Medium Access Control Schemes for Local Area Networks with Multiple Priority Function

M. GOPALAKRISHNAN*† AND L. M. PATNAIK‡

*Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, U.S.A.

‡ Department of Computer Sciences and Automation, Indian Institute of Science, Bangalore 560 012, India.

This paper deals with the development and performance evaluation of three modified versions of a scheme proposed for medium access control in local area networks. The original scheme implements a collision-free and fair medium arbitration by using a control wire in conjunction with a data bus. The modifications suggested in this paper are intended to realize the multiple priority function in local area networks.

Received May 1986, revised January 1987

1. INTRODUCTION

Local Area Networks (LAN) spanning a limited geographic area have emerged of late as a field of wide applicability. Such networks are intended to interconnect medium to high-power computers, dumb terminals, real-time data equipment and several shared resources. One of the critical issues associated with the design of an LAN is the formulation of an effective scheme for the control of access to the transmission medium. This function, commonly referred to as Medium Access Control (MAC), forms a sublayer of the Data Link communication protocol. Several MAC schemes such as the Ethernet,[1] BRAM,[2] scheme A_2,[3] etc. have been proposed in the past few years. The scheme A_2 proposed by Eswaran *et al.*[3] is attractive as it provides a simple, collision-free and fair medium arbitration by using a control wire in addition to a data bus. In view of this fact, in this paper scheme A_2 has been chosen as the basis of three new MAC protocols, two of which feature multiple packet priorities.

For the efficient utilisation of the bandwidth of the medium, it is desirable to multiplex traffic from different applications on an LAN. Such traffic typically includes packets carrying digitised voice, interactive data packets and file transfer packets carrying bulk data. However, depending on the application, the packets to be transported are to be handled subject to different time constraints.[5] It is necessary for certain types of packets to be delivered at the destination faster than the others. For instance, the end-to-end delay of voice packets in an integrated voice–data network has to be kept within tight bounds to preserve the interactive quality of speech.[7] As another example, short interactive data packets should be transmitted earlier than the bulk data packets which often contain several hundreds of bytes of information. In order to cater for the above situation, it is desirable to incorporate a multiple-packet priority function into the MAC scheme. The requirements to be met by multiple priority control strategies have been discussed rigorously in Refs 5 and 6. Tobagi's Reservation Algorith,[8] Priority Ethernet,[9] Priority Ethernet with Reassignment,[10] Mark's Proposal,[11] etc. are a few of the methods propounded to implement priority scheduling in LANs.

† To whom correspondence should be addressed.

This paper deals with three modified versions of algorithm A_2, two of which feature the multiple priority function. The pertinent details of the original scheme A_2 are provided in Section 2. Scheme A_2', presented in Section 3, overcomes certain drawbacks of A_2 and lays the foundation for the development of the priority MAC schemes presented in this paper. The various aspects of priority scheduling in LANs are discussed in Section 4. The MAC algorithm PA 1 incorporates the multiple priority function by the principle of staggered-round initiations as explained in Section 5. The delay–throughput performance of scheme PA 1 has been studied by simulation on a DEC 1090 system, using SIMULA. The choice of simulation as the tool for performance evaluation was motivated by the complexity of the problem at hand, which made it impracticable to construct an acceptably accurate analytical model. In order to assess the functionality of PA 1 in a representative application, the operation of this scheme has also been simulated in a mixed voice–data environment. The details of the simulation scenario and the results of the performance evaluation are provided in Section 5, where it is shown that the priority function of scheme PA 1 fails under heavy network loading conditions. Scheme PA 2, presented in Section 6, is basically a more sophisticated version of PA 1 in that it overcomes its shortcomings and approaches the perfection of the ideal priority discipline more closely. However, scheme PA 2 entails additional network overheads and leads to increased complexity of implementation.

2. PREVIOUS WORK

The network structure proposed by Eswaran *et al.* on which the original arbitration algorithm A_2[3] as well as the new schemes presented in this paper may be implemented, is depicted in Fig. 1 (reproduced from Ref. 3 with permission). It consists of a number N of ports (or nodes) referred to as port 1, port 2,..., port N. These ports are communication front ends of the associated data equipment. The key element of the network is a unidirectional control signal path, distinct from the data bus. Associated with port J ($\forall J \varepsilon \{1, 2, \dots, N\}$) is a flip-flop denoted by $S(J)$ and called the SEND flip-flop. The signal $P(J)$ called the OR signal, tapped at the control wire input to port J, is the inclusive OR of the SEND

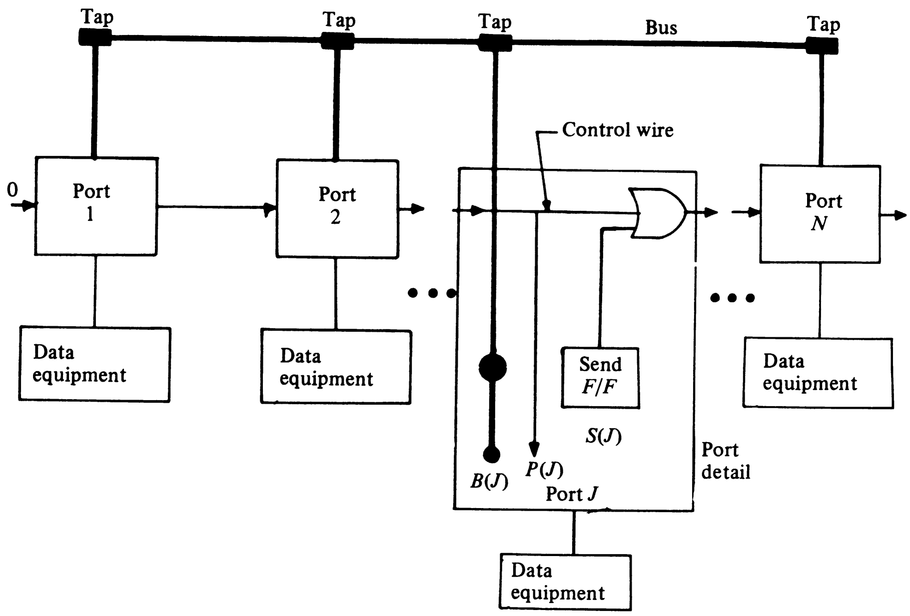


Fig. 1. Network structure (copyright © 1981 IEEE).

flip-flops of all ports to the left of port J (ports with indices less than J). Each port has the capability of detecting whether the bus is busy or not. T and R represent respectively the maximum data-bus propagation delay and control-wire delay between port 1 and port N.

As has been shown in Ref. 3, control scheme A_2 implements a collision-free and fair medium arbitration by scheduling packet transmissions in successive bus-busy periods (or rounds). However, from the arguments presented in Ref. 3, it is clear that scheme A_2 has a transient behaviour in certain situations. Consider the case whereby packets at a set of ports are awaiting the completion of the ongoing round in order to obtain bus access in the next round. If packets become available for transmission at few of the remaining ports in the brief interval at the end of the current round, it can lead to a transient condition. Under these circumstances it is possible for the bus to be forced idle for periods of the order of $N*(R+T)$ time units in the worst case. Apart from the wastage of the bus-bandwidth caused, this phenomenon also renders algorithm A_2 unsuitable for the development of a prioritised access control scheme by the methods presented in this paper. The above drawback of algorithm A_2 has been alleviated in the modified algorithm A_3 put forward by Fratta,[4] by the inclusion of a separate *COLDSTART* procedure.

3. ALGORITHM A_2': A MODIFIED VERSION OF ALGORITHM A_2

We present in this Section the MAC scheme A_2', which is a modified version of the original algorithm A_2 and which overcomes the drawback of A_2 described in the previous section. Algorithm A_2' lays the foundation for the multiple priority arbitration schemes (PA 1 and PA 2) proposed in this paper. Even though the algorithm A_2' resembles scheme A_3[4] in many respects, it does not require a separate *COLDSTART* procedure and hence is simpler. Moreover, scheme A_2' is more suitable than scheme A_2 for the realisation of priority scheduling by the methods proposed in this paper.

In addition to the details of the network structure provided in Section 2, a few other notations and their meanings need to be explained. Each port J connected to the network holds a binary variable denoted by *IDLEDET* (J, α). For any $\alpha > 0$, *IDLEDET* (J, α) is true at any time instant t if and only if port J has observed the data bus to be idle throughout the interval $(t-\alpha, t)$. The purpose of such a mechanism is to monitor the state of the data bus continuously. This enables the ports to transmit, with minimum latency, packets arriving during network-idle conditions. *TIMEOUT* (J, β) is a timer available at port J which expires β times units (for all $\beta > 0$) after its initiation. The switching delay at a network port is denoted by the parameter t_d. Delay t_d is intended to represent either the time taken by a port to drive the data bus (after the port has been granted bus access) or the time required by a port to detect a collision (Section 6). *PILOT* is an unmodulated signal transmitted on the data bus by the connected ports, for synchronisation purposes. An FCFS service discipline is maintained at each port. If a packet ('pkt') arrives at port J when the queue is empty, then 'pkt' is said to have become available to port J for transmission at the arrival instant. Otherwise, 'pkt' becomes available to port J when the transmission of the packet ahead of 'pkt' in the queue is complete. This notion of packet availability is in conformity with the one provided in Ref. 3.

3.1 Algorithm A_2'

The steps in algorithm A_2' (as executed by port J) are given below.

(1) Wait until a packet becomes available for transmission.

(2) Wait until the earliest time instant such that one of the following conditions is valid, (i) $P(J)$ is true; (ii) *IDLEDET* $(J, R+T+t_d)$ is true. If (i) occurs first, then set* $S(J)$ and go to step 4. Otherwise, set $S(J)$ and go to step 3.

* Unless otherwise specified explicitly, 'set' and 'reset' refer respectively to the assignment of a logical HIGH value and a logical LOW value to a boolean variable.

(3) Initialize *TIMEOUT* (J, $R+T+t_d$) and start transmitting *PILOT*. Wait for the earliest time instant such that one of the following conditions holds, (i) $P(J)$ is true; (ii) *TIMEOUT* (J,.) has expired. If (i) occurs first, then remove *PILOT* and go to step 4. Otherwise, go to step 5.

(4) Wait until $P(J)$ is observed to be false and then go to step 5.

(5) Transmit the packet. Reset $S(J)$ at the end of the transmission and then go to step 1.

3.2 Properties of Algorithm A_2'

The collision-free operation of algorithm A_2' within a round is identical to that of algorithm A_3, and these details are available in Ref. 4. However, the initiation of a fresh round under scheme A_2' needs further explanation. If a packet ('pkt') becomes available to port J when a port to the right of port J is transmitting, then the transmission of 'pkt' can be scheduled only in a fresh round. Every such port J waits until the end of the ongoing round. The end of a round is indicated by a bus-idle condition for a period $R+T+t_d$ (or equivalently, a true value of *IDLEDET* (J, $R+T+t_d$)). Once the end of the round has been detected, all the waiting ports execute step 3 of the algorithm in order to start a new round. It is possible for the *PILOT* signals transmitted by different ports to interfere with each other, when more than one port attempts to start a new round. However, the *TIMEOUT* mechanism ensures that every contending port, except the leftmost one, detects $P(.)$ to be true and removes *PILOT* before the transmission of its packet commences. Consequently, the leftmost member of the contending group of ports gets the exclusive control of the data bus in order to transmit the first packet. Thereafter, bus access by ports to the right of the leftmost port proceeds in a conflict-free manner according to a left-to-right ordering, as in the case of scheme A_3.[4] Fig. 2 illustrates the above phenomenon. It may be noted that even when packets become available to ports when the whole network is idle (thereby leading to a *COLDSTART* condition),[4] the above procedure is used to start a bus-busy period. The incorporation of *PILOT* and *TIMEOUT* mechanisms ensures that no port is able

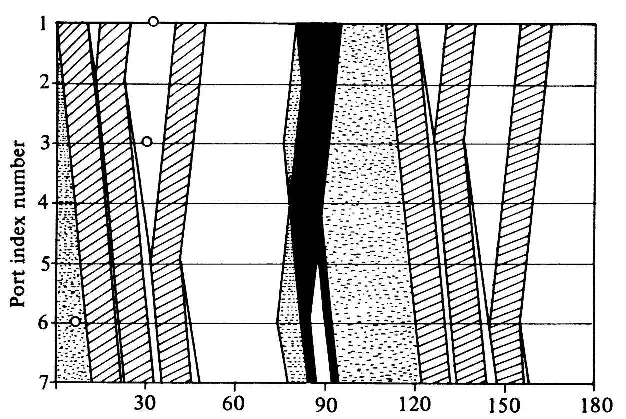


Fig. 2. Initiation of a bus-busy period under MAC scheme A_2'. Bus propagation delay, 12 μs; control wire delay, 18 μs; transmission time of each packet, 10 microseconds. ○ Packet arrival; ▩ packet transmission; ■ signal interference; ▧ pilot signal transmission.

to observe the bus to be idle for an interval longer than $R+3T+2t_d$, when the network is continuously busy. The above property has been utilised to implement the staggered round initiation procedure, which forms the basis of the prioritised MAC schemes described in the following sections. The parameter DEL_{max} denotes the worst-case bus-access delay experienced by any packet that has become available to some port for transmission. It can be shown by arguments similar to the ones used in Ref. 3 that the delay DEL_{max} satisfies the relation given by

$$DEL_{max} < (N-1)*(P_{max}+t_d)+2*(R+T)+t_d$$

Where P_{max} is the maximum transmission time of any packet.

4. PRIORITY SCHEDULING IN LANS

The issue of multiple priority function in LANs may be classified into two sub-aspects, namely the Local Priority Scheduling (LPS) and the Global Priority Scheduling (GPS). The above dichotomy is elaborated upon in the following sections.

4.1 Local Priority Scheduling

Local Priority Scheduling is concerned with the prioritised allocation of the LAN interface at each individual port to the locally generated packets. To this end, the LPS unit at each port maintains different queues of arriving packets corresponding to the different priority levels, with an FCFS discipline being maintained within each priority class. In the following we discuss two LPS protocols, LPS 1 and LPS 2, along with the relevant performance characteristics.

4.1.1. Scheme LPS 1

The scheme LPS 1 when incorporated is realized as a module that is completely decoupled from the MAC. This module is implemented as a higher-level sublayer of the Data Link Protocol as shown in Fig. 3, and it operates in coordination with the MAC as follows. LPS 1 module transfers each packet for transmission to the MAC unit (when the MAC is free) via the buffer INSERVICE. INSERVICE can hold one packet along with information regarding the priority of the packet (γ). Once INSERVICE is filled, the MAC unit proceeds to obtain medium access for the transmission of the packet held by INSERVICE. The next packet is not accepted by MAC until the transmission of the previous packet is complete. Further, the priority of the port (relevant only if the MAC module features GPS as explained in Section 4.2) remains constant (equal to γ) during the above interval. The value of γ is unaffected by the arrival of fresh packets until INSERVICE is ready to accept a new packet for transmission. When INSERVICE is ready to accept a new packet, the oldest packet of the highest level non-empty queue is transferred to INSERVICE and the operation is repeated. Further, a packet arriving at an empty port is directly accepted by INSERVICE and the MAC unit proceeds immediately to obtain medium access. The above approach relieves MAC of the queue management and hence simplifies the MAC implementation. However, LPS 1 has a certain disadvantage. This

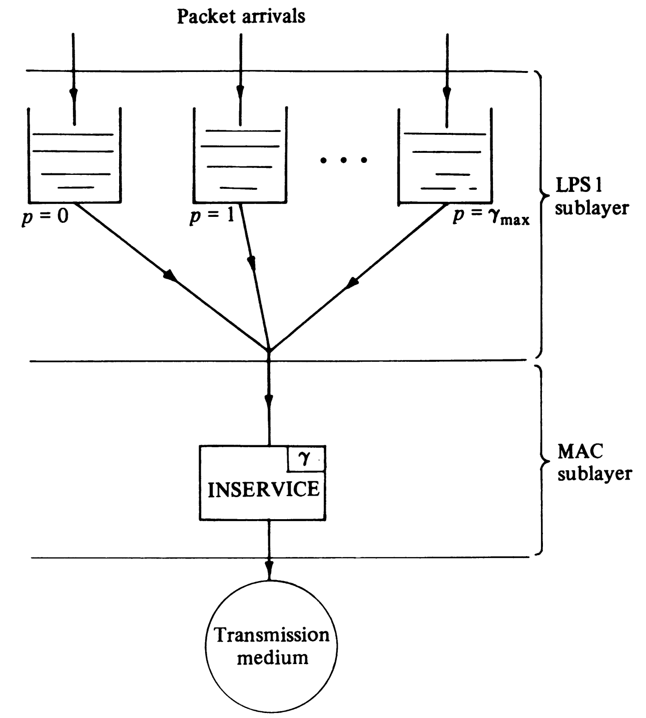


Fig. 3. Organisation of the local priority scheduler LPS 1.

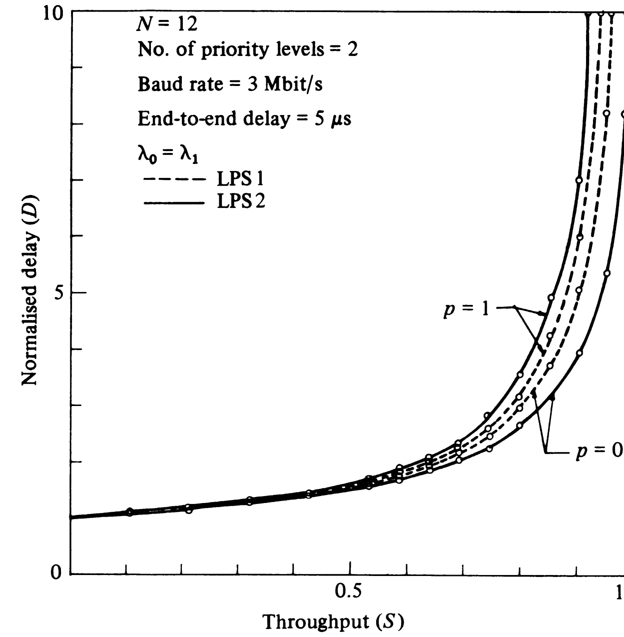


Fig. 4. Delay–throughput characteristics of MAC scheme A_2' under LPS 1 and LPS 2.

disadvantage stems from the fact that there is a time gap between the acceptance of a packet by INSERVICE and the commencement of transmission of the packet. It is possible for several packets of higher priorities to arrive at the port during the above interval. However, the transmission of any of these high-priority packets cannot be scheduled until the transmission of the impending low-priority packet is complete. This undesirable characteristic of LPS 1 is aggravated if the MAC unit features GPS (Section 4.2). This is owing to the fact that GPS can substantially delay the transmission of an obstructing low-priority packet because of the priority-based medium allocation to the ports.

4.1.2. Scheme LPS 2

The LPS 2 module and the associated prioritised packet queues are implemented as an integral part of the MAC protocol. In this case the notion of the availability of packets (Section 3) applies to each individual priority queue of the port. That is, each port may or may not have a packet available for transmission at any time corresponding to each priority class. Whenever the MAC unit observes at least one of the packet queues to be empty, it proceeds to obtain medium access. When medium access is granted, the transmission of the oldest packet of the highest-priority non-empty queue is scheduled. In addition, if the MAC features GPS (Section 4.2), then this module is required to accommodate dynamically the changes in the priority of the port due to the arrival of fresh packets. Unless the transmission of a locally generated packet is in progress, the priority of a port at any instant is equal to that of the highest-level queue (of the port) which is non-empty at that instant. However, if the transmission of a locally generated packet is ongoing, then the priority of the port is maintained constant, equal to that of the packet being transmitted. This restriction is imposed as the methods presented in this paper do not take pre-emptive priority-scheduling into consideration.[5] The scheme LPS 2 is free from the drawback of the scheme LPS 1 mentioned in the previous section. However, the choice of LPS 2 makes the implementation of the MAC module relatively more complex, as this module is required to maintain the packet queues and monitor the internal priority status dynamically.

4.1.3 Comparison of LPS 1 and LPS 2

In order to compare LPS 1 with LPS 2, the operation of these schemes has been studied by simulation. A hypothetical LAN of the form shown in Fig. 1, and with twelve ports and two priority levels, has been considered for the simulation study. Scheme A_2' (Section 3) has been employed as the MAC protocol, which operates in conjunction with one of the above to LPS schemes. The other details regarding the simulation environment and implementation are provided in Section 5.3. The results of the above performance study, depicted in Fig. 4, compare the delay–throughput characterics of the LAN using LPS 1 with that using LPS 2. Fig. 4 evinces the superiority (the rationale for this comparison is discussed in the next section) of scheme LPS 2 by virtue of the better prioritisation exhibited by this scheme.

4.2 Global Priority Scheduling

In addition to the LPS discipline within each individual station, there is a need for a global priority-scheduling mechanism. The design of the MAC protocol is directly influenced by the inclusion of GPS. The selection policy of the ideal GPS discipline should always be such that the port holding the oldest packet of the highest priority (over the entire network) obtains medium access before the others. However, the ideal service discipline can only be realised by a centralised access controller having complete knowledge of packets waiting over the whole network. In a distributed environment such as an LAN, the limited knowledge of the individual stations cannot ensure the optimal selection of ports; that is, lower-priority requests may gain medium access before higher-priority ones and 'younger' ones may gain access before 'older' ones of the same priority.[6] Moreover, the realisation of a priority-scheduling MAC requires the interchange of additional status information between the ports and consequently incurs extra overheads. Due to these two reasons, the performance of a practical MAC with GPS falls short of the performance of the ideal one.

The fundamental goal of prioritising the MAC scheme is to reduce the delay encountered by high-priority packets at the expense of an increase in the delay encountered by low-priority ones. In order to assess the effectiveness of such a mechanism, we need to formulate an appropriate performance measure. A precise approach would be to associate different costs of delay (dependent on the physical demands) with the different priority classes and to attempt to minimise the overall cost function. A simple case, yet one with a great practical significance, would be to associate a cost of unity with the high-priority class and zero with the low-priority one, in a system with two priority levels. In other words, we are willing to tolerate any amount of delay at the low-priority level to improve the high-priority delay performance. This is equivalent to specifying the separation between the delay–throughput curves of the two priority classes as the yardstick for comparison, and it is a standard approach adopted in the literature on this topic.[5,9,10] Accordingly, in this paper, the characterisation of one scheme as providing a superior prioritisation compared to another must be interpreted as the former exhibiting a greater separation of delay–throughput curves compared to the latter. Further, the priority mechanism has a discriminatory influence on other parameters such as the standard deviations of delays as well. Since we are interested in optimising the delay performance of the high-priority class, this paper examines the delay variance of only this class, under different MAC schemes.

The following sections discuss the design and performance evaluation of two MAC schemes PA 1 and PA 2 which are modified versions of the scheme A_2' (Section 3) and which incorporate the GPS. Scheme LPS 2 (Section 4.1.2) has been chosen as the local priority scheme in both cases. Further, it has been assumed that the priority $\gamma(J)$ of an active station J at any instant is equal to that of the highest-priority packet which is available to the station at the instant. The MAC schemes presented in this paper require that any port, on obtaining medium access, transmit one and only one packet of a pre-defined size. This restriction is imposed in order to ensure fairness amongst the ports, in case the loading at one or more priority levels is unsymmetric with respect to ports.

5. PA 1: A MAC SCHEME WITH NON-PRE-EMPTIVE PRIORITY SCHEDULING

We present in this section the MAC scheme PA 1, which is a modified version of scheme A_2' (Section 3) and which implements the GPS. The basic principle of operation of PA 1 is the priority-dependent staggering of initiations of bus-busy periods (Ref. 3, section 3). Each port, wishing to transmit in a fresh round, is required to wait until the bus is observed to be idle for an interval dependent on the priority of the port. In the following it is assumed that each priority level is represented by an integer-valued priority number. Successive integers represent successive priority levels in the descending order with zero signifying the highest priority class and γ_{max} signifying the lowest one. $\gamma(J)$ has been chosen to designate the current priority number of port J. Before initiating a fresh round, each port J is required to observe the bus to be idle for an interval ${}^{w}\gamma(J)$. The parameter ${}^{w}\gamma(J)$ is given by

$$^{w}\gamma(J) = (R + T + t_d)^{*}(\gamma(J) + 1),\ \gamma(J)\varepsilon\{0,\ 1, \ldots,\ \gamma_{max}\}.$$

5.1 Algorithm PA 1

The steps in the algorithm PA 1 (as executed by port J) are given below:

(1) Wait until such time that a packet is available corresponding to at least one priority class.

(2) Wait until the earliest time instant such that one of the following conditions is valid, (i) $\gamma(J) = 0$ and $P(J)$ is true, (ii) *IDLEDET* $(J, (R+T+t_d)^{*}\ (\gamma(J)+1))$ is true. If (i) occurs first, then set $S(J)$ and go to step 4. Otherwise, set $S(J)$ and go to step 3.

(3) Initialise *TIMEOUT* $(J, R+T+t_d)$ and start transmitting *PILOT*. Wait for the earliest time instant such that one of the following conditions holds, (i) $P(J)$ is true, (ii) *TIMEOUT* $(J,.)$ has expired. If (i) occurs first, then remove *PILOT* and go to step 4. Otherwise, go to step 5.

(4) Wait until $P(J)$ is observed to be false and then go to step 5.

(5) Transmit the oldest packet of priority number $\gamma(J)$ (highest-priority packet available currently). Reset $S(J)$ at the end of transmission and then go to step 1.

5.2 Features of Scheme PA 1

In order to illustrate the operation of scheme PA 1, assume that a bus-busy period is in progress. Such a period consists of transmissions by a set ϕ of ports. Also assume that packets become available to few of the ports belonging to the set ϕ^c during the ongoing round. Let γ be the priority number of the highest-level packets that become available (over the entire network) during the ongoing round. Then, steps 2 and 3 of the algorithm ensure that no port observes the data bus to be idle for an interval as long as $(R+T+t_d)^{*}(\gamma+2)$, between the ongoing round and the beginning of the next one. Hence, by step 2(ii) of the algorithm, none of the ports of priority numbers $(\gamma+1)$ and above is able to set the corresponding S flip-flop and transmit in the new round. All such ports are forced to repeat their attempts to obtain medium access at the end of the new round. As a consequence of the above staggering, ports holding packets of high priorities are able to initiate (and terminate successfully) a fresh round earlier than those holding packets of low priority. The packets of the highest priority level (priority number 0) constitute a special case. Let a packet of priority number zero become available to port J at time 't'. Then port J is permitted to transmit in the ongoing round (instead of waiting until a fresh round is initiated) if the port that is transmitting at time 't' is to the left of port J.

Scheme PA 1 has an obvious disadvantage. This stems from the fact that PA 1 is non-pre-emptive, in the sense that a bus-busy period, once initiated, cannot be pre-empted until completion. In other words, the global priority status is evaluated at the begining of each round rather than at the beginning of transmission of each packet. This prioritisation mechanism works well when the loading on the network is low or moderate, as each round may consist of few packet transmissions. However, when the network is uniformly and heavily loaded, the number of packets per round tends to be large, approaching the number of ports connected to the network. Consider the case whereby a high-priority packet becomes available to some port, when a round of packets of a lower priority is already in progress. The transmission of this high-priority packet has to be deferred until the completion of transmission of all the low-priority packets of the ongoing round. As a result, the prioritisation provided by scheme PA 1 becomes less effective under heavy loading conditions as shown in Section 5.3. The only way to circumvent the above undesirable feature is by providing a mechanism to interrupt an ongoing round when packets of higher priorities become available to one or more ports. This is basically the philosophy of design of the more sophisticated MAC scheme PA 2, presented in Section 6.

5.3 Performance of Scheme PA 1

With a view to estimating the effectiveness of the priority function of scheme PA 1 and to compare scheme PA 1 with the schemes A_2' (Section 3) and PA 2 (Section 6), a simulation study has been conducted on a DEC 1090 system using SIMULA. The simulation environment chosen consists of a bus network such as the one shown in Fig. 1, with a number N of ports equally spaced apart along the length of the bus. The end-to-end bus propagation delay T as well as the control wire delay R have been chosen to be 5 μs. The above choice of propagation delays corresponds to a maximum distance of about 1.5 km between port 1 and port N. The switching delay t_d has been assumed to be zero, as t_d is an attribute of the transmission technique used rather than that of the MAC protocol. The overhead due to packet addressing and control information also does not under the purview of the MAC protocol. Hence this overhead also has not been accounted for in the simulation study. For the above reasons, the results presented in this paper are upper limits of the performance indices of an actual network.

The details of the delay–throughput performance evaluation of scheme PA 1 in a purely data environment are presented in Section 5.3.1. Unless stated otherwise, the loading on the network has been assumed to be symmetrical. That is, corresponding to each priority number i, the total arrival rate λ_i of packets has been assumed to be equally distributed among the N ports. In order to assess the functionality of scheme PA 1 in comparison with that of the schemes A_2' and PA 2 in a typical application, the operations of these schemes have also been simulated in a mixed voice–data environment. The results of the performance study of the voice–data integration application are presented in Section 5.3.2.

5.3.1 Delay–Throughput Performance of Scheme PA 1

The transport delay of a data packet is the sum of two components, namely the queueing delay at the source port to obtain bus access and the transmission time of the packet. The delay–throughput performance evaluation has been conducted considering a network operating at 3 Mbaud and with the number N of ports set equal to 50. Each port has a packet generator corresponding to each priority level, which generates packets of 128 bytes each. The generator of priority number 'i' at each port generates packets according to a Poisson arrival process, with mean arrival rate given by λ_i/N. For the experiments discussed in the following paragraph, the simulation virtual time was chosen to be equal to that required for servicing 100 packets at each node, within each priority class. This is equivalent to the time required to service 5,000 packets over the entire network within each priority class. For validation purposes, a trial run was conducted with the virtual time increased to that required for the service of 150 packets at each node, under each priority level. The numerical results from the latter experiment tallied very well with the results from the former one, thereby asserting the validity and statistical consistency of the simulation results.

Fig. 5 illustrates the delay–throughput characteristics (for two priority levels with $\lambda_0 = \lambda_1$) of schemes PA 1, A_2' (with only LPS as discussed in Section 4) and PA 2 (section 6). From Fig. 5 it is evident that the priority function provided by scheme PA 1 is superior to that provided by scheme A_2', due to the inclusion of GPS in scheme PA 1. However, under scheme PA 1 the normalised transport delay D of packets of the higher priority shows a sharp increase as the network throughput approaches saturation. This is a consequence of the non-interruptibility of bus-busy periods, as explained in Section 5.2. Fig. 6 illustrates the delay–throughput characteristics of scheme PA 1 with two priority levels and with the ratio λ_0/λ_1 chosen as a parameter. Fig. 7 depicts the standard deviation of the normalized transport delay of the higher-

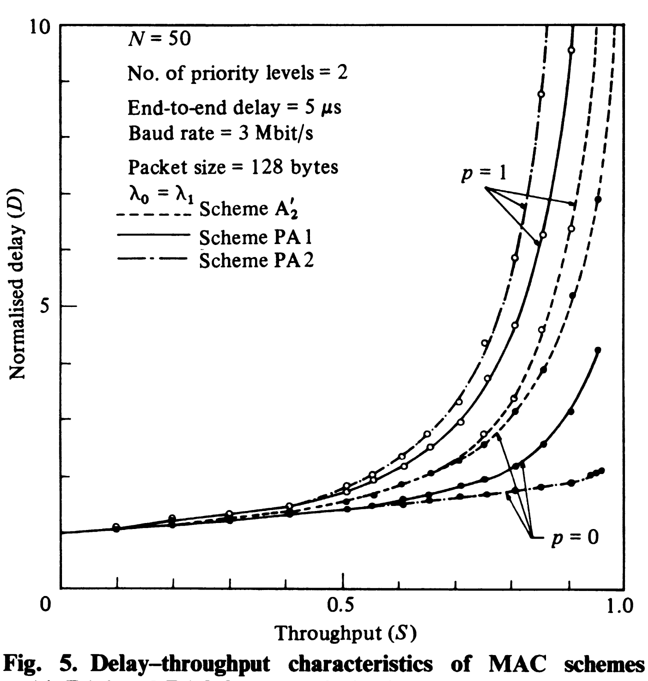


Fig. 5. Delay–throughput characteristics of MAC schemes A_2', PA 1 and PA 2 for two priority levels.

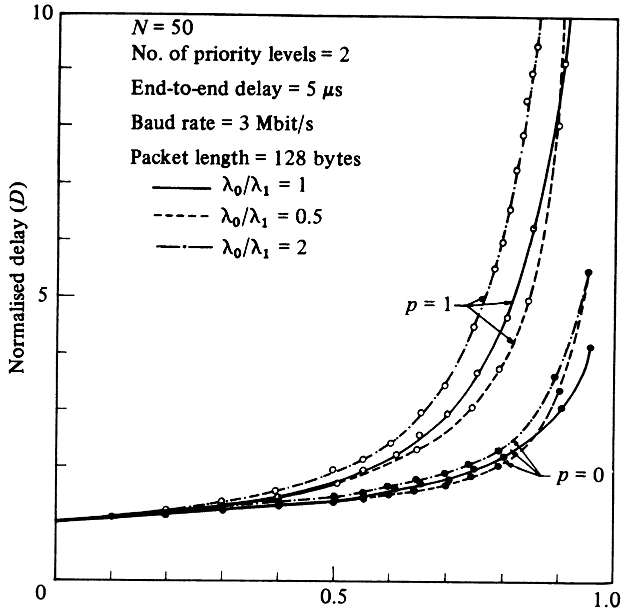


Fig. 6. Delay–throughput characteristics of MAC scheme PA 1 for different value of λ_0/λ_1.

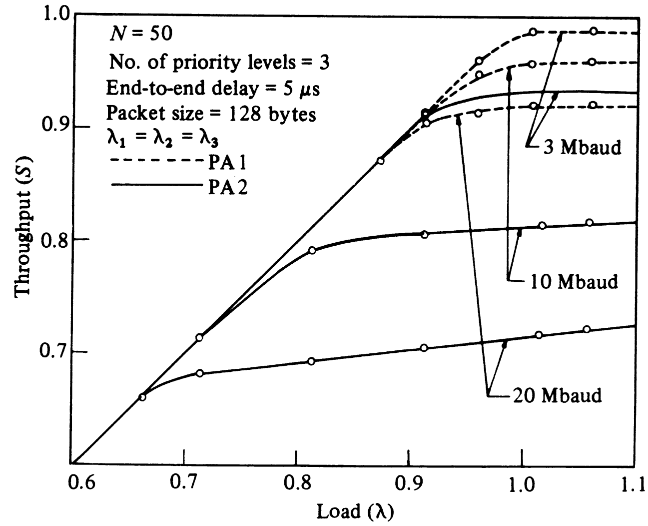


Fig. 8. Load-throughput characteristics of MAC schemes PA 1 and PA 2 for different network baud rates.

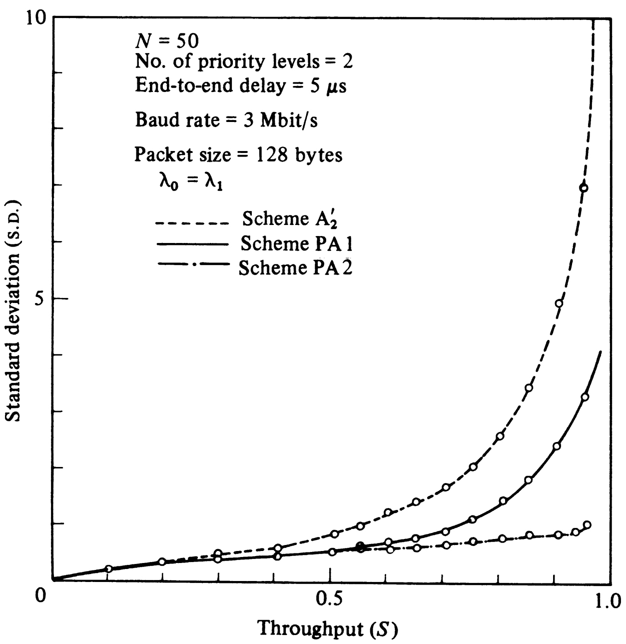


Fig. 7. Standard deviation of delays of high-priority packets.

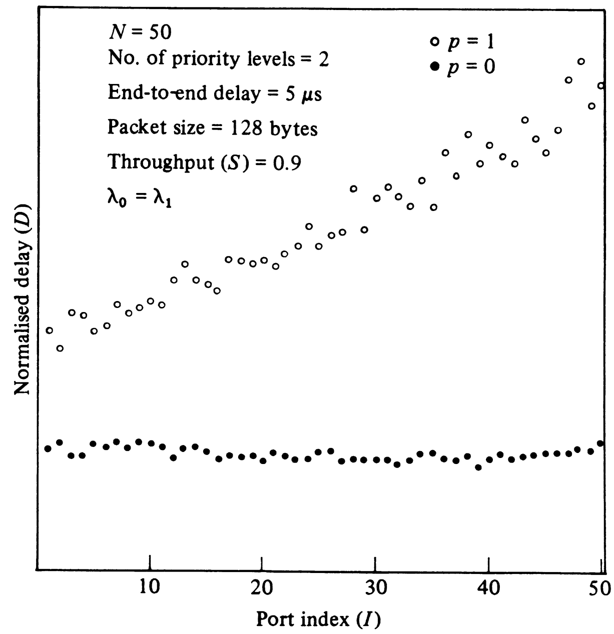


Fig. 9. Variation of normalised delay with respect to port index number under scheme PA 1.

level packets under schemes A'_2, PA 1 and PA 2. It is manifest in Fig.7 that the delays of high-priority packets under scheme PA 1 are subjected to a lower variability compared to those under scheme A'_2 and a higher variability compared to those under scheme PA 2 (Section 6). The throughput–load characteristics of schemes PA 1 and PA 2 (with three symmetrically loaded priority levels) have been plotted in Fig. 8 for three different values of network baud rates. Fig. 8, which reflects the efficiency of bandwidth utilisation, indicates that the highest throughput attainable with scheme PA 1 is more than 90%, even at transmission rates as high as 20 Mbit/s. The delay–throughput characteristics of scheme PA 1 on a network with three priority levels are shown in Fig. 10.

In order to characterise the fairness of scheme PA 1, the variation of the mean normalised packet delay D (for fixed throughput) with respect to the port index number J is shown in Fig. 9. In this case, the simulation time was chosen to be that required for the service of 500 packets per node, per priority class. It may be observed that, at the lower priority level, the value of D has a tendency to increase with increase in J. This is due to the fact that the transmission of all packets, except for the highest-priority ones, can be scheduled only in a fresh round (after each of them has become available). At the highest priority level, however, there is also a provision to schedule the transmission of a packet in the round that is in progress when the packet becomes available (Section 5.2). The chances of a high-priority packet to obtain medium access in the current round increase as the location of the port to which the packet becomes

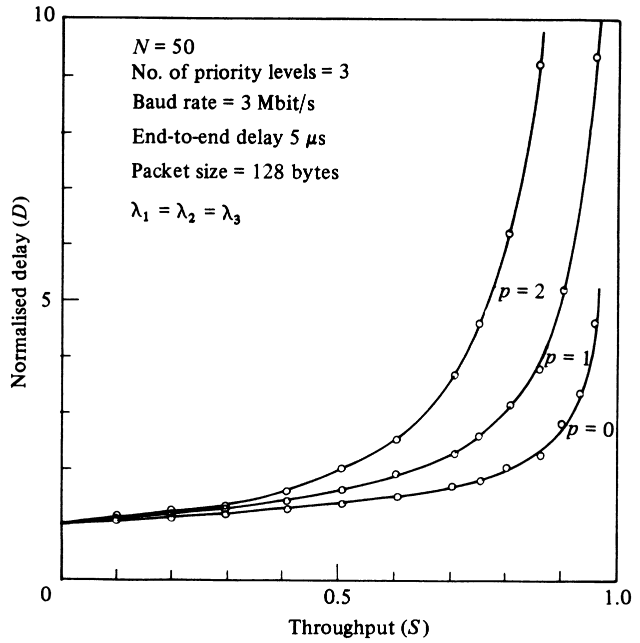


Fig. 10. Delay–throughput characteristics of MAC scheme PA 1 with three priority levels.

available shifts towards the right end of the network. As a result, the effects of the early bus access (within each round) given to ports with lower indices are offset by the higher probability of ports with higher indices to be able to transmit in an earlier round. Consequently, the value of D does not show any appreciable variation with respect to J as far as the highest-priority packets are concerned.

5.3.2. *Voice–data integration*

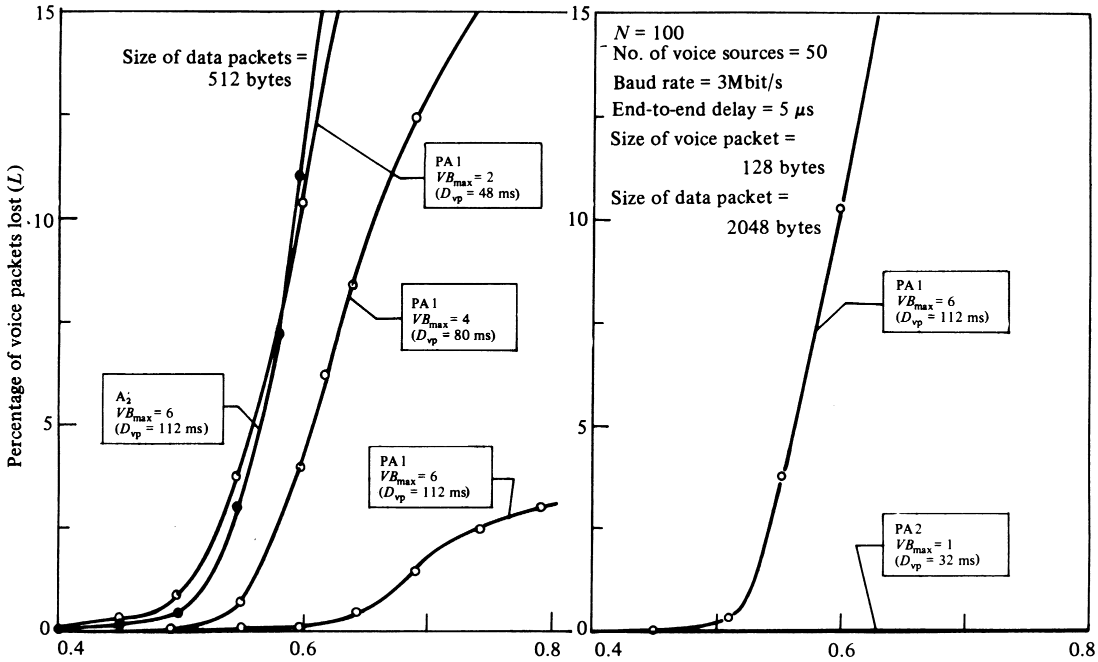
Voice signal consists of alternative periods of talkspurt and silence.[12] The analog voice to be transmitted on an LAN is digitised at a constant rate by a vocoder. A typical PCM encoder operating at 8 kHz produces one 8-bit word every 125 μs. The digital voice samples generated are collected into packets and are transported over the LAN. The transmission of voice packets is usually suppressed during the silent gaps. This step takes advantage of the 60–65 % idle time in conversations (one way) and makes extra bandwidth available for data transmission.[12] The experiments conducted by Brady[12] have indicated that in a typical conversation (one way), the average length of a talkspurt is about 1.366 seconds and that of a silent gap is about 1.802 s. Further, it has been suggested that the lengths of talkspurts and silent gaps may be modelled as geometrically distributed in terms of the number of voice packets generated.[13] In order to preserve the interactive quality of speech, the total delay from source to destination encountered by each voice packet has to be kept within tight bounds.[7] A total delay of up to 100 ms is considered acceptable. One way to impose an upper bound on the total delay is by restricting the size of the voice packet buffer at each station.[14] If the voice buffer at a port overflows due to contention for medium access, then the buffer is arranged as a FIFO queue, with the oldest sample being discarded when the new one is generated. The resulting loss (L) of voice packets (expressed as a percentage of the number of voice packets generated during the time window of observation) is an important parameter in packet voice communication. Typically, L must not exceed about 1 % for acceptable quality of the reproduced speech.[14]

The objective of the simulation study is to examine the variation of the packet loss L with the loading due to data packets. A network operating at 3 Mbaud and with 100 ports and two priority levels has been chosen as the environment of the simulation study. The higher priority level has been assigned to voice packets of 128 bytes each and the lower level to data packets of 512 bytes each. Every alternate port in the network has a generator of voice packets with the characteristics given in the previous paragraph. Thus there are 50 voice sources which correspond to 25 two-way conversations. Taking the on–off pattern of speech into account, this amounts to an average occupancy of 50 % of the bandwidth. In addition, each of the 100 ports has a generator of data packets with Poisson arrival times. The total delay D_{vp} encountered by a voice packet is the sum of three terms as given below:

$$D_{vp} = D_{vpgen} + D_{vpwait} + D_{vptx}$$

The first term D_{vpgen} is the time required to generate a voice packet and equals 16 ms for a voice-sampling rate of 8 kHz and a packet size of 128 bytes. The second term D_{vpwait} reflects the amount of time that a voice packet has to wait before obtaining medium access, and is stochastic in nature. If we assume that the size of the voice buffer at each port is limited to VB_{max}, then it is ensured that the value of D_{vpwait} is bounded above by $16 * VB_{max}$ ms. D_{vptx} is the time required to transmit a voice packet, and equals about 0.35 ms for a packet length of 128 bytes and transmission rate of 3 Mbaud. For the voice–data integration experiment, the simulation virtual time has been chosen to be 30 seconds of conversation.

Fig. 11 depicts the variation of parameter L with respect to the load λ_d which is imposed by the data traffic. Fig. 11(*a*) is concerned with the case where the data traffic is comprised of packets of 512 bytes each. The superiority of the voice-transmission reliability of scheme PA 1 to that of scheme A_2' is evident from Fig. 11(*a*). The scheme PA 1, by virture of its GPS (Section 4.2), maintains the value of L within acceptable limits, when the data traffic drives the network throughput to saturation. However, as Fig. 11(*a*) shows, scheme PA 1 requires a certain amount of buffering of voice packets at each transmitting port. This buffer is needed to accumulate the voice packets that arrive during the (uninterruptible) bus-busy periods of data packets. When each low-priority round of data packets terminates, all the voice packets queued up over the entire network are exhausted in a sequence of high-priority rounds. The amount of voice buffering required at each port is proportional to the size of the data packets. However, as mentioned previously, the buffer size at each port has to be restricted to about 6 packets in order to meet the delay constraints of voice. Consequently, the loss L tends to be unacceptably large under heavy data-loading conditions (if the data traffic consists of large packets). This is illustrated in Fig. 11(*b*), which considers a case where the size of each data packet is 2048 bytes. As described in Section 6.3, the more sophisticated scheme PA 2 overcomes the above drawback and renders the voice performance independent of the data packet size.



(a) PA 1 vs. A$'_2$ (b) PA 2 vs. PA 1

Fig. 11. Variation of loss of voice packets with respect to data load.

6. PA2: A MAC SCHEME WITH SEMI-PRE-EMPTIVE PRIORITY SCHEDULING

As mentioned earlier, the priority function of scheme PA 1 fails under heavy network-loading conditions. In this section we present the prioritised MAC scheme PA 2 that overcomes the above drawback and attains the perfect ordering of packet transmissions characterising the ideal priority discipline (Section 4.2). The idea is to include a method in the MAC to interrupt a current bus-busy period if packets of higher priorities become available at one or more ports. The scheme PA 2 is termed semi-pre-emptive, because PA 2 permits the pre-emption of a current round by higher-priority packets, but within packet boundaries. Once such an interruption has occurred, a fresh round of packets of the highest current priority commences. This may be followed by more rounds of priorities higher than the interrupted level. Although scheme PA 2 offers a very effective priority function, it entails a greater complexity of implementation. Moreover, PA 2 imposes certain restrictions on the packet format and results in extra network overhead.

Before discussing the details of scheme PA2, we describe below the special packet format and the additional capabilities required of each port. Each packet transported on the network is required to have a field called *PRIOIND* that indicates the priority number *PRIONUM* of the packet. Every active port is expected to be able to read and record the *PRIONUM* of all packets being broadcast on the bus. Each port has a mechanism to detect signal interference on the bus due to transmission by multiple ports. In order to ensure that every port on the network detects such a collision (irrespective of whether the port concerned is involved in the collision or not), the above mechanism has to be implemented based on special coding of the transmitted bits or on energy-level discrimination. Each port J maintains corresponding to each priority number 'p' a two-valued variable denoted by *PRMPT* (J,p). *PRMPT*(J,p) is equal to unity at any time if and only if the previous attempt of port J to transmit a packet of priority number 'p' had been pre-empted (as explained in Section 6.3). Further, every port J has its *PRMPT*$(J,0)$ set to zero permanently.

6.1 Algorithm PA 2

The following is the algorithm PA 2 (as executed by port J):

(1) Wait until a packet is available corresponding to at least one priority level.

(2) Wait for the earliest time instant such that one of the following conditions holds: (i) *IDLEDET*$(J, [2^*\gamma(J) - PRMPT(J,\gamma(J)+1]^*[R+T+t_d])$ is true; (ii) *PRIONUM* is equal to $\gamma(J)$ and $P(J)$ is true; (iii) *PRIONUM* is greater than $\gamma(J)$. If (i) occurs first, then set $S(J)$ and go to step 4. If (ii) occurs first, then set $S(J)$ and go to step 5. Otherwise, go to step 3.

(3) Following the completion of the current packet transmission, jam the bus with *PILOT* for an interval $R+T+t_d$. Go to step 2.

(4) Initialise *TIMEOUT* $(J,R+T+t_d)$ and start transmitting *PILOT*. Wait until one of the following conditions holds: (i) *TIMEOUT* $(J,.)$ has expired, (ii) $P(J)$ is true. If (i) occurs first, then go to step 6. Otherwise, remove *PILOT* and go to step 5.

(5) Wait until the earliest instant such that one of the following conditions is valid: (i) $\gamma(J)$ has changed (a new packet of a higher priority has become available): (ii) A collision is detected; (iii) $P(J)$ is false. If (i) occurs first, then reset $S(J)$ and go to step 3. If (ii) occurs first, then reset $S(J)$, set *PRMPT* $(J, \gamma(J))$ to *ONE* and go to step 2. Otherwise, go to step 6.

(6) Start transmission of the packet. If a collision is detected within the first $2T+t_d$ time units, then jam the bus for an interval t_d. At the end of this interval, reset $S(J)$, set $PRMPT\ (J, \gamma\ (J))$ to ONE and then go to step 2. If no collision is detected, then continue the transmission of the packet to completion. At the end of transmission, reset $S(J)$, set $PRMPT\ (J,\gamma(J))$ to $ZERO$ and then go to step 1.

6.2 Operation of Scheme PA2

The operation of scheme PA2 based on round pre-emption is as follows. If any port J with a packet (of priority number $\gamma(J)$) available observes a round already in progress, port J compares $\gamma(J)$ with the $PRIONUM$ of the packet currently being transmitted. If $\gamma(J)$ is greater than $PRIONUM$, then port J defers its transmission until a fresh round of priority number $\gamma(J)$ is initiated or until the value of $\gamma(J)$ has changed. If $\gamma(J)$ is equal to $PRIONUM$ (indicating that the priority of the available packet is equal to that of the current round), then port J schedules the transmission of its packet in that round, provided the port that is currently transmitting is to the left of port J. The above condition is indicated to port J by a $TRUE$ value of $P(J)$. A $\gamma(J)$ value less than $PRIONUM$ (indicating the availability of a packet of higher priority) causes port J to initiate a round pre-emption. In order to interrupt the current round, port J waits until the end of the packet currently being transmitted and them jams the bus with the $PILOT$ signal. The $PILOT$ signal sent by port J causes a deliberate collision with the next packet in the round, if there is any more packet transmission in the round. On detecting the occurrence of a collision, each port I waiting to transmit in the current round resets $S(I)$ to LOW, sets $PRMPT\ (I,\gamma(I))$ to unity and releases the bus. The variable $PRMPT\ (I,p)$ is used by port I to record the occurrence of a pre-emption at priority level p. A fresh round of the highest currently available priority is started by the principle of priority-dependent staggering of round initiations, as in the case of scheme PA1. The transmission of packets of the interrupted level and all the lower levels remains suspended as long as there is a packet of higher priority available at least at one port. Eventually, when a round of the interrupted priority level restarts, it has to be ensured that only those ports that have been pre-empted are permitted to transmit in that round. In other words, an interrupted round, when it restarts, should resume from the pont of interruption. Otherwise, it is possible for a port with a low index to monopolise the bus (at a particular priority level) if repeated pre-emptions occur. In order to ensure that a pre-empted round restarts properly, the stagggered delay w_p corresponding to each priority number p assumes two values depending on the state of $PRMPT\ (.,p)$. Table 1 depicts the values of w_p for a network with three priority levels.

Table 1 can be represented by the relation given by

$$w_p = (2p - PRMPT(., p) + 1)*(R+T+t_d).$$

The above relation has been used to represent the staggered delays in the description of algorithm PA2 given in Section 6.1. In order to enable the unpre-empted ports to transmit in a subsequent round, each port J that has completed transmission of a packet of priority number 'p' resets $PRMPT\ (J,p)$ to zero. The organisation of scheme PA2 as explained above is analogous to that of the multiple-priority interrupt structure of a computer operating system. As the operation of scheme PA2 is dependent on the detection of collisions, each packet is required to be as long as at least $R+T$.

Table 1. Staggering intervals for scheme PA2

P	PRMPT $(.,p)$	w_p
0	Not applicable	$R+T+t_d$
1	1	$2(R+T+t_d)$
	0	$3(R+T+t_d)$
2	1	$4(R+T+t_d)$
	0	$5(R+T+t_d)$

6.3 Performance of Scheme PA2

A performance evaluation of scheme PA2 has been conducted by simulation. The details of the simulation study have been provided in Section 5.3. The delay–throughput characteristics of scheme PA2 on a network with two priority levels is depicted in Fig. 5. It is clear from this figure that the mean normalised delay D of the high-priority packets remains low even under heavy network-loading conditions. Fig. 12 illustrates the delay–throughput curves of scheme PA2 for three priority levels. The variation of network throughput with respect to loading is shown in Fig. 8. It is evident from Fig. 8 that the maximum throughput attainable with scheme PA2 tends to decrease rapidly as the baud rate becomes large. This behaviour is attributable to the additional overhead incurred by scheme PA2 due to round pre-emptions.

Under scheme PA2, it is ensured that the transmission of a packet is delayed due to the transmission of at most one packet of a lower priority. This property of scheme PA2 makes it very well suited for voice–data integration by assigning the highest priority level to voice (Section 5.3.2). In such an application, at the end of transmission

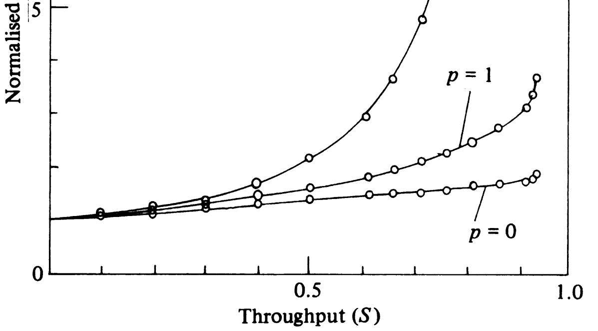


Fig. 12. Delay-throughput characteristics of MAC scheme PA2 with three priority levels.

of every data packet a sequence of high-priority rounds is initiated to exhaust the voice packets waiting over the entire network. Consequently, the loss L of voice packets (Section 5.3.2) under scheme PA 2 is not affected by the size of data packets. In this respect, the performance of scheme PA 2 is superior to that of scheme PA 1 in a mixed voice–data environment. This is illustrated in Fig. 11(*b*).

7. CONCLUSIONS

This paper is concerned with the development and performance evaluation of three modified versions (schemes A_2', PA 1 and PA 2) of an algorithm, proposed for medium access control in local area networks. The scheme A_2' has been designed in order to overcome certain drawbacks of the original algorithm. The schemes PA 1 and PA 2, based on A_2', are intended to implement the feature of multiple priority function in local area networks. PA 1 realises multiple priority scheduling by the principle of staggered-round initiations. However, due to certain inherent features, the priority function of PA 1 fails under heavy network-loading conditions. In order to circumvent the shortcomings of scheme PA 1 and approach the perfection of the ideal priority discipline closely, the more sophisticated control scheme PA 2 has been proposed. However, scheme PA 2 entails a greater complexity of implementation and increases the overhead of network operation.

Acknowledgement

The authors wish to express their deep gratitude for the constructive suggestions of the Referees.

REFERENCES

1. R. M. Metcalfe and D. R. Boggs, Ethernet: distributed packet switching for local computer networks. *Comm. ACM* **19**, 395–404 (1976).
2. I. Chlamtac, W. R. Franta and K. D. Levin, BRAM: the broadcast recognizing access method. *IEEE Trans Comm.* **27**, 1183–1190 (1979).
3. K. P. Eswaran, V. C. Hamacher and G. S. Shedler, Collision-free access control for computer communication bus networks. *IEEE Trans. Software Eng.* **7**, 574–582 (1981).
4. L. Fratta, An improved access protocol for data communication bus networks with control wire. *Proc. ACM SIGCOMM* 83, Symposium on Communications Architectures and Protocols, pp.219–225 (1983).
5. R. Rom and F. A. Tobagi, Message-based priority functions in local multiaccess communication systems. *Computer Networks* **5**, 273–286 (1981).
6. IITB, *Interim Report for Costll BIS: Performance Analysis of LAN's*, IITB Report no. A1323–9716 (1983).
7. C. J. Coviello, O. I. Lake and G. R. Redinbo, System design implications of packetized voice. *Proc. IEEE Int. Conf. Comm.* **3**, 38.3–49 to 38.3–53 (1977).
8. F. A. Tobagi, Performance analysis of message-based priority functions. *IEEE NTC* '80 *Conf. Rec.*, p. 37-5 (1980).
9. I. Iida, M. Ishizuka, Y. Yasuda and M. Onoe, Random access packet switched local computer network with priority function. *IEEE NTC* '80 *Conf. Rec.*, p. 37-4 (1980).
10. I.Iida, Y. Yasuda and Y. Komachi, Modified CSMA/CD local area network with message-based priority function. *Proc. IEEE INFOCOM* 83, pp. 472–477 (1983).
11. J. W. Mark, Distribution scheduling conflict-free multiple access for local area communication networks. *IEEE Trans. Comm.* **28**, 1968-1976 (1980).
12. P. T. Brady, A statistical analysis of on-off patterns in 16 conversations. *Bell Syst. Tech. J.* **47** (1) 73-91 (1968).
13. N. Janakiraman, B. Pagurek and J. E. Neilson, Multiplexing voice sources. *IEEE NTC* '81 *Conf. Rec.* **4**, G4.3.1–G4.3.5 (1981).
14. T. A. Gonsalves, Packet–voice communication on an Ethernet local computer network: an experimental study. *Proc. ACM SIGCOMM* '83, Symposium on Communications Architectures and Protocols, pp. 178–185 (1983).
15. M. Gopalakrishnan and L. M. Patnaik, Collision-free bus access schemes for local area networks with multiple packet priorities. *Proc. IEEE Int. Conf. on Computers, Systems and Signal Processing, Bangalore, India*, **2**, 829–833 (1984).