω). In practice, propagation delays can only be ignored in highly localised networks or when mean frame lengths are large.

For the constant frame lengths assumed in the simulation experiments, the first two moments and the Laplace transform of the frame transmission delay distribution are given by:

$$E[T_p] = (L_p+L_h)/\omega,$$

$$E[T_p^2] = E^2[T_p]$$

and $$F_p^*(\lambda) = \exp(-\lambda[L_p+L_h]/\omega).$$

The maximum value of the normalised mean throughput is given by:

$$E[X]_{max} = \frac{100(L_p + L_h + \omega T_g)}{L_p + L_h + (1 + 2e)\tau\omega}$$

In the previous analysis it was assumed that all of the frames generated by the LLC sublayer are accepted for transmission by the MAC sublayer. This implies the presence of a buffer which is sufficiently large to store frames generated whilst transmissions are in progress. However, in practice, each station incorporates only a unit-capacity MAC buffer, so that there is a finite probability that the buffer will be full (viz "overflow") at the time a frame is offered to the MAC sublayer. In this case, the frame is either rescheduled by the LLC sublayer for later transmission, or aborted. The normalised mean throughput is related to the mean aggregate frame generation rate by:

$$E[X] = 100\lambda(E[T_p] + T_g).(1-P_b(\lambda))$$

No closed-form expression exists for the buffer overflow probability, $P_b(\lambda)$, but numerical solutions may be obtained by iteratively solving the Markov balance equations(ref.5).

APPENDIX III

THE TOKEN RING PROTOCOL

The state diagram of the Token Ring (ANSI 802.5) MAC protocol, illustrated in Figure III.1, depicts four operational states.

The operation of the protocol will be presented in a simpler form compared to the standard(ref.13). Only those states and transitions incorporated in simulation model are discussed. Hence, state 4 (*transmit zeros and modify stacks*) and state 5 (*transmit fill and strip start-of-frame sequence*), which are only entered when multiple levels of frame priority are employed, are not described. For the same reason, some transitions are not described. Transitions which occur due to changes of priority level are not considered. Neither are the *repeat* state "bit-flipping loop", or the transitions that occur upon expiry of the Return-to-Repeat-State (TRR) timer, considered, because the protocol is assumed to operate without errors.

Stations attached to the physical medium (a ring) can only transmit when they receive a permission token. A station resides in state 0 (*repeat*) until it receives a token and it has one or more frames queued and ready for transmission. In the *repeat* state, a station always re-transmits all of the information it receives. (It may change some of the bits, but this is not relevant to the the model.) If the destination address matches the station address, the information is also copied to a buffer for removal by the LLC sublayer. This action corresponds to the data receiving process. When a station in the *repeat* state having no ready frames receives a token, it retransmits the token without modification. Note that only one token can be on the ring at any time and that frames (and the token) circulate around the ring in only one and the same direction.

A station enters state 1 (*transmit-frames*) when it receives a token and it has one or more frames ready to transmit. It resets its token holding timer (THT), which commences to count down, and transmits the first frame. If after the first frame has been transmitted there are more frames to send, the amount of time required to transmit the second frame is computed and the THT is interrogated. Provided there is sufficient time left to transmit the frame before the THT expires, the first bit of the second frame is transmitted without delay after the last bit of the first frame. This process repeats until either the THT expires or there are no further frames in the buffer. At this stage state 2 (*transmit-fill*) is entered.

Whilst in the *transmit-fill* state, a station transmits an indeterminate pattern of bits (or "fill") until it receives and recognises its own address in the source address field of the last frame it has transmitted. The station then enters state 3 (*strip-frame*) and transmits a token to the down-stream station, followed by more fill, until it receives and recognises the end of the last frame it has transmitted. It then re-enters the *repeat* state.

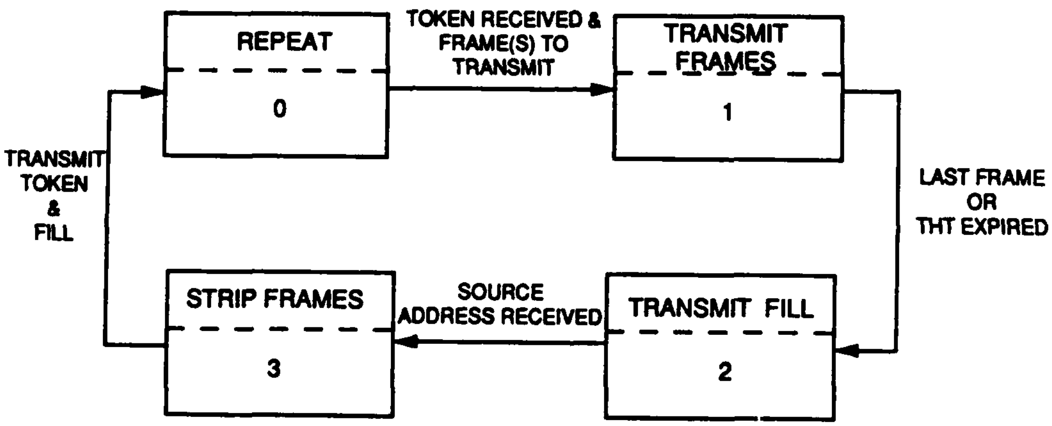

Figure III.1 State diagram of Token Ring MAC protocol

APPENDIX IV

ANALYTIC RESULTS FOR TOKEN RING PERFORMANCE

A general analytic formula for the mean frame transfer delay is given by Bux(ref.2) who modified Konheim and Meister's discrete-time formula(ref.23) to obtain a continuous-time approximation.

Under steady-state conditions the mean throughput is equal to the mean aggregate frame generation rate, λ, and the normalised mean throughput is given by

$$E[X] = 100\lambda E[T_p].$$

Expressions for the the mean frame transmission delay, $E[T_p]$, and the normalised mean transfer delay, $E[D]$, appear in Appendix II.

Analytic results exist only for the restricted case in which all stations generate the same amount of traffic, viz the frame generation rate distributions are identical. A simplified solution is obtained by assuming that all of the frames generated by the LLC sublayer are accepted for transmission by the MAC sublayer. This implies that the buffer in the MAC sublayer of each station is large enough to store all of the frames prior to transmission. The normalised mean throughput can achieve a maximum of 100%.

Bux's simplified solution for the mean transfer delay, t_f, additionally assumes Poisson distributed frame generation rates and equal station latencies:

$$t_f = \frac{\lambda E[T_p]E[T_p^2]}{2(1-\lambda E[T_p])E[T_p]} + E[T_p] + \frac{\gamma(1-\lambda E[T_p]/N)}{2(1-\lambda E[T_p])} + \tau/2$$

where N is the number of stations, γ is the station latency ($\gamma=\tau+q$, where q is the minimum token holding time) and the other symbols have the same meanings as in Appendix II. This formula only applies over the equilibrium range of frame generation rates $0\le\lambda<1/E[T_p]$. Within this range, individual frame transfer delays are bounded.

In the simulation experiment the values of L_p are assumed to be exponentially distributed with a mean of $E[L_p]$, hence the first two moments of the frame transmission delay distribution are given by:

$$E[T_p] = (E[L_p] + L_h)/\omega, \text{ and}$$
$$E[T_p^2] = E^2[T_p] + E^2[L_p]/\omega^2.$$

DISTRIBUTION

	Number of copies
Defence Science and Technology Organisation	
Chief Defence Scientist	)
First Assistant Secretary, Science Policy	) 1
Director General Science and Technology Programs	)
Counsellor Defence Science London	Cnt Sht Only
Counsellor Defence Science Washington	Cnt Sht Only
DSR - Bangkok	Cnt Sht Only
SA to DRC - Kuala Lumpur	Cnt Sht Only
Weapons Systems Research Laboratory	
Director	1
Chief, Combat Systems Division	1
Head, Combat Systems Integration	1
Head, Combat Systems Effectiveness	1
J.G. Schapel, Combat Systems Integration	1
S.J. Miller, Combat Systems Integration	1
M.L. Scholz, Combat Systems Integration	3
Electronics Research Laboratory	
Director	1
Head, Switching Signalling and Networks	1
Manager, Information Systems Branch	1
Head, Networks	1
Libraries and Information Services	
Defence Central Library, Campbell Park	1
Document Exchange Centre, Defence Information Services and Science Liaison Branch for:	
Microfilming	1
National Library of Australia	1
United Kingdom, Defence Research Information Centre	2
Canada, Director Scientific Information Services	1

New Zealand, Ministry of Defence	1
United States, Defense Technical Information Center	12
Main Library, Defence Science and Technology Organisation, Salisbury	2
Library, DSD, Melbourne	1
Library, DSTO, Sydney	1
Library, Aeronautical Research Laboratory	1
Library, Materials Research Laboratory	1
Australian Defence Force Academy Library	1
Department of Defence	
Director of Departmental Publications	1
Joint Intelligence Organisation (DSTI)	1
Navy Office	
Navy Scientific Adviser	Cnt Sht Only
Director, Naval Combat Systems Engineering	1
Director, Submarine Warfare Systems Centre	Cnt Sht Only
Deputy Director, Development, SWSC	1
Other Departments and Organisations	
SA Institute of Technology	
Head, School of Mathematics and Computer Studies	1
Dr D.M. Panton, Maths and Computer Studies	1
Director, Digital Communications Group	1
Dr J. Asenstorfer, Digital Communications Group	1
Director, Teletraffic Research Centre, Univ of Adelaide	1
Manager, LAN Products Group, AWA Defence and Aerospace, North Ryde NSW	1
UNITED KINGDOM	
British Library, Document Supply Centre (UK)	1
Institution of Electrical Engineers (UK)	1
Spares	3
Total number of copies	55

DOCUMENT CONTROL DATA SHEET

Security classification of this page : UNCLASSIFIED

1 DOCUMENT NUMBERS

AR Number :	AR-005-937
Series Number :	WSRL-TR-45/89
Other Numbers :	

2 SECURITY CLASSIFICATION

a. Complete Document :	Unclassified
b. Title in Isolation :	Unclassified
c. Summary in Isolation :	Unclassified

3 DOWNGRADING / DELIMITING INSTRUCTIONS

4 TITLE

SIMULATING LOCAL AREA NETWORK PROTOCOLS WITH THE GENERAL PURPOSE SIMULATION SYSTEM (GPSS)

5 PERSONAL AUTHOR (S)

M.L. Scholz

6 DOCUMENT DATE

March 1990

7

7.1 TOTAL NUMBER OF PAGES	59
7.2 NUMBER OF REFERENCES	37

8 8.1 CORPORATE AUTHOR (S)

Weapons Systems Research Laboratory

8.2 DOCUMENT SERIES and NUMBER
Technical Report
45/89

9 REFERENCE NUMBERS

a. Task :	NAV 87/226.2
b. Sponsoring Agency :	NAVY

10 COST CODE

779235

11 IMPRINT (Publishing organisation)

Defence Science and Technology Organisation

12 COMPUTER PROGRAM (S)
(Title (s) and language (s))

13 RELEASE LIMITATIONS (of the document)

Approved for Public Release

Security classification of this page : UNCLASSIFIED

Security classification of this page : UNCLASSIFIED

14 ANNOUNCEMENT LIMITATIONS (of the information on these pages)

No limitation

15 DESCRIPTORS

a. EJC Thesaurus Terms	GPSS (programming language) Simulation languages Local area networks
b. Non - Thesaurus Terms	Ethernet protocol (computers) Token Ring protocol (computers)

16 COSATI CODES

1207

17 SUMMARY OR ABSTRACT
(if this is security classified, the announcement of this report will be similarly classified)

(U) Simulation models of ANSI 802.3 (Ethernet) and ANSI 802.5 (Token Ring) transmission control protocols have been constructed to assess the usefulness of the General Purpose Simulation System (GPSS) simulation language for modelling the performance of local area networks. The models are described in detail and the data obtained from them are compared with analytic results. Model verification and other problems associated with analysing simulation output are discussed.

Security classification of this page : UNCLASSIFIED