Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes

LEONARD KLEINROCK, FELLOW, IEEE, AND MICHEL O. SCHOLL

Abstract—We study new access schemes for a population of geographically distributed data users who communicate with each other and/or with a central station over a multiple-access broadcast ground radio packet-switching channel. We introduce and analyze alternating priorities (AP), round robin (RR), and random order (RO) as new *conflict-free* methods for multiplexing buffered users *without control from a central station*. These methods are effective when the number of users is not too large; as the number grows, a large overhead leads to a performance degradation. To reduce this degradation, we consider a natural extension of AP, called minislotted alternating priorities (MSAP) which reduces the overhead and is superior to fixed assignment, polling, and known random access schemes under heavy traffic conditions. At light input loads, only random access schemes outperform MSAP when we have a large population of users. In addition, and of major importance, is the fact that MSAP does not require control from a central station.

I. INTRODUCTION

THE constantly growing need for access to computers, data communication channels, and distributed computer communication networks creates a formidable problem of allocating these large, expensive resources among an ever-increasing number of users. In this paper, we restrict our attention to the allocation of a data communication channel to a set of local data sources. The demands placed upon this finite-capacity resource are unpredictable and *bursty* [1] and are made by a population of geographically scattered and (possibly) mobile users.

One of the major problems in data communications is to provide local access from users terminals to information processing systems available from a local computer or from a network of computers. The users (terminals) are connected to the computer (or network) by means of a *centralized* communication network; that is, all channel demands are made either by terminals which need access to the computer or by the computer which must be connected to a terminal. In such networks, the central node controls the transmission from any user.

We will also consider the communication channel as a media providing direct communication among the users (terminals, minicomputers) themselves. In such point-to-point communication networks, it is often true that a central node can no longer efficiently and reliably control the transmission of all the network's users. On the contrary, control may be distributed among the users themselves. We are concerned with efficiently providing access from the users to the (centralized or point-to-point) communication network.

At the end of the 1960's the store and forward *packet-switched* technology emerged as a cost-effective alternative to the widespread expensive circuit switching technology [2]. In the packet-switched technology [3, vol. II], the communication links are statistically shared by messages from different source-destination pairs. In addition, each message can be broken into packets of information with the addresses of the source and destination attached to each packet. Packets are individually routed through the network to their destination by "hopping" from one node to another.

Originally applied in distributed computer-communications networks, packet-switching techniques have more recently been effectively used in radio communications (both satellite and ground radio channels) [3, vol. II]. One of the first packet-radio communication systems was the ALOHA system developed at the University of Hawaii [4]. The Advanced Research Projects Agency (ARPA) of the Department of Defense has developed an experimental packet-radio broadcast network as an interface between a point-to-point wire network (like the ARPANET) and a number of geographically scattered terminals [5]. Furthermore, there is currently an immense worldwide interest in the development of satellite communications systems [6]-[8] which may be operated as multiaccess radio broadcast channels.

In this paper, we focus our attention on data communication over packet-switched *ground radio* systems as an attractive means for data transmission among users (here we exclude the study of satellite communications systems). For such data communication among users, broadcast radio communications is chosen as an effective alternative to conventional wire communications for the following reasons: 1) in a broadcast mode, any number of users may access the channel and the transmission of a signal by a user may be received over a geographically wide area by any number of receivers; 2) a broadcast mode is particularly suitable when the users are mobile or are located in remote regions where a wire connection is not easy to implement; 3) the design of a broadcast system is flexible; and 4) given that all users are in line of sight (LOS) and within range of each other, the provision of a completely connected network topology by a radio channel eliminates complex topological design and routing problems [9], [10].

Of interest to this paper is the consideration of a single-

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received July 20, 1977; revised October 22, 1979. This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-77-C-0272.

L. Kleinrock is with the Department of Computer Science, University of California, Los Angeles, CA 90024.

M. Scholl was with the Department of Computer Science, University of California, Los Angeles, CA 90024. He is now with IRIA, Domaine de Voluceau–Rocquencourt, Le Chesnay, France.

broadcast high-speed radio channel shared in some multiaccess fashion and in a packet-switched mode. The radio channel as considered in the following is characterized as a wide-band channel with a maximum propagation delay τ between any source-destination pair which is only a very small fraction a of the packet transmission time.[1]

The problem we are faced with is how to share and how to control access to the channel in a fashion which provides an acceptable level of performance. Several multiple access techniques which attempt to resolve some of these issues have previously been implemented or proposed. These fall into the following categories:

1) fixed assignment, e.g., time division multiple access (TDMA) and frequency division multiple access (FDMA) [11]

2) random access schemes, e.g., ALOHA [6], [8], [13], [14] and carrier sense multiple access (CSMA) [15], [16]

3) reservation techniques, e.g., roll call polling [11], [12], carrier sense split-reservation multiple-access (CS SRMA), [17] dynamic conflict-free reservation schemes [18].

With TDMA and FDMA, the performance is very sensitive to the number of users, and we observe a poor delay performance at low loads due to the inherent burstiness of the traffic [1]. On the other hand, ALOHA and CSMA provide excellent delay performance at low input rates, even for a very large number of users (e.g., $N = 1000$ users) at the price of collisions increasing with the load; however, at higher loads, this results in a poor channel efficiency. Random access modes lend themselves to distributed control for access to the channel, while reservation schemes require special control signals.

We inquire whether such random access techniques are optimal for the distributed channel access control of a small number ($N \leqslant 20$) of (possibly) buffered users. To answer this question, in this paper we recommend an approach different from the conflict-prone random access modes. This approach (which was reported upon in [21]) is as follows.

1) We choose a distributed *dynamic channel assignment* algorithm known to each user which is *conflict-free*. By avoiding collisions, we ensure a high channel utilization under heavy traffic conditions. By a distributed algorithm, we mean the following. If user i is presently transmitting a packet over the channel, an assignment scheme or priority rule *common to all users* designates user j (possible $= i$) to transmit next. Thus, all users know from this priority rule to which user (j) the channel has next been assigned.

2) As in CSMA, we use the carrier-sensing capability of each user, i.e., the capability of each user to listen to the carrier due to another user's transmission.[2] After a propagation time τ, all other users may start detecting the presence or absence of the carrier due to user j's transmission. In case the carrier is absent (user j had no packet to transmit), they all know from the priority rule which user (say k) is next chosen, and user k may then start transmission immediately. They all listen to the carrier for the next τ seconds, after which, if the carrier is absent, the next assigned user may start transmitting a packet, etc. Therefore, even though only one user has a packet to transmit, after a worst case of N attempts, this user will be chosen again. Two classes of protocols are presented and analyzed in this paper. Alternating priorities (AP), round robin (RR), and random order (RO) belong to the first class and are the subject of Section II in which we present the protocols, discuss the assumptions, establish the throughput-delay performance, and finally compare them to the performance of existing multiple access modes. AP, RR, and RO are shown to be suitable for multiple access by a small number of (possibly) buffered users without the control of a central station. However, the performance degrades badly as the number of users increases. As an example of the second class of protocols, in Section III we introduce and analyze minislotted alternating priorities (MSAP) and compare its performance to that of the other competitive access schemes. MSAP is adapted to *distributed* multiple access to a ground radio channel with a small number of buffered users, and is shown to accept a larger number of users ($N \leqslant 50$) without serious performance degradation. In spite of its sensitivity to carrier sensing errors, MSAP is shown to outperform roll call polling, and it is one of the few schemes known which, under heavy traffic conditions, performs as well as $M/D/1$ (perfect scheduling) to within only a multiplicative constant.

II. AP, RR, AND RO PROTOCOLS AND THEIR THROUGHPUT-DELAY CHARACTERISTICS

A. Transmission Protocols and System Assumptions

At any instant, a user is said to be *ready* if he has a packet ready for transmission (otherwise he is said to be *idle*). The four protocols considered below (HOL, AP, RR, and RO) differ by the priority assignment rule common to all users. This rule, based on which user has transmitted the last packet, establishes a priority order among the N users for the next packet transmission. That user who is ith in priority order ($1 \leqslant i \leqslant N$) (assuming he is ready) transmits the next packet if and only if all higher priority users are idle.

All users are assumed to be in line-of-sight (LOS) and within range of each other. Therefore, we assume that any user has the ability to sense the carrier of any other transmission on the (common frequency) channel. Let τ be the maximum propagation time between any source-destination pair. The time required to detect the carrier due to a packet transmission is considered to be negligible (or, which is equivalent, is included in τ). All packets are of constant length and are transmitted over an assumed noiseless channel; the probability of false carrier detection is considered to be negligible. The system assumes no multipath effect (the effect of multipath on a signal is to spread the signal duration due to echoes). We assume that the time axis is slotted. A slot consists of three parts (see Fig. 1):

1) an overhead of ($N - 1$) "minislots," each of duration τ, where N is the number of users;

[1] Consider, for example, 1000 bit packets transmitted over a radio channel operating at a speed of 100 kbits/s. If the maximum distance between any source-destination pair is 10 mi, then the (speed of light) packet propagation delay is of the order of 54 μs. Therefore, $a = 0.005$. On the contrary, when one considers satellite channels [14], a may be three or four orders of magnitude larger.

[2] In the context of packet radio channels, sensing carrier prior to transmission was originally suggested by D. Wax of the University of Hawaii in an internal memorandum dated March 4, 1971. The practical problems of feasibility and implementation of carrier sensing are not addressed here.

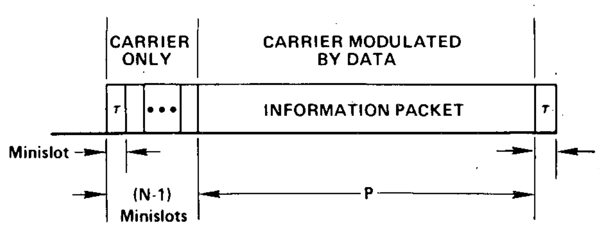


Fig. 1. HOL, AP, RR, RO: slot configuration.

2) the packet transmission time of duration P;

3) one minislot (of duration τ) which accounts for the (propagation) time between the end of transmission and the end of reception.

The N users are ordered in each slot (of duration $P + N\tau$: see Fig. 1) by the priority rule which characterizes the protocol. For all priority rules (and thus for all protocols), the N users are synchronized[3] in each slot as follows.

1) If the highest priority user is ready, he need not sense the channel and synchronizes his packet's transmission as follows.

a) At the beginning of the slot, he begins transmission of the carrier (with no data modulation). After one minislot at most, all other users know whether the slot is reserved (carrier detected) or not (carrier absent).

b) $(N - 1)$ minislots later he transmits his data packets. Otherwise, (if he is idle), he remains quiet until the end of the slot.

2) If the ith user in priority $(1 \leqslant i \leqslant N)$ is ready, he senses the channel for $(i - 1)$ minislots.

a) If no carrier is detected after $(i - 1)$ minislots, then at the beginning of the ith minislot, he transmits his carrier, and $(N - i)$ minislots later, he transmits his packet.

b) Otherwise (idle user or carrier detected earlier), he waits for the next slot and the process is repeated (with a possibly different priority order).

Under all protocols, a slot is unused if and only if all N users are idle at the beginning of this slot.

We first consider head of the line (HOL) [3, vol. II]. This protocol is devised for a population of N users on which a fixed priority structure is imposed; the priority among users remains constant in time, i.e., the ordering of the N users does not change from one packet transmission to the next.

We next consider alternating priorities (AP), named after the priority queueing system studied in [19]. In such a protocol, once a user seizes the channel, he keeps transmitting packets until he goes idle. More precisely, AP obeys the following rule. The N users are numbered in a given sequence (say $1, 2, \cdots, N$). The highest priority is assigned to that user (say user i) who transmitted the last packet. Priority then decreases in cyclic order around the numbered users, that is, the priority ordering is, in decreasing order,

$$i,\ [i \bmod N] + 1,\quad [(i+1) \bmod N] + 1, \cdots,\quad [(i+N-2) \bmod N] + 1.$$

[3] The practical problems imvolved in synchronizing users are not addressed in this paper (multipath, false carrier detection, variable propagation time due to variable distance between the various source-destination pairs, etc.). Clearly, the impact of such implementation problems on performance is nonnegligible, and the analytical results throughout the paper represent only an upper bound on performance.

The third protocol is called round robin (RR). As in AP, the users are numbered according to a given sequency (say $1, 2, \cdots, N$). In this protocol, the highest priority is assigned in a round-robin cyclic fashion among the users. That is, the highest priority in a given slot is assigned to that user whose number (mod N) *follows* that of the user (say user i) who had highest priority in the previous slot. This is true even if user i was idle (and therefore did not transmit) in the previous slot. Thus, the priority ordering in the current slot, in decreasing order, would be

$$[i \bmod N] + 1,\quad [(i+1) \bmod N] + 1, \cdots,\quad [(i+N-2) \bmod N] + 1, i.$$

In the fourth protocol, called random order (RO), the priority order of the N users is chosen at random, i.e., each user generates the same pseudorandom permutation of $1, 2, \cdots, N$ which gives the priority (in decreasing order) of the N users for the current slot. No matter who uses the current slot, each user generates a new permutation (the same for all users) which gives the priority order of the N users for the next slot.

B. Traffic Model and Channel Capacity

Here we characterize the traffic source, define some variables, and give the first important performance measure, namely, the channel capacity.

We assume that our traffic source consists of a finite number N of buffered users, each with unlimited buffer space. Each user generates new packets independently of the others according to a homogeneous Poisson point process. We assume that the full packet is instantaneously generated at those points. The aggregate packet generation rate is denoted by λ (packets/second). If N is not too large, each user may generate packets frequently enough so that the interarrival time between successive packets at a given user is less than the delay incurred by a packet from arrival to the end of transmission. Thus, a user may have more than one packet requiring transmission at any time, and those will be transmitted on a first-come first-served basis from his queue.

In addition, we characterize the traffic as follows. Each packet (of constant length) requires P seconds for transmission. Let $S = \lambda P$. S is the total average number of packets generated per transmission time, i.e., it is the input rate normalized with respect to P. In equilibrium, S can also be referred to as the *channel utilization* [14], [16]. Indeed, if we were able to perfectly schedule the packets into the available channel space with absolutely no control overhead, we could achieve a maximum throughput equal to 1 (packet/slot). Because there are N minislots wasted (for sensing the carrier) between the successive transmission of two packets (see Fig. 1), the maximum achievable throughput (the maximum channel utilization), called the *channel capacity* [14], [16] under a given protocol and denoted by C, is less than one; the maximum rate of packets transmitted per slot is always equal to one, but the slot size increases with N. Since within each slot, $N\tau$ seconds are lost for control, the channel capacity of HOL, AP, RR, and RO is

$$C = \frac{1}{1 + Na} \tag{1}$$

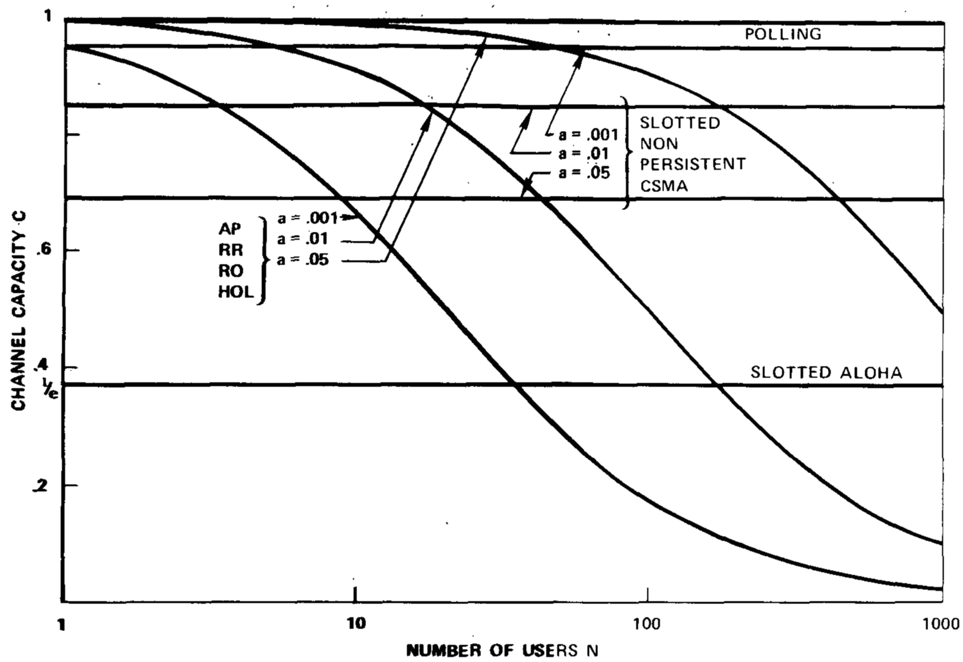


Fig. 2. HOL, AP, RR, RP: effect of number of users on channel capacity.

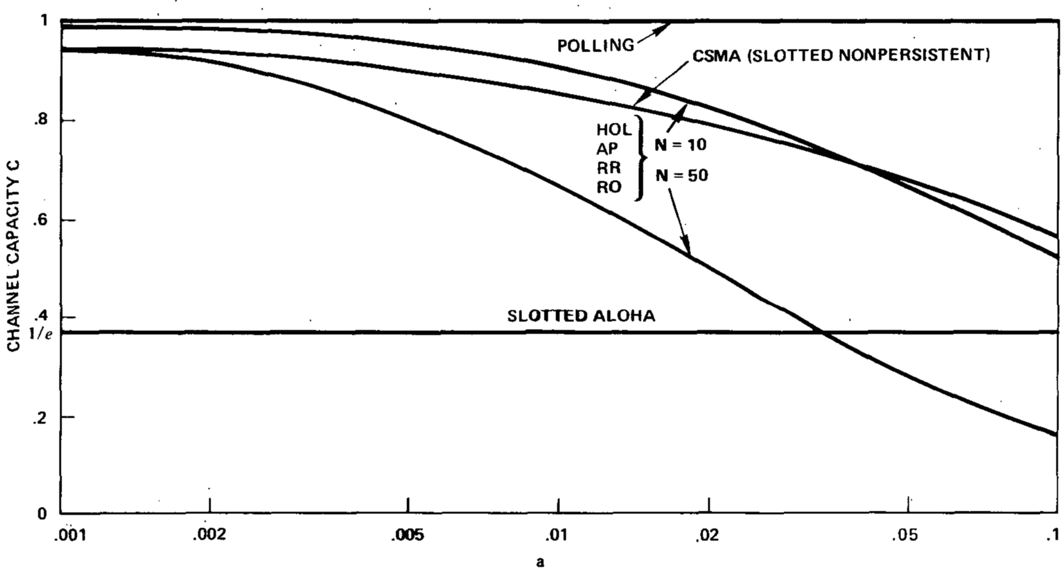


Fig. 3. HOL, AP, RR, RO: effect of propagation delay on channel capacity.

where $a = \tau/P$.

In Fig. 2 we plot the capacity C versus the number of users N for various values of a. On the same figure are plotted the capacity of (roll call) polling, CSMA, and slotted ALOHA. With polling [12], one can always achieve a theoretical throughput of 1, since when one user transmits heavily over the channel, he keeps transmitting at a rate of one packet per packet transmission time. If his buffer never empties, there is no waste of the channel due to switching to (polling) another user. The capacity of slotted ALOHA is known to be $1/e$ [14]. The slotted nonpersistent CSMA protocol provides the highest capacity among all CSMA protocols [16]; the CSMA capacity is plotted for $a = 0.05, 0.01$, and 0.001. As N increases at fixed a, the capacity of HOL, AP, RR, and RO decays very quickly below the slotted nonpersistent CSMA capacity [16] (e.g., $N = 44$ for $a = 0.001$, $N = 18$ for $a = 0.01$, $N = 10$ for $a = 0.05$) and is worse than the slotted ALOHA mode [14] for $N > 172$ if $a = 0.01$ (or $N > 34$ if $a = 0.05$). However, when a is very small, say $a = 0.001$,[4] the capacity is large (>90 percent) for $N < 110$.

In Fig. 3 we plot the capacity C versus a for various values of N. As a increases, for a small number of users ($N = 10$), the protocols studied in this section have a channel capacity which

[4] We obtain $a = 0.001$, if, for example, all users are less than 2 mi apart and transmit 1000 bit packets over a channel operating at a speed of 100 kbits/s.

is higher than all CSMA protocols for values of a not larger than 0.038, is fairly good when $a \leqslant 0.02$ ($\geqslant 83$ percent), and drops below slotted ALOHA when $a > (e - 1)/N$. When the number of users is larger, C quickly decays as a increases, e.g., if $N = 50$ for values of $a > 0.35$, the capacity even drops below that of slotted ALOHA.

C. Delay Analysis

Together with the channel capacity, the expected packet delay T is a critical performance measure. T is defined as the average time, normalized with respect to P, elapsing from the generation of a packet until the end of its transmission. The expected packet delay normalized with respect to one slot duration is denoted by D: $T = D(1 + Na)$.

We first establish a conservation law which gives us, as a main result, the expected packet delay under a broad class of protocols for all values of the number of users N. When all users have the same input rate (λ/N), the main result is that AP, RR, and RO produce the *same* expected delay. The next two sections are devoted to analytical results concerning delay under HOL and AP (the latter for $N = 2$). Following that is a section dedicated to some simulation results when the input rate varies from one user to another. Since we cannot discriminate AP, RR, and RO on the basis of average delay (they produce the same expected delay when all users have the same input rate), it is necessary to investigate the delay variance. This is the subject of the last section.

1) Conservation Law: We model our multiple access schemes as an $M/G/1$ priority queueing system with rest period[5] where the service time and the rest period have the same deterministic distribution of length one slot[6] and where each user belongs to one of N priority groups. In addition, our queueing disciplines are work-conserving (the server neither creates nor destroys work in that he never stands idle in the face of a nonempty queue, i.e., a slot is unused if and only if all users are idle—and all users must be fully served before departing), nonpreemptive, and service independent. We may then extend Kleinrock's conservation law [20] to include the case of priority queueing systems with rest period.

Theorem (Kleinrock [20]): For any $M/G/1$ system in equilibrium and any nonpreemptive work-conserving queueing discipline,

$$\sum_{i=1}^{N} \rho_i W_i = \frac{\rho}{1-\rho} W_0, \qquad \rho < 1 \tag{2}$$

where arriving customers belong to one of a set of N different priority classes, customers from priority group i arrive in a Poisson stream at rate λ_i, and each customer from this group has a mean service time $\bar{x}_i$ and a service time second moment $\overline{x_i^2}$. Customers from group i incur an average waiting time (in queue) W_i, and ρ_i, ρ, and W_0 are defined as follows:

$$\rho_i = \lambda_i \bar{x}_i \tag{3}$$

$$\rho = \sum_{i=1}^{N} \rho_i \tag{4}$$

$$W_0 = \sum_{i=1}^{N} \rho_i \frac{\overline{x_i^2}}{2\bar{x}_i} = \sum_{i=1}^{N} \lambda_i \frac{\overline{x_i^2}}{2}. \tag{5}$$

W_0 represents the expected residual life of the customer found in service upon an arrival's entry. Thus, the weighted sum of the average waiting times W_i never changes, whatever the (conservative) queueing discipline.

This result is easily extended to the case of an $M/G/1$ queueing system with rest period and any work-conserving and nonpreemptive queueing discipline. Denote by $\bar{T}_0$ the mean rest period duration and by $\overline{T_0^2}$ the rest period second moment. Then we have (see the Appendix)

$$\sum_{i=1}^{N} \rho_i W_i = \frac{\rho}{1-\rho} W_0 + \rho \frac{\overline{T_0^2}}{2\bar{T}_0}, \qquad \rho < 1 \tag{6}$$

where ρ_i, ρ, and W_0 are defined in (3), (4), and (5).

Furthermore, one can easily show that when the order of service is independent of service time, then the distribution of the total number of customers in the system is independent of the queueing discipline. The approach for showing this statement is exactly the same used in a regular $M/G/1$ priority queueing system (see [3, vol. II, p. 113]). Using Little's result (see [3, vol. I], for example), we have that $\lambda_i W_i = \bar{N}_{iq}$ = average number of customers from group i in the queue and, clearly, $\Sigma_{i=1}^{N} \bar{N}_{iq} = \bar{N}_q$ = average number of customers (of all groups) in the queue. If the order of service is also independent of service time (in particular, we have $\bar{x}_i = \bar{x}$, $\overline{x_i^2} = \overline{x^2}$ for all i and $W_0 = \lambda\overline{x^2}/2$ where $\lambda = \Sigma_{i=1}^{N} \lambda_i$), then (6) becomes

$$\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} W_i = \sum_{i=1}^{N} \frac{\bar{N}_{iq}}{\lambda} = \frac{\bar{N}_q}{\lambda} = W = \frac{\lambda\overline{x^2}}{2(1-\rho)} + \frac{\overline{T_0^2}}{2\bar{T}_0}, \quad \rho < 1 \tag{7}$$

with $\rho = \lambda\bar{x}$. The right-hand side of (7) is also equal to the average waiting time W in an $M/G/1$ queue with rest period and FCFS order of service [19], [21].

Thus, the conservation law puts a linear equality constraint on the set of average waiting times W_i; any attempt to modify the queueing discipline so as to reduce one of the W_i's forces a change in some of the other W_i's in a way which balances the result.

2) Expected Packet Delay Under AP, RR, and RO: Denote by Q_i the expected waiting time (time from a packet's generation to the beginning of its transmission) at user i, normalized with respect to a slot (of duration $[1 + Na]P$). Recall that in

[5] An $M/G/1$ queue with rest period (see [19]) is identical to an $M/G/1$ queue [3, vol. I] in all respects, except that at the end of a busy period, the server takes a rest period whose duration has an arbitrary distribution (independent of the arrival and service processes). At the end of the rest period, he starts serving the backlog accumulated during his absence. If there is no backlog at the end of the rest period, the server takes another rest period independent of the first one, etc.

[6] Since the transmission is slotted, a packet which upon generation finds the system empty (no packet waiting for transmission at any user; no packet being transmitted) must wait until the beginning of the following slot to be a candidate for transmission.

our model, all service times are constant, that is, $\overline{x^2} = (\bar{x})^2 = T_0^2 = (\bar{T}_0)^2 = [(1 + Na)P]^2$. For any protocol (HOL, AP, RR, or RO) and for any set of the input rates $\{\lambda_i\}$, this extended conservation law (7) states that

$$\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} Q_i = \frac{\rho}{2(1-\rho)} + \frac{1}{2} = \frac{1}{2(1-\rho)} \qquad \rho < 1 \tag{8}$$

where $\rho = \lambda[1 + Na]P$ is the total normalized input rate (packets/slot). Denoting by D_i the expected normalized packet delay (waiting time in queue plus transmission time) of a packet generated at user i normalized with respect to one slot (i.e., $D_i = Q_i + 1$), we then have

$$\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} D_i = \frac{1}{2(1-\rho)} + 1 \qquad \rho < 1. \tag{9}$$

Equations (8) and (9) are true regardless which of our protocols is used!

Furthermore, the total average number of packets in system $(\bar{N})$ is (by Little's result) $\bar{N} = \lambda(W + \bar{x})$ and is independent of the protocol used (see previous section) and is given by

$$\bar{N} = \frac{\rho}{2(1-\rho)} + \rho \qquad \rho < 1. \tag{10}$$

When the input rate is the same at all users ($\lambda_i = \lambda/N$ for all i), then obviously $D_i = D_j$ for all $i \neq j \in \{1, \cdots, N\}$ under AP, RR, RO (referred to as symmetric protocols). Then, from (9) we have

$$D_i = \frac{1}{2(1-\rho)} + 1 \qquad \text{for all } i. \tag{11}$$

Thus, we see that when the packet generation rate is the same at all users, then the expected packet delay is independent of which (symmetric) protocol we use. In order to compare AP, RR, and RO and decide which one is the best in terms of delay, we must compare the delay *variance* under these various protocols.

First, however, let us try to solve for the average delay for each group (D_i) in the general case, i.e., when the input rate λ_i is not the same at all queues. For HOL, a simple mean value analysis is available, the results of which we present in the next section. For AP, RR, and RO, the problem is not easy; below we present the results of an exact analysis of AP in the special case of $N = 2$ users. No analysis is currently available for RR and RO.

3) Head of the Line (HOL)–Average Packet Delay: The average delay of a packet generated at user i ($i = 1, \cdots, N$) expressed in slots is given by

$$D_i = 1 + \frac{1}{2(1-\sigma_i)(1-\sigma_{i+1})} \tag{12}$$

where

$$\sigma_i = \sum_{j=1}^{N} \lambda_j [1 + Na]P \tag{13}$$

and where we choose, without loss of generality, an external priority structure such that group $i (i = 2, \cdots, N)$ has higher priority than group $i - 1$.

Proof: One easily extends Cobham's result [22] to an $M/G/1$ system with rest period and a HOL queueing discipline, and therefore to one HOL protocol [21]. Equation (12) has also been directly derived by a different, although longer, approach [23]. One can verify that the conservation law holds by substituting D_i as given by (12) into (11).

4) Average Packet Delay Under AP($N = 2$; $\lambda_1 \neq \lambda_2$): The expected delay under AP is given by (11) when the traffic is equally distributed among all users ($\lambda_i = \lambda/N$, $i = 1, \cdots, N$). When the λ_i's are not equal, solving for delay is a more complicated problem and does not seem to result in a closed form for $N > 2$. For the case of two users accessing the channel under the AP protocol, it is shown in [21] that the expected packet delays for user 1 (D_1) and for user 2(D_2) expressed in slots are given by

$$D_1 = 1 + \frac{\rho_1}{2(1-\rho_1)} + \frac{\rho_2(1-\rho_1)^2 + \rho_1\rho_2^2}{2(1-\rho_1)(1-\rho)[(1-\rho_1)(1-\rho_2) + \rho_1\rho_2]} + \frac{1}{2}\left[\frac{\lambda_1}{\lambda} + \frac{\lambda_2(1-\rho_1) - \lambda_1\rho_2(1-2\rho_2)}{\lambda[(1-\rho_1)(1-\rho_2) + \rho_1\rho_2]}\right] \tag{14}$$

and

$$D_2 = 1 + \frac{\rho_2}{2(1-\rho_2)} + \frac{\rho_1(1-\rho_2)^2 + \rho_2\rho_1^2}{2(1-\rho_2)(1-\rho)[(1-\rho_1)(1-\rho_2) + \rho_1\rho_2]} + \frac{1}{2}\left[\frac{\lambda_2}{\lambda} + \frac{\lambda_1(1-\rho_2) - \lambda_2\rho_1(1-2\rho_1)}{\lambda[(1-\rho_1)(1-\rho_2) + \rho_1\rho_2]}\right] \tag{15}$$

where

$$\begin{cases} \rho_i = \lambda_i[1 + Na]P, & i = 1, 2 \\ \rho = \rho_1 + \rho_2. \end{cases} \tag{16}$$

In addition, we check the conservation law and find that (9) is verified, i.e.,

$$\rho_1 D_1 + \rho_2 D_2 = \frac{\rho}{2(1-\rho)} + \rho, \qquad \rho < 1.$$

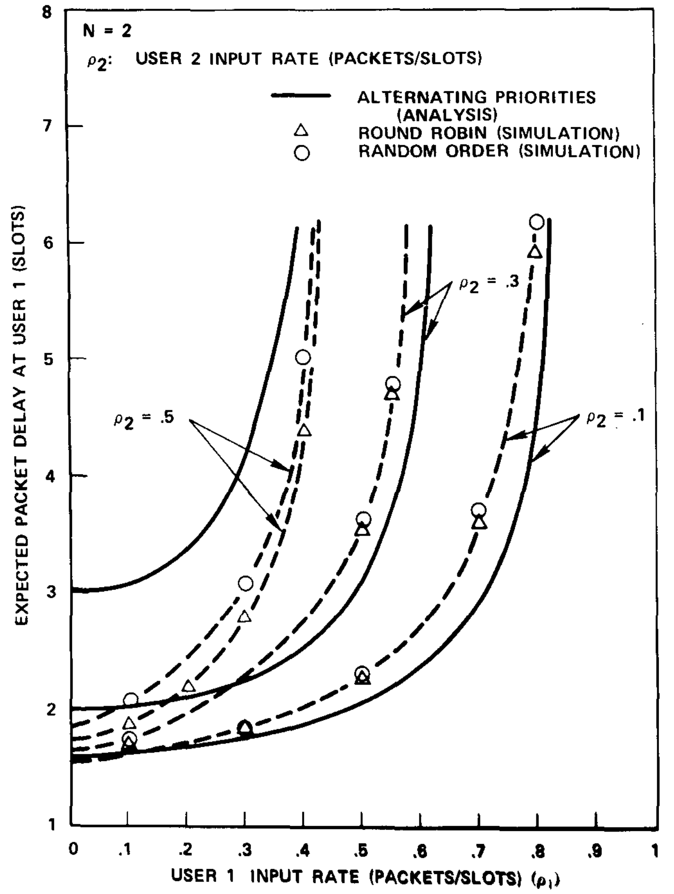


Fig. 4. AP, RR, and RO, $N = 2$ users: effect of user 2's input rate on user 1's average delay-throughput performance.

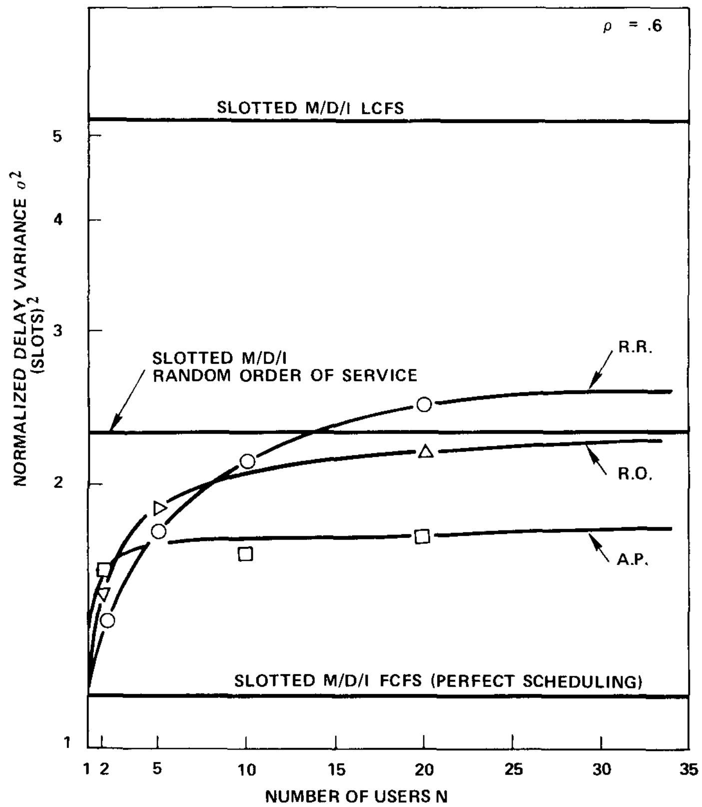


Fig. 5. AP, RR, and RO: effect of the number of users on the delay variance.

5) Comparison of AP, RR, and RO: The three protocols are equivalent in terms of average delay throughput performance when the N users have identical input rates (11). This is no longer true for asymmetric input rates. Intuitively, one expects that RR favors the packets generated at users with lower input rates. This is verified by simulation and illustrated in Fig. 4 where we take an example of $N = 2$ users. The expected packet delay D_1 at user 1 is plotted versus user 1's input rate ρ_1 (packets/slot) in Fig. 4 for various values of user 2's input rate for AP (14), RR (simulation), and RO (simulation). We observe that if $\rho_1 = \rho_2$, the three protocols provide the same delay; this we know from the previous section. When $\rho_1 < \rho_2$, RR and AP provide, respectively, the smallest and the largest expected delay at user 1; but as expected from the conservation law, the expected delay at user 2 is largest under RR and shortest under AP. When $\rho_2 > \rho_1$, the situation is, of course, reversed (Fig. 4).

6) Delay Variance Under AP, RR, and RO: Since the average packet delay is constant for these protocols in the case of identically loaded users ($\lambda_i = \lambda/N$ for all i), we are interested in comparing the higher moments of the delay distribution under AP, RR, and RO.

In Fig. 5 the delay variance, as obtained from simulation under AP, RR, and RO, is plotted versus the number of users for a normalized input rate $\rho = 0.6$ (packets/slot). These values are compared to the delay variance of three special disciplines in an $M/D/1$ slotted system: first-come first-served (FCFS), last-come first served (LCFS), and random order of service (ROS). These three systems are special cases of $M/G/1$ queues with rest period with, respectively, FCFS, LCFS, and random order of service. The second moments of waiting time for these have been derived in [21] and are shown to be related by the following equation:

$$\overline{W_{\text{FCFS}}^2} = (1-\rho)\overline{W_{\text{LCFS}}^2} = \left(1-\frac{\rho}{2}\right)\overline{W_{\text{ROS}}^2}. \qquad (17)$$

This is precisely the relationship found in [24] in the case of regular $M/G/1$ queues with no rest period. Obviously, the variance under the FCFS, ROS, and LCFS disciplines is independent of the number of users. When N increases (Fig. 5), it is expected that AP will have the smallest variance (among AP, RR, and RO), and that RR will have the largest, the difference between the two being less than 1 (slot)^2 (for $\rho = 0.6$), while the variance under RO converges to that of an $M/D/1$ slotted queue with ROS; this result was to be expected since, when N is very large, there is, with high probability, at most one packet waiting at any given user. Therefore, to randomly select which of the ready users will transmit a packet is equivalent to randomly selecting one packet among all packets present in one queue.

Thus, we may conclude that the three protocols are quite

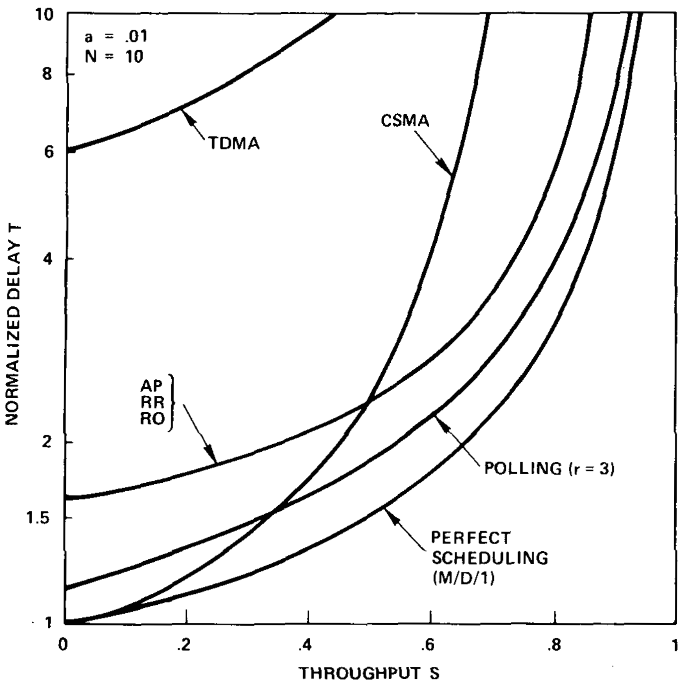


Fig. 6. AP, RR, and RO, average packet delay versus throughput: comparison to existing techniques ($N = 10, a = 0.01$).

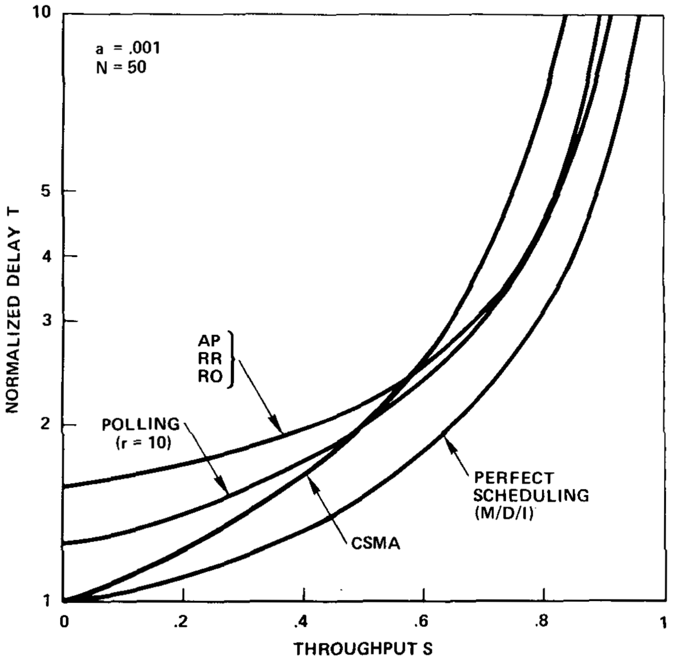


Fig. 7. AP, RR, and RO, T versus S: comparison to existing techniques ($N = 50, a = 0.001$).

equivalent in terms of mean delay-throughput performance and differ only slightly with respect to variance.

D. Discussion

Let us compare this mean delay-throughput performance to that of the best among the existing access techniques over a ground radio channel which were mentioned in the Introduction. In Figs. 6, 7, and 8 we plot the average packet delay T normalized with respect to the packet transmission time P versus the throughput for CSMA, roll call polling, and for our new schemes (AP, RR, and RO).

An analysis of (roll call) polling can be found in [12] where stationary distributions for queue lengths and waiting times are derived. These results are applied in [17] to packet radio. The expected packet delay is given by [17]

$$T = 1 + \frac{S}{2(1-S)} + \frac{a}{2}\left(1 - \frac{S}{N}\right)\left(1 + \frac{Nr}{1-S}\right) \tag{18}$$

where $r(\geqslant 3)$ represents the total time (in minislots of duration τ) spent in interrogating (polling) a user.[7] $r = 2 + (T_p/\tau)$ where T_p is the transmission time of the polling message (containing the user's identification); r is chosen to be $r = 3$.

Fig. 6 depicts an example of performance for $N = 10$ users. The performance of TDMA is plotted in the same figure. The expected delay under TDMA [16, footnote 2], [23][8] is given

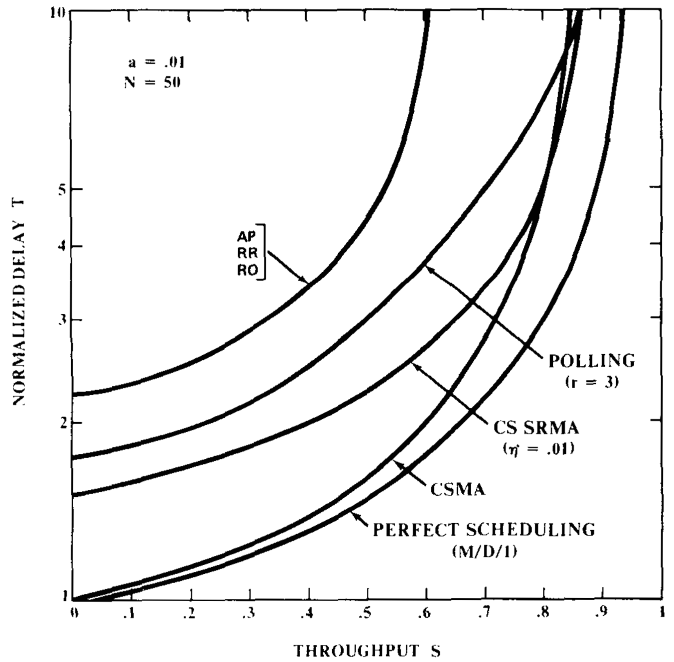


Fig. 8. AP, RR, and RO, T versus S: comparison to existing techniques ($N = 50, a = 0.01$).

by

$$T = 1 + N\left[\frac{S}{2(1-S)} + \frac{1}{2}\right].$$

For these parameters, we find that the channel capacity is equal to 0.91 [see (1)] for AP, RR, and RO, and the delay under those protocols is by far lower than with TDMA and slightly higher than with polling. Let us now compare AP, RR, and RO to CSMA. The (slotted nonpersistent) CSMA performance is

[7] It takes one minislot for the polling packet to reach the user, and the station has to wait an additional minislot (propagation time from the user to the station) before it can decide whether to allocate the channel to the polled user or poll the next user in sequence. For a detailed description of the roll call polling protocol, see [12] and [17].

[8] As a matter of fact, the expected packet delay under TDMA, as given in [28, eq. (1)] is incorrect, and should be reduced by $(N - 1)$.

plotted as predicted by the infinite population model [16]. It was shown in [25] that the performance predicted by this model is a very good approximation of the performance of $N = 10$ buffered users contending for the channel under CSMA. We note from Fig. 6 that at light traffic, CSMA provides the smallest delay. However, when S is greater than 0.5, the new schemes perform much better than CSMA.

In Fig. 7, $a = 0.001$. AP, RR, and RO produce a performance comparable to that under polling for an even larger number of users ($N = 50$). In Fig. 7 we plot the CSMA performance as predicted by the infinite population model [16]. This performance (unstable channel: no steady state over an infinite time horizon) is likely an upper bound for the steady-state performance (stable channel) for $N = 50$ users (whose performance has been studied [26], but was not available for $N = 50$ and $a = 0.001$). Under heavy traffic conditions ($S \geqslant 0.6$), the new schemes provide lower delays than CSMA, although they achieve the same channel capacity, $C \cong 0.95$.

In the last example, we choose $N = 50$ and $a = 0.01$. The steady-state performance (stable channel) of CSMA is plotted in Fig. 8, as predicted by the finite population model studied in [26]. The delay is significantly higher with the new protocols. Even at very light traffic ($S \cong 0$), the delay is 2.25 (1.5 slots times 1.5 packet transmission times since the slot size is equal to $(1 + Na)T$). The capacity for the channel is only 2/3 under the new protocols (1), while it is 0.84 for CSMA [26], greater than 0.9 for CS SRMA [17], and 1 for polling.

In summary, for a small number of users,[9] AP, RR, and RO provide a good channel capacity ($C = 1/(1 + Na) > 0.9$) and a delay-throughout performance close to that with polling (Fig. 6). When all users are very close to each other (a small, e.g., $a = 0.001$), AP, RR, and RO accept a significant number of users ($N \leqslant 50$) without performance degradation (Fig. 7); under heavy traffic conditions, they perform better than CSMA. But, as with CSMA, they require all users to be in line-of-sight and within range of each other, while polling does not have such a requirement. However, the new schemes, as in CSMA, have the advantage of not requiring control from a master user (central station), while polling does. Thus, we conclude that the new protocols are particularly suitable for multiple access from a small number of buffered users without control from a central station. When a is not too small (e.g., $a = 0.01$), the performance degrades with the number of users (Fig. 8). Indeed, in each slot, the overhead is proportional to the number of users. The protocols AP, RR, and RO can be modified to decrease this overhead somewhat [21]. In the next section, we consider a natural extension of AP which reduces the control overhead significantly.

III. MINISLOTTED ALTERNATING PRIORITIES (MSAP)

In this section, we introduce and analyze a *conflict-free* scheme referred to as minislotted alternating priorities (MSAP) which also allows buffering capabilities and *does not require control from a central station*. We solve for the average packet delay under MSAP and show that MSAP performs better than CSMA under heavy traffic conditions for all numbers of users, and performs better than roll call polling for all traffic levels and all numbers of users.

A. Protocol

As with the former schemes, we use the carrier sense capability of each user. However, we now reduce the channel time lost in control, i.e., the overhead due to carrier sensing in order to "steal" a slot "assigned" to an idle user. Without loss of generality, we assume that users grasp the channel according to a fixed order, say $1, 2, \cdots, N$. The protocol still obeys the alternating priorities rule, i.e., once a user, say user i, seizes the channel, he keeps transmitting packets until he is idle.

By carrier sensing, at most one slot later, all users detect the end of transmission of user i (absence of carrier[10]); in particular, so does the next user in sequence (user $(i \bmod N) + 1$). Then

1) either user $(i \bmod N) + 1$ starts transmission of a packet; in this case, one minislot after the beginning of his transmission, all others detect the carrier. They wait until the end of this packet's transmission and then operate as above.

2) or user $(i \bmod N) + 1$ is idle; in this case, one minislot later, all other users detect no carrier; they then know that it is the turn of the next user in sequence, i.e., user $[(i + 1) \bmod N] + 1$ and operate as above.

When all users are idle, the "turn" keeps changing at each minislot until it is the turn of a nonidle user.

In Fig. 9 we consider an example with four users. Two minislots after the end of user 3's transmission, user 1 starts transmission since he detected that user 4 was idle. He transmits three packets, followed by user 2 then user 3.

Three remarks are noteworthy.

1) There is no *positive* means for a user to know when his turn has come: he has to keep counting the idle minislots. In the previous example (see Fig. 9), users 1 and 2, respectively, count two and one minislots after the last transmission (respectively, users 3 and 1). The integrity of this sequence count is directly affected by carrier sensing errors (see footnote 3). Overlapping may then occur between the transmission from two different users, and some recovery procedure must be implemented. For these reasons, the results presented below are only an upper bound on this protocol's performance.

2) We could have chosen the random order rule or the round robin rule. The latter is suitable for unbalanced traffic, i.e., for users with a smaller input rate than others, the round robin rule (only one packet per user per cycle) provides more frequent access to the channel than the alternating priorities rule does. However, the alternating priorities rule is chosen here in order to minimize the "changeover" time between users. This changeover time, which is lost for packet transmission, is shorter with alternating priorities than it is with round robin. This overhead (one minislot per switchover from one user to another) is very small compared to that incurred with the schemes studied in Section II (for which N minislots are lost at each packet transmission time). The maximum channel

[9] In particular, when the product Na is small. A typical value is $Na \leqslant 0.1$, e.g., $N = 10, a \leqslant 0.01$ or $N = 20, a \leqslant 0.005$.

[10] All users know whose turn has come at most one minislot (of duration τ) after the end of transmission of a packet.

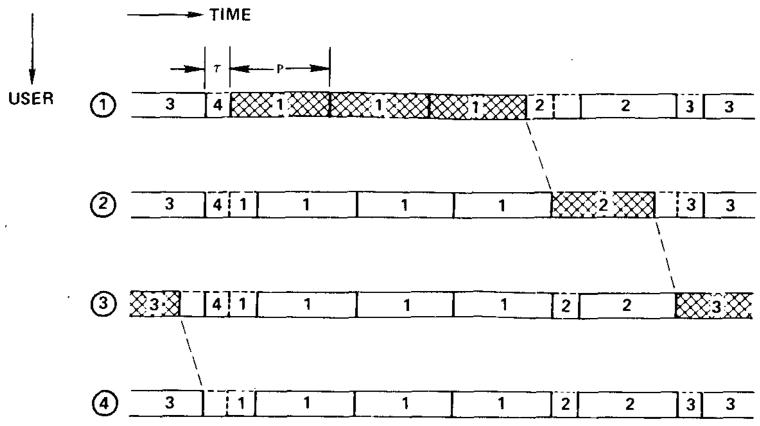


Fig. 9. Minislotted alternating priorities (MSAP), example of 4 users (cross-hatching indicates a transmission).

utilization is obtained with the alternating priorities rule which allows the system to achieve full utilization of the channel. When one queue is saturated and keeps the channel for its own use, there is no changeover, and therefore the throughput is $S = 1$ packet/packet transmission time. Thus, the *capacity of MSAP is equal to 1.*

3) In roll call polling, the channel is assigned to the users according to the same rule. The only difference is that the polling time or changeover time between two users is equal to the polling message transmission time of length $b \triangleq T_p/\tau$ minislots ($b \geqslant 1$) plus twice the propagation time between users and the station [17]. If we denote this changeover time by r, we then have for

$$\text{a) polling: } r = b + 2, \quad \text{b) MSAP: } r = 1. \tag{20}$$

Since the polling message contains the identification of the user which is polled, b will increase with N (in particular, it must grow in proportion to $\log_2 N$).[11] Also, b depends on the parameter a. If a increases, b will decrease down to a minimum of 1. From the last statement and (20), it is evident that the changeover time is much smaller with MSAP than with polling.

We see that with MSAP, the N users are really passing around a token which gives them permission to transmit. The token is silence!

B. Expected Delay

We may apply the results of Konheim and Meister [12] for roll call polling to MSAP by choosing the "polling" time r equal to 1 in (18). This equation gives the expected normalized delay in ground radio polling [17]. Then, with MSAP, the expected normalized packet delay is given by

$$T = 1 + \frac{S}{2(1-S)} + \frac{a}{2}\left(1 - \frac{S}{N}\right)\left(1 + \frac{N}{1-S}\right). \tag{21}$$

In particular, at very light traffic ($S \approx 0$), the ratio of the expected packet delay under polling to the expected packet delay under MSAP increases as $\log_2 N$ as $N \to \infty$.

C. Discussion

The performance of MSAP (expected packet delay normalized with respect to P versus the throughput) is compared to that of (roll call) polling and CSMA in Figs. 10 and 11 for $N = 50$ and $N = 100$, respectively, for a value of $a = 0.01$. The CSMA performance is plotted as obtained in [26].

In comparing MSAP to CSMA, we note from Figs. 10 and 11 that at light traffic, the larger N is, the better is the CSMA performance as compared to the performance of MSAP. In particular, at very light traffic ($S \approx 0$), the expected packet delay under MSAP [see (21)] is $[1 + a/2\ (1 + N)]$, whereas for CSMA, it is 1.

However, under heavy traffic conditions, MSAP always performs better then CSMA. For $N = 50$, the crossover point occurs at an input rate S_0 equal to 0.7. The variation of the "crossover" value of the input rate (S_0) (if $S > S_0$, then $T_{MSAP} < T_{CSMA}$) as a function of the number of users is illustrated in Fig. 12. When N grows large, S_0 reaches 0.85 which is the channel capacity of CSMA. Indeed, for $S > 0.85$, the channel is saturated under CSMA (infinite delays), while finite delays are observed under MSAP (which has a capacity of 1). The contour S_0 versus N is obtained by comparing the MSAP delay performance to that of CSMA as obtained, respectively, in [26] (solid line) and in [16] (dashed lines). While the first CSMA performance [26] is optimistic for $N \leqslant 20$ (since the model that predicts this performance assumes no buffering capabilities at the users), the second CSMA performance [16] is pessimistic in the range $20 \leqslant N \leqslant 1000$.

In summary, the delay-throughput performance of MSAP

[11] An alternative method of polling, called "hub go-ahead" polling or simply "hub" polling, used in wire communications, is not readily applicable to our radio system, but is advantageous on long line communications: the computer addresses only the user at the end of the line, say A. Assume A is idle. Then A sends the poll to his neighbor B. Assume B is ready. He sends his packets. The station receives them and then resumes polling at B's neighbor, say user C. Thus, the main advantage lies in lessening the number of line turnarounds [11].

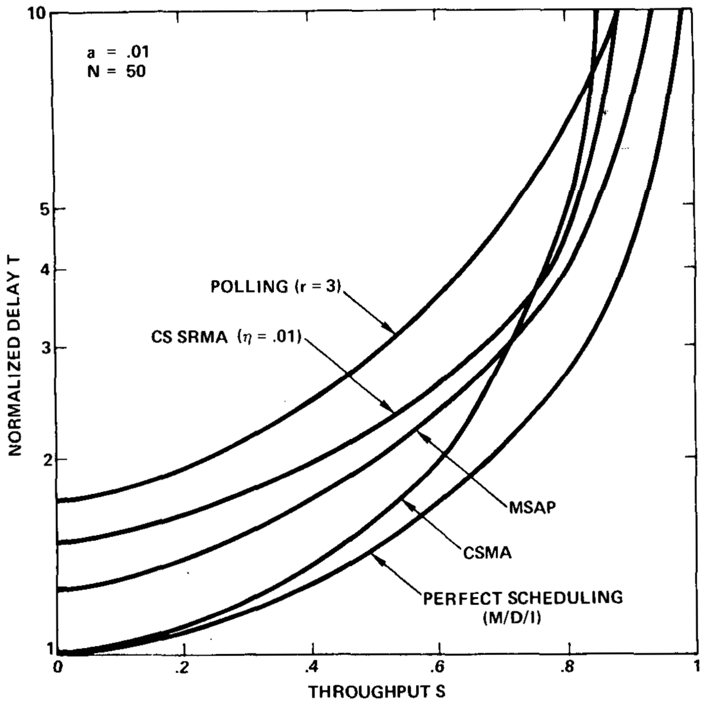


Fig. 10. MSAP, T versus S: comparison to existing techniques ($N = 50$).

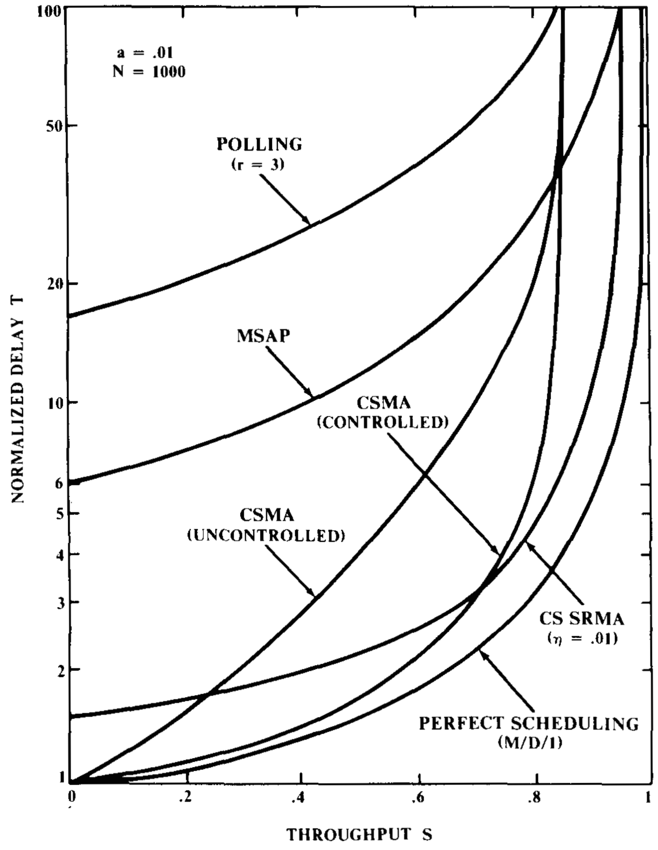


Fig. 11. MSAP, T versus S: comparison to existing techniques ($N = 1000, a = 0.01$).

exceeds that of polling for all values of N and a. Under heavy traffic conditions, MSAP provides the best performance of all techniques discussed here for all values of Na.[12] Under light traffic conditions, random access techniques outperform MSAP (by far if N is large). However, MSAP is more suitable than CSMA for a small number of (possibly) buffered users, since then even when the traffic is very light, the expected delay under MSAP is only slightly larger than that observed with CSMA (at $S = 0$, the difference is equal to $Na/2$).

So far, in our comparison between MSAP and CSMA, we have considered a small value of the parameter a ($a \leqslant 0.01$). As a increases, the performance of CSMA–and therefore that of CS SRMA–is known to decay below that of slotted ALOHA [16]. How does MSAP perform compared to CSMA, TDMA, and slotted ALOHA when a increases? When a is small ($a \leqslant 0.1$), we know that MSAP performs better than TDMA even for a small number of users [see (8) and (10)]. This is no longer true when a is large ($a > 0.5$) and N is small.

To complete this discussion, let us fix the input rate S and let us consider the regions of the $N \times a$ plane shown in Figs. 13 and 14 in which either TDMA (19) or slotted ALOHA [14], [15] or (slotted nonpersistent) CSMA[13] or MSAP (21) provides the lowest expected delay T. When the traffic is light (Fig. 13: $S = 0.3$), four regions are shown in which each of the access schemes, respectively, CSMA, ALOHA (N and a large), TDMA (N very small and a large), and MSAP produce the lowest delay. As might have been expected for a large population of users, random access techniques perform better, and when a increases, CSMA's performance decays below that of ALOHA. It is clear also from Fig. 13 that for all values of a (<0.6), MSAP performs the best when N is not too large. In addition, when a is small ($a = 10^{-3}$), even with a significant number of users ($N < 100$), MSAP provides the lowest delay.

Under heavier traffic conditions (Fig. 14: $S = 0.6$), the ALOHA region disappears since the maximum achievable throughput under slotted ALOHA is $S = 1/e$. When a is large, TDMA is the best scheme. It is interesting to note that the bound on a beyond which TDMA performs better is $a = 1$ for most values of N ($N > 10$), and that MSAP performs better than CSMA with a significantly larger number of users than under light traffic conditions. In particular, observe that for $a > 0.1$, MSAP is always better than CSMA since one cannot achieve a throughput of $S = 0.6$ with CSMA if $a > 0.1$. Further, for all traffic levels (Figs. 13 and 14) and for all values of N, MSAP produces a lower delay than CSMA when a goes to zero (although the difference in delay performance between the two also goes to zero).

On the boundary line between two regions, two schemes

[12] However, the reservation scheme introduced in [18] performs better than MSAP for large values of a.

[13] For all values of N, the performance of CSMA used in our comparison has been chosen as that predicted by the infinite population model studied in [16]. As mentioned earlier, this is only a lower bound on the performance predicted by more realistic models. For example, 1) when $N \leqslant 5$ [25], 2) when $20 \leqslant N < 1000$. (However, it is a tight bound when $N \approx 1000$) [26].

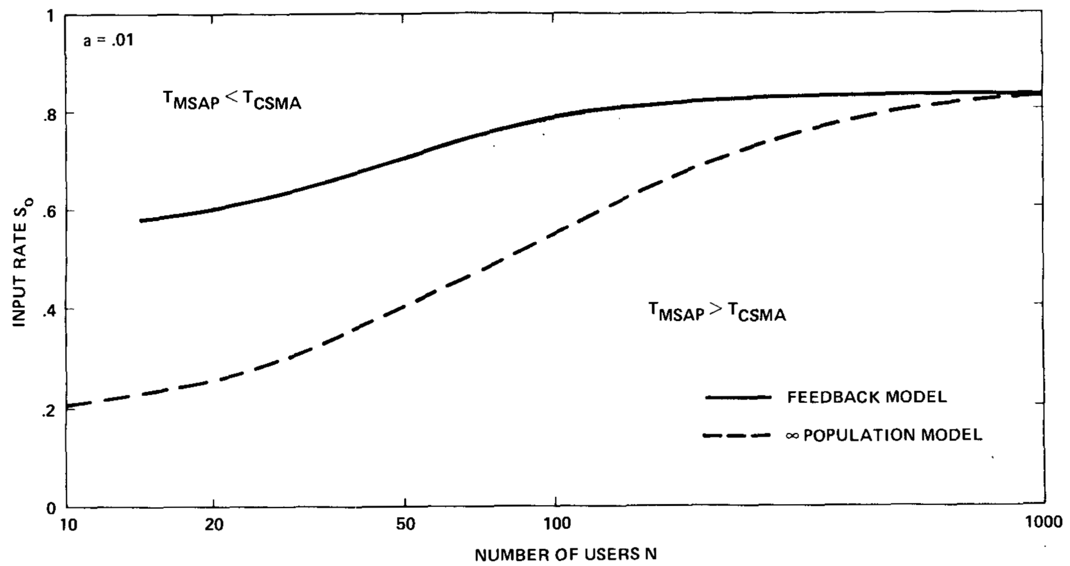


Fig. 12. MSAP and CSMA: S_0 versus N.

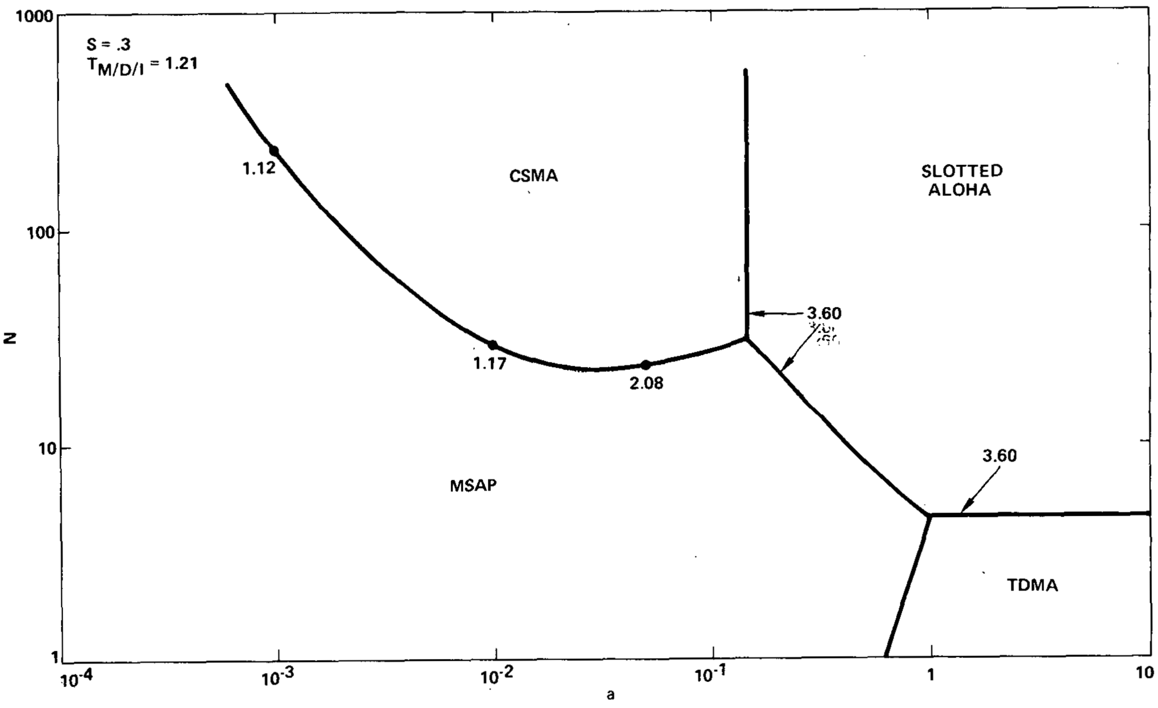


Fig. 13. N versus a (S = 0.3).

produce the same delay performance. The ratio of this delay performance over ($M/D/1$) perfect scheduling performance (for $S = 0.3$, $T_{M/D/1} = 1.21$, and for $S = 0.5$, $T_{M/D/1} = 1.75$) is represented at various points by the numerical labels in the figures along the boundary. As an example, consider Fig. 14 ($S = 0.6$). For $a < 1$, two regions are shown. Above the contour (N large, $a < 0.1$), CSMA provides the lowest delay. Below the contour, MSAP performs better. For $N \cong 170$ and $a = 0.01$, both schemes produce the same expected delay which is 2.18 times the expected delay in an $M/D/1$ queue.

Finally, in Fig. 15 the contours $T_{MSAP} = \alpha T_{CSMA}$ are plotted in the $N \times a$ plane for $S = 0.5$. On such a contour, the ratio α of the expected delay under MSAP to the expected delay under CSMA as obtained in [16] is constant. Such a contour defines two regions in the plane, below which ($T_{MSAP} < \alpha T_{CSMA}$) one can, and above which ($T_{MSAP} > \alpha T_{CSMA}$) one cannot achieve a delay lower with MSAP than α times that produced by CSMA. As might be expected, when a is small ($a < 5 \times 10^{-3}$), one cannot achieve with MSAP a significantly lower delay than with CSMA. Observe that the contour $\alpha = 1$ and the axis $a = 0$ (asymptotic to the contour $\alpha = 1$) represent the boundary on which both schemes produce the same delay. When $a = 0$, $T_{MSAP} = T_{CSMA} = T_{M/D/1} = 1.5$. For a large value of N (e.g., $N = 250$), as a increases, α first decreases (from 1: $a = 0$; to 0.92: $a \cong 5.10^{-5}$), then increases (up to 2: $a = 2.10^{-2}$), and finally decreases to zero ($a = 0.165$).

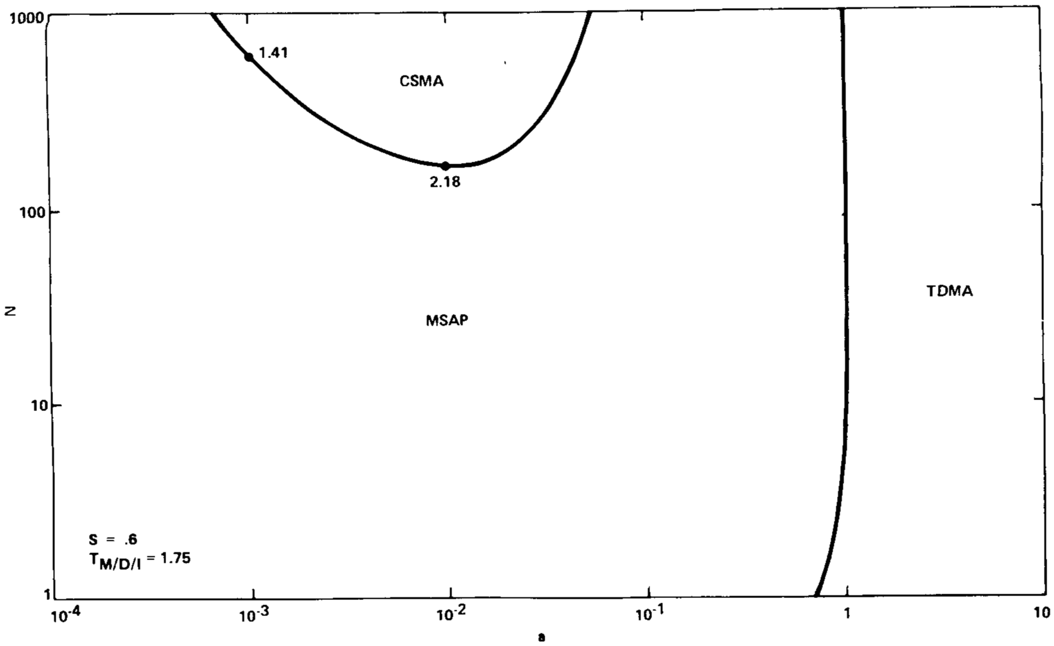


Fig. 14. N versus a (S = 0.6).

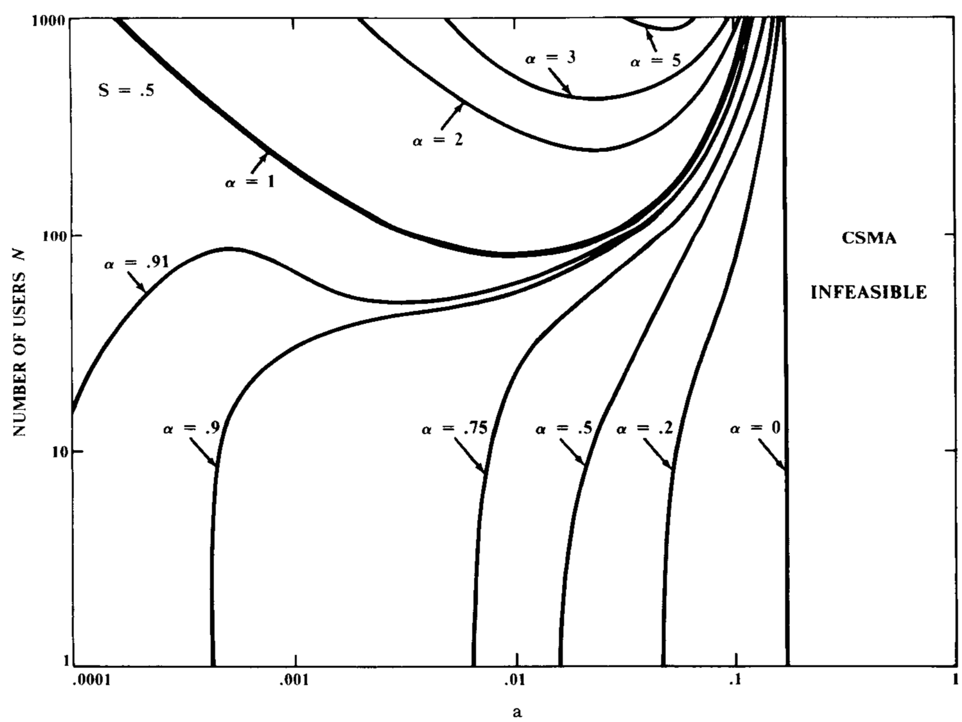


Fig. 15. CSMA versus MSAP: contours $T_{MSAP} = \alpha T_{CSMA}$($S$ = 0.5).

Thus, when a is large, MSAP produces a significantly lower delay than does CSMA. For $a > 0.165$, a throughput of $S = 0.5$ cannot be achieved under CSMA (the contour $\alpha = 0$ (i.e., $a = 0.165$) is asymptotic to all α contours and corresponds to infinite delay with CSMA). Finally, observe that for $N \leqslant 80$ (and S = 0.5: Fig. 15), MSAP provides a lower delay than CSMA for all a ($\alpha < 1$ for all a).

IV. SUMMARY

When we have a small number of users requiring buffer space for more than one packet, random access techniques (e.g., CSMA) are not necessarily optimal for multiple access to a single ground radio channel. AP, RR, and RO were introduced as new *conflict-free* multiplexing techniques suitable for a small number of (possibly) buffered users (Section II). As with random access schemes, these new schemes do not require control from a central station. However, they are sensitive to carrier sensing errors. When the number of users is small (e.g., $N < 20$ if $a = 0.01$), the channel capacity achieved under the new protocols is larger than that achieved under CSMA (Fig. 2) and they produce a delay performance comparable to that of roll-call polling [12] and better than CSMA under heavy traffic conditions (Figs. 6 and 7). A restrictive assumption for AP, RR, and RO is that all users must be in LOS. Furthermore, the performance of AP, RR, and RO degrades badly as N increases (Fig. 8).

To reduce this degradation, we considered a natural extension of AP, called MSAP (Section III), which is also conflict-free, reduces the overhead, and provides very good performance under heavy traffic conditions. MSAP achieves a capacity of 1, and was shown to perform better than roll-call polling (Fig. 10). Compared to random access schemes, MSAP is less sensitive than CSMA to an increase in the ratio of the propagation time to the packet transmission time (Fig. 15); however, a degradation of the delay was observed at light traffic as N increases (Fig. 11). As with CSMA, MSAP assumes that all users are in LOS, while polling does not require such an assumption, and MSAP is sensitive to carrier sensing errors. However, MSAP offers the advantage of not requiring control from a central station (so too, with CSMA), while polling does.

In conclusion, MSAP is one of the few schemes known [18] which, under heavy traffic conditions, performs as well as $M/D/1$ (perfect scheduling) to within a multiplicative constant (21). MSAP, which was introduced for distributed multiple access to a ground radio channel with a small number of buffered users, handles a larger number of users (e.g., $N = 50$ if $a = 0.01$) without serious degradation.

APPENDIX

PROOF OF THE CONSERVATION LAW FOR $M/G/1$ WITH REST PERIOD

The proof follows the argument by Kleinrock almost exactly [3], [20]. Let $U(t)$ denote the total unfinished work present in the system at time t. This is the time required to empty the system of all customers present at t if no new customers are allowed to enter the system after t. $U(t)$ increases by an amount x_i (the service time) when a customer of class i arrives. During the rest period $U(t)$ remains constant between arrivals. When the server ends its rest period, $U(t)$ decreases at *unit rate as long as $U(t)$ is positive*. Once having reached zero, it remains there until the next arrival (at which instant it jumps up by a service time). Observe that $U(t)$ is not affected by the queueing discipline used as long as it is work conserving.

For a regular (no rest period) $M/G/1$ priority queueing system, we have (see [3, eq. (3.17)])

$$\bar{U} = W_0 + \sum_{i=1}^{N} \rho_i W_i \tag{A.1}$$

where $\bar{U}$ is the expected unfinished work and ρ_i and W_0 are given by (3) and (5).

As a matter of fact, (A.1) holds for any conservative and nonpreemptive $G/G/1$ queueing system such that the queueing discipline is independent of any measure of the service time. No Poisson assumption is required for the arrival process (see [28]). By using exactly the same argument [3], [28], we may extend the result (A.1) to the case of a conservative, nonpreemptive, and service-independent queueing discipline in a $G/G/1$ queue with rest period. However, let us restrict ourselves to the $M/G/1$ queue with rest period.

Since $\bar{U}$ is independent of the order of service, we may as well calculate $\bar{U}$ for a first-come first-served discipline (FCFS).

For an arrival occurring at time t, the following is true.

1) If the arrival occurs during a period when the server is busy (with probability ρ[14]), then

$$W_{\text{FCFS}}(t) = U(t).$$

Indeed, if service is given in order of arrival, the time a customer has to wait for service if he arrived at time t, $W_{\text{FCFS}}(t)$ is precisely the backlog of work at this instant, i.e., $U(t)$.

2) If the arrival occurs during a rest period (with probability $(1 - \rho)$), then

$$W_{\text{FCFS}}(t) = U(t) + \theta_0$$

where θ_0 is the residual life of the rest period upon arrival.

Taking expectations, we may write

$$\bar{W}_{\text{FCFS}} = \bar{U} + (1 - \rho)\tau_0 \tag{A.2}$$

where τ_0 is the expected residual life of the rest period: $\tau_0 = \overline{T_0^2}/2\bar{T}_0$. Using the value of $\bar{U}$ from (A.2) in (A.1), we have

$$\bar{W}_{\text{FCFS}} - (1 - \rho)\tau_0 = W_0 + \sum_{i=1}^{N} \rho_i W_i. \tag{A.3}$$

Equation (A.3) is true regardless of the order of service. In particular, if the discipline is FCFS, $W_i = \bar{W}_{\text{FCFS}}$ for all i, we then obtain from (A.3)

$$\bar{W}_{\text{FCFS}} = \frac{W_0}{1 - \rho} + \tau_0. \tag{A.4}$$

Substituting the value of $\bar{W}_{\text{FCFS}}$ as given by (A.4) into (A.3), we have the conservation law given in (6).

The same argument may be applied to conservative, nonpreemptive, and service-independent queueing disciplines in $M/G/1$ queueing systems with an initial setup time. In such queueing systems, [19], [29], there is no rest period (the idle period ends with the arrival of the first customer). However, there is a setup time y associated with the beginning of a busy period. y is drawn from an arbitrary distribution independent of the arrival and service processes, with first and second moments $\bar{y}$ and $\overline{y^2}$. One can easily show that (A.1) holds for such a queueing system (with a conservative, nonpremptive, and service-independent queueing discipline), and this yields the following conservation law (see [21]):

$$\sum_{i=1}^{N} \rho_i W_i = \frac{\rho W_0}{1 - \rho} + \rho \frac{2\bar{y} + \lambda \overline{y^2}}{2(1 + \lambda \bar{y})}, \qquad \rho < 1.$$

ACKNOWLEDGMENT

The authors take great pleasure in acknowledging the assistance of C. W. Tseng who created Figs. 13 and 14.

[14] It is shown in [19] and [21] that P [server is busy] = ρ in an FCFS $M/G/1$ queueing system with rest period.

REFERENCES

[1] P. E. Jackson and C. D. Stubbs, "A study of multiaccess computer communications," in *Spring Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 34, 1969, pp. 491–504.

[2] L. G. Roberts, "Data by the packet," *IEEE Spectrum*, vol. 11, pp. 46–51, Feb. 1974.

[3] L. Kleinrock, *Queueing Systems, Vol. I, Theory* and *Vol II, Computer Applications*. New York: Wiley-Interscience, 1975 and 1976.

[4] N. Abramson, "The Aloha system—Another alternative for computer communications," in *Fall Joint Comput. Conf., AFIPS Conf. Proc.*, vol. 37, 1970, pp. 281–285.

[5] R. E. Kahn, "The organization of computer resources into a packet radio network," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44, 1975, pp. 177–186.

[6] N. Abramson, "Packet switching with satellites," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42, 1973, pp. 695–702.

[7] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 42, 1973, pp. 711–716.

[8] S. S. Lam, "Packet switching in a multi-access broadcast channel with application to satellite communication in a computer network," School Eng. and Appl. Sci., Univ. California, Los Angeles, UCLA-ENG-7429, Apr. 1974.

[9] H. Frank and W. Chou, "Topological optimization of computer networks," *Proc. IEEE*, vol. 60, pp. 1385–1396, Nov. 1972.

[10] M. Gerla, W. Chou, and H. Frank, "Computational considerations and routing problems for large computer communication networks," in *Proc. Nat. Telecommun. Conf.*, Atlanta, GA, Nov. 1973, pp. 2B-1–2B-5.

[11] J. Martin, *Teleprocessing Network Organization*. Englewood Cliffs, NJ: Prentice-Hall, 1970.

[12] A. G. Konheim and B. Meister, "Waiting lines and times in a system with polling," *J. Ass. Comput. Mach.*, vol. 21, pp. 470–490, July 1974.

[13] L. G. Roberts, "ALOHA packet system with and without slots and capture," ARPA Network Inform. Cen., Stanford Res. Inst., Menlo Park, CA, ASS Note 8 (NIC 11290), June 1972.

[14] L. Kleinrock and S. S. Lam, "Packet switching in a multi-access broadcast channel: Performance evaluation," *IEEE Trans. Commun.*, vol. COM-23, pp. 410–423, Apr. 1975.

[15] F. A. Tobagi, "Random access techniques for data transmission over packet switched radio networks," School Eng. Appl. Sci., Univ. California, Los Angeles, UCLA-ENG-7499, Dec. 1974.

[16] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[17] F.A.Tobagi and L. Kleinrock, "Packet switching in radio channels: Part III—Polling and (dynamic) split-channel reservation multiple access," *IEEE Trans. Commun.*, vol. COM-24, pp. 832–845, Aug. 1976.

[18] L. Kleinrock, "Performance of distributed multi-access computer-communication systems," in *IFIP Congr. '77 Proc.*, Toronto, Ont., Canada, Aug. 1977, pp. 547–552.

[19] L. W. Miller, "Alternating priorities in multi-class queues," Ph.D. dissertation, Cornell Univ., Ithaca, NY, 1964.

[20] L. Kleinrock, "A conservation law for a wide class of queueing disciplines," *Naval Res. Logistics Quart.*, vol. 12, pp. 181–192, 1965.

[21] M. Scholl, "Multiplexing techniques for data transmission over packet switched systems," School Eng. Appl. Sci., Univ. California, Los Angeles, UCLA-ENG-76123, Dec. 1976.

[22] A. Cobham, "Priority assignment in waiting line problems," *Oper. Res.*, vol. 2, pp. 70–76, 1954.

[23] J. Spragins, "Loop transmission systems—Mean value analysis," *IEEE Trans. Commun.*, vol. COM-20, pp. 592–602, June 1972.

[24] L. Takács, "Delay distributions for one line with Poisson input, general holding times, and various orders of service," *Bell Syst. Tech. J.*, pp. 487–503, Mar. 1963.

[25] F. A. Tobagi and L. Kleinrock, "On the analysis and simulation of buffered packet radio systems" in *Proc. 9th Hawaii Int. Conf. Syst. Sci.*, Univ. Hawaii, Honolulu, Jan. 1976, pp. 42–45.

[26] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-25, pp. 1103–1119, Oct. 1977.

[27] D. R. Cox, *Renewal Theory (Monographs on Applied Probability and Statistics)*. London: Chapman and Hall, 1962.

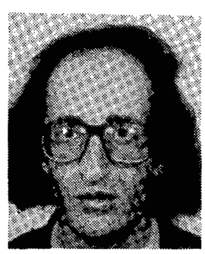
[28] L. Schrage, "An alternative proof of a conservation law for the queue $G/G/1$," *Oper. Res.*, vol. 18, pp. 185–187, 1970.

[29] P. D. Welch, "On a generalized $M/G/1$ queueing process in which the first customer of each busy period receives exceptional service," *Oper. Res.*, vol. 12, pp. 736–752, 1964.

★

Leonard Kleinrock (S'55–M'64–SM'71–F'73), for a photograph and biography, see p. 574 of the April 1980 issue of this TRANSACTIONS.

★



Michel O. Scholl received the Diplome d'Ingénieur civil from ENST, Paris, France, in 1966, the Docteur-Ingénieur degree in computer science from Toulouse University, Toulouse, France, in 1971, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1976.

From 1969 to 1972 he was with IRIA, Le Chesnay, France, where he successively worked in research projects on computer-assisted instruction and information retrieval. From 1973 to 1976 he was a Post Graduate Research Engineer in the Department of Computer Science, UCLA. His research activity was within the area of modeling and performance evaluation of computer communication systems. Since 1977 he has again been associated with IRIA. His current research interests include modeling and performance evaluation of database management systems and problems related to database machines.