A Carrier Sense Multiple Access Protocol for Local Networks

Simon S. Lam

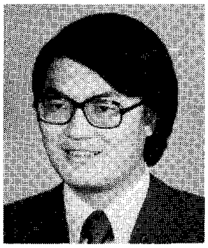
Department of Computer Sciences, University of Texas at Austin, Austin, Texas 78712, USA

A consequence of bursty traffic in computer communications is that among a large population of network users, at any one time only a small number of them have data to send (ready users). In this environment, the performance of an access protocol for a broadcast network depends mainly upon how quickly one of the ready users can be identified and given sole access to the shared channel. The relative merits of the access protocols of polling, probing and carrier sense multiple access (CSMA) with respect to this channel assignment delay in local networks are considered. A central controller is needed for polling and probing while CSMA employs distributed control. A specific CSMA protocol is defined which requires that "collisions" in the channel be detected and that the users involved in a collision abort their transmissions quickly. In addition, it is assumed that the contention algorithm is adaptive and gives rise to a stable channel. An analytic model is developed. Our main result is the moment generating function of the distributed queue size (number of ready users). Mean value formulas for message delay and channel assignment delay are also derived. These results on queue size and delay are the major contribution of this paper, since they are not available in prior CSMA models in closed analytical form. Numerical results are given to illustrate the performance of the CSMA protocol. When the channel utilization is light to moderate, the mean channel assignment delay of the CSMA protocol is significantly less than that of both polling and probing; consequently, the mean message delay is much smaller. It is also shown that when queueing of messages is permitted at individual users, the maximum channel throughput of CSMA approaches unity in the limit of very long queues. Finally, simulation results of several adaptive control algorithms are presented. The accuracy of our analytic formulas was carefully studied and found to be very good in all cases considered.

Keywords: Local Networks, broadcast channel, multiple access protocols, distributed queueing, performance analysis, queueing theory.

1. Introduction

Multipoint networks have been widely used in local networking for the interconnection of terminals to a central site: either a central computing facility or a gateway to a resource sharing computer network. The terminals are typically unintelligent and access to the shared data path (channel) is managed by the central site using a polling protocol [1]. With increasing interest in local networking and the availability of inexpensive microprocessors, other interconnection topologies, transmission media and access protocols have been proposed and investigated. They include loop networks with centralized control [2] or distributed control [3], a digital cable network using time-division multiple access [4], the ALOHANET [5] and the Packet Radio Network [6]; the last two pioneered the use of radio channels and contention protocols for multiple access. Recently, considerable interest has been revived in multipoint cable networks



Simon S. Lam received the B.S.E.E. degree (with Distinction) from Washington State University, Pullman, Washington, in 1969, and the M.S. and Ph.D. degrees in engineering from the University of California at Los Angeles, in 1970 and 1974, respectively. From 1972 to 1974, he was with the ARPA Network project at UCLA and did research on satellite packet communication. From 1974 to 1977, he was a research staff member with the IBM Thomas J. Watson Research Center, Yorktown Heights, New York where he worked on performance analysis problems of packet switching networks, SNA and satellite networks. Since September 1977, he has been with the University of Texas at Austin where he is now Associate Professor of Computer Sciences. His current research interests include computer systems modeling and analysis, computer-communication networks and protocols. At the University of California at Los Angeles, he held a Phi Kappa Phi Fellowship from 1969 to 1970, and a Chancellor's Teaching Fellowship from 1969 to 1973. In 1975, he received the Leonard G. Abraham Award for the best paper of the year in the field of Communication Systems published in IEEE Transactions on Communications. Simon is a member of Tau Beta Pi, Sigma Tau, Phi Kappa Phi, Pi Mu Epsilon, the Association for Computing Machinery, and IEEE.

* This work was supported by National Science Foundation under Grant No. ENG78-01803.

© North-Holland Publishing Company
Computer Networks 4 (1980) 21–32

The author is pleased to acknowledge the programming assistance of Luke Lien. He is also indebted to an anonymous referee for his helpful comments.

(based upon CATV technology) employing a variety of multiple access protocols [7–12].

The multiple access problem in multipoint networks is addressed in this paper. A multipoint cable network such as those in [8–11] can be viewed upon as a broadcast channel shared by a population of distributed users. Suppose there are N users. There are two problems to be addressed by the access protocol: (1) among the N users, identify those with data who desire access to the channel, the *ready users*, and (2) assign channel access to exactly one of the ready users if at least one exists. We recognize the second problem to be a mutual exclusion problem with the broadcast channel being the shared resource [13]. The difficulty here is that the broadcast channel is also the only means of communication for coordinating the distributed users.

We can think of ready users as customers in a "distributed queue" waiting to use the broadcast channel. A consequence of the conservation law in queueing theory [14] is that the average message delay performance of an access protocol is independent of the order of service but depends mainly upon the amount of time wasted for assigning channel access. Thus, when access protocols are compared solely on the basis of average message delay performance for a given channel throughput level, the above two problems reduce to just the following: whenever the channel is free and there are one or more ready users, how quickly can channel access be assigned to a ready user? The amount of time needed will be referred to as the *channel assignment delay*.

Two major categories of multiple access protocols may be used: polling and contention protocols [15]. Polling protocols require a central controller. On the other hand, with contention protocols each network user makes his own decision according to an algorithm which is driven by observable outcomes in the broadcast channel. We shall consider multipoint networks that have short propagation delays between users relative to the transmission time of a message. In a short propagation delay environment, carrier sense multiple access (CSMA) protocols may be used. Under CSMA protocols, an individual user must sense the channel for the presence of any transmission in progress and is allowed to contend for the channel only if the channel is sensed idle. With carrier sensing, the probability of success of a transmitted message is much higher than that for pure random access i.e. ALOHA protocols [14–20]. However, "collisions" of transmissions by different users cannot be avoided entirely since the channel propagation delay between any two users is nonzero.

CSMA protocols have been studied extensively in the past within a packet radio network environment by Kleinrock and Tobagi [16,17] and later by Hansen and Schwartz [18]. Analytic results in these references are mainly concerned with the maximum channel "throughput" achievable by various protocols. Characterization of the number of ready users and message delay is limited to approximate numerical solutions or simulation results.

1.1. Summary of this paper

The main contribution of this paper is an accurate analytic model of a CSMA protocol. The protocol is defined in Section 2. In Section 3, the performance of various multiple access strategies under two extreme traffic conditions are compared. The figure of merit used is the mean value of channel assignment delay introduced earlier. The CSMA protocol is shown to possess some desirable performance characteristics. In Section 4, our CSMA analysis is presented. The moment generating function of the number of ready users is obtained. Formulas for the average message delay and average channel assignment delay are also derived. In Section 5, numerical results are plotted to illustrate the performance of the CSMA protocol, which is also compared with the performance of polling. A key assumption in our analysis is the presence of an effective adaptive control algorithm for contention. In section 6, several simple yet practical algorithms are given as well as simulation results of their performance. The accuracy of our analytic formulas is examined in detail. We conclude in the final section by discussing possible extensions and improvements of this work.

2. The CSMA Protocol

The main difference between the CSMA protocol studied in this paper and the p-persistent CSMA protocol of Kleinrock and Tobagi [16,17] is as follows. We assume here that collisions in the channel are detected and that users involved in a collision abort their transmissions immediately upon detecting the collision. Mechanisms for detecting collisions and aborting collided transmissions have been implemented in several multipoint cable networks [8,9, 11]. However, it appears to be much more difficult to

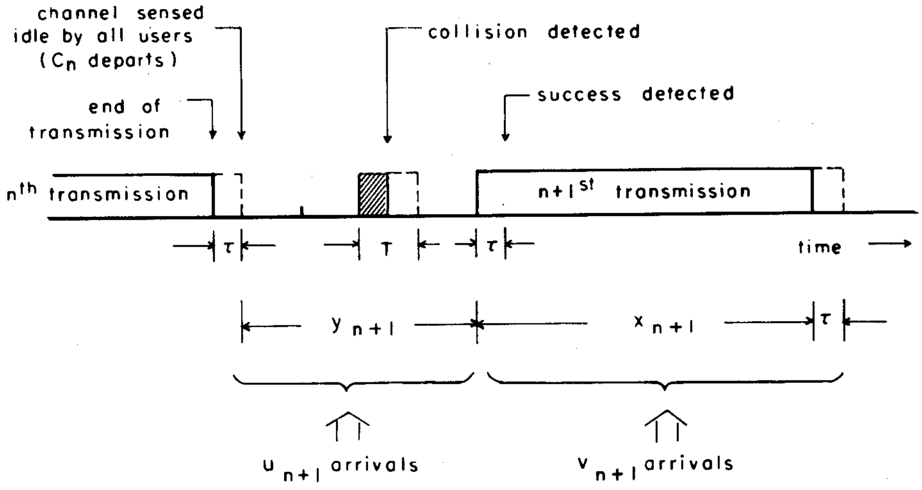


Fig. 1. A snapshot of the broadcast channel.

implement a "collision abort" capability in the radio environment of interest in [16,17].

Like the p-persistent protocol in [16,17], network users are assumed to be time synchronized so that following each successful transmission, the channel is slotted in time. (See Fig. 1.) Users can start transmissions only at the beginning of a time slot. Let τ be the amount of time from the start of transmission by one user to when all users sense the presence of this transmission. It is equal to the maximum propagation delay between two users in the network plus carrier detection time. (The latter depends upon the modulation technique and channel bandwidth. It was considered to be negligible relative to the propagation delay in [17].) In order to implement the collision abort capability described above, the minimum duration of a time slot is $T = 2\tau$, so that within a time slot if a collision is detected and the collided transmissions are aborted immediately, the channel will be free of any transmissions at the beginning of the next time slot.

The slotted channel assumption is made to simplify our analysis. The practical problem of time synchronizing all users in the network is a classical one and beyond the scope of this paper. (See [21].) It is reasonable to say that slotting can be accomplished fairly easily if the slot duration T is made much larger than 2τ; in other words, large guard times are used. In a real system, either a slotted or unslotted channel may be implemented.

Our CSMA protocol is defined by the following two possible courses of action for ready users:

(P1) Following a successful transmission, each ready user transmits with probability 1 into the next time slot.

(P2) Upon detection of a collision, each ready user uses an adaptive algorithm for selecting its transmission probability (<1) in the next time slot.

It should be clear at this point that we have effectively reduced the contention problem in CSMA to a slotted ALOHA problem. Slotted ALOHA has been studied extensively in the past [19,20,22–27], from which we learned that to prevent channel saturation (with zero probability of a successful transmission), the transmission probability of each ready user must be adaptively adjusted. Various control strategies have been proposed and studied. Experimental results have shown that a slotted ALOHA channel can be adaptively controlled to yield an equilibrium throughput rate S close to the theoretical limit of 1/e (=0.368) for a large population of users [23–26]. With an asymmetric strategy, the achievable S will be even higher [27].

3. A Comparison of Multiple Access Strategies

Using the mean value of channel assignment delay as a figure of merit, we shall examine in this section the performance of various multiple access strategies under two extreme traffic conditions. We shall first consider strategies using a central controller for "passive" users, and then strategies using distributed control for "active" users.

When a central controller is used, users are passive in the sense that they transmit only in response to a query from the central controller. In conventional polling protocols, the users are queried one by one, which is just a linear search procedure for identifying the ready users. To find out the status of all users (ready or not) in the population, the number of queries needed is N, which is just the number of users. This overhead is independent of the number of ready users actually present. Consider now two extreme traffic cnditions. First, suppose only one of the N users is ready (traffic condition 1). The mean number of queries needed to find the ready user is $(N+1)/2$, assuming all users equally likely to be the ready user. The other extreme case is when all N users are ready (traffic condition 2). In this case, the number of queries needed to find a ready user (anyone) is 1.

The method of probing i.e. polling a group of users all at once, was recently proposed and studied by Hayes [28]. Each ready user responds to a probe by transmitting some noise energy in the channel. The central controller can only determine whether or not there is a response. He cannot tell, however, the exact number of ready users in the group probed. The key idea is as follows. If a group of users is probed and none responds because no one is ready, the whole group can be eliminated. If probing a group produces a positive response, it is subdivided into two (or more) groups which are then probed separately. When the network is lightly loaded, with few ready users, significant overhead reduction results through probing and eliminating groups of non-ready users all at once. Probing can be used in a tree search procedure for identifying ready users. For ease of exposition, suppose $N = 2^n$ for some positive integer n so that each user corresponds to a leaf in a binary tree. Suppose also that the search procedure begins with probing the whole tree. Again we consider the two extreme traffic conditions. First, if only one out of N users is ready, then it takes $2(\log_2 N) + 1$ queries to find out the status of all users; the mean number of queries needed to find the ready user is $\frac{3}{2}(\log_2 N) + 1$, assuming all users equally likely to be the ready user. On the other hand, if all N users are ready, then it takes $N^2 - 1$ queries to find out the status of all users and $(\log_2 N) + 1$ queries to find a ready user (anyone).

We next consider multiple access strategies based upon distributed control. Each ready user must actively seek channel access and make his own decisions according to an algorithm driven by observable outcomes in the broadcast channel. The contention protocol defined by (P1) and (P2) in the previous section is such a strategy. Under this protocol, when there is exactly one ready user and the channel is free, the channel assignment delay is 0. On the other hand, if all users are ready, then the mean channel assignment delay is 2.72 time slots. This last result is obtained under the assumption of an optimal algorithm for controlling the transmission probability in (P2) [19,20,22–27].

A linear search procedure can also be implemented using time slots and distributed control. An example is the MSAP protocol of Kleinrock and Scholl [29]. A time slot can be used to find out the status of a user by giving him sole access right to use the time slot. A user who is ready and has the access right to transmit in a time slot must do so. Suppose a sequential ordering of all users is known to individual users and synchronized in advance among all users. Each user can determine for himself the identity of the one with the access right to use the next time slot. With a linear search procedure, the mean channel assignment delay is $(N-1)/2$ time slots for traffic condition 1 and is zero for traffic condition 2.

Similarly, a tree search procedure can be implemented using time slots and distributed control. An example is Capetanekis' algorithm [30]. Since his algorithm was originally designed for a satellite channel, we shall consider a variation of it suitable for our short propagation delay environment here. Suppose, as before, each user corresponds to a leaf in a binary tree. Also, a user who is ready and is given the access right to transmit in a time slot must do so. By assigning the access right to use a time slot to all users in a tree (or subtree), the time slot will have 3 possible outcomes (empty, success, collision) indicating the number of ready users in the tree (or subtree) to be 0, 1 or more than 1. Suppose, initially, all users are given the access right to use the next time slot. If exactly one user is ready (traffic condition 1), he transmits and the channel assignment delay is zero. If two or more users are ready, a collision results. The tree is then divided into two subtrees, which can be searched one after the other. It is easy to see that when all N users are ready (traffic condition 2), the channel assignment delay is $(\log_2 N) + 1$ time slots.

The above results are summarized in Table 1. For comparison between strategies for passive and active users, we need to compare the mean time $\overline{w}$ needed to complete a query and the duration T of a time slot. We learned earlier that $T \geqslant 2\tau$. How close T can be made to approach 2τ depends upon the amount of guard time needed by the time synchronization tech-

Table 1
Channel assignment delays of multiple access strategies under two extreme traffic conditions

	Traffic Condition 1: 1 out of N users ready	Traffic Condition 2: all N users ready
Linear search using polling	$\frac{N+1}{2}$ queries (mean value)	1 query
Binary tree search using probing	$\frac{3}{2}(\log_2 N) + 1$ queries (mean value)	$(\log_2 N) + 1$ queries
Contention using time slots	0	2.72 time slots (mean value assuming optimal symmetric adaptive control)
Linear search using time slots	$\frac{N-1}{2}$ time slots (mean value)	0
Binary tree search using time slots	0	$(\log_2 N) + 1$ time slots

nique. On the other hand, $\bar{w}$ must include the round-trip propagation delay between the central controller and the user queried, plus transmission times of the query message and response. Because of the transmission times, it is likely that in most cases $\bar{w} > T$ but they will probably be of the same order of magnitude.

Referring to Table 1, we shall next compare the strategies in terms of the number of steps (queries or time slots) needed by each strategy. The two traffic conditions represent a lightly utilized channel (condition 1) and a heavily utilized channel (condition 2). Under traffic condition 1, the linear search procedures have the worst performance. Binary tree search using probing is much better than linear search, especially when N is large. Contention and binary tree search using time slots have the best performance (zero channel assignment delay).

Under traffic condition 2, the linear search procedures have the best performance while both binary tree search procedures perform poorly. Thus binary tree search procedures need to adapt to the level of traffic and dynamically switch to a linear search procedure beyond a certain level of channel utilization. Some such adaptive algorithms are discussed in [28,30]. The contention protocol also requires adaptive control for stable operation in order to achieve the mean channel assignment delay of 2.72 time slots as discussed earlier.

To properly evaluate the relative merits of the different access strategies, we need to have a measure of how often the actual traffic condition approximates traffic condition 1 or traffic condition 2. One of the results of our analysis below shows that traffic condition 1 does in fact prevail a great deal of the time even when the channel utilization is moderately high. This demonstrates the superiority of the contention and tree search procedures (given adaptive control). The performance difference between these strategies and linear search increases as N becomes large.

Lastly, we note that either a slotted or an unslotted channel can be used with the contention protocol while time slotting is needed for the binary tree search procedure with distributed control.

4. Analysis of the CSMA Protocol

For our analysis of CSMA based upon the contention protocol defined by (P1) and (P2), we shall assume that in (P2) a suitable adaptive algorithm is used so that the probability of a successful transmission (slotted ALOHA throughput) in the next time slot is equal to a constant S. This assumption is an approximation but has been found to be a very good one in simulation studies [23–26].

We shall further assume that errors due to random noise are insignificant relative to errors due to collisions and can be neglected. The source of traffic to the broadcast channel consists of an infinite population of users who collectively form an independent. Poisson process with an aggregate mean message generation rate of λ messages per second. This approximates a large but finite population in which each user generates messages infrequently; each message can be transmitted in an interval much less than the average time between successive messages generated by a given user. Each user is allowed to store and attempt to transmit at most one message at a time. Thus the generation of a new message is equivalent to increasing the number of ready users by one. The effect of queueing messages at individual users is discussed later.

Finally, we assume that the transmission time of each message is an independent identically distributed (i.i.d.) random variable with the probability distribution function (PDF) $\beta(x)$, mean value b_1, second moment b_2 and Laplace transform $\beta^*(s)$.

The ready users can be considered to form a distributed queue with random order of service for the broadcast channel. We are interested in obtaining the equilibrium moment generating function of the distributed queue size. We shall use an imbedded Markov chain analysis. Under the assumptions of Poisson arrivals and that messages arrive and depart one at a time, the moment generating function of queue size obtained for the imbedded points is valid for all points in time.

A snapshot of the channel is illustrated in Fig. 1. We define the following random variables:

q_n = number of ready users left behind by the departure of the nth transmission, C_n,

y_{n+1} = time from the departure of C_n to the beginning of the next successful transmission,

u_{n+1} = number of new (Poisson) arrivals during y_{n+1},

x_{n+1} = transmission time of C_{n+1},

v_{n+1} = number of new (Poisson) arrivals during $x_{n+1} + \tau$.

We assumed earlier that x_{n+1} has the PDF $\beta(x)$. We shall let $B(x)$ be the PDF of $x_{n+1} + \tau$. The corresponding Laplace transform is thus

$$B^*(s) = \beta^*(s)\,\mathrm{e}^{-s\tau}.$$

The random variable y_{n+1} is the sum of two independent random time intervals

$$y_{n+1} = (I_{n+1} + r_{n+1})\,T,$$

where T is the duration of a slot, I_{n+1} is the number of slots in an idle period immediately following the departure of C_n, and r_{n+1} is the number of slots in the contention period following a collision until the next successful transmission. The slot containing the initial collision is included in r_{n+1}. We note that I_{n+1} is nonzero only if $q_n = 0$. Also, if there has been no collision when C_{n+1} begins, $r_{n+1} = 0$.

Let p_j be the probability of j new arrivals (ready users) in a time slot.

$$p_j = \frac{(\lambda T)^j\,\mathrm{e}^{-\lambda T}}{j!}\,, \quad j = 0, 1, 2, \ldots$$

At the start of the next time slot, each new arrival executes (P1) or (P2) in exactly the same manner as all other ready users.

Given our earlier assumption, we have

$$\mathrm{Prob}\,[I_{n+1} = k/q_n = 0] = (1 - p_0)\,p_0^{k-1}, \quad k = 1, 2, \ldots$$

Also,

$$\mathrm{Prob}\,[r_{n+1} = k/\text{collision occurred}] = S(1 - S)^{k-1} \quad k = 1, 2, \ldots$$

From this last result, the Laplace transform of the probability density function (pdf) of a contention period (given a collision occurred) is

$$C^*(s) = \frac{S\,\mathrm{e}^{-sT}}{1 - (1 - S)\,\mathrm{e}^{-sT}}\,,$$

which has a mean of T/S and a second moment of

$$T^2\left(1 + \frac{2(1 - S)}{S^2}\right).$$

The following important relationship is evident from Fig. 1.

$$q_{n+1} = q_n + u_{n+1} + v_{n+1} - 1\,, \qquad (1)$$

where v_{n+1} is an independent random variable with the z-transform $B^*(\lambda - \lambda z)$, while u_{n+1} depends upon q_n in the following manner as a consequence of (P1) and (P2). Given

(1) $q_n = 0$,

$$u_{n+1} = \begin{cases} 1 & \text{with prob. } \dfrac{p_1}{1 - p_0} \\ j + \text{number of arrivals during a contention period} & \text{with prob. } \dfrac{p_j}{1 - p_0} \quad j = 2, 3, \ldots \end{cases}$$

(2) $q_n = 1$, $u_{n+1} = 0$

(3) $q_n \geqslant 2$, u_{n+1} = number of arrivals during a contention period. (2)

Given the occurrence of a collision, the number of new arrivals during a contention period is an independent random variable with the z-transform $C^*(\lambda - \lambda z)$.

The equilibrium queue length probabilities

$$Q_k = \lim_{n \to \infty} \mathrm{Prob}\,[q_n = k]\,, \quad k = 0, 1, 2, \ldots$$

exist if $\lambda(b_1 + \tau + T/S) < 1$ (see [14]). Define the z-transform

$$Q(z) = \sum_{k=0}^{\infty} Q_k z^k\,.$$

By considering Eqs. (1) and (2) and taking the $n \to \infty$ limit, we obtain after some algebraic manipulations the following important result: *

$$Q(z) = B^*(\lambda - \lambda z)\Big\{Q_1 z[1 - C^*(\lambda - \lambda z)] + \frac{Q_0}{1 - p_0}\,[p_1 z(1 - C^*(\lambda - \lambda z)) - C^*(\lambda - \lambda z)(1 - e^{-\lambda T(1-z)})]\Big\} \times \Big(z - B^*(\lambda - \lambda z)\, C^*(\lambda - \lambda z)\Big)^{-1}, \qquad (3)$$

where

$$Q_0 = \frac{1 - \lambda(b_1 + \tau + T/S)}{\lambda T\left(\dfrac{1}{1 - p_0} - \dfrac{1}{B^*(\lambda)\, S}\right)} \qquad (4)$$

and

$$Q_1 = \left(\frac{1}{B^*(\lambda)} - \frac{p_1}{1 - p_0}\right) Q_0 \qquad (5)$$

Using Eqs. (3)–(5), we can obtain the mean queue size. Application of Little's result [14] yields the mean message delay (time of arrival to time of departure) to be

$$D = \bar{x} + \frac{T}{S} + \frac{T}{2} - \frac{1 - p_0}{2[B^*(\lambda)\, S - (1 - p_0)]} \times \left(\frac{2}{\lambda} + ST - 3T\right) + \frac{\lambda[\overline{x^2} + 2\bar{x}(T/S) + T^2(1 + 2(1 - S)/S^2)]}{2[1 - \lambda(\bar{x} + T/S)]}, \qquad (6)$$

where

$$\bar{x} = b_1 + \tau$$

and

$$\overline{x^2} = b_2 + 2b_1\tau + \tau^2 .$$

We next consider the channel assignment delay, that is, given that the channel is free and that there is at least one ready user, we want the pdf of the time from when the above conditions are satisfied to the start of the next successful transmission. Let d_n be a random variable representing the channel assignment delay immediately prior to the n^{th} transmission and

$$d = \lim_{n \to \infty} d_n$$

It can be readily shown that

$$\text{Prob}\,[d = kT] = \begin{cases} Q_0\, \dfrac{p_1}{1 - p_0} + Q_1, & k = 0 \\ \left\{Q_0\left(1 - \dfrac{p_1}{1 - p_0}\right) + \displaystyle\sum_{i=2}^{\infty} Q_i\right\} S(1 - S)^{k-1}, & k = 1, 2, \ldots \end{cases} \qquad (7)$$

The mean channel assignment delay is thus

$$\bar{d} = \frac{1}{S}\left(1 - Q_0\, \frac{p_1}{1 - p_0} - Q_1\right) T . \qquad (8)$$

Note that $Q_0\, p_1/(1 - p_0) + Q_1$ is the fraction of transmissions that incur zero assignment delay in gaining channel access (no collision).

5. Performance Observations

An important performance parameter is the ratio of the carrier sense time τ to the mean message transmission time b_1:

$$\alpha = \frac{\tau}{b_1} . \qquad (9)$$

The *throughput* of the CSMA channel is defined to be the fraction of channel time utilized by data messages, which is

$$\rho = \lambda b_1$$

under equilibrium conditions. The requirement of $\lambda(\bar{x} + T/S) < 1$ gives rise to the following upper bound on the channel throughput

$$\rho < \frac{S}{2\alpha + (1 + \alpha)\, S} . \qquad (10)$$

In Fig. 2, we show the delay performance of the CSMA channel as a function of α and ρ. The normalized delay D/b_1 is plotted and it is assumed that messages are of constant length. Observe that the delay performance of CSMA improves significantly as α becomes small. A small α may come about either by decreasing the carrier sense time τ or by increasing the duration b$_1$ of each user transmission.

In these numerical calculations, the probability S of a successful transmission during contention periods is assumed to be $1/e$ which is the optimum slotted ALOHA throughput rate for an infinite population model.

* The derivation of $Q(z)$ does not depend upon the specific form of $C^*(s)$ as long as the contention period duration is an independent positive random variable.

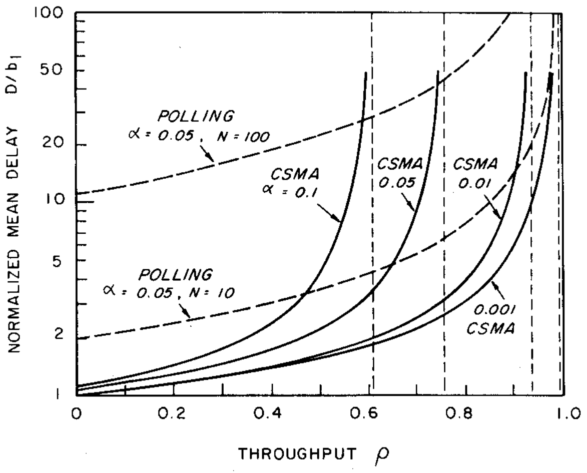


Fig. 2. Mean delay versus throughput.

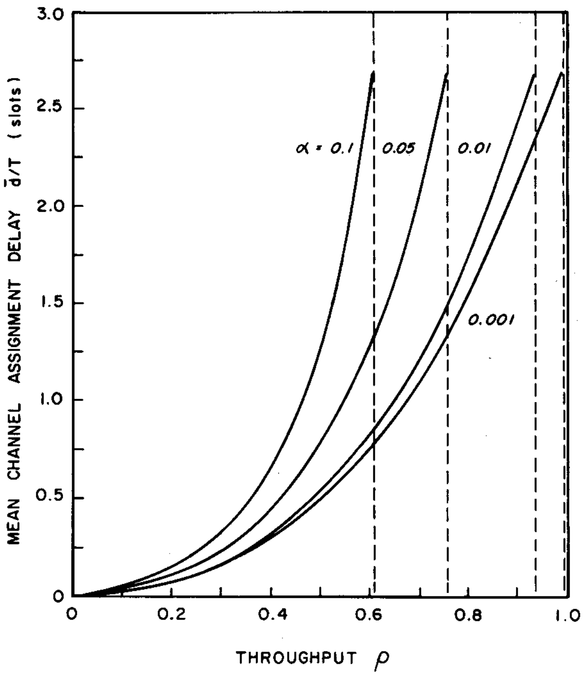


Fig. 3. Mean channel assignment dealy versus throughput.

The delay-throughput performance of roll-call polling is also shown using the delay formula in [31]. The delay results shown for polling assume Poisson message arrivals and constant message length. The ratio of propagation delay to message transmission time is $\alpha = 0.05$. The ratio of data to polling message length is 10. Queueing of messages at individual users is assumed; hence the maximum channel throughput is one. Delay-throughput curves for both 10 users and 100 users are shown. Note that the corresponding delay-throughput performance of CSMA at $\alpha = 0.05$ is independent of the number of users. It also permits no queueing of messages at individual users; hence the maximum throughput is less than 1. We observe that CSMA is superior to polling when the channel throughput is low but becomes inferior when the channel throughput is increased to one. However, if queueing of messages is possible at individual users for CSMA, more than one message may be transmitted every time a user gains channel access. Hence, as the network load ρ is increased from 0 to 1, the delay performance of CSMA is first given by the $\alpha = 0.05$ curve at a small channel throughput but switches to the $\alpha = 0.01$ curve and then the $\alpha = 0.001$ curve and so on as the channel throughput increases and queues become long. The channel throughput of CSMA is one in the limit of infinitely long queues at individual users.

In Fig. 3, we show the mean channel assignment delay $\bar{d}$ as a function of α and ρ. Note that $\bar{d}$ decreases to zero when ρ is small. This is because (P1) in the CSMA protocol permits a ready user to access the channel immediately. In Fig. 4 we plot the fraction of transmissions that incur zero delay in gaining channel access given that the channel is free. For comparison, recall that when only one ready user is present, the mean channel assignment delay is

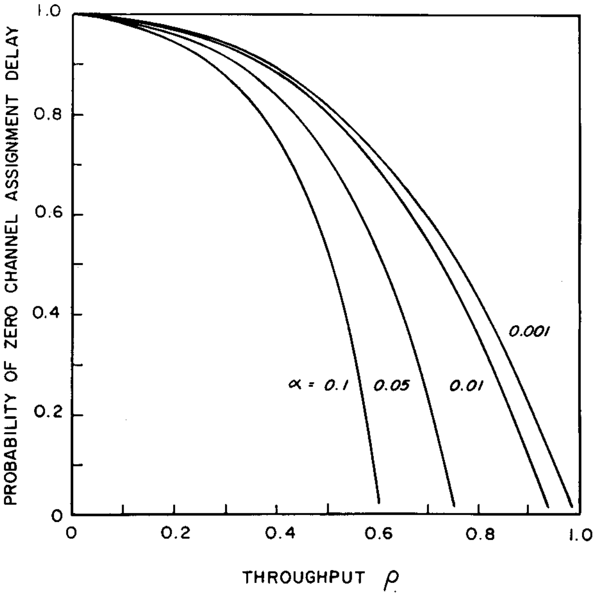


Fig. 4. Probability of zero channel assignment delay versus throughput.

$\frac{1}{2}(N+1)\,\bar{w}$ for conventional polling and $[\frac{3}{2}(\log_2 N) + 1]\,\bar{w}$ for probing.

Referring again to Figure 3, observe that as ρ is increased, $\bar{d}/T$ increases to the maximum value of $1/S$. This desirable property is a consequence of the presence of an adaptive algorithm that we assumed in (P2) which guarantees channel stability during contention periods.

6. Simulation Results

In this section, we compare the above analytic results with experimental results from simulation. We assume, as before, the aggregate arrival process of new messages to be a Poisson process. For simplicity, a deterministic message transmission time is used.

To carry out the simulation, it is now necessary to specify the adaptive algorithm in (P2) with which a ready user selects its transmission probability for the next time slot. Many adaptive algorithms have been proposed and studied in the past; see [8,23–27,30]. Since this is not the primary concern of this study, we limited ourselves to algorithms which are easy to implement, but may not be optimal. Specifically, the following class of algorithms that depend only upon locally available information was considered.

Each ready user transmit into a time slot with probability P_k where $k = 0, 1, 2, \ldots$ is the accumulated number of collisions incurred by the user. Each user determines his P_k according to one of the following algorithms:

(1) $P_k = (0.5)^k, \quad k = 0, 1, 2, \ldots$.

k is reset to 0 when the ready user has a successful transmission.

(2) $$P_k = \begin{cases} 1, & k = 0 \\ p, & k = 1, 2, 3 \\ 0, & \text{otherwise} \end{cases}$$

k is reset to 0 following each successful transmission in the channel (by anyone). When k reaches 4, the ready user discards his message and becomes a nonready user.

We note that both algorithms (1) and (2) are not optimal. They also deviate slightly from the idealized properties of (P2) that we assumed in our analysis. Algorithm (1) was first proposed in [8]. Note that it does not reset k to 0 following every successful transmission in the channel as is expected by (P1). Algorithm (2) permits messages to be discarded occasionally in times of congestion. Algorithm (2) was first proposed and studied in [23–25].

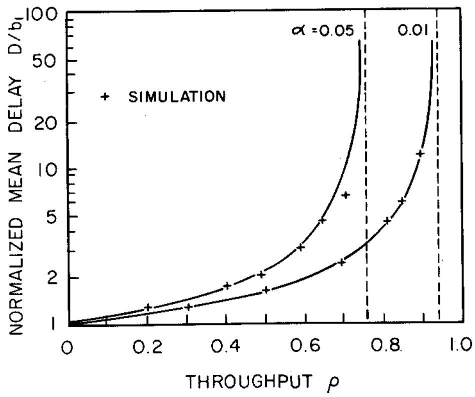


Fig. 5. Mean delay versus throughput for algorithm (1).

In Figures 5 and 6, we compare analysis and simulation results for mean message delay. We see from Figure 5 that the agreement between simulation and analysis is excellent for algorithm (1). Results for algorithm (2) are shown in Figure 6 for two values of p (0.1 and 0.2). We learned from our previous work that $p = 0.2$ is close to optimal while $p = 0.1$ is nonoptimal [20,23]. We see from Figure 6, that $p = 0.1$ indeed gives rise to delay results slightly higher than the theoretical (optimum) values. Delay results for $p = 0.2$ are actually smaller than the theoretical values when the channel is heavily loaded. This is because the simulated mean delay values in Figure 6 are calculated only for successfully transmitted messages; the delays incurred by discarded messages are not accounted for. With $p = 0.2$, the collision probability is higher than that with $p = 0.1$ and will thus give rise to more discarded messages. The loss probability for the two cases is illustrated in Figure 7.

Comparing algorithms (1) and (2), note that while algorithm (2) discards some messages when the

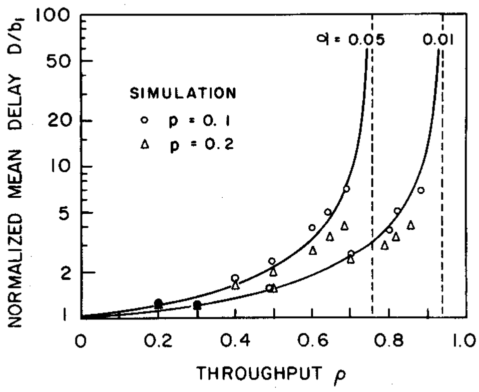


Fig. 6. Mean dealy versus throughput for algorithm (2).

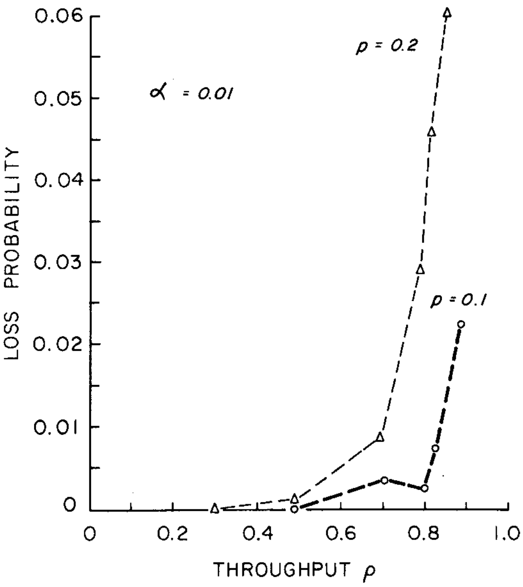


Fig. 7. Loss probability versus throughput for algorithm (2).

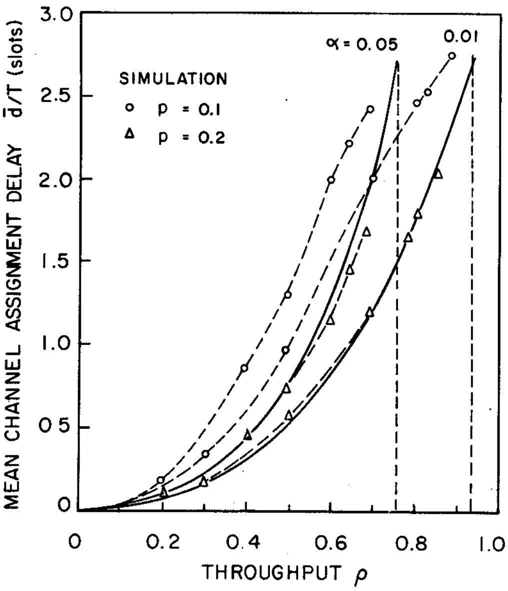


Fig. 9. Mean channel assignment dealy versus throughput for algorithm (2).

channel is heavily loaded, algorithm (1) does not. As a result, we observed that the variance in message delay of algorithm (1) is typically one or more orders of magnitude higher than that of algorithm (2) when the channel is moderately to heavily loaded.

In Figures 8 and 9, we plotted simulation and analysis results for the mean channel assignment delay. Here, the nonoptimality of algorithm (1) and algorithm (2) with $p = 0.1$ is clearly shown. However, with a small α, the mean channel assignment delay is much smaller than the mean waiting time in the (distributed) queue. Hence, our main analytic results are robust; they are applicable for heuristic nonoptimal algorithms as long as they give rise to a stable channel.

In Figures 10 and 11, we plotted simulation and analysis results for the probability of zero delay in

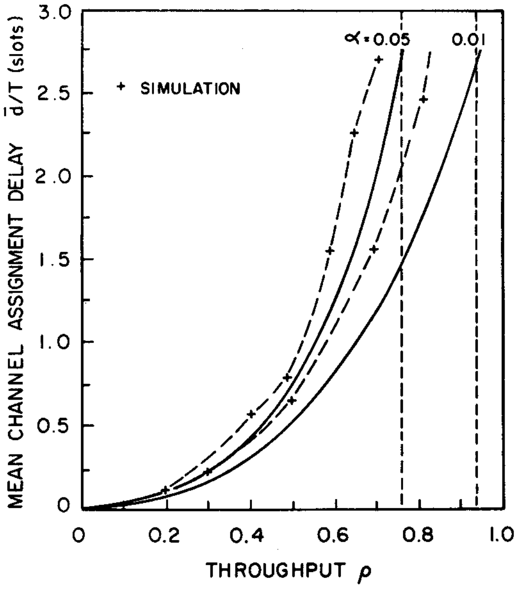


Fig. 8. Mean channel assignment delay versus throughput for algorithm (1).

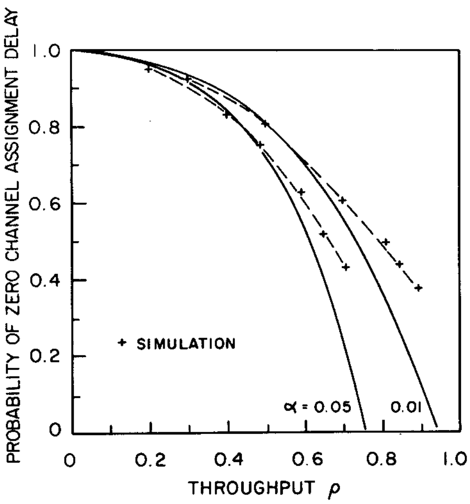


Fig. 10. Probability of zero channel assignment delay versus throughput for algorithm (1).

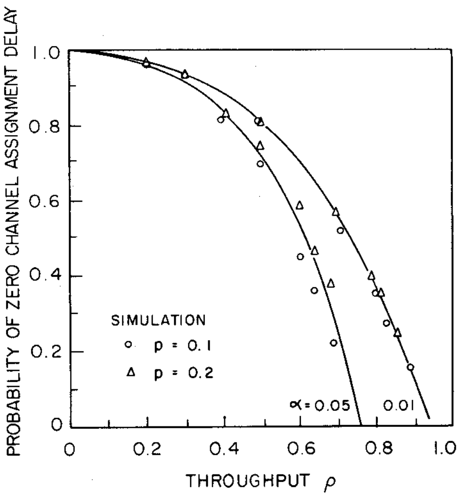


Fig. 11. Probability of zero channel assignment delay versus throughput for algorithm (2).

assigning channel access to a ready user. Note in Figure 11 that simulation and analysis agree very well for algorithm (2). In the case of Figure 10, the simulation results are actually better than the analysis results when the channel is heavily loaded. This is because algorithm (1) deviates from protocol (P2) by not requiring ready users to transmit into a time slot with probability 1 following each successful transmission in the channel. This requirement would produce a collision with a high probability when the channel is heavily loaded.

7. Conclusions

We proposed and analyzed a CSMA protocol as a distributed control technique for a population of users to share a broadcast channel. Specifically we considered local networks with short propagation delays. The capability of aborting collided transmissions is the main difference between our model and previous models of CSMA. It is also assumed that the channel is stable during contention periods (presence of an adaptive control algorithm). Our main results include the moment generating function of the number of ready users, as well as mean value formulas for message delay and channel assignment delay. These results are new. Characterization of the distributed queue size and message delay has previously been limited to numerical solutions or simulations.

We found that the CSMA protocol as defined in this paper has the desirable property that when the channel is lightly utilized, the channel assignment delay is extremely short. The performance of CSMA when the channel is heavily utilized depends upon the ratio α in Eq. (9). We make the following observation. If the number of users is finite and queueing of messages is permitted at individual users, then as $\rho \to 1$, we must have $\alpha \to 0$, since the transmission time of each user increases as a result of long queues. In this case, the maximum channel throughput of CSMA is one (the same as polling with queueing permitted at individual users).

In this paper, the performance measures of interest are average message delay and channel throughput. These are appropriate for a homogeneous population of users. We have neglected scheduling issues of fairness, discrimination, explicit and implicit priorities etc. These are important considerations in operational networks handling diverse traffic types with dissimilar performance goals (high throughput, low delay, real-time applications etc.) Some of these issues are addressed in [32].

References

[1] M. Schwartz, Computer-Communication Network Design and Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[2] A. Fraser, "A Virtual Channel Network," Datamation, Vol. 21, Feb. 1975.

[3] D.J. Farber, and K.C. Larson, "The System Architecture of the Distributed Computer System – the Communications Systems," Proc. Symp. Computer-Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, April 1972.

[4] D.G. Willard, "Mitrix: A Sophisticated Digital Cable Communications System," Conf. Rec. National Telecommunications Conference, Nov. 1973.

[5] R. Binder, et al., "ALOHA Packet Broadcasting – A Retrospect," AFIPS Conf. Proc. Vol. 44, 1975.

[6] R.E. Kahn, "The Organization of Computer Resources into a Packet Radio Network," IEEE Trans. on Commun., Vol. COM-25, Jan. 1977.

[7] V.A. DeMarines, and L.W. Hill, "The Cable Bus in Data Communications," Datamation, August 1976.

[8] R.M. Metcalfe, and D.R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Communications of the ACM, 19, No. 7, July 1976.

[9] A. West, and A. Davison, "CNET – A Cheap Network for Distributed Computing," Department of Computer Science and Statistics, Queen Mary College, University of London, Report TR 120, March 1978.

[10] J.E. Thornton, et al., "A New Approach to Network Storage Management," Computer Design, Nov. 1975.

[11] K.J. Biba, and J.W. Yeh, "FordNet: A Front-End Approach to Local Computer Networks," Proc. Local Area Communications Network Symposium, Boston, May 1979.

[12] J. Yeh, P. Lehot and G. Steddum, "Hierarchical Design of a Family of Local Computer Networks," Proc. Pacific Telecommunications Conference, Honolulu, Jan. 1980.

[13] P. Brinch Hansen, Operating System Principles, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[14] L. Kleinrock, Queueing Systems, Vol. I, 1975; Vol. II, 1976, Wiley-Interscience, N.Y.

[15] S.S. Lam, "Multiple Access Protocols," Dept. of Computer Sciences, University of Texas at Austin, Tech. Rep. TR-88, Jan. 1979; to appear in Computer Communication: State of the Art and Direction for the Future, edited by W. Chou, Prentice-Hall.

[16] F.A. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Network," Ph.D. Dissertation, Computer Science Department, University of California at Los Angeles, Dec. 1974.

[17] L. Kleinrock, and F.A. Tobagi, "Packet Switching in Radio Channels: Part 1 – Carrier Sense Multiple Access Modes and their Throughput-Delay Characteristics," IEEE Trans. on Comm. Vol. COM-23, December 1975.

[18] L.W. Hansen and M. Schwartz, "An Assigned-Slot Listen-Before-Transmission Protocol for a Multiaccess Data Channel," Conf. Rec. International Conference on Communications, Chicago, June 1977.

[19] N. Abramson, "Packet Switching with Satellites," AFIPS Conf. Proc., Vol. 42, AFIPS Press, Montvale, New Jersey, 1973.

[20] L. Kleinrock, and S.S. Lam, "Packet-switching in A Slotted Satellite Channel," AFIPS Conf. Proc., Vol. 42, AFIPS Press, Montvale, N.J., 1973.

[21] C. Ellingson, and R. Kulpinski, "Dissemination of System Time," IEEE Trans. on Commun., Vol. COM-21, May 1973.

[22] L. Kleinrock and S.S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. on Commun., Vol. COM-23, April 1975.

[23] S.S. Lam, "Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network," Ph.D. Dissertation, Computer Science Department, Univ. of California, Los Angeles, March 1974.

[24] S.S. Lam, and L. Kleinrock, "Dynamic Control Schemes for a Packet Switched Multi-access Broadcast Channel," AFIPS Conf. Proc., Vol. 44, AFIPS Press, Montvale, N.J., 1975.

[25] S.S. Lam, and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," IEEE Trans. on Commun., Vol. COM-23, Sept. 1975.

[26] M. Gerla and L. Kleinrock, "Closed Loop Stability Controls for S-Aloha Satellite Communications," Proc. Fifth Data Communications Symposium, Snowbird, Utah, September 1977.

[27] L. Kleinrock, and Y. Yemini, "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication," Conf. Rec. ICC'78, Toronto, June 1978.

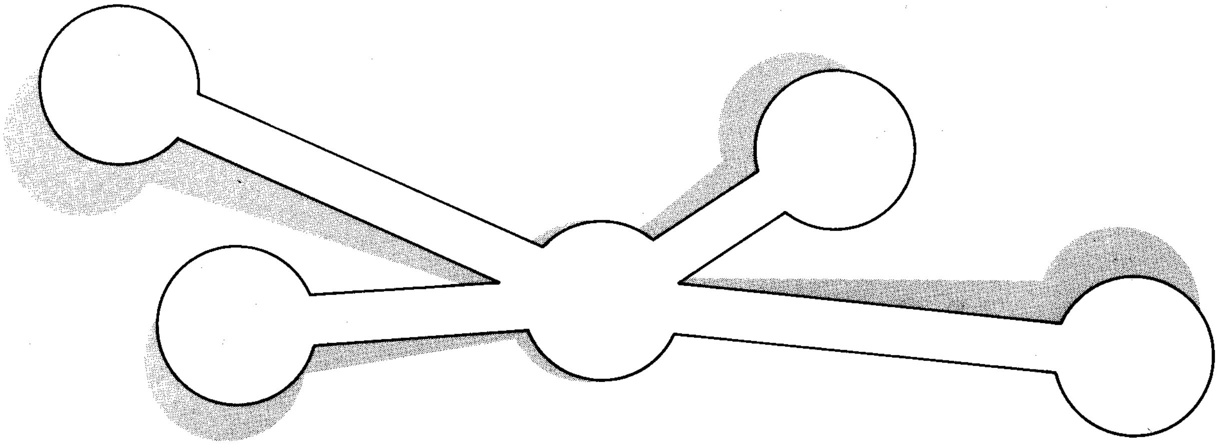
[28] J.F. Hayes, "An Adaptive Technique for Local Distribution," IEEE Trans. on Commun., Vol. COM-26, August 1978.

[29] L. Kleinrock, and M. Scholl, "Packet Switching in Radio Channels: New Conflict-Free Multiple Accese Techniques for a Small Number of Data Users," Conf. Rec. Int. Conf. on Communications, Chicago, June 1977.

[30] J.I. Capetanekis, "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations," Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Aug. 1977.

[31] A.G. Konheim, and B. Meister, "Waiting Lines and Times in A System with Polling," J.ACM, Vol. 21, July 1974.

[32] S.S. Lam, "A Packet Network Architecture for Local Interconnection", Conf. Rec. Int. Conf. on Communications, Seattle, June 1980.



Corrigendum

"A Carrier Sense Multiple Access Protocol for Local Networks," *Computer Networks,* Vol. 4, February 1980, pp. 21-32.

The formula for the second moment of the contention period in this article should be

$$T^2(2\text{-}S)/S^2$$

instead of the expression on page 26. This mistake was brought to my attention by M. Niktash of Carleton University and also by Professor Mischa Schwartz of Columbia University. Since the average message delay formula in this article had been used by other authors (such as W. Bux) in performance comparisons of local area networks, Professor Schwartz wondered if those comparisons would be affected.

To answer his question, we performed numerical calculations of the average message delay for different values of α and b_1. (As in the numerical examples of the article, a constant message length was assumed.) We studied the error in the average message delay due to the incorrect expression for the second moment of the contention period. We found that the error increases with α and with the channel utilization but is independent of b_1. For $\alpha < 0.1$, the error is less than 1%. For $\alpha = 0.1$, the error is less than 1% in most cases but goes up to 3% for a heavily utilized channel. We conclude that previous performance comparisons utilizing the results of this article were not affected in any significant way.

I would like to thank C. H. Chow of the University of Texas at Austin who carried out the numerical calculations.