4

Multiple Access Protocols

SIMON S. LAM
The University of Texas at Austin
Austin, Texas

4.1 INTRODUCTION

Packet-broadcasting networks may be defined as packet-switching networks in which the connectivity requirements of the network users are furnished by a broadcast medium (channel). Two examples of broadcast media are satellite and terrestrial radio channels. A less obvious example is that of a multipoint cable. Although multipoint networks have been in use for many years as broadcast networks, it was the ALOHANET using radio channels and a novel contention protocol [ABRA 70, BIND 75a] that first called attention to the concept of packet broadcasting as we know it now. This concept clearly emerged with the packet satellite project [ABRA 73, KLEI 73, ROBE 73, CROW 73, BUTT 74, JACO 77, JACO 78, WEIS 78] and the packet radio project [KAHN 77, KAHN 78] of the Advanced Research Projects Agency. Currently, packet broadcasting networks appear to be important in two areas. The first is satellite data networks. The rapid pace of development of commercial satellite systems in recent years has enabled substantial reductions in satellite system costs [ABRA 75]. Meanwhile, another marketplace is expanding rapidly: local area networks, which although outside the computer room, are confined to local environments (e.g., office complex, manufacturing plant, etc.). Such networks, based on CATV technology, are said to offer real advantages to computing facilities which serve many terminals and/or employ distributed processing [WILL 74, THOR 75, DEMA 76, METC 76, WEST 78]. An extensive bibliography of local computer networks can be found in [CLAR 78, THUR 79].

Of interest in this chapter are protocols for sharing a single broadcast channel by a population of users. The users have random traffic requirements as well as delay constraints. The central problem is that of *conflict resolution* among users desiring channel access. The methods for conflict resolution will be referred to as *multiple access protocols*. We present a classification of such protocols, illustrate each class with description of specific protocols, and compare their performance under various traffic and channel assumptions. To delimit the scope of this chapter, we consider only networks in which everyone in the population of users can "hear" everyone else over the broadcast channel (i.e., we have a one-hop broadcast network). Multi-hop networks with packet repeaters such as those described in [KAHN 78, LAM 80b] are not considered.

The key measure of performance of a multiple access (MA) protocol is its channel throughput versus average delay trade-off characteristic. The throughput of a channel is defined as follows. Let C be the channel transmission rate in bits per second (bps) and let there be on the average P bits in a transmitted block of data. The *channel throughput* S is defined to be the ratio of the rate of successfully transmitted data blocks to the rate C/P. Thus channel throughput is a normalized quantity between 0 and 1. It includes as useful throughput overhead bits contained in data blocks for bit synchronization, addressing, error control, and other network control functions; these overheads are not, however, directly attributable to the MA protocol. Other overheads that are directly attributable to the implementation of an MA protocol are accounted for in the calculation of channel throughput; the maximum achievable channel throughput is thus less than 1 and is of interest as a gross measure of performance.

We next discuss briefly traditional multiple access protocols which are channel-oriented rather than packet-oriented (of interest here). A model for characterizing traffic sources is then presented. Following that, the characteristics of different categories of multiple access protocols are surveyed. Specific protocols and their performance are examined in some detail in Secs. 4.2, 4.3, and 4.4.

4.1.1 Traditional Techniques

The problem of multiple access in the design of satellite systems, for example, has been solved in the past with voice communications in mind. The design objective is to maximize the number of (voice-grade) channels for given constraints of power and bandwidth. The common satellite multiple access techniques are frequency-division multiple access (FDMA), time-division multiple access (TDMA), and code-division multiple access (CDMA) [SCHW 73]. CDMA is also called spread-spectrum multiple access. It is by far the least efficient and has been used mainly for military systems with antijam and security requirements. TDMA is generally more efficient than FDMA; the price paid is an increase in the cost and complexity of the equipment of each user [PRIT 77].

Multiple access protocols have traditionally been channel-oriented. The transmission capacity available is subdivided into separate channels (with FDMA or TDMA). The basic unit for allocation is thus a channel. Channels can be either (1) fixed assigned, or (2) demand assigned to users. With demand assignment, a channel needs to be set aside for signaling among users. (Access to the signaling channel is another multiple access problem! A typical solution for this is TDMA with fixed assignment.) Demand assignment can then be accomplished with either a central controller or a distributed control algorithm [PRIT 77, PUEN 71].

The channel-oriented MA protocols are suitable for voice traffic and may also be suitable for some data traffic. Data communications in general, however, have very diverse requirements, ranging from inquiry–response systems with intermittent traffic to file transfers with large volumes of data. In this chapter we are interested mainly in the class of new packet-oriented MA protocols for time sharing a single broadcast channel. The broadcast channel under consideration for the shared use of a population of users may possibly have been derived at a higher level of resource allocation via FDMA, TDMA, or CDMA. (The allocation scheme at this level will be of no concern to us.) We concentrate on the conflict resolution aspect of the multiple access problem and its performance in terms of throughput and delay. Other aspects of the multiple access problem (such as modulation, coding, and clock synchronization) are classical problems and are beyond the scope of this chapter; see, for example, [ELLI 73, JACO 74, GARD 80].

4.1.2 Traffic Model

The following model will be used to represent the traffic characteristics and transmission requirements of users of a message (or packet)-oriented communication network. Examples of users are human operators of computer terminals, computer programs interacting with such human operators or with other programs/data bases in other machines, and data concentrators for a multiplicity of such traffic sources. We view a user simply as a traffic source that can be modeled as a random point process with instants of message arrivals being the points of interest. A message is defined to be a block of data that has a time-delay constraint associated with it for delivery to a destination user. (In a packet-oriented network, a message may be transported in one or more packets.) The definition above is quite general. For example, a message may be a computer data file that needs to be delivered within a period of hours. It may be a line of characters in an inquiry–response system that needs to be delivered in a fraction of a second. It may be a digital voice sample (8 bits) that has a delay constraint dictated by the real-time voice sampling rate (e.g., 8000 samples per second).

Computer data traffic sources are often described as "bursty." Traffic burstiness is an important characteristic that influences the design of packet communication systems. This concept is briefly reviewed and a quantitative measure of burstiness, called the *bursty factor*, as defined in [LAM 78A], is presented below.

The bursty nature of a data traffic source stems from more than just randomness in message generation time and size. The user-specified message delay constraints to be met for these traffic sources are actually the single most important factor in determining if data traffic sources behave in a bursty manner. Suppose that we are given a traffic source with

$$T = \text{average interarrival time between messages}$$

and

$$\delta = \text{average message delay constraint}$$

where δ can be estimated in practice from the performance specifications of the intended network users. Generally, a user-specified source–destination delay constraint can be broken up into several parts, each part becoming a constraint for a segment of the communication path. δ is defined here to be the constraint for the message transmission time plus any necessary conflict resolution and queueing delays; it excludes, however, propagation delays through the network as well as message-processing delays at the source and destination.

The bursty factor β of the traffic source is defined to be

$$\beta = \frac{\delta}{T} \tag{4.1}$$

Note that β depends only on δ and T, which are inherent user characteristics. (The delay constraint δ is indeed an inherent source characteristic in the eyes of the network designer. Failing to satisfy it means that the user will take his business elsewhere!) Next consider a traffic source that is formed by merging together N sources with different statistics and delay constraints. The bursty factor of the aggregate source is defined to be the sum of the bursty factors of the individual sources:

$$\beta = \beta_1 + \beta_2 + \dots + \beta_N \tag{4.2}$$

The usefulness of β is due to the following observation [LAM 78A]. Suppose that a communication channel is dedicated to a traffic source with bursty factor β and that all delay constraints are met. Then the resulting ratio of peak to average channel data rates (PAR) satisfies

$$\text{PAR} \geq \frac{1}{\beta} \tag{4.3}$$

and the channel throughput S satisfies

$$S \leq \beta \tag{4.4}$$

In other words, β gives an upper bound on the duty cycle of a traffic source. This information is useful to the network designer and is available independent of the communication system eventually provided. A traffic source with a small β ($\ll 1$) is said to be *bursty*.

The result above also says that for bursty users, channel-oriented MA protocols using either fixed assignment or demand assignment (over a period of time $\gg \delta$) are going to be very inefficient ($S \ll 1$). To improve the throughput of a broadcast channel shared by users with random bursty traffic, it is desirable to dynamically allocate transmission capacity on a per message (or packet) basis. This benefits from the multiplexing effect of Eq. (4.2) as well as the scaling effect of large systems discussed in [KLEI 76, sec. 5.1]. The key to realizing these gains is to design MA protocols for resolving channel-access conflicts without excessive overhead.

We note that the definition of bursty factor in Eq. (4.1) does not involve the

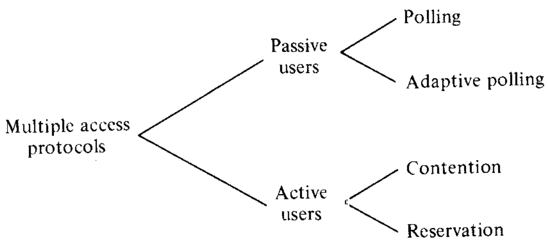

Figure 4.1. Classification of multiple access protocols.

average length of messages generated by a traffic source. Thus it is possible for a traffic source to generate very long messages but is still considered to be very bursty by definition. This average-message-length parameter, we shall see, also strongly affects the performance of MA protocols.

4.1.3 Network Assumptions

We consider a population of N users sharing use of a single broadcast channel. The ith user has a message generation rate of λ_i messages per second. Each message may give rise to one or more fixed-length packets with a mean number of L. Each packet carries a destination address so that when the packet is transmitted over the channel, with no interference from another user, it will be received by the proper addressee(s). Errors due to channel noise will be ignored. Such errors affect MA protocols equally and their effect can be evaluated separately. Notice that the specific connectivity requirements of the population of users are relevant only indirectly through the resulting set of message rates. Given the set of λ_i, it does not matter, for example, whether users want to communicate with each other or they all want to talk to a specific central site. Thus our only concern is the access problem of the broadcast channel.

Each user is capable of sending and receiving data at the channel transmission rate of C bps. In a number of MA protocols, the users are time synchronized so that the channel can be viewed as a sequence of time slots (just as in TDMA). Each time slot can accommodate one data packet. Minislots may also be interleaved with the data slots to accommodate small control packets.

When given access to the broadcast channel, a user can send data to any other user. Thus the MA protocol is simply an algorithm (possibly distributed as well as nondeterministic) for determining the channel-access rights of the users. In some protocols, the access right is not uniquely determined and it is possible for packet transmissions from different users to "collide" in the channel. It is assumed that the collisions are always destructive and that none of the packets involved in a collision can be correctly received. Each packet contains parity bits for error detection.

Suppose that it takes R seconds following a packet transmission for a user to find out whether it suffered a collision. R will be referred to as the *collision detection* *time*. In many systems [ABRA 73, METC 76], the user can find out the outcome of a transmission for himself by monitoring the broadcast channel. In this case, R is approximately equal to the propagation delay of the channel. (The propagation delay is always taken to be the maximum value between any pair of sender and receiver in the network.) If this collision detection mechanism is not possible, then R corresponds to the timeout period of a positive acknowledgment protocol [ABRA 70, BIND 75A].

4.1.4 A Classification of Protocols

The gamut of packet-oriented MA protocols can be classified as shown in Fig. 4.1 and may be illustrated by the following analogy.* Consider a group of students in a classroom who want to say something. Assume that if two or more students talk at the same time, the resulting speech will be unintelligible.

Passive versus Active Talkers

The students may be "passive." A student will talk only when specifically asked to do so by a teacher acting as a central controller. The teacher determines who wants to talk by polling the students one by one. Or the students may be "active." Whoever has something to say will try to do something about getting the opportunity to talk. Among protocols for active talkers, we further differentiate between two categories as follows.

Contention versus Reservation Protocols

Under a contention protocol, there is no attempt to coordinate the talkers. Each student tries to talk whenever he has something to say. In doing so, he might exercise some caution to try to minimize interference with other talkers by observing past activities in the room. The alternative is to use reservation protocols which may use one of the following two types of control.

Centralized versus Distributed Control

Under a reservation protocol, each student who wants to talk raises his hand to signal a request to talk. A teacher may be present to serve as a central controller to determine who should talk next. Alternatively, each student monitors the requests of all students and exercises a distributed algorithm to determine who should talk next.

In the next three sections we illustrate each class of protocols introduced above with descriptions of specific protocols. Performance implications of various channel and traffic parameters are discussed. In particular, a number of the protocols rely upon a short channel propagation delay for their efficiency and are thus not suitable for satellite channels.

*Another way to classify MA protocols, discussed in [LAM 80A] is to think of users desiring channel access as customers forming a distributed queue. The objective of a MA protocol is to identify such customers.

4.2 PROTOCOLS FOR PASSIVE USERS

This class of protocols is commonly known as *polling protocols*. The presence of a central controller is required. Users are passive in the sense that they may access the channel only when specifically polled by the central controller. The controller has two functions: (1) to identify users with data to send (the *ready users*), and (2) to schedule use of the channel by these users.

4.2.1 Conventional Polling Protocols

In conventional polling protocols, the central controller performs the foregoing two functions by polling the population of users one after the other. Upon the arrival of a polling message, the user polled transmits all messages accumulated in his buffer. The time needed to poll every user once and transmit their accumulated data is said to be the polling cycle time t_C. This is just the time between successive polls of a specific user.

A very important parameter determining the efficiency of polling protocols is the total "walk time" in a polling cycle. This is the portion of cycle time attributable to necessary overheads such as channel propagation delay, transmission time of polling and response messages, modem synchronization time, and so on. A detailed breakdown of time elements in a polling cycle can be found in [MART 72].

Minor implementation variations of the foregoing description exist, such as, prioritized polling list, some users polled more than once in a cycle, the amount of data that a user can transmit at a time is limited, and others [MART 72]. These will not be considered.

The two conventional polling protocols most widely used are: roll-call polling and hub polling [SCHW 77, chap. 12]. In *roll-call polling*, the central controller sends a polling message to the user polled. The user transmits his waiting data (if any) and passes control back to the central controller before the next user is polled. In *hub polling*, the central controller initiates a polling cycle by polling the user at the top of its polling list. The user transmits his waiting data (if any) and passes the polling message on to the next user directly. The user at the bottom of the list passes control back to the central controller to complete a polling cycle. The obvious advantage of hub polling over roll-call polling is that the average walk time $\bar{w}$ between users is typically less. However, the individual users need to be more intelligent and reliable than those of roll-call polling.

To illustrate the key performance trade-offs in the design of polling protocols, we consider here some results from a queueing model of N identical users [KONH 74]. The same model applies to both roll-call and hub polling. (A detailed analysis of polling that takes into account various system features and response-time elements can be found in [CHOU 78].)

Suppose that each user has an arrival rate of λ messages per second and an average message transmission time of $1/\mu$ sec.

Define

$$\rho = \frac{\lambda}{\mu}$$

Let $\bar{w}$ be the average walk time between users and $\bar{t}_C$ be the average cycle time. Then it can be shown that [KONH 74]

$$\bar{t}_C = \frac{N\bar{w}}{1 - N\rho} \tag{4.5}$$

The waiting time of a message is defined here to be the interval from its time of arrival to the start of its transmission. An expression for the average waiting time W is derived in [KONH 74]. It can be easily shown that

$$W \geq \frac{\bar{t}_C}{2}(1 - \rho)$$

Thus the simple expression in Eq. (4.5) can be used as an indirect measure of the average delay of a message, which is

$$D = W + \frac{1}{\mu}$$

Equation (4.5) has the usual queueing theory form, with $N\rho$ being the traffic intensity of the broadcast channel. We have in the limit $\bar{t}_C \uparrow \infty$ as $N\rho \uparrow 1$. Observe that $\bar{t}_C$ is also directly proportional to the total walk time $N\bar{w}$ in a polling cycle. Consequently, the delay performance may be greatly degraded if $\bar{w}$ is large (e.g., satellite channel) or N is large.

We next consider a performance booby trap of polling protocols. Suppose that N is increased but a higher-speed channel is used such that the channel utilization $N\rho$ remains the same. This change will in general tend to decrease the average delay D as a result of the scaling effect [KLEI 76, sec. 5.1]. However, with a conventional polling protocol, the same change will affect D adversely as well as follows. Increasing N while keeping $N\rho$ constant increases the $\bar{t}_C/2$ portion of the average delay due to the increased polling overhead $N\bar{w}$!

So far we have assumed users with plenty of buffer capacity (i.e., infinite waiting room for queues). Under this assumption the maximum channel throughput of polling is 1 in the limit of infinite queues.

Consider now users that have limited buffering capacity, so that each user can transmit at most one message at a time. One consequence is that $\bar{t}_C$ is always finite. In addition, the channel throughput S can be easily shown to be bounded as follows:

$$S \leq \frac{1/\mu}{\bar{w} + (1/\mu)} \tag{4.6}$$

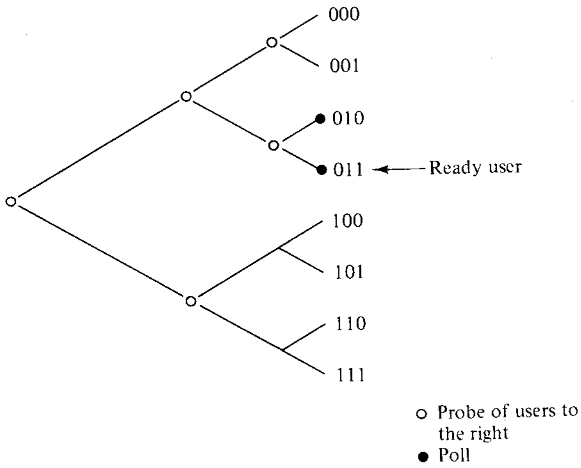

Figure 4.2. Inquiries (probe or poll) needed in an example of eight users.

The upper bound in Eq. (4.6) is the maximum channel throughput of polling for such users. In this case polling becomes extremely inefficient when the traffic consists of very short messages (e.g., terminals doing character-at-a-time transmission [WEST 72]).

4.2.2 Adaptive Polling

With conventional polling, when the network is lightly loaded, it is still necessary to poll every one of the users, although few users are *ready* (have data to send). The mean delay D in this situation is mainly determined by the polling overhead and not by the channel traffic intensity!

For a lightly loaded network, an adaptive polling protocol has been proposed by Hayes [HAYE 78] which significantly reduces the polling overhead. The key idea introduced is called *probing*. When a group of (two or more) users are probed, a polling message is sent by the central controller to all users in the group. Any users with data to send respond by transmitting some signal in the channel. To avoid cancellation by interference, the signal can be some noise energy. Note that if a group of users is probed and none responds, the whole group can be ignored until the next polling cycle. Thus when the network is lightly loaded, significant polling overhead reduction results through probing groups of users. The following implementation is suggested [HAYE 78].

Assume a total of $N = 2^n$ users. Each user is addressed by an n-bit binary number. The central controller can probe a group of users by sending the common prefix of the addresses of the users. At the beginning of each polling cycle, the central controller divides the users into 2^{n-j} groups of 2^j users each. The determination of j will be discussed later. Each of the 2^{n-j} groups is probed separately. If probing a group produces a positive response, each group is divided into two groups, which are then probed separately. This process is repeated until all ready users are identified and serviced.

A poll (of a single user) or a probe (of two or more users) will both be referred to as an *inquiry*. An example is illustrated in Fig. 4.2 with a group of eight users. Only one of them (address 011) is ready. Starting with probing all users ($j = 0$), a total of seven inquiries (five probes and two polls) are needed to complete the polling cycle.

Returning to our general problem, let us consider the following special cases.

Case 1. Pure polling, $j = 0$. The number of inquiries required is 2^n, which is independent of the number of ready users.

Case 2. Pure probing, $j = n$. Pure probing is excellent under very light loading. For instance, if there is exactly one ready user, the number of inquiries is $2n + 1$. However, pure probing is penalized under heavy loading. If all users are ready, the number of inquiries will be $2^{n+1} - 1$.

From these extreme cases, we see that the group size 2^j to be selected at the beginning of each polling cycle is crucial to the performance of the protocol. It should be chosen to minimize the expected number of inquiries in the current polling cycle. Let

$$p = \text{Prob [a user has data to send in the current polling cycle]}$$

and

$$\bar{I}(j) = E[\text{number of inquiries}/j]$$

We want to determine j^* such that

$$\bar{I}(j^*) = \min_{0 \leq j \leq n} \bar{I}(j)$$

Let us compare the probing of a group of 2^j users and probing two groups of 2^{j-1} users. If we probe 2^j users, the probability that none will respond to the probe is

$$\text{Prob[no response}/j] = (1 - p)^{2^j}$$

In this event, the choice of j instead of $j - 1$ saves one inquiry. On the other hand, in the event that there is a positive response from the group of 2^j users, the choice of j instead of $j - 1$ spends one more inquiry. Thus we have

$$\begin{aligned}\Delta\bar{I}(j) &= \bar{I}(j) - \bar{I}(j-1)\\ &= 2^{n-j}[(-1)(1-p)^{2^j} + (1 - (1-p)^{2^j})]\\ &= 2^{n-j}[1 - 2(1-p)^{2^j}]\end{aligned}$$

for $j = 1, \ldots, n$. Note that the term $1 - 2(1 - p)^{2^j}$ increases monotonically with j. Thus $\bar{I}(j)$ is minimized by the largest j such that[1]

[1]This is a simpler proof than the one found in [HAYE 78]. It is interesting to note that in general if addresses use base m numbers so that each node of the tree in Fig. 4.2 has m branches, the optimality condition is $(1 - p)^{m^j} > 1/m$.

$$1 - 2(1 - p)^{2^j} < 0$$

or

$$(1 - p)^{2^j} > \frac{1}{2} \tag{4.7}$$

Equation (4.7) tells the central controller how to determine j at the beginning of each polling cycle as a function of p. An estimate of p is

$$p = 1 - e^{-\lambda t_c} \tag{4.8}$$

where λ is the Poisson rate of message arrivals to a user, and t_c is the duration of the last polling cycle. Equation (4.8) assumes that messages arriving during one polling cycle do not respond positively to an inquiry until the next polling cycle. (This gives a pessimistic view of the operation of the system.) From Eqs. (4.7) and (4.8), we have

$$2^j < \frac{\ln 2}{\lambda t_c} \tag{4.9}$$

At the beginning of each polling cycle, the central controller can determine j^* as a function of λ and t_c using Eq. (4.9)

Performance Summary

The mean cycle time for the adaptive polling protocol is still basically Eq. (4.5) but with the polling overhead term $N\overline{w}$ considerably reduced, especially under light loading. For instance, if there is only one ready user in a polling cycle, the number of inquiries is $2(\log_2 N) + 1$ instead of N. The adaptivity in the protocol assures that there is no penalty during periods of heavy load. These above observations are illustrated in Fig. 4.3, which compares the mean number of inquiries per polling cycle as a function of p for pure polling, pure probing, and the optimum adaptive strategy using Eq. (4.7).

4.3 PROTOCOLS FOR ACTIVE USERS

We consider next two classes of protocols that require ready users to seek channel access actively instead of waiting to be polled. These are called contention and reservation protocols. Under *contention protocols*, there is no attempt to coordinate the ready users to avoid collisions entirely. Instead, each ready user makes his own decision regarding when to access the channel: he exercises caution, however, to minimize interference with other ready users as much as possible. Under *reservation protocols*, a reservation channel is provided for ready users to communicate among themselves such that only one ready user is scheduled for channel access at a time. Since users are geographically distributed, the multiple access problem has not really disappeared; it now exists in the access of the reservation channel for the transmission of small reservation requests.

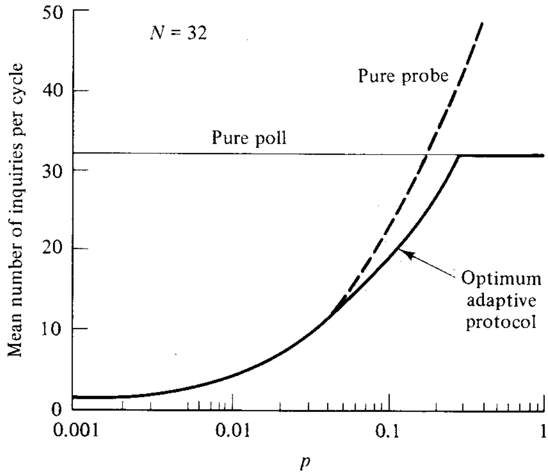

Figure 4.3. Performance of adaptive polling.

4.3.1 Contention Protocols

Unlike polling protocols, the overhead incurred by contention protocols for assigning channel access to ready users is independent of N but instead is dependent on the level of channel traffic. Thus pure contention protocols are suitable for a large population of bursty users (i.e., $N \uparrow \infty$ as $\beta \downarrow 0$ for each user but $N\beta$ remains constant). For a lightly loaded network, the delay performance of contention protocols can be far superior to polling protocols. Below we describe two pure contention protocols (ALOHA and slotted ALOHA), two contention protocols which include elements of reservation (R-ALOHA and CSMA), and an adaptive protocol (URN). We discuss, in the context of slotted ALOHA, the stability problem of contention protocols and the need for adaptive control in these protocols. We then describe a tree algorithm for resolving contention which guarantees channel stability. The algorithm is based on essentially the same idea as that of the adaptive polling algorithm described earlier.

4.3.1.1 The ALOHA Protocol [ABRA 70, BIND 75A]

Under the ALOHA protocol, users are not synchronized in any way. Each user transmits a data packet whenever one is ready. In the event that two or more packets collide (i.e., overlap in time), each user involved realizes this after R seconds (the collision detection time) and retransmits his packet after a randomized delay. As we discuss below, this randomized delay turns out to be crucial to the stability behavior and thus the throughput-delay performance of all contention-based protocols.

In [ABRA 70], Abramson first derived the maximum channel throughput of the ALOHA protocol in the limit of an infinite user population ($N \uparrow \infty$ and for each user $\beta \downarrow 0$). All messages consist of single packets. Hence the aggregate packet "birth process" is a Poisson process at a rate of S packets per packet transmission time.

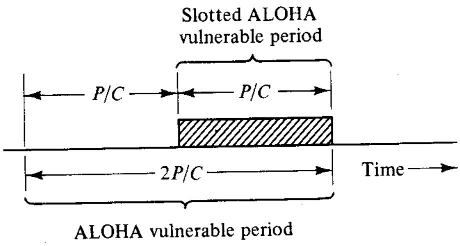

Figure 4.4. ALOHA and slotted ALOHA vulnerable periods of a transmitted packet.

Abramson also made the assumption that the sum of new transmissions and retransmissions in the channel (called *channel traffic*) can be approximated as a Poisson process at a rate of G packets per packet time. Furthermore, statistical equilibrium is assumed. The probability that a transmitted packet is successful is

$$\frac{S}{G} = e^{-2G} \tag{4.10}$$

which is obtained from consideration of Fig. 4.4; each transmitted packet has a vulnerable period of two packet time durations. It will be successful only if no other packet begins transmission within the vulnerable period.

From Eq. (4.10) the maximum possible ALOHA channel throughput is obtained at $G = 0.5$ and for an infinite-user population model (under the foregoing assumptions) is

$$C_A = \frac{1}{2e} \simeq 0.184 \tag{4.11}$$

4.3.1.2 The Slotted ALOHA Protocol [ROBE 72, ABRA 73, KLEI 73]

The slotted ALOHA protocol is just like ALOHA with the additional requirement that the channel is slotted in time. Users are required to synchronize their packet transmissions into fixed-length channel time slots. By requiring synchronization of packet start times, packet collisions due to partial overlaps are avoided and the vulnerable period of a transmitted packet is just the duration of a time slot (see Fig. 4.4). Under the same assumptions given above for ALOHA, we have

$$\frac{S}{G} = e^{-G} \tag{4.12}$$

where S is maximized at $G = 1$. The resulting slotted ALOHA maximum channel throughput for an infinite-user population model is twice that of the unslotted case:

$$C_{SA} = \frac{1}{e} \simeq 0.368 \tag{4.13}$$

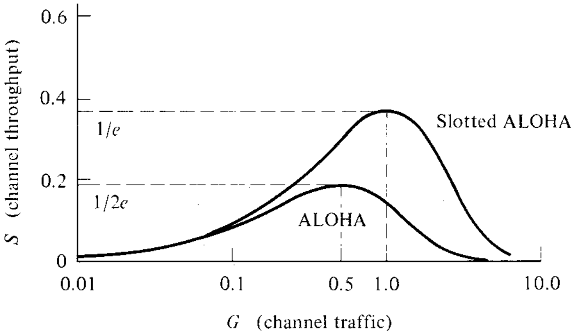

Figure 4.5. ALOHA and slotted ALOHA throughout curves.

Equations (4.10) and (4.12) are plotted in Fig. 4.5.

Two methods of scheduling retransmissions following the detection of a collision have been considered [METC 73, LAM 74]:

1. *Geometrically distributed delay.* A collided packet is retransmitted in a time slot with probability $p < 1$. With probability $1 - p$, the packet is delayed and the Bernoulli trial is repeated in the next time slot.
2. *Uniformly distributed delay.* A time slot is selected for retransmitting a collided packet from the next K time slots chosen equally likely.

For purposes of performance modeling, simulation studies indicate that in many cases the mean value $\bar{k}$ (instead of the exact probability distribution) of the retransmission delay is sufficient for predicting the behavior of a slotted ALOHA channel [LAM 74].

4.3.1.3 Performance Considerations

The analysis of ALOHA and slotted ALOHA given above was based upon three assumptions: (A1) statistical independence of channel traffic, (A2) infinite user population, and (A3) statistical equilibrium. More detailed studies of slotted ALOHA investigated the validity of these assumptions. In particular, it was shown that a necessary condition for (A1) is that the mean randomized delay $\bar{k}$ for retransmissions must be large ($\bar{k} \rightarrow \infty$). However, simulations showed that the maximum channel throughput results given by Eqs. (4.11) and (4.13) are robust; they are already quite accurate if $N \geq 10$ and $\bar{k} \geq 5$ [KLEI 73, LAM 74]. In these same references, the delay performance of slotted ALOHA was first studied for finite values of $\bar{k}$. When the traffic distribution is unbalanced with a mixture of high-rate as well as low-rate users, it was shown that the slotted ALOHA maximum channel throughput of $1/e$ can be considerably improved [ABRA 73].

The validity of assumption (A3) was also examined. It was shown that the slotted ALOHA protocol, without adaptive control, is potentially unstable [METC 73, LAM 74, KLEI 74, KLEI 75A]. (We might have deduced this from the curves in Fig. 4.5, which show two equilibrium values of G for each value of S!) Statistical fluctuations may cause the channel to drift into a saturation state—the channel is filled up with collisions resulting in zero throughput. For an unstable channel, equilibrium conditions assumed earlier exist only for a finite period of time before channel saturation occurs. A Markov chain formulation of the infinite population slotted ALOHA model shows that it is always unstable, in the sense that a stationary probability distribution does not exist [LAM 74, KLEI 75A, FAYO 77].

Fortunately, N must be finite in a real network. In this case slotted ALOHA channels may exhibit stable or unstable behavior, depending upon the parameters N, $\bar{k}$, and S (channel input rate). In [LAM 74, KLEI 74, KLEI 75A], a theory of channel stability behavior is formulated. Specifically, a method for characterizing stable and unstable channels and a quantitative measure of instability for unstable channels are introduced. A theoretical treatment of the adaptive control of unstable channels using a Markov decision model can be found in [LAM 75A]. Various heuristic control algorithms and their performance are presented in [LAM 75B, LAM 79A, LAM 80A]. Several effective feedback control algorithms are proposed in [GERL 77A]. Simulations showed that these techniques are effective means of achieving stability, for initially unstable channels, at the expense of a small amount of delay-throughput performance degradation relative to lower-bound values.

Control algorithms for preventing channel saturation are based upon one or both of the following mechanisms: (1) reducing the probability of retransmitting a collided packet in a time slot (thus increasing the effective $\bar{k}$), and (2) revoking the access right of some users for a period of time (thus reducing the effective N). These two mechanisms have been referred to as retransmission control and input control, respectively [LAM 74, LAM 75A]. Recall that the slotted ALOHA channel throughput S is maximized when the channel traffic rate is 1. The goal of most adaptive control algorithms is to achieve the $G = 1$ condition in the channel. The main difficulty is the acquisition of global network status information by individual users. If, for example, the total number n of ready users can be made known to individual users instantaneously, then an adaptive strategy for realizing the $G = 1$ condition is to have each ready user transmit into the next time slot with probability $1/n$ (provided that $n \geq 1$).

Next we observe that the delay-throughput performance of contention protocols is not entirely independent of N, as we said earlier. When N is increased (accompanied by a decrease in β), slotted ALOHA channels tend to be less stable [LAM 74, KLEI 75A], which affects the delay-throughput performance. However, this is only a second-order effect.

We have discussed the stability problem and adaptive control techniques within the context of slotted ALOHA, which is a pure contention protocol. However, the stability problem is present in all contention-based protocols (many to be described in this section as well as the next section on reservation protocols) and must not be forgotten.

4.3.1.4 The R-ALOHA Protocol [CROW 73, LAM 78B, LAM 80C]

The traffic environment suitable for ALOHA and slotted ALOHA is that of a large population of low-rate bursty users with short messages (one packet per message). The R-ALOHA protocol was originally proposed by Crowther et al. [CROW 73] and is suitable for users who generate long multipacket messages or users with steady input traffic and queueing capability.

The R-ALOHA protocol requires, in addition to time slotting, that time slots are organized into frames. Time slots are identified by their positions in the frame. The duration of a frame must be greater than the maximum channel propagation time between any two users in the network. (This is of concern in practice only for a satellite channel.) Consequently, each user is aware of the usage status of time slots in the previous frame. The network operates without any central control but requires each user to execute the same set of rules for transmitting packets into a time slot depending on the outcome in the same time slot of the previous frame.

A time slot in the previous frame is *unused* if it was empty or contained a collision. Slots unused in the previous frame are available for contention by ready users in exactly the same manner as slotted ALOHA. A slot that had a successful transmission by a user in the previous frame is *used* and is reserved for the same user in the current frame. As a result, such a user now has the equivalent of an assigned TDMA channel for as long as he has traffic to send in it.

We further differentiate between two slightly different protocols depending on whether an end-of-use flag is included in the header of the last packet before a user gives up his reserved slot:

(P1) end-of-use flag not included

(P2) end-of-use flag included

Under (P1), a time slot is always wasted when a user gives up his reserved slot. The trade-off for adopting (P2) is some additional packet processing overhead by each user.

Under the assumption of equilibrium conditions, the channel throughput S_{RA} of R-ALOHA can be expressed in terms of the slotted ALOHA throughput S_{SA} for the contention portion of the channel [LAM 80C]:

$$S_{RA} = \frac{S_{SA}}{S_{SA} + 1/\bar{v}} \qquad \text{under (P1)}$$

or

$$S_{RA} = \frac{S_{SA}}{S_{SA} + [(1 - S_{SA})/\bar{v}]} \qquad \text{under (P2)} \tag{4.14}$$

where $\bar{v}$ is the average number of packets that a user transmits before he gives up a

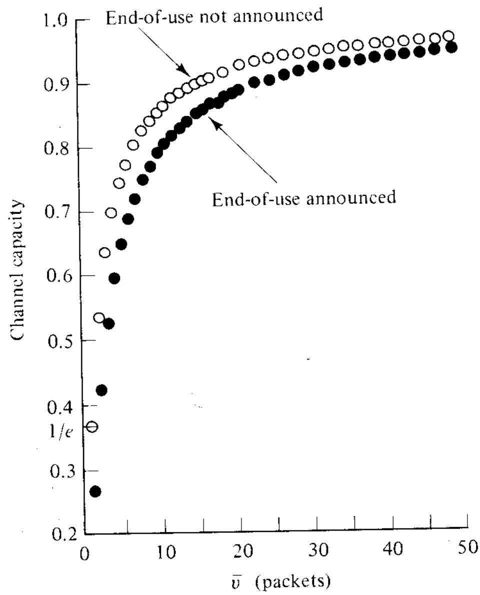

Figure 4.6. Maximum channel throughput versus v for R-ALOHA.

reserved slot. Note that S_{RA} is a monotonic function of S_{SA}, so that the maximum channel throughput of R-ALOHA is

$$C_{RA} = \frac{C_{SA}}{C_{SA} + 1/\bar{v}} \qquad \text{under (P1)}$$

or

$$C_{RA} = \frac{C_{SA}}{C_{SA} + [(1 - C_{SA})/\bar{v}]} \qquad \text{under (P2)}$$

Assuming that $C_{SA} = 1/e$, C_{RA} is plotted in Fig. 4.6 as a function of $\bar{v}$. Note that $\bar{v}$ range from 1 to infinity; thus we have

$$\frac{1}{1+e} \leq C_{RA} \leq 1 \qquad \text{under (P1)}$$

or

$$\frac{1}{e} \leq C_{RA} \leq 1 \qquad \text{under (P2)}$$

Notice that the R-ALOHA protocol adapts itself to the nature of the traffic. At one extreme, it behaves like slotted ALOHA when the users are bursty ($\bar{v}$ = 1). At the other extreme, the channel throughput of R-ALOHA approaches 1 in the $\bar{v} \to \infty$

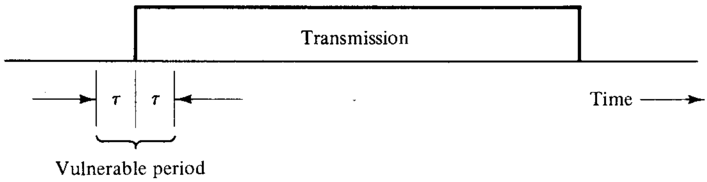

Figure 4.7. Unslotted CSMA vulnerable period.

limit; this is the case for high-rate users who can accommodate very long queues. The reader is referred to [LAM 78B, LAM 79A, LAM 80C] for a queueing and message delay analysis of R-ALOHA as well as simulation results.

4.3.1.5 CSMA Protocols [TOBA 74, KLEI 75B, METC 76, HANS 77, LAM 79B, LAM 80A, TOBA 79]

For broadcast channels with a *short propagation delay*, collisions in the channel can be significantly reduced by requiring each user to sense the channel for the presence of any ongoing transmission before accessing it, hence the name Carrier-Sense Multiple Access (CSMA). A ready user (user with data to send) transmits his data into the channel only if the channel has been sensed idle. Let τ seconds 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 on the modulation technique and channel bandwidth. It was found to be negligible relative to the propagation delay in some cases [KLEI 75B].)

Given that a user sensed the channel to be idle and began a transmission, the vulnerable period of that transmission to collisions is 2τ seconds (see Fig. 4.7). In a short propagation delay environment, this vulnerable period is significantly smaller that those of ALOHA and slotted ALOHA. The channel may also be slotted into minislots of duration τ seconds each, thereby reducing the vulnerable period further to just τ seconds. (Note, however, that τ is the absolute minimum time-slot duration. A larger value is necessary in practice. The reader is referred to [ELLI 73, GARD 80] for a discussion of some techniques for time-synchronizing distributed users.)

Suppose that $\bar{x}$ is the average duration of a successful transmission. The maximum channel throughput of a CSMA protocol depends strongly upon the ratio

$$\alpha = \tau/\bar{x}$$

Specifically, we find out below that C_{CSMA} goes up to 1 if α is decreased to zero (by either letting $\tau \downarrow 0$ or $\bar{x} \uparrow \infty$; see below).

CSMA protocols have been studied extensively in the past within a packet radio network environment by Kleinrock and Tobagi [KLEI 75B, TOBA 74] and later by Hansen and Schwartz [HANS 77]. CSMA protocols were also implemented and analyzed within a multipoint cable network environment [METC 76, WEST 78,

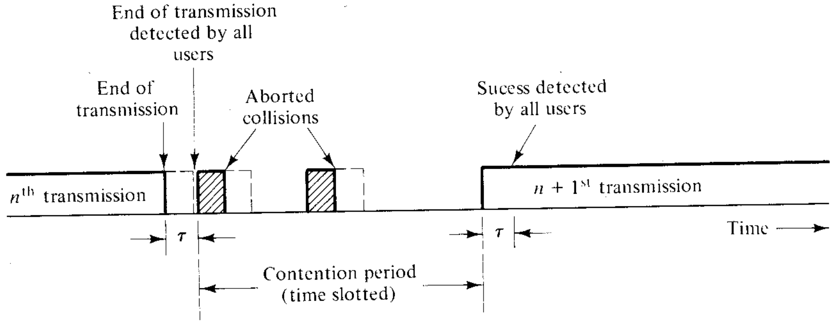

Figure 4.8. Illustration of the CSMA protocol.

LAM 79B, LAM 80A, TOBA 79].The main difference between the two environments is as follows. In cable networks, collisions in the broadcast channel can be detected easily. Thus users involved in a collision can abort their transmissions immediately upon detecting the collision. Mechanisms for detecting collisions and aborting transmissions have been implemented in at least two networks [METC 76, WEST 78]. It appears, however, that this "collision abort" capability is not easily implementable in packet radio networks of interest in [TOBA 74, KLEI 75B].

We present next the CSMA protocol defined and analyzed in [LAM 80A] This particular version of CSMA clearly shows the relationship of CSMA to the slotted ALOHA protocol (in pretty much the same manner as R-ALOHA is related to slotted ALOHA). The analysis of this CSMA protocol is fairly complete. In addition to maximum channel throughput, explicit formulas for average message delay as well as the moment generating function of the number of ready users are available.

A description of the protocol follows; (see also Fig. 4.8). Network users are time synchronized so that following each successful transmission, the channel is slotted in time. To implement the collision abort capability discussed above, the minimum duration of a time slot is 2τ, so that within a time slot if a collision is detected and the collided transmissions are aborted immediately, the channel will be clear of any transmission at the beginning of the next time slot. (In practice, the duration of a time slot needs to be larger than 2τ. The slotted channel assumption is made mainly to facilitate the analysis. In a real network, either a slotted or unslotted channel may be implemented.) The protocol is defined by the two possible courses of action by ready users:

(C1) Following a successful transmission, each ready user transmits with probability 1 into the next time slot.

(C2) Upon detection of a collision, each ready user exercises an adaptive algorithm for selecting its transmission probability (less than 1) in the next time slot.

The adaptive algorithm in (C2) is not specified. Howevewewever, it should be clear that the contention problem in the minislots is exactly the slotted ALOHA problem. An adaptive control algorithm (such as those considered for slotted ALOHA) is needed to guarantee that a successful transmission occurs within a finite number of slots following a collision. The first successful transmission terminates the contention period (see Fig. 4.8).

The maximum channel throughput of the CSMA protocol defined can be obtained as a function of the equilibrium slotted ALOHA throughput S_{SA}. Let C_{SA} be the maximum achievable value of S_{SA}. It is shown in [LAM 80A] that

$$C_{CSMA} = \frac{C_{SA}}{2\alpha + C_{SA}(1+\alpha)} \tag{4.15}$$

We note that C_{CSMA} is equal to one in the $\alpha \downarrow 0$ limit. Recall that $\alpha = \tau/\bar{x}$. In both ground radio and cable environments, τ is typically very short, say 0.001 to 0.1 of a packet transmission time. Furthermore, if we consider users who are capable of accommodating long queues, so that $\bar{x}$ is now the average time to empty a user's queue instead of a single packet transmission time, the channel throughput will be one in the $\bar{x} \rightarrow \infty$ limit.

The CSMA protocol defined above has the desirable property that when the channel is lightly utilized, the delay in identifying and assigning channel access to a ready user is extremely short and is independent of N (unlike polling and probing, considered earlier also for a short-propagation-delay environment). In particular, when there is exactly one ready user, the delay is zero.

In Fig. 4.9 we plot the fraction of transmissions that incur zero delay in gaining channel access (given that the channel is free) as a function of α and channel throughput. These are analytic results obtained with $S_{SA} = 1/e$ in [LAM 80A], where a queueing and message delay analysis of the foregoing CSMA protocol can be found. A comparison of the delay-throughput performance of CSMA and polling is shown in Fig. 4.14 and discussed in Sec. 4.4. The stability problem and adaptive control of CSMA have also been addressed [TOBA 77, HANS 77]. An alternative method of solution to these problems is to view the contention periods as a slotted ALOHA channel; this is the approach taken here as well as in [METC 76].

4.3.1.6 *The URN Protocol* [KLEI 78, YEMI 78]

The slotted ALOHA protocol is an important component of both R-ALOHA and CSMA protocols. We also learned earlier that the slotted ALOHA protocol needs to be adaptively controlled. Furthermore, the goal of most adaptive control algorithms is to achieve the $G = 1$ condition in the channel. Let us assume for the moment that the number n of ready users at any time is in fact known to individual users instantaneously. (We discuss later how to estimate n.) Then an adaptive strategy for achieving the $G = 1$ condition is to have each ready user transmit into the next time slot with probability $1/n$, where $n \geq 1$.

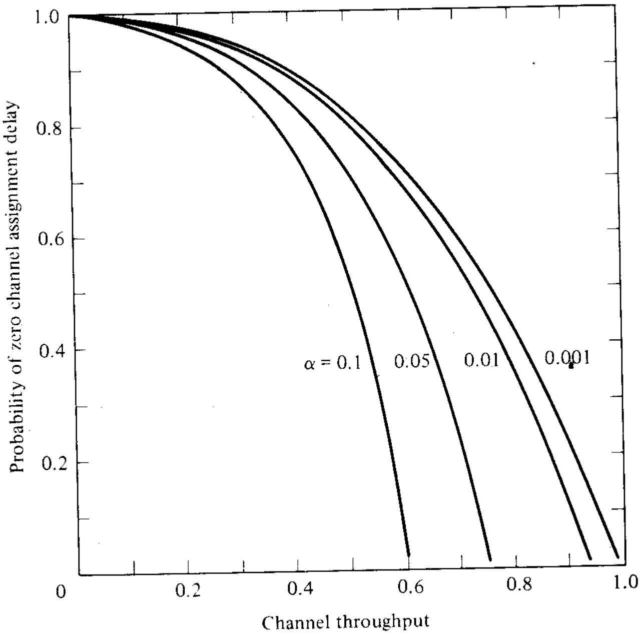

Figure 4.9. Probability of zero channel assignment delay versus throughput.

Kleinrock and Yemini [KLEI 78] proposed an alternative "pure" strategy for ready users to determine whether or not to transmit in the next time slot: the probability of transmission is either 1 or 0. In other words, some users have full channel-access rights, whereas others have none. (Consequently, the URN protocol is said to be *asymmetric*.) A user who has a channel-access right and is also ready transmits into the next time slot. It is possible to prove that optimal strategies are always pure strategies and therefore asymmetric [YEMI 78].

The URN protocol is described using the following URN model. Consider each user as a colored ball in an urn: black for ready, white for not ready. The access protocol is essentially a rule to sample balls from the urn. Let k be the number of balls drawn from the urn. The probability of a successful transmission (throughput) is that of getting exactly one black ball in the sample. The probability is

$$\text{Prob [throughput]} = \frac{\binom{n}{1}\binom{N-n}{k-1}}{\binom{N}{k}} \tag{4.16}$$

where N is the number of balls, and n is the number black balls. Equation (4.16) is maximized when $k = \lfloor N/n \rfloor$ where $\lfloor x \rfloor$ gives the integer part of x. Not only does this value of k maximize the probability of selecting exactly one black ball, but it also gives that the average number of black balls selected is equal to 1 (i.e., $G = 1$).

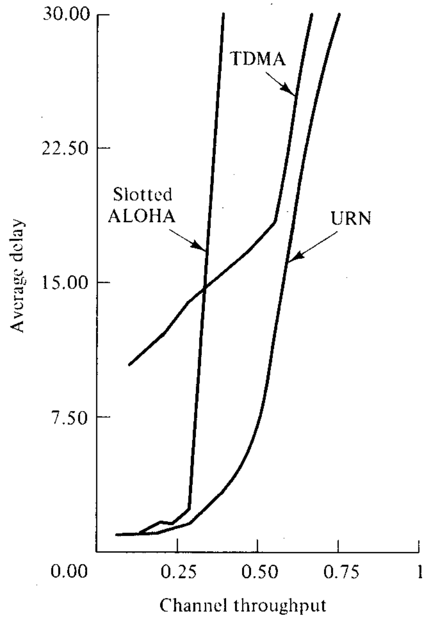

Figure 4.10. Simulated delay-throughput performance of URN.

The URN protocol adapts smoothly to network load fluctuations. When the network is lightly loaded, a large number of users get channel access rights. For instance, $n = 1$ gives risk to $k = N$; all users get access rights, but only one (the lone ready user) is going to make use of it. As the network load increases, n increases and the number k of users getting access rights is reduced. When $n > N/2$, then $k = 1$ and the URN protocol becomes effectively a TDMA protocol (which is most suitable for a heavy load). The maximum channel throughput of URN is thus unity. In Fig. 4.10 simulated delay-throughput results of URN obtained in [KLEI 78] are shown, illustrating the desirable traffic adaptivity of the protocol.

Two questions arise in the implementation of the URN protocol: How does an individual user obtain the up-to-date value of n? How does the protocol obtain coordination of the distributed decisions of individual users?

A solution for estimating n with high accuracy at the expense of a small overhead is proposed in the references. Briefly, it consists of a binary erasure reservation subchannel. An idle user who becomes ready (n increases by 1) sends a message of a few bits in the subchannel. When a ready user turns idle (n decreases by 1), the condition is detected by other users from examining his last packet or its positive acknowledgment in the broadcast channel. An erasure (collision) in the subchannel means that two or more users become ready in the same time slot. In this case, the increase in n is assumed to be 2 (an approximation). The resulting error in the estimate of n was found to be negligible since the probability of more than two users

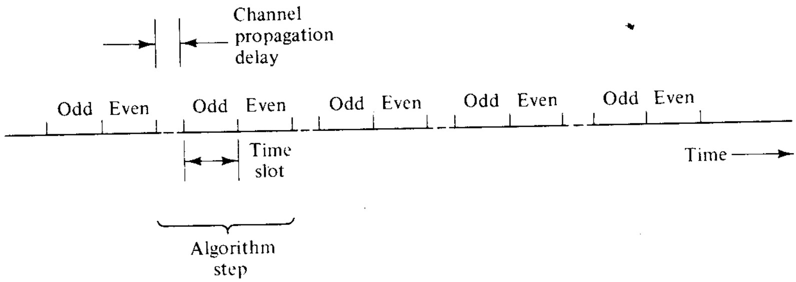

Figure 4.11. Tree algorithm steps.

becoming ready in the same time slot is very small. Furthermore, the estimate of n is corrected every time the network goes idle ($n = 0$). Other heuristic algorithms for estimating n can be found in [LAM 74, LAM 75A, LAM 75B].

Implementation of the URN protocol must also ensure that individual users agree on k, the number of users with access rights, as well as their identity. The optimal $k = \lfloor N/n \rfloor$ to be used is determined by each user from n estimated as described above. The selection of which k users should get access rights may be achieved via identical pseudo-random number generators at individual users, or via a window mechanism as well as other methods. The reader should consult the references for details.

4.3.1.7 A Tree Algorithm for Contention Resolution [CAPE 77, CAPE 79]

A tree algorithm for scheduling retransmission of packets involved in a collision was proposed by Capetanakis [CAPE 77]. A very desirable property of the algorithm is channel stability. The key idea of the algorithm is similar to that of adaptive polling considered earlier. The difference between the two is in implementation. Adaptive polling relies on a central controller probing passive users for status information. On the other hand, implementation of the tree algorithm is distributed, thus requiring users to be able to observe outcomes in the broadcast channel and make decisions. To do so, the channel needs to be time slotted. Furthermore, each algorithm step requires at least a channel propagation time plus two time slots to execute. The channel propagation time in each algorithm step is necessary to enable users to find out the outcomes of the previous algorithm step (see Fig. 4.11). For a satellite channel, some special technique is needed to avoid wasting the large propagation time interval between slot pairs. One method is to time multiplex the channel into subchannels that can be used independently by different user populations. For example, if P/C is the duration of a time slot, R is the channel propagation time, and m is an integer such that

$$\frac{2mP}{C} > R$$

then the satellite channel can be time multiplexed into $m + 1$ subchannels, each accessed by a population of users with the tree algorithm. For a high-speed satellite channel (m is large), much of the benefit from statistically averaging user demands discussed in Sec. 4.1.2 will be lost. Another idea suggested in the reference is to use a tree algorithm on $1/(m + 1)$ of the channel to make reservations for the other $m/(m + 1)$ of the channel (see Sec. 4.3.2).

The binary tree algorithm is described next. Suppose that each user corresponds to a leaf in a binary tree, such as that illustrated in Fig. 4.2 earlier (in the context of probing). Each user handles at most one packet at a time. The number of users can be finite or infinite. An infinite user population corresponds to a Poisson source of packet arrivals; in this case the binary tree extends to infinity. Time slots in the channel are paired into odd and even slots as shown in Fig. 4.11. To facilitate description of the algorithm, a last-in-first-out stack is assumed. (It is obviously not necessary.) Each algorithm step consists of the following three action steps. Initially, the stack is empty and step 3 is executed.

1. Remove the binary tree (or subtree) from top of stack and divide it into two subtrees. Users with access rights in the first subtree transmit in the odd slot; users with access rights in the second subtree transmit in the even slot.
2. Observe outcomes in the two slots. For each slot, discard the subtree if the slot is empty or contains a successful transmission; put the subtree back on the stack if the slot contains a collision.
3. If the stack is empty, give access rights to all new packets that have arrived in the meantime and put the entire tree on stack; go to step 1.

Note that new packet arrivals are only given access rights when the stack empties, which marks the end of one "epoch" and the beginning of the next. Arrivals during one epoch must wait to be served in the next epoch.

We make two observations. First, the objective of the algorithm is to divide the population of users into partitions, each containing at most one contending user. Therefore, the order of the tree search is not important. (In other words, the stack does not have to be last-in-first-out.) Second, the assignment of binary addresses to users does not have to be predetermined. Specifically, following a collision, a contending user may with equal probability join any one of the two resulting subtrees.

For the Poisson source model, Capetanakis [CAPE 77] showed that the tree that minimizes the number of time slots to process η packets, where η is a Poisson random variable, is binary everywhere except for the root node. The optimum degree d_0^* of the root node depends on the mean value of η.

An adaptive version of the tree algorithm (analogous to adaptive polling) can be designed to adjust dynamically the degree of the root node at the beginning of each epoch as a function of the mean number of accumulated packets λh where h is the number of time slots in the previous epoch and λ is the Poisson source rate. The optimum degree of the root node is

$$d_0^* = \begin{cases} 1 & \lambda h \leq 1.70 \\ n & 1.70 + 1.15(n-2) < h \leq 1.70 + 1.15(n-1) \end{cases}$$

The adaptive tree algorithm has a maximum channel throughput of 0.430 packet per slot and guarantees channel stability if $\lambda < 0.430$ packet per slot.

The tree algorithm can be used for contention resolution in CSMA instead of adaptively controlled random retransmission delays for ready users. The trade-off is the tree algorithm's requirement for time slotting; time-synchronizing distributed users to achieve small time slots is a nontrivial problem [ELLI 73, GARD 80].

4.3.1.8 Concluding Remarks

In this section we began with pure contention protocols (ALOHA, slotted ALOHA) and then moved on to more sophisticated contention-based protocols (R-ALOHA, CSMA, URN) with improved delay-throughput performance. A tree algorithm for contention resolution was also shown. These protocols all have distributed control. Each user makes his own decision regarding channel access based solely on observable outcomes in the broadcast channel. In the URN protocol implementation, however, a reservation subchannel is provided for users to communicate with each other in a limited fashion. In Sec. 4.3.2 we describe reservation protocols that require users to cooperate with each other to avoid collisions entirely through use of a reservation subchannel.

4.3.2 Reservation Protocols

The objective of reservation protocols is to avoid collisions entirely. Since users are distributed, a reservation subchannel is necessary for users to communicate with each other. There are two key problems to be solved for most reservation protocols: (1) implementation of the reservation subchannel, and (2) implementation of a queue for the entire population of distributed users (distributed global queue). We also examine some protocols for a short propagation delay environment which do not require a global queue.

4.3.2.1 The Reservation Channel

The original broadcast channel can be either time or frequency multiplexed into a reservation channel and a data channel. An advantage of time-multiplexed channels is that the partition may be variable. With a variable partition, the maximum throughput of data messages can be made very close to one under a heavy network load when users have long queues. With a fixed partition, the overhead lost to the reservation channel is a fixed fraction of the original channel capacity.

With a global queue, there is no conflict in the use of the data channel. However, the multiple access problem of the distributed users has not disappeared. It exists now in the access of the reservation channel. Any of the previously described multiple access protocols can be used. However, for simplicity, most proposed reservation protocols adopt either a fixed assigned TDMA protocol or some version of the slotted

ALOHA protocol. We are faced with the same trade-off as before. A TDMA protocol performs poorly for a large population of bursty users. On the other hand, a slotted ALOHA protocol is independent of N but needs to be adaptively controlled for stable operation.

The high achievable channel throughput of reservation protocols (relative to pure contention) comes about from substantially reducing the volume of traffic requiring conflict resolution, and the concomitant overhead; the reduction is from the totality of data messages to just one short reservation packet per data message (or less). Note that R-ALOHA and CSMA derive their efficiency from essentially the same principle.

Part of the price that one pays for the gain in channel throughput of reservation protocols over contention protocols is an increase in message delay. The minimum delay incurred by a message, excluding message transmission time, is more than twice the channel propagation time. This consideration is important for satellite channels. The minimum delay can be reduced, however, if one can anticipate future arrivals and make reservations in advance! One possible example is packetized digital speech traffic.

4.3.2.2 The Distributed Global Queue

There are two approaches to implement a global queue of requests for a population of distributed users. One is to employ a central controller which tells the ready users when to access the channel; an additional subchannel for controller-to-user traffic is typically required. On the other hand, a distributed control implementation is more interesting and probably more desirable. In this approach each user maintains information on the status of the global queue and makes his own decision as to when is his turn to access the channel. An important problem here is the synchronization of queue status information of all users. This means that reservation packets broadcasted in the reservation channel need to be received correctly by *all* users. In the event of an error, an individual user must be able to detect the presence of error in his queue status information. Any such user who is out of synchronization, as well as new users who have just joined the network, must be able to acquire *queue synchronization* from observing the reservation and data channels within a reasonable duration of time.

Various queue disciplines may be used for the global queue [e.g., random selection, first-in-first-out (FIFO), round-robin, priority based upon type or delay constraint, etc.]. The processing requirement of a sophisticated scheduling algorithm may be quite substantial.

4.3.2.3 A Protocol with Distributed Control [ROBE 73]

The ideas of a reservation subchannel and a distributed global queue were first proposed by Roberts [ROBE 73]. A satellite channel was considered; the reservation protocol proposed, however, is applicable to other broadcast media. It is assumed that the channel is time slotted and that time slots are organized into frames. Each frame consists of a data subframe and a reservation subframe; each slot in the reservation subframe is further subdivided into V smaller slots. The small slots are for reservation packets as well as possibly positive acknowledgment packets and small data packets, to be used on a contention basis with the slotted ALOHA protocol.[2]

Roberts' protocol makes use of the broadcast capability of the channel; a reservation packet successfully transmitted with no interference is received by all users. Each reservation request is for a position in the global queue for a group of packets. The queue discipline proposed by Roberts is FIFO according to the order reservation requests are received. We shall refer to this as the *FIFO protocol.* Each user maintains his copy of the queue status information. It is sufficient for each user to record only the queue length (in number of packets) as well as the queue positions of his own reservations.

It is necessary for a currently inactive user who wants to join the queue, to acquire queue synchronization. In this protocol the queue length information may be supplied in the header of each data packet transmitted. Alternatively, it may be announced periodically by a "master" controller. Note that such queue length information is one propagation delay old when received. To acquire queue synchronization, a user must update the queue length information so received with reservation requests received within a channel propagation time just prior to receiving the queue-length information.

To maintain synchronization among the users, it is necessary and sufficient that each reservation packet that is received correctly by any user be received correctly by all users. This condition may be assured by properly encoding the reservation requests. A simple strategy proposed by Roberts is to send parity-checked copies of requests in triplicate within a reservation packet.

As a result of the distributed nature of queue management, the impact of an error in a user's queue status is to cause some collisions in data slots and to delay some data packets momentarily. However, no catastrophic failure occurs. Users involved in such collisions must declare themselves to be out of synchronization. A user who receives a reservation packet with unrecoverable error must also do the same. Users who are out of synchronization discard their acquired reservations and reacquire queue synchronization in the same fashion as newly activated users described above.

4.3.2.4 Maximum Channel Throughput

The maximum channel throughput of a reservation protocol is $1 - \gamma$, where γ is the minimum fraction of the original channel capacity needed to accommodate the reservation request traffic. Consider the FIFO protocol above. Let L be the average number of data packets per reservation request. Since there are V small slots in a data slot, the ratio of data bits to reservation request bits transmitted is equal to

[2]Our description varies slightly from the reference. In Roberts' original proposal, each reservation subframe is just one data slot long. He also considered a technique for adaptively changing the ratio of the subframe sizes as a function of traffic load. When the queue length is zero, the whole frame is used for making reservations.

$$\bar{v} = VL$$

Suppose that the reservation channel is used by reservation packets only (excluding acknowledgment and small data packets mentioned above) and its maximum channel throughput is C_{SA} under the slotted ALOHA protocol. It can be easily shown that in this case

$$\gamma = \frac{1}{1 + C_{SA}\bar{v}} \tag{4.17}$$

The maximum channel throughput of the reservation protocol is $1 - \gamma$. We make several observations. First, the maximum throughput expression is the same as that for R-ALOHA, with just a slightly different interpretation for $\bar{v}$. Second, it is independent of N. Third, $\bar{v}$ is assumed to be a fixed parameter; thus the maximum channel throughput value is fixed and strictly less than 1.

Now suppose that the partition between data and reservation slots in a frame can be dynamically varied. Also, reservation requests can be piggybacked in the header of scheduled data packets. In this case, when the network is heavily loaded with long message queues at individual users, $\bar{v}$ will become very large; the maximum channel throughput becomes 1 in the $\bar{v} \to \infty$ limit.

Alternatively, if the frame is fixed partitioned and a fixed assignment TDMA protocol is used for the reservation subchannel, we have

$$\gamma = \frac{N}{MV} \tag{4.18}$$

where N is the number of users and M is the number of data slots in a frame.

A fixed assignment TDMA protocol is sometimes preferable to slotted ALOHA for the reservation subchannel since it is simple to implement; unlike slotted ALOHA, adaptive control is not needed. However, Eq. (4.18) shows that it is applicable only for a small user population. One can always increase M to increase channel throughput; a large M is undesirable, however, since the delay of reservation packets becomes large and hence the message delay as well.

The maximum throughput $1 - \gamma$ is plotted in Fig. 4.12 versus N for the two cases given by Eqs. (4.17) and (4.18).

4.3.2.5 Other Protocols with Distributed Control

Another reservation protocol with distributed control was proposed by Binder [BIND 75B]. The channel is divided into time slots which are organized into frames. Let M be the number of slots in a frame. The frame duration is required to be larger than the channel propagation time. Also, the number N of users is required to be less than or equal to M. Each user is fixed assigned a time slot within the frame and sends information concerning his current queue length in the header of each data packet

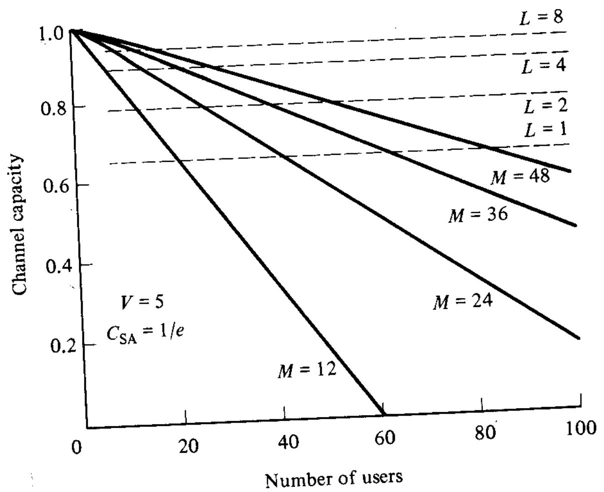

Figure 4.12. Maximum channel throughput versus number of users for reservation protocols.

that he transmits into his fixed assigned slot. Note that this is equivalent to a fixed assigned TDMA reservation subchannel.

The global queue is implemented via distributed control. The global queue status consists of the queue lengths of all ready users (those with nonempty queues). Each user exercises the following rules for channel access. A user may send data in his fixed assigned slot at any time. Any unassigned as well as unused slots (assigned to currently idle users) within a frame are used by the ready users in a round-robin fashion. A user who has been idle can transmit in his fixed assigned (owned) slot to deliberately generate a collision. Such a collision is noted by all users. Another rule dictates that following a collision, only the owner of the slot can use it in the next frame. Thus a special feature of this protocol is that even if a user is in the process of acquiring queue synchronization, he still has the use of a fixed assigned TDMA channel. A disadvantage is that the number of users N must be less than M. The problems of maintaining and acquiring queue synchronization are similar to those of the FIFO protocol above.

Recently, the PODA (Priority-Oriented Demand Assignment) protocol was proposed and implemented in SATNET, a prototype packet satellite network [JACO 77, JACO 78, WEIS 78]. Logically, PODA is an extension of Roberts' FIFO protocol with a much more sophisticated scheduling algorithm designed to handle the requirements of a general-purpose packet satellite network. These requirements include: multiple delay constraints, multiple priority levels, variable message length, fairness, efficient message acknowledgments, and the accommodation of "stream traffic." Stream traffic denotes a class of traffic sources typified by digital voice. Each such traffic source generates a stream of messages with a small interarrival-time variance. Also, the maximum acceptable delay for each message is only slightly larger than the channel propagation time. These characteristics are quite different from what we have considered so far. Thus stream traffic requires special handling apart from individual data messages (datagrams).

Like the FIFO protocol, PODA divides channel time into frames. Each frame consists of an information subframe and a reservation subframe. If slotted ALOHA is used for multiple access in the reservation subframe, the protocol is said to be contention-based and is called CPODA; if fixed assignment TDMA is used, the protocol is called FPODA. In addition to using the reservation subframe, a reservation request can also be piggybacked into the header of a scheduled message. While the total frame size is fixed, the reservation subframe is allowed to grow or shrink according to network loading. If the global queue is empty, the reservation subframe occupies the entire frame. The two types of traffic are distinguished when reservations are made. An explicit reservation is sent for each datagram while a reservation is sent only once for all messages of a particular stream. Each stream reservation contains information defining the stream repetition interval, desired maximum delay relative to this interval, and priority. Whenever the interval starting time is near, a reservation is automatically created and entered into the scheduling queue. The scheduling discipline of the global queue is very elaborate and depends on explicit priority, urgency, and fairness. Some measurement and simulation performance results of CPODA are reported in [GERL 77B, CHU 78]. Queue synchronization is addressed in [HSU 78, WEIS 78]. More on reservation protocols can be found in [BALA 79, BORG 77, BORG 78].

4.3.2.6 Short-Propagation-Delay Environment

The reservation protocols described above all maintain a global queue of reservations, one reservation for a group of packets, for channel access. Consider now networks with a very short propagation delay relative to the transmission time of a packet. Instead of a global queue, the following approach for conflict resolution has been proposed for a ground radio network environment by Kleinrock and Scholl [KLEI 77].

The broadcast channel is divided into minislots interleaved with data slots. Each data slot is preceded by N minislots, where N is the user population size. We shall refer to this class of protocols as the *minislotted protocols*. Before the start of every data slot, a priority ordering exists among the N users. Three priority disciplines were considered in [KLEI 77]: alternating priorities, round robin, and random order. The priority ordering determines the assignment of minislots: one per user. Ready users make their presence known by transmitting (carrier only) into their assigned minislots while idle users keep quiet. The first ready user appearing in a minislot gets the following data slot. Since there are N minislots associated with every data slot, the maximum channel throughput is

$$\frac{1}{1 + N\alpha}$$

where α is the ratio of minislot to data slot duration. Obviously, the performance of this class of protocols is acceptable only if $N\alpha << 1$. The delay-throughput-performance of minislotted protocols was found to be inferior to roll-call polling in many cases [KLEI 77].

To reduce the number of minislots needed for each data slot, the MSAP protocol was proposed. At the end of a packet transmission, the protocol operates as follows:

1. The user who transmitted the last packet is given priority.
2. The priority user can transmit immediately; other users defer via carrier sensing.
3. If the priority user is sensed idle, then one minislot later, access priority is passed on to the next user in sequence, and steps 2 and 3 are repeated.

The number of minislots in between contiguous packet transmissions ranges from 0 to $N - 1$. It should be clear that MSAP is analogous to conventional polling in its scheduling discipline but has distributed control and active users instead of centralized control and passive users. However, a minislot of duration τ (same as that in CSMA) in MSAP is typically much smaller than the corresponding average walk time $\bar{w}$ in polling. This is because the transmission time of a polling message needs to be included in $\bar{w}$ but no corresponding overhead is needed in τ. The delay-throughput performance of MSAP was found to be better than roll-call polling in all cases considered in [KLEI 77]. However, MSAP requires the nontrivial task of time synchronizing users to implement minislots.

Independently, a similar idea was explored by Rothhauser and Wild [ROTH 76] for a bit-synchronous broadcast channel (e.g., a data bus). As a result of bit synchronism, a single bit is sufficient for a user to indicate his status (ready or idle). With a population of N users, N bits are sufficient for identifying all ready users immediately preceding each data transfer phase using the "one-out-of-N" code. During the data transfer phase, ready users can transmit according to a priority sequence known to all users. A multilevel code structure was also proposed in [ROTH 76] that can substantially reduce the number of bits for identifying ready users [to a minimum of $\ln(N)$ bits when only one ready user is present]. This protocol is called *multilevel multiple access* (MLMA). A closer look shows that the multilevel code structure is based on the same "tree search" idea as Hayes' algorithm and the tree algorithm described earlier.

No provision for traffic adaptivity was mentioned in the MLMA proposal. Traffic adaptivity was found to be necessary for efficiency in times of heavy traffic in the other two algorithms.

4.3.2.7 Protocols with Centralized Control

The presence of a central controller eliminates the queue synchronization problem discussed earlier for distributed control protocols. Instead, the central controller manages the global queue, accepts reservations, and informs users when to access the channel. An additional subchannel is typically required for controller-to-user traffic. Since the multiple access problem of the reservation subchannel remains essentially the same as discussed above for distributed control protocols, we will not describe specific protocols.

Protocols with centralized control have been proposed and studied by Tobagi and Kleinrock for radio networks [TOBA 76], by Mark for a data bus [MARK 78], and by Ng and Mark for a satellite network with on-board processing that serves as the central controller [NG 77]. Recently, Mark and Ng proposed a coding scheme, called CMAP, that offers significant overhead reduction over the one-out-of-N code for identifying ready users [MARK 79].

4.4 PERFORMANCE COMPARISONS

Our focus in this chapter is mainly on the sharing and conflict resolution aspect of the multiple access problem. A broadcast channel of C bps is assumed. The performance criteria of interest are channel throughput and message delay. Engineering considerations such as modulation, clock synchronization, coding, and random-noise errors are not within the scope of this chapter. Different broadcast media are differentiated mainly by the effect of their channel propagation delay on an MA protocol's delay-throughput performance. (We note, however, that some protocols do have special requirements and can be implemented in some broadcast media but not others.)

Even within our limited scope of performance, it should be clear to the reader by now that there is no single protocol that is optimum. The performance of a multiple access protocol is strongly dependent on the traffic model and network loading. In general, some traffic characteristics do favor one class of protocols more than others. We list some of them below.

Traffic Model	Multiple Access Protocols Favored
Nonbursty users	Fixed assigned channels (TDMA, FDMA)
Bursty users, short messages	Pure contention
Bursty users, long messages, large N	Reservation protocols with contention reservation channel
Bursty users, long messages, small N	Reservation protocols with fixed TDMA reservation channel

For traffic models that are a fixed or time-varying combination of the models above, "mixed" protocols (such as R-ALOHA, CSMA) and adaptive protocols (such as URN, adaptive polling) may be suitable. To illustrate some of the observations above, we show the delay-throughput performance of representatives of the different classes of protocols under various specific traffic assumptions.

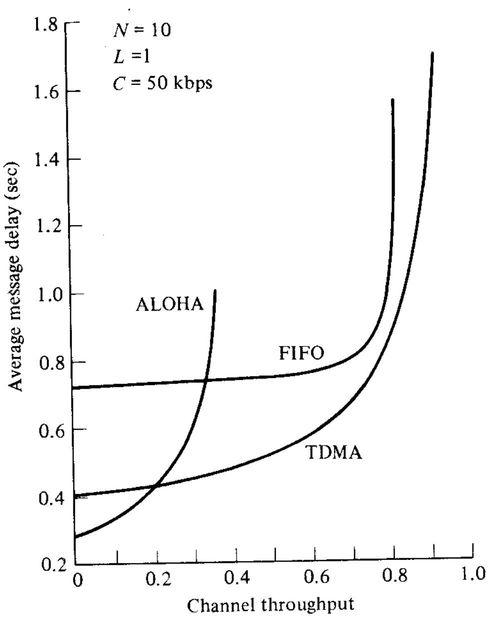

Figure 4.13. Delay-throughput trade-off for 10 users and short messages (one packet per message).

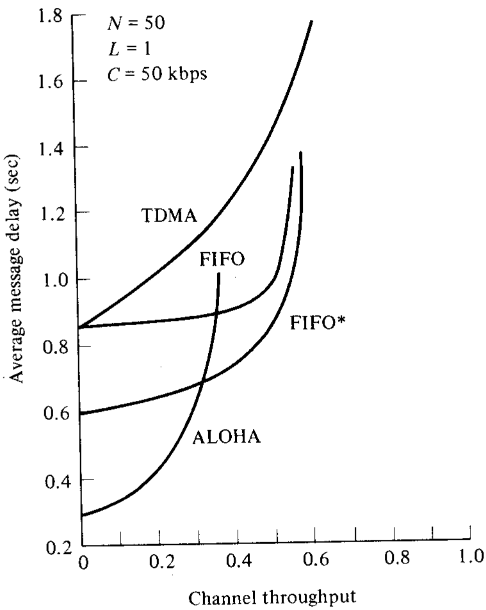

Figure 4.14. Delay-throughput trade-off for 50 users and short messages (one packet per message).

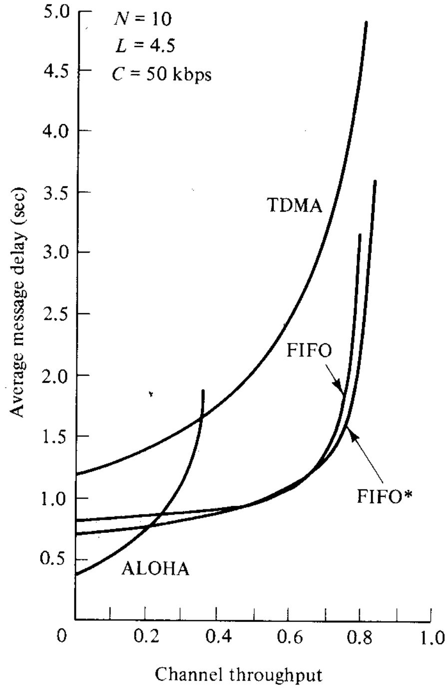

Figure 4.15. Delay-throughput trade-off for 10 users and multipacket messages.

In Figs. 4.13, to 4.15, four protocols are considered: (1) fixed assigned TDMA channels, (2) slotted ALOHA, (3) FIFO modified to use a fixed TDMA reservation channel, and (4) FIFO with a slotted ALOHA reservation channel; they are labeled as TDMA, ALOHA, FIFO, and FIFO*, respectively, in the figures. Delay formulas for TDMA and slotted ALOHA from [LAM 77B, KLEI 73] are used. The expected delay of a message for the reservation protocols is taken to be the sum of the expected delay D_1 incurred by the reservation request and the expected delay D_2 incurred by the message itself after the reservation has been made. To calculate D_1 and D_2, we regard the original broadcast channel at C bps to be split up into two separate channels: a data channel at $(1 - \gamma)C$ bps and a reservation channel at γC bps. D_1 and D_2 are then calculated separately.

In Fig. 4.13 we show results obtained assuming 10 users sharing a 50-kbps satellite channel with a propagation delay of 0.27 second. All messages consist of single packets of 1125 bits each. In the case of FIFO, a frame size $M = 12$ is assumed with $V = 5$ and $\gamma = 1/6$. Observe that the performance of fixed assigned TDMA is the best for this traffic model except when the channel throughput is less than 0.2, where slotted ALOHA gives a smaller delay.

Now suppose that we consider a larger population of more bursty users than the traffic model above. Let N be increased from 10 to 50 (so that the bursty factor of each user is now 1/5 of that of the model above). Figure 4.14 shows that TDMA has the worst performance for this new traffic model. FIFO, with $M = 24$ and $\gamma = 5/12$, also has poor performance, due to the large reservation channel overhead needed for 50 users. FIFO*, with $\gamma = 0.4$ assuming $C_{SA} = 0.3$, has better performance than FIFO, but the reservation channel overhead is still quite large since each message consists of a single packet only. The delay-throughput performance of slotted ALOHA is independent of the change from the first to the second traffic model and appears to be the most suitable protocol for the second traffic model, provided that a channel throughput of about 0.3 or less is acceptable. If a channel throughput of more than 0.3 is desired, FIFO* should be employed.

Now suppose that we are back to having 10 users but that the data traffic consists of long messages (eight packets each) as well as short messages (one packet each) in equal number. The average message length L is increased from 1 to 4.5 packets. Delay-throughput results for the four protocols are shown in Fig. 4.15. The delay curve for slotted ALOHA is plotted using the delay formula for multipacket messages in [LAM 74]. For this traffic model, the reservation protocols have the best performance except at a channel throughput of less than 0.2, where slotted ALOHA is better. Slotted ALOHA, as we know, was not designed to take advantage of the presence of multipacket messages. Simulations also showed that when a large number of multipacket messages are present in the input traffic, the slotted ALOHA channel becomes more unstable [LAM 74].

Some protocols are designed to take advantage of a short-propagation-delay environment. They include CSMA protocols that are contention-based, the reservation protocols MSAP and MLMA, as well as polling protocols. Reservation protocols that are not contention-based and polling protocols have similar delay-throughput characteristics since the conflict resolution overhead in each case is proportional to N, the number of users. In particular, MSAP and polling are characterized by essentially the same delay formula; any difference in performance is just a consequence of different values for the minislot duration τ in MSAP and the corresponding average walk time $\bar{w}$ in polling [KLEI 77]. There are, of course, differences that do not show up explicitly in the delay-throughput performance, such as distributed control in MSAP versus centralized control in polling, and the need for time-synchronizing distributed users in MSAP but not in polling.

The delay-throughput performance of CSMA and roll-call polling are compared in Fig. 4.16 using the CSMA delay formula from [LAM 79B] and polling delay formula from [KONH 74]. The delay results shown for polling assume Poisson message arrivals and one packet per message. The ratio of propagation delay to the packet transmission time is $\alpha = 0.05$. The ratio of data to polling message length is $\bar{v} = 10$. Queueing of messages at indivdual users is assumed; hence the channel throughput approaches 1 when queues become very long under heavy traffic. Delay-throughput curves are shown for both 10 users and 100 users. The corresponding delay-throughput performance of CSMA at $\alpha = 0.05$ is independent of the number of users. Since the analysis in [LAM 79B] assumes that individual users handle one packet at a time, the maximum channel throughput is less than 1. As before, we observe that CSMA, being a contention-based protocol, is superior to polling when

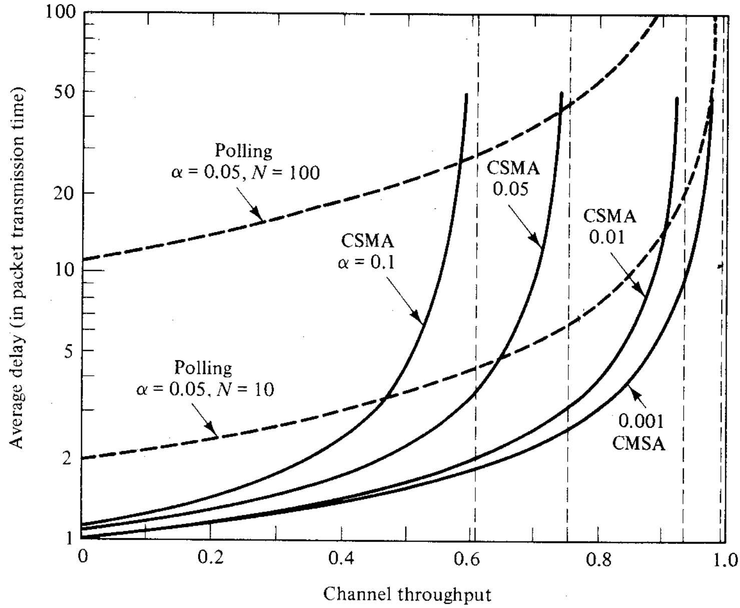

Figure 4.16. Comparison of CSMA and polling delay-throughout performance.

the channel throughput is low but becomes inferior when the channel throughput is increased to near 1. 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; the amount of overhead per message is reduced. Hence as the network load is increased from 0 to 1, the delay performance of CSMA if 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 approaches 1 in the limit of infinitely long queues at individual users.

4.5 CONCLUSIONS

We considered the problem of interconnecting distributed users via a broadcast channel and surveyed a wide class of multiple access protocols. The throughput-delay performance characteristics of these protocols were examined and compared. Our emphasis has been on packet-oriented protocols. Channel-oriented protocols, however, are currently prevalent in existing systems and are in many cases more cost effective than packet protocols. Specifically, in broadcast networks with a large population of users, the total user interface cost, proportional to N, tends to dominate the network cost. Packet protocols become important only when the cost of user interface to a (shared) wideband broadcast medium is reduced to the point that a large population of users can be economically interconnected directly via the broadcast medium (i.e., without the user of a traffic multiplexor/concentrator to justify the cost of an interface). Trends in the costs of electronics and computing seem to be most encouraging toward packet protocols in the foreseeable future. Some specific observations follow.

With the current cost ($\geq$ \$100,000) and size (antenna size $\geq$ 5 m) of satellite earth stations*, it would not be economical to install a separate earth station at a packet network node unless it has a substantial amount of traffic (100 kbps or more [ROBE 78]). At such a traffic level, traditional channel-oriented multiple access protocols will perform very well; there will be no need for the more complex packet protocols. However, the advent of satellite systems at the higher frequencies (e.g., 14/12, 30/20 GHz) than are currently available will facilitate the development of small, inexpensive earth stations (say, 1-to-3-m antenna size) that can be sited almost anywhere for low-rate users. The packet protocols described herein will be very important in this new environment. Other developments in satellite communications that will affect the development of multiple access protocols include multibeam satellites (thus the broadcast network assumption in this paper needs to be modified) and the availability of on-board processing capability [PRIT 77, JACO 78].

In recent years, packet communication has made significant advances to reduce the cost of long-distance backbone networks. Local distribution and collection of data (between a central site and a population of terminals), however, remains by far the most expensive portion of many communication networks. We are addressing local area networks which, although outside the computer room, are confined to local environments, such as office complexes and manufacturing plants, without the use of common-carrier facilities. Local area networks serve several functions: for local interconnection, for access to local computing facilities, for access to a gateway into a backbone network, and so on. The broadcast media of interest here include CATV, optical fiber, and radio. All of these are currently under intensive research and development. In this environment, packet protocols will again be important in networks that can economically interconnect a large population of users. With the availability of inexpensive microprocessors, the key to implementing such packet broadcast networks reduces to the development of inexpensive transceivers for the specific broadcast medium. The economic viability of packet broadcast networks using CATV-based technology has already been demonstrated at data rates of several Mbps [DEMA 76, METC 76, CLAR 78, WEST 78].

In conclusion, we have examined the basic principles and performance tradeoffs of multiple access protocols for broadcast networks. For the sake of clarity, we have often omitted specific implementation details. Therefore, the protocols described herein form the basic elements of more sophisticated operational protocols. In the future, more flexible and reliable protocols will be needed to accommodate the increasing volume and diversity of data traffic, including digital voice, facsimile, and

*These are 1978 figures when this chapter was first written [LAM 79C].

word processing systems, in addition to the present transaction-based terminal and file transfer traffic.

ACKNOWLEDGMENTS

The final manuscript has benefited tremendously from the comments and suggestions of the editor. This work was supported by the National Science Foundation under Grant ECS78-01803.

REFERENCES

ABRA 70 Abramson, N., "THE ALOHA SYSTEM—Another Alternative for Computer Communications," *AFIPS Conf. Proc.*, Vol. 37, AFIPS Press, Montvale, N.J., 1970, pp. 281–285.

ABRA 73 Abramson, N., "Packet Switching with Satellites," *AFIPS Conf. Proc.*, Vol. 42, AFIPS Press, Montvale, N.J., 1973, pp. 695–702.

ABRA 75 Abramson, N., and E. R. Cacciamani, "Satellites: Not Just a Big Cable in the Sky," *IEEE Spectrum*, Vol. 12, September 1975.

BALA 79 Balagangadhar, M. M., and R. L. Pickholtz, "Analysis of a Reservation Multiple Access Technique for Data Transmission via Satellites," *IEEE Trans. Commun.*, October 1979.

BIND 75A Binder, R., et al., "ALOHA Packet Broadcasting—A Retrospect," *Nat. Comput. Conf., AFIPS Conf. Proc.*, Vol. 44, 1975, pp. 201–215.

BIND 75B Binder, R., "A Dynamic Packet-Switching System for Satellite Broadcast Channels," *ICC '75 Conf. Rec.*, IEEE, New York, 1975, pp. 41-1 to 41-5.

BORG 77 Borgonovo, F., and L. Fratta, "A New Technique for Satellite Broadcast Communication," *Proc. 5th Data Commun. Symp.*, Snowbird, Utah, September 1977.

BORG 78 Borgonovo, F., and L. Fratta, "SRUC: A Technique for Packet Transmission on Multiple Access Channel," *Proc. ICCC*, Kyoto, Japan, September 1978.

BUTT 74 Butterfield, S. C., R. D. Rettberg, and D. C. Walden, "The Satellite IMP for the ARPA Network," *Proc. 7th Hawaii Int. Conf. Syst. Sci.*, Computer Nets Supplement, January 1974.

CAPE 77 Capetanakis, J. I., "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations," Ph.D., dissertation, Department of Electrical Engineering and Computer Science, M.I.T., August 1977.

CAPE 79 Capetanakis, J. I., "Generalized TDMA: The Multi-accessing Tree Protocol," *IEEE Trans. Commun.*, October 1979.

CHOU 78 Chou, W., "Terminal Response Time on Polled Teleprocessing Networks," *Proc. Comput. Networking Symp.*, December 1978.

CHU 78 Chu, W. W., and W. E. Naylor, "Measurement and Simulation Results of C-PODA Protocol Performance," *NTC '78 Conf. Rec., Natl. Telecommun. Conf.*, Birmingham, Ala., December 1978.

CLAR 78 Clark, D. D., K. T. Pogram, and D. P. Reed, "An Introduction to Local Area Networks," *Proc. IEEE*, Vol. 66, November 1978.

CROW 73 Crowther, W., et al., "A System for Broadcast Communication: Reservation-ALOHA," *Proc. 6th HICSS*, University of Hawaii, Honolulu, January 1973.

DEMA 76 DeMarines, V. A., and L. W. Hill, "The Cable Bus in Data Communications," *Datamation*, August 1976.

ELLI 73 Ellingson, C., and R. Kulpinski, "Dissemination of System Time," *IEEE Trans. Commun.*, Vol. COM-21, May 1973.

FAYO 77 Fayolle, G., et al., "Stability and Optimal Control of the Packet Switching Broadcast Channel," *J. ACM*, Vol. 24, July 1977.

GARD 80 Gardner, F. M., and W. C. Lindsey, eds., *Special Issue on Synchronization, IEEE Trans. Commun.*, August 1980.

GERL 77A Gerla, M., and L. Kleinrock, "Closed Loop Stability Controls for S-ALOHA Satellite Communication," *Proc. 5th Data Commun. Symp.*, Snowbird, Utah, September 1977.

GERL 77B Gerla, M., L. Nelson, and L. Kleinrock, "Packet Satellite Multiple Access: Models and Measurements," *Conf. Rec. Natl. Telecommun. Conf.*, Los Angeles, December 1977.

HANS 77 Hansen, L. W., and M. Schwartz, "An Assigned-Slot Listen-before-Transmission Protocol for a Multiaccess Data Channel," *Conf. Rec. Int. Conf. Commun.*, Chicago, June 1977.

HAYE 78 Hayes, J. F., "An Adaptive Technique for Local Distribution," *IEEE Trans. Commun.*, Vol. COM-26, August 1978.

HSU 78 Hsu, N., and L. Lee, "Channel Scheduling Synchronization for the PODA Protocol," *Conf. Rec. ICC '78*, Toronto, June 1978.

JACO 74 Jacobs, I. M., "Practical Applications of Coding," *IEEE Trans. Inf. Theory*, Vol. IT-20, May 1974.

JACO 77 Jacobs, I., et al., "CPODA—A Demand Assignment Protocol for Satnet," *Proc. 5th Data Commun. Symp.*, Snowbird, Utah, September 27–29, 1977.

JACO 78 Jacobs, I. M., R. Binder, and E. V. Hoversten, "General Purpose Packet Satellite Networks," *Proc. IEEE*, Vol. 66, November 1978.

KAHN 77 Kahn, R. E., "The Organization of Computer Resources into a Packet Radio Network," *IEEE Trans. Commun.*, Vol. COM-25, January 1977.

KAHN 78 Kahn, R. E., et al., "Advances in Packet Radio Technology," *Proc. IEEE*, Vol. 66, November 1978.

KLEI 73 Kleinrock, L., and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel," *AFIPS Conf. Proc.*, Vol. 42, AFIPS Press, Montvale, N.J., 1973, pp. 703–710.

KLEI 74 Kleinrock, L., and S. S. Lam, "On Stability of Packet Switching in a Random Multi-access Broadcast Channel," *7th Hawaii Int. Conf. Syst. Sci.*, Honolulu, January 1974; *Proc. Spec. Subconf. on Comput. Nets*, 1974.

KLEI 75A Kleinrock, L., and S. S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," *IEEE Trans. Commun.*, Vol. COM-23, April 1975, pp. 410–423.

KLEI 75B Kleinrock, L., and F. A. Tobagi, "Packet Switching in Radio Channels, Part 1: Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics," *IEEE Trans. Commun.*, Vol. COM-23, No. 12, December 1975, pp. 1400–1416.

KLEI 76 Kleinrock, L., *Queueing Systems*, Vol. 2: *Computer Applications,* Wiley-Interscience, New York, 1976.

KLEI 77 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. Commun.*, Chicago, June 1977.

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

KONH 74 Konheim, A. G., and B. Meister, "Waiting Lines and Times in a System with Polling," *J. ACM*, Vol. 21, July 1974.

LAM 74 Lam, S. S., "Packet Switching in a Multi-access Broadcast Channel with Application to Satellite Communication in a Computer Network," Ph.D. dissertation, Computer Science Department, University of California, Los Angeles, March 1974.

LAM 75A Lam, S. S., and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," *IEEE Trans. Commun.*, Vol. COM-23, September 1975, pp. 891–904.

LAM 75B Lam, S. S., 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, pp. 143–153.

LAM 77A Lam, S. S., "Satellite Multiaccess Schemes for Data Traffic," *Conf. Rec. Int. Conf. Commun.*, Chicago, June 1977, pp. 37.1-19 to 37.1-24.

LAM 77B Lam, S. S., "Delay Analysis of a Time Division Multiple Access (TDMA) Channel," *IEEE Trans. Commun.*, Vol. COM-25, December 1977, pp. 1489–1494.

LAM 78A Lam, S. S., "A New Measure for Characterizing Data Traffic," *IEEE Trans. Commun.*, Vol. COM-26, January 1978.

LAM 78B Lam, S. S., "An Analysis of the R-ALOHA Protocol for Satellite Packet Switching," *Conf. Rec. ICC 78, Int. Conf. Commun.*, Toronto, June 1978.

LAM 79A Lam, S. S., "On Protocols for Satellite Packet Switching," *Conf. Rec. Int. Conf. Commun.*, Boston, June 1979.

LAM 79B Lam, S. S., "A Study of the CSMA Protocol in Local Networks," *Proc. 4th Berkeley Conf. Distributed Data Manage. Comput. Networks*, San Francisco, August 1979.

LAM 79C Lam, S. S., "Satellite Packet Communication—Multiple Access Protocols and Performance," *IEEE Trans. Commun.*, October 1979.

LAM 80A Lam, S. S., "A Carrier Sense Multiple Access Protocol for Local Networks," *Comput. Networks*, February 1980.

LAM 80B Lam, S. S., "A Packet Network Architecture for Local Interconnection," *Conf. Rec. Int. Conf. Commun.*, Seattle, June 1980.

LAM 80C Lam, S. S., "Packet Broadcast Networks—A Performance Analysis of the R-ALOHA Protocol," *IEEE Trans. Comput.*, July 1980.

MARK 78 Mark, J. W., "Global Scheduling Approach to Conflict-Free Multiaccess via a Data Bus," *IEEE Trans. Commun.*, Vol. COM-26, September 1978.

MARK 79 Mark, J. W., and S. F. W. Ng, "A Coding Scheme for Conflict-Free Multiaccess Using Global Scheduling," *IEEE Trans. Commun.*, September 1979.

MART 72 Martin, J., *Systems Analysis for Data Transmission*, Prentice-Hall, Englewood Cliffs, N.J., 1972.

METC 73 Metcalfe, R. M., "Steady-State Analysis of a Slotted and Controlled ALOHA System with Blocking," *Proc. 6th Hawaii Int. Conf. Syst. Sci.*, Honolulu, January 1973.

METC 76 Metcalfe, R. M., and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Commun. ACM*, Vol. 19, No. 7, July 1976.

NG 77 Ng, S. F. W., and J. W. Mark, "A Multiaccess Model for Packet Switching with a Satellite Having Some Processing Capability," *IEEE Trans. Commun.*, Vol. COM-25, January 1977, pp. 128–135.

PRIT 77 Pritchard, W. L., "Satellite Communication—An Overview of the Problems and Programs," *Proc. IEEE*, Vol. 65, March 1977.

PUEN 71 Puente, J. G., W. G. Schmidt, and A. M. Werth, "Multiple-Access Techniques for Commercial Satellites," *Proc. IEEE*, Vol. 59, February 1971.

ROBE 72 Roberts, L. G., "ALOHA Packet System with and without Slots and Capture," ARPANET Satellite System Note 8, (NIC 11290), June 1972; reprinted in *Comput. Commun. Rev.*, Vol. 5, April 1975.

ROBE 73 Roberts, L., "Dynamic Allocation of Satellite Capacity through Packet Reservation," *AFIPS Conf. Proc.*, Vol. 42, AFIPS Press, Montvale, N.J., 1973, pp. 711–716.

ROBE 78 Roberts, L., "The Evolution of Packet Switching," *Proc. IEEE*, Vol. 66, November 1978.

ROTH 76 Rothhauser, E. H., and D. Wild, "MLMA: A Collision-Free Multi-access Method," IBM Research Report RZ802, November 1976.

SCHW 73 Schwartz, J. W., and M. Muntner, "Multiple-Access Communications for Computer Nets," *Computer-Communication Networks*, N. Abramson and F. Kuo, eds., Prentice-Hall, Englewood Cliffs, N.J., 1973.

SCHW 77 Schwartz, M., *Computer-Communication Network Design and Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1977.

THOR 75 Thornton, J. E., et al., "A New Approach to Network Storage Measurement," *Computer Design*, November 1975.

THUR 79 Thurber, K. J., and H. A. Freeman, "A Bibliography of Local Computer Network," *Comput. Commun. Rev.*, April 1979.

TOBA 74 Tobagi, F. A., "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," Ph.D. dissertation, Computer Science Department, University of California, Los Angeles, Calif., December 1974.

TOBA 76 Tobagi, F. A., and L. Kleinrock, "Packet Switching in Radio Channels: Part III. Polling and (Dynamic) Split-Channel Reservation Multiple Access," *IEEE Trans. Commun.*, Vol. COM-24, No. 8, pp. 832–845, August 1976.

TOBA 77 Tobagi, F. A., and L. Kleinrock, "Packet Switching in Radio Channels: Part IV—Stability Considerations and Dynamic Control in Carrier Sense Multiple Access," *IEEE Trans. Commun.*, Vol. COM-25, October 1977.

TOBA 79 Tobagi, F. A., and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," *Proc. Local Area Commun. Network Symp.*, Boston, May 1979.

WEIS 78 Weissler, R., et al., "Synchronization and Multiple Access Protocols in the Initial Satellite IMP," *Proc. Fall COMPCON '78*, Washington, D.C., September 1978.

WEST 72 West, L. P., "Loop-Transmission Control Structures," *IEEE Trans. Commun.*, Vol. COM-20, No. 3, Part II, June 1972, pp. 531–539.

WEST 78 West, A., and A. Davison, "CNET—A Cheap Network for Distributed Computing," Department of Computer Science and Statistics, Queen Mary College, University of London, Rep. TR120, March 1978.

WILL 74 Willard, D. G., "A Time Division Multiple Access System for Digital Communication," *Comput. Des.*, June 1974.

YEMI 78 Yemini, Y., "On Channel Sharing in Discrete-Time Packet Switched, Multi-access Broadcast Communication," Ph.D. dissertation, Computer Science Department, University of California, Los Angeles, Calif., 1978.