To appear in Proc. IMACS Congress, Montreal, August 1982.

ON THE PERFORMANCE OF MULTIPLE ACCESS PROTOCOLS FOR LOCAL NETWORKS*

Simon S. Lam
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Abstract

Local communication networks that interconnect distributed users via a broadcast channel are considered. A taxonomy of multiple access protocols for such networks is illustrated. A method for comparing multiple access protocols using two simple performance criteria is shown. Queueing models for the performance analysis of such networks are briefly discussed.

INTRODUCTION

Local networks interconnect computers and terminals within a local environment, such as an office building, typically without the use of common-carrier communications facilities. The distances spanned by such networks range from hundreds to thousands of meters. The users (i.e., terminals and computers) typically communicate via the exchange of messages or packets instead of through shared memory. The interconnection of users and resources (processors, memory, programs, data, etc.) within a local environment is extremely useful. The users can communicate with one another, as well as share use of the entire collection of resources. The network may also provide the function of local collection and distribution of data traffic to and from the outside world (via gateways to long-haul networks). Various applications of local networks have been considered. But by far the most important one seems to be office automation. It is projected that in a few years, local networks for office automation will be a business in the hundreds of millions of dollars [1].

Most local networks are based upon three communications architectures: broadcast cable, ring network, and broadcast radio. In each case there is a single communication channel that needs to be shared by the entire population of distributed network users. The set of rules and algorithms that define the method of sharing is referred to as the multiple access protocol. Multiple access protocols that can be used for the three communications architectures are conceptually similar. However, the performance of a multiple access protocol depends upon various network parameter values which may be different in different architectures. Also, certain protocols assume specific hardware capabilities that are available in one architecture but cannot be easily implemented in another architecture. For the purpose of this paper, we shall consider the communications architecture simply as a broadcast channel, with the property that the transmission of any user in the channel (if it is the only transmission present) is observable and can be received by all other network users after a small propagation delay.

The balance of this paper is organized as follows. First, we give a taxonomy of multiple access protocols. Specifically, three basic multiple access techniques are described. We assert that most multiple access protocols that have been proposed are based upon one or a hybrid combination of these techniques. A method for comparing multiple access protocols using two very simple performance criteria is presented. Queueing models for the performance analysis of multiple access protocols are then considered. We conclude that much more work needs to be done in the development of such queueing models.

* This work was supported by National Science Foundation Grant No. ECS78-01803.

A TAXONOMY OF MULTIPLE ACCESS PROTOCOLS

Consider the sharing of a single broadcast channel by a population of N distributed users. Traditional techniques for solving the multiple access problem are based upon dividing the broadcast channel into a pool of subchannels, using either frequency-division or time-division. These subchannels may be fixed-assigned to the population of users; in this case, N subchannels are needed. The subchannels may also be demand-assigned to the population of users; in this case, fewer than N subchannels are needed. Demand assignment may be achieved with either a central controller or a distributed control algorithm. These techniques are said to be channel-oriented since each unit of allocation is one subchannel [2].

It is well-known in queueing theory that a single high-speed server is often preferable to multiple servers whose aggregate service capacity is the same as the single server [3]. This is particularly true in data communications where traffic sources are typically very bursty. Multiple access protocols of interest in the literature and to be considered below are primarily packet-oriented, i.e., packets from different users are scheduled in some fashion to use the entire broadcast channel [2].

The multiple access problem is thus simply the problem of forming a queue among a population of distributed users [4]. To form such a distributed queue, there are two subproblems. (As shown below, these subproblems can be solved with either a central controller or a distributed control algorithm.)

1. Identify those users with data to send and who desire channel access (the ready users).
2. Assign the channel to a ready user according to some scheduling discipline.

Instead of identifying all ready users present in the population and employing an explicit scheduling discipline, the following variation may be employed. If the channel is idle, it is next assigned to the first ready user identified. In this case, the scheduling discipline is implicit in the algorithm for finding a ready user.

There are three basic techniques for finding ready users: linear search, tree search and contention. We believe that most multiple access protocols having been proposed or implemented are based upon one of these techniques or a hybrid combination of them.

Linear Search

A prime example of a multiple access protocol based upon linear search is roll-call polling. A central controller is employed and the population of users is queried according to some sequential ordering of the users.

Distributed control may also be used. In hub-polling, a special message, called a token, is passed among the users according to some sequential order. When the token reaches a user who is ready, he is assigned access to the channel.

Suppose that the users are time synchronized so that the channel is divided into time slots following the end of a transmission (see Figure 1). A linear search of the users can be performed by assigning one user to each time slot according to a predefined sequential order. A user can indicate in his time slot whether or not he is ready. It is assumed that at the end of a time slot the outcome of the time slot is globally known.

Suppose that following the end of a transmission, the channel is assigned next to the first ready user found. The scheduling discipline may be implicitly defined by specifying the user who begins the sequence following a user n transmission; see Figure 1. For example, if the sequence is always 1, 2, 3, ..., then we have a priority discipline with user 1 having the highest priority and user N the lowest priority. If the sequence is n+1, n+2, n+3, ..., then we have a round-robin discipline [5].

Tree search

Protocols that find ready users by a tree search can also be implemented with a central controller or by means of time slots similar to what has been described earlier. Suppose that the network users correspond to the leaves of a tree. Without loss of generality, consider a binary tree. The tree is traversed starting from the root node and the ready users are identified using a divide-and-conquer type of algorithm.

A central controller, if used, can "probe" a group of users simultaneously instead of polling them one after another. When probed, all ready users in the group will respond. The central controller can distinguish between two outcomes: (1) no response, (2) response (at least one ready user). In the tree traversal, when a node is visited, the central controller probes the group of users that are the descendants of the visited node. If there is no response, the entire subtree is "pruned." On the other hand, if there is a response, then each of the two sons of the node will have to be visited subsequently in the tree traversal [6].

The tree traversal can also be carried out using time slots (see Figure 1). If the users exercise the same tree traversal algorithm and observe the same outcomes in the time slots, they will arrive at the same conclusion about the status of all users. In this case, the node visited next in the tree traversal is assigned the next time slot. It is assumed that when a node is visited, all users who are descendants of that node will transmit into the assigned time slot. Suppose that three possible outcomes can be distinguished in a time slot: (1) no one ready, (2) a single user ready and his identity is indicated, (3) a collision (two or more users ready). When a node is visited, the occurrence of either of the first two outcomes will provide information on the status of all its descendants [7]. Hence, the subtree may be pruned. It should be clear that the ability to distinguish the three outcomes of a time slot instead of the two outcomes of a probe will give rise to a more efficient search. In practice, the ability to recognize these and, perhaps, other outcomes will depend upon the communications technology being used. As before, if the channel is next assigned to the first ready user found in the tree traversal, then the scheduling discipline is implicitly specified by the tree traversal algorithm.

Contention

The third basic technique for finding ready users is based solely upon distributed control. In its purest form, each user simply transmits a newly generated packet into the channel immediately, hoping that it will be successfully received without colliding with someone else's transmission. A packet that has unfortunately collided with another packet is retransmitted after a random delay. A random delay is necessary to minimize the probability of collision with the same packet again [8, 9]. This random delay determines the throughput-delay performance of the system and affects its stability behavior [10]. Protocols based upon contention should always be implemented in conjunction with some adaptive control algorithm to avoid unstable behavior [4, 10, 11]. Contention protocols in general give rise to a random order of service. However, it is possible to favor some users over others by specifying different retransmission delays.

For Table 1 shown below (to be introduced in the next section), we assume a time-slotted channel for the contention protocol. At the end of a time slot, it is globally known whether it is empty, contains a single user's transmission, or contains a collision. Following the end of a transmission (see Figure 1), it is assumed that each user transmits in the next time slot with probability 1. A packet that has had a collision since then is retransmitted in the next time slot with a probability (<1) that is adaptively controlled.

PERFORMANCE CONSIDERATIONS

The three basic multiple access techniques may be compared by considering two extreme traffic conditions: (1) exactly one out of the N users is ready, and (2) all of the N users are ready. In Table 1, from [4], we show the mean assignment delay (amount of channel time wasted) for each of the techniques to identify the ready user under traffic condition 1 or an arbitrary ready user under traffic condition 2. The ready user under traffic condition 1 is assumed to be chosen randomly. We also assume that those time slots that initiate a new transmission contain data (they are not wasted). Hence, some entries in Table 1 are equal to zero.

Given a large N, Table 1 shows that linear search performs well under traffic condition 2 but performs poorly under traffic condition 1. The opposite is true for tree search and contention. Note that the actual performance of each technique in a specific system depends upon system parameters which determine the time slot size or query duration. Whether one technique is better than another (over a wide range of channel utilizations) depends upon the specific parameter values under consideration.

In Figure 2, the general behavior of the mean delay versus throughput characteristic of a multiple access protocol is shown. Note that the maximum throughput S_{max} and the minimum mean delay D_{min} in Figure 2 can be predicted from the entries in Table 1 for a given multiple access protocol. Specifically, the mean channel assignment delay of a multiple access technique under traffic condition 1 determines D_{min} while the mean channel assignment delay under traffic condition 2 determines S_{max}.

Queueing Models

In most local networks, users have the capability to sense the presence of an ongoing transmission in the channel and are required to keep quiet until the transmission terminates. (This is referred to as a deference protocol.) Such a network can be modeled as a single-server multi-queue system as follows. Upon the departure of a customer, a (random) delay is incurred to find and schedule the next customer into service, if one is present. We shall refer to such a delay as the channel assignment delay. The delay duration depends upon the multiple access protocol as well as the state of the system (distribution of customers among the queues).

We next briefly examine two available queueing models for multiple access protocols. The first is the model of Konheim and Meister for a polling system [12]. Conceptually, their model can be used for any multiple access protocol employing a linear search technique and a round-robin discipline.

The second model is one developed by this author for a CSMA/CD protocol based upon contention [4]. The channel assignment delay was assumed to depend upon the number of ready users in a simple fashion so that an imbedded Markov chain analysis was applicable. The moment generating function of the number of ready users and mean value formulas for packet delay and channel assignment delay were derived.

A comparison of the mean delay versus throughput performance of the CSMA/CD and polling protocols is shown in Figure 3 (reprinted from [4]). In this figure, b_1 is the mean transmission time of a ready user, and α is the ratio of the (maximum) channel propagation delay to b_1. In this particular comparison, CSMA/CD is better than polling (assuming a certain overhead for each query). However, with a different set of parameter values (query overhead and time slot size) it is possible for a linear search protocol to be better than a contention protocol.

Further Work

Much of the research on multiple access protocols has been concerned with the design of such protocols for different networks with different capabilities. (The reader is referred to [2] for a tutorial treatment of multiple access protocols.) The performance of most multiple access protocols have been characterized primarily by their maximum achievable throughputs.

Few queueing models have been developed for the analysis of the following classes of important protocols:

1. protocols based upon tree search as well as a combination of tree search and linear search [6, 7];
2. protocols based upon a combination of contention and linear search [13];
3. scheduling disciplines that are either explicitly specified or implicit in search algorithms;
4. protocols for network users with diverse traffic loads and response time requirements.

REFERENCES

[1] Datamation, Vol. 28, March 1982, pp. 115-116.

[2] Lam, S.S., "Multiple Access Protocols," Technical Report TR-88, Dept. of Computer Sciences, Univ. of Texas at Austin, January 1979; to appear as a chapter in Computer Communications, Vol. I: Principles, edited by W. Chou, Prentice-Hall, June 1982.

[3] Kleinrock, L., Queueing Systems Vol. 2: Computer Applications, Wiley-Interscience, New York, 1976.

[4] Lam, S.S., "A Carrier Sense Multiple Access Protocol for Local Networks," Computer Networks, Feb. 1980.

[5] Kleinrock, L. and M. Scholl, "Packet Switching in Radio Channels: New Conflict-free Multiple Access Schemes for a Small Number of Data Users," Conf. Rec. Int. Conf. on Commun., Chicago, June 1977.

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

[7] Capetanakis, J.I., "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations," Ph.D. Thesis, Dept. of Elec. Eng. and Comp. Sci., M.I.T., Aug. 1977.

[8] Abramson, N., "THE ALOHA SYSTEM--Another Alternative for Computer Communications," AFIPS Conf. Proc. Vol. 37, AFIPS Press, Montvale, NJ, 1970, pp. 281-285.

[9] Kleinrock, L. and S.S. Lam, "Packet Switching in a Slotted Satellite Channel," AFIPS Conf. Proc., Vol. 42, AFIPS Press, Montvale, NJ, 1973, pp. 703-710.

[10] Kleinrock, L. and S.S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. on Commun., Vol. COM-23, April 1975, pp. 410-423.

[11] Lam, S.S. and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," IEEE Trans. on Commun., Vol. COM-23, Sept. 1975, pp. 891-904.

[12] Konheim, A.G. and B. Meister, "Waiting Lines and Times in a System with Polling," JACM, Vol. 21, July 1974.

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

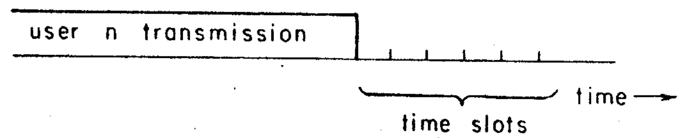

Fig. 1. A time-slotted channel.

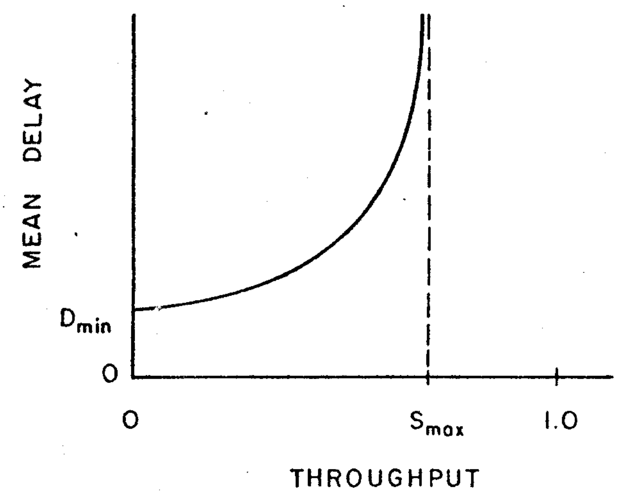

Fig. 2. General delay-throughput characteristic

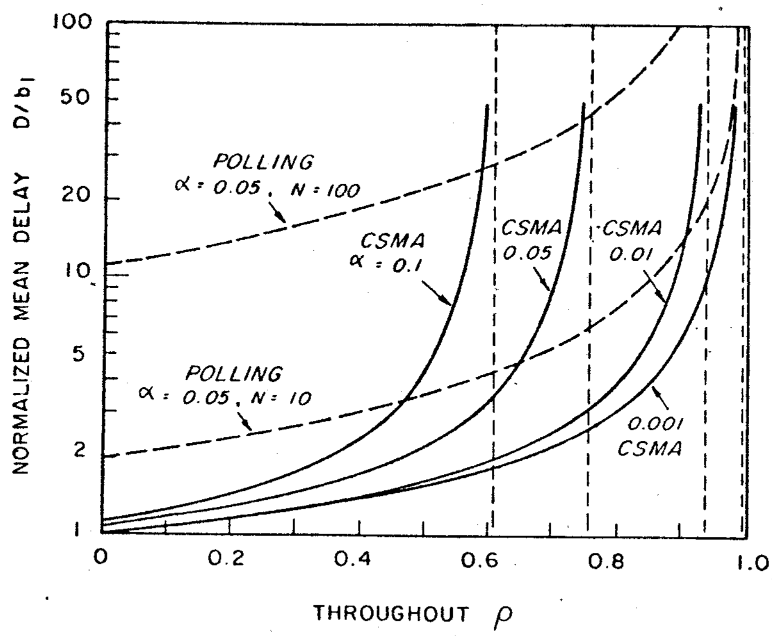

Fig. 3. A comparison of the CSMA/CD and polling protocols.

	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	$(\log_2 N)$ time slots

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