MULTIPLE ACCESS SCHEMES

Communication channels are major components of computer communication networks. They provide the physical mediums over which signals representing data are transmitted from one node of the network to another node. Communication channels can be classified into two main categories: point-to-point channels and shared channels. Typically, the backbone of wide area networks (WAN) consists of point-to-point channels, whereas local area networks (LAN) use shared channels.

Point-to-point channels are dedicated to connecting a pair of nodes of the network. They are usually used in fixed topology networks, and their cost depends on many parameters such as distance and bandwidth. An important characteristic of these channels is that nodes do not interfere with each other; in other words, transmissions between a pair of nodes has no effect on the transmissions between another pair of nodes, even if a node is common to the two pairs.

Shared channels are used when point-to-point channels are not economical or not available or when dynamic topologies are preferable. In a shared channel, called also a broadcast channel, several nodes can potentially transmit and/or receive messages at the same time. Shared channels appear naturally in radio networks, satellite networks, and some local area networks (e.g., Ethernet). Their deployment is usually easier than point-to-point channels. An important characteristic of shared channels is that transmissions of different nodes interfere with each other; specifically, one transmission coinciding in time with another may cause none of them to be received. This means that the success of a transmission between a pair of nodes is no longer independent of other transmissions.

To have successful transmissions in shared channels, interference must be avoided or at least controlled. The channel allocation among the competing nodes is critical for proper operation of the network. This article focuses on access schemes to such channels known as multiple access schemes. These schemes are nothing more than channel allocation rules that determine who goes next on the channel, aiming at some desirable network performance characteristics. Multiple access schemes belong to a sublayer of the data link layer called the medium access control layer (MAC), which is especially important in LANs.

Multiple access schemes are natural not only in communication systems but also in many other systems such as computer systems, storage facilities, or servers of any kind, where resources are shared by a number of nodes. In this article we mainly address shared communication channels.

One way to classify multiple access schemes is according to the level of contention that is allowed among the nodes of the network. On the one hand, there are the conflict-free schemes that ensure that each transmission is successful, namely, it will not be interfered with by any other transmission. On the other hand, there are the contention-based schemes that do not guarantee that a transmission will be successful, namely, it might be interfered with by another transmission.

Conflict-free transmissions can be achieved by allocating the shared channel in an adaptive or nonadaptive (static) manner. Two common static allocations are the time division multiple access (TDMA), where the entire available bandwidth is allocated to a single node for a fraction of the time, and the frequency division multiple access (FDMA), where a fraction of the available bandwidth is allocated to a single node for all the time. Adaptive allocations are usually based on demands so that nodes that are idle use only little of the shared channel, leaving the majority of their share to other more active nodes. Adaptive allocations can be done by various reservation schemes using either central or distributed network control. Polling algorithms illustrate central control, whereas ring networks generally use distributed control based on token-passing mechanisms. It is important to note that idle nodes consume their portion of the shared channel when conflict-free schemes are used. The aggregate channel portion of idle nodes becomes significant when the number of potential nodes in the system is very large to the extent that conflict-free schemes might become impractical.

When contention-based schemes are used, it is essential to devise algorithms that resolve conflicts when they occur, so that messages are eventually transmitted successfully. Conflict-resolution algorithms can be either adaptive or nonadaptive (static). Static resolution can be deterministic using some fixed priority that is assigned to the nodes, or it can be probabilistic when the transmission schedule for interfered nodes is chosen from a fixed distribution as is done in Aloha-type schemes and the various versions of carrier-sensing multiple access (CSMA) schemes. Adaptive resolutions attempt to track the system evolution and exploit the available information. For example, resolution can be based on time of arrival, giving highest (or lowest) priority to the oldest message in the system as is done in some tree-based algorithms. Alternatively, resolution can be probabilistic but such that the statistics change dynamically according to the extent of the interference. This category includes estimating the multiplicity of the interfering nodes and the exponential back-off scheme of the Ethernet standard. Note that when the population of potential nodes in the system increases beyond a certain amount and conflict-free schemes are useless, contention-based protocols are the only possible solution.

The goal of this article is to survey typical examples of multiple access schemes. These examples include TDMA, FDMA, Aloha, polling, and tree-based schemes. The allocated space for the topic of multiple access schemes in the encyclopedia (which is yet another shared resource) is just too tiny to include all the ingenious multiple access schemes that have been designed by researchers over the years. Interested readers should refer to books on the subject (e.g., Rom and Sidi (23), Hammond and O'Reilly (22), and to the international journals that have published papers on the subject).

BASIC MODEL

When multiple access schemes are devised, a collection of nodes that communicate with each other or with a central node via a single shared channel is considered. In general, the ability of a node to hear the transmission of another node depends on the transmission power used, on the distance between the two nodes, and on the sensitivity of the receiver at the receiving node. We assume single-hop topologies in which all nodes hear one another, and whenever messages are transmitted successfully they arrive at their destinations.

The shared channel is the medium through which data are transferred from their sources to their destinations. The total transmission rate possible in the channel is C bits/s. We consider an errorless collision channel. Collision is a situation in which, at the receiver, two or more transmissions overlap in time wholly or partially. A collision channel is one in which all the colliding transmissions are not received correctly and must be retransmitted until they are received correctly. We assume that nodes can detect collisions. The channel is errorless in the sense that a single transmission heard at a node is always received correctly. Other possible channels include the noisy channel in which errors may occur even if only a single transmission is heard at a node; furthermore, the channel may be such that errors between successive transmissions are not independent. Another channel type is the capture channel in which one or more of the colliding transmissions captures the receiver and can be received correctly. Yet another case is a channel in which coding is used so that even if transmissions collide the receiver can still decode some or all of the transmitted information.

The basic unit of data generated by a node is a message. It is possible, though, that because of its length, a message cannot be transmitted in a single transmission and must therefore be broken into smaller units called packets, each of which can be transmitted in a single channel access. A message consists of an integral number of packets, although the number of packets in a message can vary randomly. Packet size is measured by the time required to transmit the packet after access to the channel has been granted. Typically, all packets are of equal size, say L bits.

The number of nodes that share the channel is denoted by M. When M becomes very large, the population of nodes is referred to as infinite population. Only contention-based schemes can cope with an infinite node population. The aggregate arrival process of new packets is assumed to be Poisson with rate Λ packets/s. When the population is finite, the arrival rate to each node is $\lambda = \Lambda/M$ packets/s.

Nodes are generally not assumed to be synchronized and are capable of accessing and transmitting their messages on the shared channel at any time. Another important class of systems is that of slotted systems in which there is a global clock that marks discrete intervals of time called slots whose length is usually the time required to transmit a packet (i.e., $T = L/C$ s). In these systems, transmissions of packets start only at slot starts. The slot length is therefore $T = L/C$ s. Other operations, such as determining activities on the channel, can be done at any time.

In some models, nodes can tell if the shared channel is in use before trying to use it. If the channel is sensed as busy, no node will attempt to use it until it goes idle in order to reduce interference. Naturally, additional hardware is required at each node to implement the sensing ability. In other models, nodes cannot sense the channel before trying to use it. They just go ahead and transmit according to their access scheme. Only later can they determine whether or not the transmission was successful via the feedback mechanism. Feedback in general is the information available to the nodes regarding activities on the shared channel at prior times. This information can be obtained by listening to the channel, or by explicit acknowledgment messages sent by the receiving node. For every scheme, there exist some instants of time (typically slot boundaries or end of transmissions) in which feedback information is available. Common feedback information indicates whether a message was successfully transmitted or a collision took place or the channel was idle. Feedback mechanisms do not consume the shared channel sources because they usually use a different channel or are able to determine the feedback locally. Other feedback variations include indication of the exact or the estimated number of colliding transmissions, or providing uncertain feedback (e.g., in the case of a noisy channel).

The important performance measures of multiple access schemes are their throughput and delay. The throughput of the channel is the aggregate average amount of data that is transported successfully through the channel in a unit of time. The throughput equals the fraction of time in which the channel is engaged in the successful transmission of node data and will be denoted by S, and it is obvious that $S \leq 1$. In conflict-free access schemes, the throughput is also the total or offered load on the shared channel. However, in contention-based access schemes, the offered load on the shared channel includes transmissions of new packets as well as retransmissions of packets that collide with each other. The offered load is denoted by g (measured in packets per second) and, obviously, $g \geq \Lambda$. The normalized offered load [i.e., the rate (per packet transmission time) packets are transmitted on the channel] is denoted by $G = g \cdot T$ and, obviously, $G \geq S$.

Delay is the time from the moment a message is generated until it arrives successfully across the shared channel. Here one must distinguish between the node and the system measures because it is possible that the average delay measured for the entire system does not necessarily reflect the average delay experienced by any of the nodes. In "fair" or homogeneous systems, we expect these to be almost identical. The average delay is denoted by D seconds, and its normalized version, grouped into units of packet transmission times, is denoted by $\mathscr{D}$ (i.e., $\mathscr{D} = D/T = D \cdot C/L$).

Another important performance criterion is system stability. Unfortunately, some schemes' characteristics may be such that some message-generation rates, even smaller than the maximal transmission rate in the channel, cannot be sustained by the system for a long time. Evaluation of those input rates for which the system remains stable is therefore essential.

Ideal Access Scheme

Before introducing the various multiple access schemes, let us consider an ideal scheme to use the shared channel. Ideally, transfer of the channel from one node to another can be accomplished instantaneously, without cost. Furthermore, whenever a node has data to transmit, some ingenious central controller knows this instantaneously and assigns the channel to that node in case the channel is idle. If the channel is busy, packets that arrive at the nodes are queued. For our purposes, the order in which packets of different nodes are served is not important. The performance of the ideal scheme serves as a bound to what can be expected from any practical access scheme.

The way the ideal scheme operates is identical to the operation of a single queue that is served by a single server, because packets do not interfere and because no time is wasted in transferring the channel use from one node to another. Because arrivals of new packets are according to a Poisson process and time is slotted, the performance of the ideal scheme is that of an $M/D/1$ queue. The throughput of an $M/D/1$ queue is just the utilization factor of the server as long as $S < 1$ (the stability condition), and it equals the offered load, in other words,

$$S = G = \Lambda T = \frac{\lambda \cdot M \cdot L}{C} \tag{1}$$

The normalized average delay of an $M/D/1$ queue is given by (as long as $S < 1$)

$$\mathscr{D} = 1 + \frac{S}{2(1-S)} = \frac{2-S}{2(1-S)} \quad (2)$$

The unit in the expression $\mathscr{D}$ is the normalized transmission time of a packet, whereas $S/[2(1-S)]$ is the normalized waiting time of a packet until being transmitted.

No access scheme can achieve throughput higher than S given in Eq. (1), and no access scheme can provide normalized average delays lower than $\mathscr{D}$ given in Eq. (2). These quantities will serve as yardsticks in the sequel.

CONFLICT-FREE SCHEMES

Conflict-free schemes are designed to ensure that a transmission, whenever made, is not interfered with by any other transmission and is therefore successful. This is achieved by allocating the channel to the nodes without any overlap between the portions of the channel allocated to different nodes. An important advantage of conflict-free access protocols is the ability to ensure fairness among nodes and the ability to control the packet delay—a feature that may be essential in real-time applications.

We consider both fixed-assignment schemes and dynamic schemes that guarantee no conflicts. In fixed-assignment schemes the channel allocation is predetermined (typically at network design time) and is independent of the demands of the nodes in the network. The most well-known fixed-assignment schemes are the frequency division multiple access and the time division multiple access. For both FDMA and TDMA, no overhead, in the form of control messages, is incurred. However, because of the static and fixed assignment, parts of the channel might be idle even though some nodes have data to transmit. Dynamic channel allocation schemes attempt to overcome this drawback by changing the channel allocation based on the current demands of the nodes. These schemes use some kind of reservation strategies based on either centralized or distributed polling.

Fixed Assignment

Both FDMA and TDMA are the oldest and most understood access schemes, widely used in practice. They are the most common implementation of fixed-assignment schemes.

With FDMA the entire available frequency band is divided into bands, each of which is used by a single node. Every node is therefore equipped with a transmitter for a given, predetermined frequency band and a receiver for each band (which can be implemented as a single receiver for the entire range with a bank of band-pass filters for the individual bands). With TDMA the time axis is divided into time slots, preassigned to the different nodes. Every node is allowed to transmit freely during the slot assigned to it; that is, during the assigned slot the entire shared channel is devoted to that node. The slot assignments follow a predetermined pattern that repeats itself periodically; each such period is called a frame. In most TDMA implementations, every node has exactly one slot in every frame.

The main advantage of both FDMA and TDMA is that each transmission is guaranteed to be successful and no control messages are required. An additional advantage of FDMA is its simplicity—it does not require any coordination or synchronization among the nodes because each can use its own frequency band without interference. However, both FDMA and TDMA are wasteful, especially when the load is momentarily uneven, because when one node is idle, its share of the channel cannot be used by other nodes. Another drawback of FDMA and TDMA is that they are not flexible; adding a new node to the network requires equipment or software modification in every other node. In addition, both waste some portion of the channel to ensure no overlap (either in time or in bandwidth) in the transmissions of different nodes. FDMA uses guard bands between the subchannels, and TDMA uses guard times to separate the nodes.

Neglecting the channel waste resulting from guard bands or times, the throughput of FDMA and TDMA is identical to that of the idealized schemes, because packets are never transmitted more than once. Therefore, we have for both

$$S = G = \Lambda T = \frac{\lambda \cdot M \cdot L}{C}$$

The delay characteristics of FDMA and TDMA are different. With FDMA the transmission rate of each node is C/M bits/s; therefore, the time to transmit a packet is $M \cdot L/C$ seconds. Each node can be modeled as an $M/D/1$ queue with arrival rate $\lambda = \Lambda/M$ and service time $M \cdot L/C$. The normalized average delay is, therefore,

$$\mathscr{D} = M\left[1 + \frac{S}{2(1-S)}\right] = M\frac{2-S}{2(1-S)}$$

which is M times larger than the normalized average delay of the ideal scheme.

With TDMA the transmission rate of each node is C bits/s, and the time to transmit a packet is L/C seconds. Each node can be modeled as an $M/D/1$ queue with arrival rate $\lambda = \Lambda/M$, but service is granted to the node only once a frame, namely every $M \cdot L/C$ seconds. The normalized average delay is therefore

$$\mathscr{D} = 1 + \frac{M}{2(1-S)}$$

Comparing the throughput delay characteristics of FDMA and TDMA, we note that

$$\mathscr{D}_{\mathrm{FDMA}} = \mathscr{D}_{\mathrm{TDMA}} + \frac{M}{2} - 1$$

We thus conclude that for any reasonable parameters, the TDMA-normalized average delay is always less than that of FDMA and the difference grows linearly with the number of nodes and is independent of the load. The difference stems from the fact that the actual transmission of a packet in TDMA takes only a single slot, whereas in FDMA it lasts the equivalent of an entire frame. This difference is somewhat offset by the fact that a packet arriving at an empty node may need to wait until the proper slot when a TDMA scheme is employed, whereas in FDMA transmission starts right away. It must be remembered, though, that at high throughput the dominant factor in the normalized average delay is inversely proportional to $(1 - S)$ in both TDMA and FDMA; therefore,

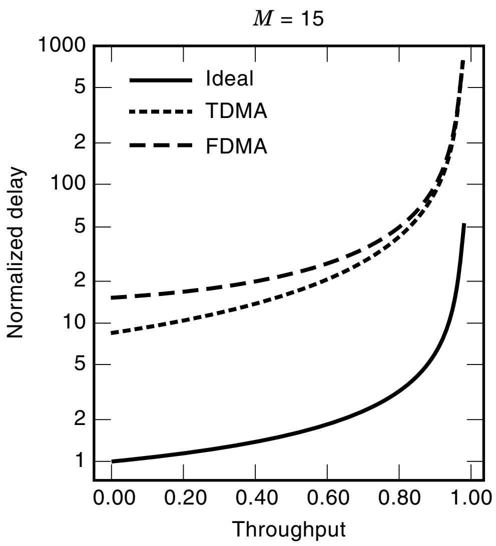

Figure 1. TDMA and FDMA performance.

the ratio of the normalized average delays between the two schemes approaches unity when the load increases. Figure 1 depicts the delay-throughput characteristics for TDMA and FDMA and the ideal access scheme for 50 users.

Further Reading. Many texts treating FDMA and TDMA are available [e.g., Martin (1) and Stallings (2)]. A good analysis of TDMA and FDMA can be found in Ref. 3. A sample path comparison between FDMA and TDMA schemes is carried out in Ref. 4 where it is shown that TDMA is better than FDMA not just on the average. A TDMA scheme in which the packets of each node are serviced according to a priority rule is analyzed by De Moraes and Rubin (5). The question of optimal allocation of slots to the nodes in generalized TDMA (in which a node can have more than one slot in a frame) is addressed in Itai and Rosberg (6), where the throughput of the network is maximized (assuming single buffers for each node), Hofri and Rosberg (7), where the expected packet-delay in the network is minimized. Message delay (as opposed to packet delay) for generalized TDMA is analyzed by Rom and Sidi (8).

Dynamic Assignment

Static conflict-free protocols such as FDMA and TDMA schemes do not use the shared channel very efficiently, especially when the network is lightly loaded or when the loads of different nodes are asymmetric. The static and fixed assignment in these schemes cause portions of the channel to remain idle even though some nodes have data to transmit. Dynamic channel allocation schemes are designed to overcome this drawback. With dynamic allocation strategies, the channel allocation changes with time and is based on current (and possibly changing) demands of the various nodes. The better and more responsive use of the shared channel achieved with dynamic schemes does not come for free; it requires control overhead that is unnecessary with fixed-assignment schemes and consumes a portion of the channel.

To ensure conflict-free operation, it is necessary to reach an agreement among the nodes on who transmits in a given slot. This agreement entails collecting information as to which nodes have packets to transmit and an arbitration scheme that selects one of these nodes to transmit in the slot. Both the information collection and the arbitration can be achieved using centralized control or distributed control.

A representative example of schemes that use centralized control are polling schemes. The basic feature of polling schemes is the operation of a central controller that polls the nodes of the network in some predetermined order (the most common being round-robin) to provide access to the shared channel. When a node is polled and has packets to transmit, it uses the whole shared channel to transmit its backlogged packets. With an exhaustive policy, the node empties its backlog completely, whereas with a gated policy it transmits only those packets that reside in its queue upon the polling instant. The last transmitted packet contains an indication that the central controller can poll the next node. If a polled node does not have packets to transmit, the next node is polled. In between polls, nodes accumulate the arriving packets in their queues and do not transmit until polled.

The control overhead of polling schemes is a result of the time required to switch from one node to the next. The switching time, denoted by w, includes all the time necessary to transfer the poll (channel propagation delay, transmission time of polling and response packets, etc.). We let $\hat{w} = w/T$ denote the normalized switching time.

The throughput of a polling scheme is identical to that of an ideal scheme and is given by Eq. (1). The normalized average delay is given by

$$\mathscr{D} = 1 + \frac{S}{2(1-S)} + \frac{M\hat{\omega}(1-S/M)}{2(1-S)}$$

We note that the first two terms are just the normalized average delay of the ideal scheme and the third term reflects the overhead resulting from the switching times from one node to the next.

As an example of a distributed dynamic conflict-free scheme we use the mini slotted alternating priority (MSAP) scheme (9). The MSAP scheme allows the nodes to determine in a distributed manner the order in which they'll use the shared channel, assuming the nodes are ordered according to some priority rule. Either the priority rule can be static or it can change in a round-robin manner in each slot.

MSAP is based on distributed reservations. To describe its operation, we need to define the slot structure. Let τ (seconds) denote the maximum system propagation delay, that is, the longest time it takes for a signal emitted at one end of the network to reach the other end. The quantity τ plays a crucial role in multiple access schemes. Its normalized version is denoted by $a = \tau/T$. Let every slot consist of initial $M - 1$ reservation minislots, each of duration τ, followed by a data transmission period of duration T, followed by another minislot. Only those nodes wishing to transmit in a slot take any action: a node that does not wish to transmit in a given slot remains quiet for the entire slot duration. Given that every node wishing to transmit knows its own priority, they behave as follows. If the node of the highest priority wishes to transmit in this slot, then it starts immediately. Its transmission consists of an unmodulated carrier for a duration of $M - 1$ minislots followed by a packet of duration T. A node of the ith priority ($2 \leq i \leq M$) wishing to transmit in this slot will do so only if the first $i - 1$ minislots are idle. In this case, it will transmit $M - i$ minislots of unmodulated carrier followed

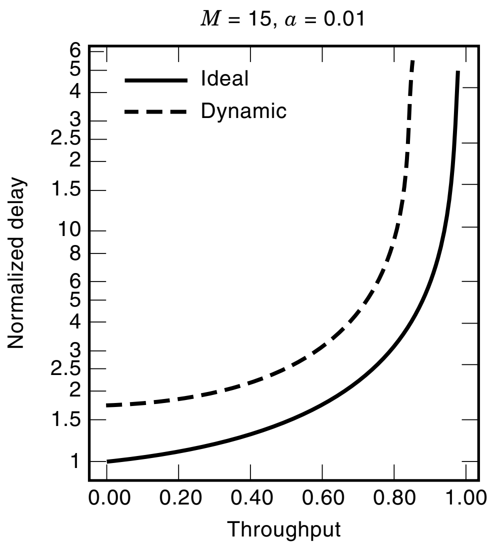

Figure 2. Dynamic access.

by a packet of duration T. The specific choice of the minislot duration ensures that when a given node transmits in a minislot all other nodes know it by the end of that minislot allowing them to react appropriately. The additional minislot at the end allows the data signals to reach every node of the network. This is needed to ensure that all start synchronized in the next slot, as required by the reservation scheme.

The fraction of slots in which transmissions take place is ΛT. Because a fraction of $M\tau/(T + M\tau)$ of every slot is overhead, we conclude that the throughput of this scheme is

$$S = \Lambda T \frac{T}{T + M\tau} = \Lambda T \frac{1}{1 + Ma}$$

The normalized average delay is obtained by using standard analysis of priority queues, and it is given by

$$\mathscr{D} = (1 + Ma)\left\{1 + \frac{1}{2[1 - (1 + Ma)S]}\right\}$$

Figure 2 depicts the delay-throughput characteristics for the dynamic-access schemes for 50 users.

Further Reading. The variants of polling schemes are numerous. Reference 10 contains the analysis of most of the basic schemes with a long list of references that is complemented in Ref. 11. In Ref. 12 more advanced schemes are described along with some optimization considerations in the operations of polling schemes, such as the determination of the poll order of the nodes.

The MSAP scheme described previously represents an entire family of schemes that guarantees conflict-free transmissions using distributed reservation. All these schemes have a sequence of preceding bits serving to reserve or announce upcoming transmissions (this is known as the reservation preamble). In MSAP there are $M - 1$ such bits for every transmitted packet. An improvement to the MSAP scheme is the bit-map protocol described by Tanenbaum (13). The idea is to use a single reservation preamble to schedule more than a single transmission; using the fact that all participating nodes are aware of the reservations made in the preamble. The bit-map scheme requires synchronization among the nodes that is somewhat more sophisticated than the MSAP scheme, but the overhead paid per transmitted packet is less than the overhead for MSAP. Another variation of a reservation scheme has been described by Roberts (14). There, every node can make a reservation in every minislot of the reservation preamble, and if the reservation remains uncontested, that reserving node will transmit. If there is a collision in the reservation minislot, all nodes but the "owner" of that minislot will abstain from transmission. Altogether, this is a standard TDMA with idle slots made available to be grabbed by others. Several additional reservation and TDMA schemes are also analyzed by Rubin (4). One of the most efficient reservation schemes is the broadcast recognition access method (BRAM) (15). This is essentially a combination between the bit-map and the MSAP schemes. As with MSAP, a reservation preamble serves to reserve the channel for a single node, but unlike the MSAP the reservation preamble does not necessarily contain all $M - 1$ minislots. The idea is that nodes start their transmission with a staggered delay not before they ensure that another transmission is not ongoing [Kleinrock and Scholl (9) also refers to a similar scheme]. Under heavy load BRAM reduces to regular TDMA.

CONTENTION-BASED SCHEMES

With the conflict-free schemes discussed earlier, every scheduled transmission is guaranteed to succeed. With contention-based schemes success of a transmission is not guaranteed in advance because whenever two or more nodes are transmitting on the shared channel simultaneously, a collision occurs and the data cannot be received correctly. This being the case, packets may have to be transmitted and retransmitted until eventually they are correctly received. Transmission scheduling is therefore the focal concern of contention-based schemes.

Pure and Slotted Aloha

The Aloha family of schemes is probably the richest family of multiple access protocols. First of all, its popularity is the result of seniority because it was the first contention-based scheme introduced (16). Second, many of these schemes are so simple that their implementation is straightforward. Many local area networks of today implement some sophisticated variants of this family of schemes.

The pure Aloha scheme is the basic scheme in the family and it is very simple (16). It states that a newly generated packet is transmitted immediately upon generation, hoping for no interference by others. If two or more nodes transmit so that their packets overlap (even partially) in time, interference results, and the transmissions are unsuccessful. In this case every colliding node, independently of the others, schedules its retransmission to a random time in the future. This randomness is required to ensure that the same set of packets does not continue to collide indefinitely.

The Aloha scheme is very well suited to bursty traffic because a node does not hold the shared channel when it has no packets to transmit. The drawback of this scheme is that network performance deteriorates significantly as a result of excessive collisions at medium and high traffic intensities. The Aloha scheme is a completely distributed scheme that allows every node to operate independently of the others.

The exact characterization of the offered load to the channel for the pure Aloha scheme is extremely complicated. To overcome this complexity, it is standard to assume that the offered load forms a Poisson process (with rate g, of course). This flawed assumption is an approximation (as has been shown by simulation) that simplifies the analysis of Aloha-type schemes considerably and provides some initial intuitive understanding of the ALOHA scheme. Consider a packet (new or retransmitted) whose transmission starts at time t. This packet will be successful if no other packet is transmitted in the interval $(t - T, t + T)$ (this period of duration $2T$ is called the vulnerable period). The probability of this happening, that is, the probability of success P_s is the probability that no packet is transmitted in an interval of length $2T$. Because the transmission points correspond to a Poisson process, we have

$$P_s = e^{-2gT}$$

Now, packets are scheduled at a rate of g per second, of which only a fraction P_s are successful. Thus, the rate of successfully transmitted packets is gP_s. When a packet is successful, the channel carries useful information for a period of T seconds; in any other case, it carries no useful information at all. Because the throughput is the fraction of time that useful information is carried on the shared channel, we have

$$S = gTe^{-2gT} = Ge^{-2G}$$

This relation between S and G is typical to many Aloha-type schemes. For small values of G (light load), the throughput is approximately the offered load. For large values of G (heavy load), the throughput decreases rapidly because of excessive amount of collisions. For pure Aloha we note that for $G = \frac{1}{2}$, S takes on its maximal value of $1/2e \approx 0.18$. This value is referred to as the capacity of the pure Aloha channel. Figure 3 depicts the load-throughput characteristics for the Aloha-type schemes.

We recall that for a system to be stable the long-term rate of input must equal the long-term rate of output meaning that stability requires $S = \Lambda T$. Larger values of Λ clearly cannot result in stable operation. Note, however, that even for smaller values of Λ there are two values of G to which it corresponds—one larger and one smaller than $\frac{1}{2}$. The smaller one is (conditionally) stable, whereas the other one is conditionally unstable, meaning that if the offered load increases beyond that point the system will continue to drift to higher load and lower throughput. Thus, without additional measures of control, the stable throughput of pure Aloha is 0 (17). It is appropriate to note that this theoretical instability is rarely a severe problem in real systems, where the long-term load, including, of course, the "off-hours" load, is fairly small, although temporary problems may occur.

The delay characteristic of the Aloha scheme can be approximated as follows. For each packet, the average number of transmission attempts until the packet is transmitted successfully is $G/S = e^{2G}$. Thus, the average number of unsuccessful transmission attempts is $G/S - 1 = e^{2G} - 1$. If a collision occurs, the node reschedules the colliding packet for some random time in the future. Let the average rescheduling time be B (seconds). Each successful transmission attempt requires T seconds and each unsuccessful transmission attempt requires $T + B$ seconds on the average. Therefore, the average delay is given by

$$D = T + (G/S - 1)(T + B) = T + (e^{2G} - 1)(T + B) \quad (3)$$

and in a normalized form

$$\mathscr{D} = 1 + (e^{2G} - 1)(1 + B/T)$$

With pure Aloha, even if the overlap in time between two transmitted packets is very tiny, both packets are destroyed. The slotted Aloha variation overcomes this drawback, and it is simply pure Aloha with a slotted channel. Thus, two (or more) packets can either overlap completely or do not overlap at all, and the vulnerable period is reduced to a single slot. In other words, a slot will be successful if and only if exactly one packet is transmitted in that slot. Therefore,

$$S = gTe^{-gT} = Ge^{-G}$$

This relation is very similar to that of pure Aloha, except of increased throughput. Channel capacity is $1/e \approx 0.36$ and is achieved at $G = 1$. These results were first derived by Roberts (14). Similar to the pure Aloha scheme, the normalized average delay for the slotted Aloha scheme is

$$\mathscr{D} = 1 + (e^{G} - 1)(1 + B/T)$$

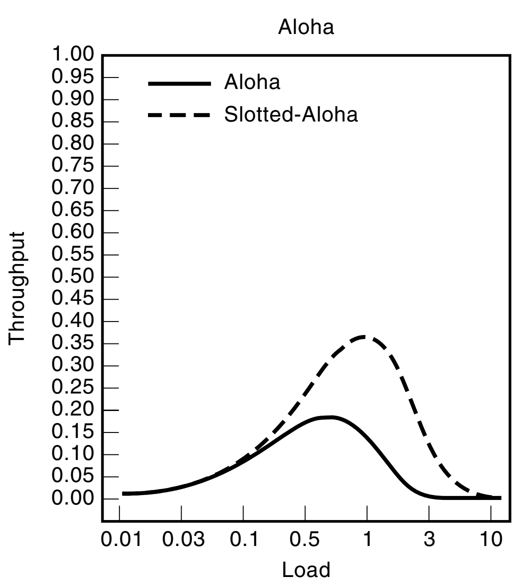

Figure 3. Throughput of Aloha and slotted Aloha.

Carrier-Sensing Protocols

The Aloha schemes exhibit fairly poor performance, which can be attributed to the "impolite" behavior of the nodes, namely, whenever one has a packet to transmit it does so without consideration of others. It is clear that in a shared environment even little consideration can benefit all. Consider a listen-before-talk behavior wherein every node, before attempting any transmission, listens whether somebody else is already using the channel. If this is the case, the node will refrain from transmission to the benefit of all; its packet will clearly not be successful if transmitted; furthermore, disturbing another node will cause the currently transmitted packet to be retransmitted, possibly disturbing yet another packet.

The process of listening to the shared channel is not that demanding. Every node is equipped with a receiver anyway, and every node can monitor the channel because it is shared. Moreover, to detect another node's transmission does not require receiving the information; it suffices to sense the carrier that is present when signals are transmitted. The carrier-sensing family of schemes is characterized by sensing the carrier and deciding according to it whether another transmission is ongoing.

Carrier sensing does not yield conflict-free operation. Suppose that the channel has been idle for a while and that two nodes concurrently generate a packet. Each will sense the channel, discover that it is idle, and transmit the packet to result in a collection. "Concurrently" here does not really mean at the very same time; if one node starts transmitting it takes some time for the signal to propagate and arrive at the other node. Hence concurrently actually means within a time window of duration equal to signal propagation time. The maximum propagation time in the network is τ, and its normalized version is a, an important parameter that affects the performance of carrier sensing schemes. The larger this quantity is, collisions are more likely and the performance becomes worse.

All the carrier sensing multiple access schemes share the same philosophy: when a node generates a new packet, the channel is sensed, and if found idle the packet is transmitted without further ado. When a collision takes place, every transmitting node reschedules a retransmission of the collided packet to some other time in the future (chosen with some randomization to avoid repeated collisions) at which time the same operation is repeated. The variations on the CSMA scheme are caused by the behavior of nodes that wish to transmit and find (by sensing) the channel busy. Most of the basic variations were introduced and analyzed by Kleinrock and Tobagi (18–20).

In the nonpersistent versions of CSMA (NP-CSMA) a node that generated a packet and found the channel busy refrains from transmitting the packet and behaves exactly as if its packet collided [i.e., it schedules (randomly) the retransmission of the packet to some time in the future]. With NP-CSMA, there are situations in which the channel is idle although one or more nodes have packets to transmit. The 1-persistent CSMA (1P-CSMA) is an alternative to NP-CSMA because it avoids such situations by being a bit more greedy. This is achieved by applying the following rule. A node that senses the channel and finds it busy persists to wait and transmits as soon as the channel becomes idle. Consequently, the channel is always used if there is a node with a packet. With the 1-persistent scheme, a collision may occur not only because of nonzero propagation delays but also when two nodes become ready to transmit in the middle of another node's transmission. In this case, both nodes will wait until that transmission ends and will begin transmission simultaneously, resulting in a collision.

For slotted operation, CSMA schemes use time slot of duration τ seconds, which is usually much smaller than the slot size of duration T seconds, used with slotted Aloha. However, like slotted Aloha, all nodes using slotted CSMA schemes are forced to start transmission at the beginning of a slot.

Beside the ability to sense the carrier, some local area networks (such as Ethernet) have an additional feature, namely, that nodes can detect interference among several transmissions (including their own) while transmission is in progress and abort transmission of their collided packets. If this can be done sufficiently fast, then the duration of an unsuccessful transmission would be shorter than that of a successful one, thus improving the performance of the scheme. Together with carrier sensing, this produces a variation of CSMA that is known as CSMA/CD (Carrier Sensing Multiple Access with Collision Detection). The operation of all CSMA/CD schemes is identical to the operation of the corresponding CSMA schemes, except that if a collision is detected during transmission, the transmission is aborted and the packet is scheduled for transmission at some later time. For Ethernet networks this random delay is doubled (at most 16 times) each time the packet collides—a scheme known as binary exponential backoff. To ensure that all network nodes indeed detect a collision when it occurs, a consensus reenforcement procedure is used. This procedure is manifested by jamming the channel with a collision signal for a duration of τ_{cr} seconds, which is usually much larger than the time necessary to detect a collision. We let $\gamma = \tau_{cr}/\tau$.

The analysis of the throughput of CSMA schemes is rather complicated. It is based on computations of average lengths of idle and transmission periods. For NP-CSMA we have

$$S = \frac{gTe^{-g\tau}}{g(T+2\tau)+e^{-g\tau}} = \frac{Ge^{-aG}}{G(1+2a)+e^{-aG}}$$

For slotted NP-CSMA, we have

$$S = \frac{aGe^{-aG}}{1-e^{-aG}+a}$$

For 1P-CSMA, we have

$$\begin{aligned} S &= \frac{gTe^{-g(T+2\tau)}[1+gT+g\tau(1+gT+g\tau/2)]}{g(T+2\tau)-(1-e^{-g\tau})+(1+g\tau)e^{-gT+\tau}} \\ &= \frac{Ge^{-G(1+2a)}[1+G+aG(1+G+aG/2)]}{G(1+2a)-(1-e^{-aG})+(1+aG)e^{-G(1+a)}} \end{aligned}$$

For slotted 1P-CSMA, we have

$$S = \frac{Ge^{-G(1+a)}[1+a-e^{-aG}]}{(1+a)(1-e^{-aG})+ae^{-G(1+a)}}$$

For nonpersistent CSMA/CD, we have

$$S = \frac{Ge^{-aG}}{Ge^{-aG}+\gamma aG(1-e^{-aG})+2aG(1-e^{-aG})+2-e^{-aG}}$$

For slotted nonpersistent CSMA/CD, we have

$$S = \frac{Ge^{-aG}}{Ge^{-aG}+\gamma aG(1-e^{-aG}-aGe^{-aG})+(2-e^{-aG}-aGe^{-aG})}$$

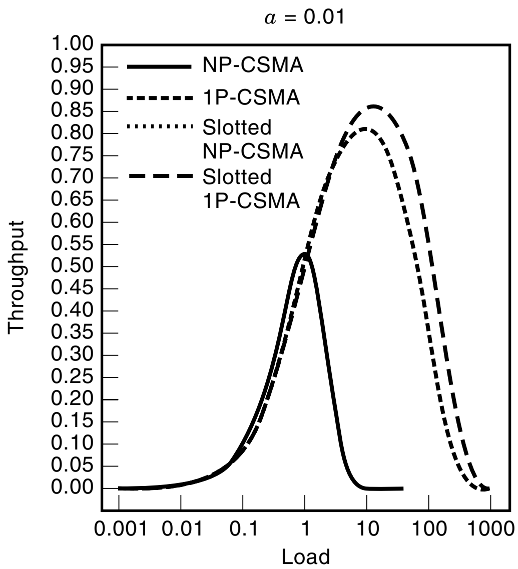

Figure 4. Throughput of CSMA versions.

Figure 4 depicts the load-throughput characteristics for the CSMA-type schemes.

Further Reading. Numerous variations on the environment under which the Aloha and CSMA schemes operate have been addressed in the literature (see, e.g., Refs. 3, 13, and 21–23). For instance, various packet length distributions were considered by Abramson (24) and Ferguson (25) for Aloha and by Tobagi and Hunt (26) and for CSMA.

The assumption that, whenever two or more packets overlap at the receiver, all packets are lost is overly pessimistic. In radio networks the receiver might correctly receive a packet despite the fact that it is time-overlapping with other transmitted packets. This phenomenon is known as capture and it can happen as a result of various characteristics of radio systems. Most studies (27,28) considered power capture (the phenomenon whereby the strongest of several transmitted signals is correctly received at the receiver). Thus, if a single high-powered packet is transmitted, then it is correctly received regardless of other transmissions. Hence, channel use increases.

Reservation schemes that allow contentions are designed to have the advantages of both the Aloha and the TDMA approaches. Examples of reservation schemes appear in Ref. 29, where the knowledge of the number of users is needed, or in Refs. 14 and 30, where they do not require this knowledge. Approximate analysis of a reservation Aloha protocol can be found in Lam (31).

Approximate analysis of the delay was presented by Ferguson (32) for Aloha and by Beuerman and Coyle (33) for CSMA schemes. Instability issues of the Aloha protocol were first identified by Carleial and Hellman (34) and Lam and Kleinrock (35). Later, similar issues were identified for the CSMA family of protocols by Tobagi and Kleinrock (20).

COLLISION RESOLUTION SCHEMES

The original Aloha scheme and its CSMA derivatives are inherently unstable in the absence of some external control. Looking into the philosophy behind the schemes, it is obvious that there is no sincere attempt to resolve collisions among packets as soon as they occur. Instead, the attempts to resolve collisions are always deferred to the future, with the hope that things will then work out, somehow, but they never do.

Another type of contention-based schemes with a different philosophy are collision resolution schemes (CRS). In these schemes the efforts are concentrated on resolving collisions as soon as they occur. Moreover, in most versions of these schemes, new packets that arrive to the network are inhibited from being transmitted while the resolutions of collisions is in progress. This ensures that if the rate of arrival of new packets to the system is smaller than the rate at which collisions can be resolved (the maximal rate of departing packets—throughput), then the system is stable. The basic idea behind these schemes is to exploit in a more sophisticated manner the feedback information that is available to the nodes in order to control the retransmission process so that collisions are resolved more efficiently.

The most basic collision resolution scheme is called the binary-tree CRS (or binary-tree scheme) and was proposed by Capetanakis (36), Hayes (37), and Tsybakov and Mikhailov (38). According to this scheme, when a collision occurs, in slot k say, all nodes that are not involved in the collision wait until the collision is resolved. The nodes involved in the collision split randomly into two subsets, by (for instance) each flipping a coin. The nodes in the first subset, those that flipped 0, retransmit in slot $k + 1$, whereas those that flipped 1 wait until all those that flipped 0 transmit their packets successfully. If slot $k + 1$ is either idle or contains a successful transmission, the nodes of the second subset (those that flipped 1) retransmit in slot $k + 2$. If slot $k + 1$ contains another collision, then the procedure is repeated (i.e., the nodes whose packets collided in slot $k + 1$ flip a coin again and operate according to the outcome of the coin flipping, and so on). A node having a packet that collided (at least once) is backlogged.

The operation of the scheme can also be described by a binary-tree in which every vertex corresponds to a time slot. The root of the tree corresponds to the slot of the original collision. Each vertex of the tree also designates a subset (perhaps empty) of backlogged nodes. Vertices whose subsets contain at least two nodes indicate collisions and have two outgoing branches, corresponding to the splitting of the subset into two new subsets. Vertices corresponding to empty subsets or subsets containing one node are leaves of the tree and indicate an idle and a successful slot, respectively. For instance, consider a collision that occurs in slot 1. At this point it is neither known how many nodes nor who are the nodes that collided in this slot. Each of the colliding nodes flip a coin, and those that flipped 0 transmit in slot 2. By the rules of the scheme, no newly arrived packet is transmitted while the resolution of a collision is in progress, so that only nodes that collided in slot 1 and flipped 0 transmit in slot 2. Another collision occurs in slot 2, and the nodes involved in that collision flip a coin again. In this example, all the colliding nodes of slot 2 flipped 1 and therefore slot 3 is idle. The nodes that flipped 1 in slot 2 transmit again in slot 4, resulting in another collision and forcing the nodes involved in it to flip a coin once more. One node flips 0 and transmits (successfully) in slot 5 causing all nodes that flipped 1 in slot 4 to transmit in slot 6. In this example, there is one such node, and therefore slot 6 is a successful one. Now that the collision among all nodes that flipped 0 in slot 1 has been resolved, the nodes that flipped 1 in that slot transmit (in slot 7). Another collision occurs, and the nodes involved in it flip a coin. Another collision is observed in slot 8, meaning that at least two nodes flipped 0 in slot 7. The nodes that collided in slot 8 flip a coin and, as it happens, there is a single node that flipped 0, and it transmits (successfully) in slot 9. Then, in slot 10, transmit the nodes that flipped 1 in slot 8. There is only one such node, and its transmission is, of course, successful. Finally, the nodes that flipped 1 in slot 7 must transmit in slot 11. In this example, there is no such node; hence slot 11 is idle, completing the resolution of the collision that occurred in slot 7 and, at the same time, the one in the first slot.

It is clear from this example that each node, including those that are not involved in the collision, can construct the binary-tree by following the feedback signals corresponding to each slot, thus knowing exactly when the collision is resolved. A collision is resolved when the nodes of the network know that all packets involved in the collision have been transmitted successfully. The time interval starting with the original collision (if any) and ending when this collision is resolved is called collision resolution interval (CRI). In the preceding example the length of the CRI is 11 slots.

The binary-tree protocol dictates how to resolve collisions after they occur. To complete the description of the protocol, one must specify when newly generated packets are transmitted for the first time. One alternative, which is assumed all along (known as the obvious-access scheme), is that new packets are inhibited from being transmitted while a resolution of a collision is in progress. That is, packets that arrive to the system while a resolution of a collision is in progress, wait until the collision is resolved, at which time they are transmitted. In the example above all new packets arriving to the system during slots 1 through 11 are transmitted for the first time in slot 12.

Let L_n be the expected length of a CRI that starts with the transmission of n packets. From the operation of the scheme, it is clear that as long as the arrival rate of new packets into the system is smaller than the ratio n/L_n (for large n), the system is stable. When fair coins are used for splitting the users upon collisions, one can show that for every n,

$$L_n \leq 2.886n + 1$$

yielding stable system for arrival rates that are smaller than 0.346.

The performance of the binary-tree protocol can be improved in two ways. The first is to speed up the collision resolution process by avoiding certain, avoidable, collisions. The second is based on the observation that collisions among a small number of packets are resolved more efficiently than collisions among a large number of packets. Therefore, if most CRIs start with a small number of packets, the performance of the protocol is expected to improve.

Consider again the example above. In slots 2 and 3 a collision is followed by an idle slot. This implies that in slot 2 all users (and there were at least two of them) flipped 1. The binary-tree protocol dictates that these users must transmit in slot 4, although it is obvious that this will generate a collision that can be avoided. The modified binary-tree protocol was suggested by Massey (39), and it eliminates such avoidable collisions by letting the users that flipped 1 in slot 2 in the preceding example, flip coins before transmitting in slot 4. Consequently, the slot in which an avoidable collision would occur is saved. In this case, fair coins yield a stable system for arrival rates smaller than 0.375, and biased coins increase this number up to 0.381.

When the obvious access is employed, it is very likely that a CRI will start with a collision among a large number of packets when the previous CRI was long. When the system operates near its maximal throughput, most CRIs are long; hence collisions among a large number of packets must be resolved frequently, yielding nonefficient operation. Ideally, if it were possible to start each CRI with the transmission of exactly one packet, the throughput of the system would have been 1. Because this is not possible, one should try to design the system so that in most cases a CRI starts with the transmission of about one packet. There are several ways to acheive this goal by determining a first-time transmission rule (i.e., when packets are transmitted for the first time). One way, suggested by Capetanakis (36), is to have an estimate on the number of packets that arrived in the previous CRI and divide them into smaller groups, each having an expected number of packets on the order of one and handling each group separately. Another way, known as the epoch mechanism has been suggested by Gallager (40) and Tsybakov and Mikhailov (41). According to this mechanism, time is divided into consecutive epochs each of length Δ slots. The ith arrival epoch is the time interval $[i\Delta, (i+1)\Delta]$. Packets that arrive during the ith arrival epoch are transmitted for the first time in the first slot after the collision among packets that arrived during the $(i-1)$st arrival epoch is resolved. The parameter Δ is chosen to optimize the performance of the system. When $\Delta = 2.68$, the system is stable for arrival rates up to 0.429 if slots of sure collisions are not saved, and up to 0.462 if they are. A final enhancement of the epoch mechanism is to start a new epoch each time a collision is followed by two successful transmissions. This guarantees that each CRI will start with an optimal number of packets, and it yields the highest stable throughput known for multiple access systems—0.487.

Further Reading. Numerous variations of the environment under which collision resolution protocols operate have been addressed in the literature and excellent surveys on the subject appear in Refs. 42 and 43. Books by Bertsekas and Gallager (21) and Rom and Sidi (23) are also excellent sources on collision resolution protocols. Considerable effort has been spent on finding upper bounds to the maximum throughput that can be achieved in an infinite population model with Poisson arrivals and ternary feedback. The best upper bound known to date is 0.568 and is the work of Tsybakov and Likhanov (44). Practical multiple access communication systems are prone to various types of errors. Collision resolution protocols that operate in the presence of noise errors, erasures, and captures have been studied in Refs. 45–49. Collision resolution protocols yielding high throughputs for general arrival processes (even if their statistics are unknown) were developed by Cidon and Sidi (50) and Greenberg et al. (51). The expected packet delay of the binary-tree protocol has been derived by Fayolle et al. (52) and Tsybakov and Mikhailov (38). Bounds on the expected packet delay of the algorithm with the epoch mechanism have been obtained in Refs. 41 and 53, and bounds on the packet delay distribution have been obtained in Refs. 54 and 55.

BIBLIOGRAPHY

1. J. Martin, *Communication Satellite Systems,* Englewood Cliffs, NJ: Prentice-Hall, 1978.
2. W. Stallings, *Data and Computer Communications,* New York: Macmillan, 1985.
3. J. F. Hayes, *Modeling and Analysis of Computer Communications Networks,* New York: Plenum Press, 1984.
4. I. Rubin, Access control disciplines for multi-access communications channels: Reservation and TDMA schemes, *IEEE Trans. Inf. Theory,* **IT-25**: 516–536, 1979.
5. L. F. M. De Moraes and I. Rubin, Message delays for a TDMA scheme under a Nonpreemptive Priority Discipline, *IEEE Trans. Commun.,* **COM-32**: 583–588, 1984.
6. A. Itai and Z. Rosberg, A golden ratio control policy for a multiple-access channel, *IEEE Trans. Autom. Control,* **AC-29**: 712–718, 1984.
7. M. Hofri and Z. Rosberg, Packet delay under the golden ratio weighted TDM policy in a multiple access channel, *IEEE Trans. Inf. Theory,* **IT-33**: 341–349, 1987.
8. R. Rom and M. Sidi, Message delay distribution in generalized time division multiple access (TDMA), *Probability Eng. Inf. Sci.,* **4**: 187–202, 1990.
9. L. Kleinrock and M. Scholl, Packet switching in radio channels: New conflict-free multiple access schemes, *IEEE Trans. Commun.,* **COM-28**: 1015–1029, 1980.
10. H. Takagi, *Analysis of Polling Systems,* Cambridge, MA: MIT Press, 1986.
11. H. Takagi, Queueing analysis of polling models, *ACM Comp. Surv.,* **20** (1): 5–28, 1988.
12. H. Levy and M. Sidi, Polling systems: Applications, modeling and optimization, *IEEE Trans. Commun.,* **38**: 1750–1760, 1990.
13. A. S. Tanenbaum, *Computer Networks,* 3rd ed., Englewood Cliffs, NJ: Prentice-Hall International Editions, 1996.
14. L. G. Roberts, ALOHA packet system with and without slots and capture, *Comput. Commun. Rev.,* **5** (2): 28–42, 1975.
15. I. Chlamtac, W. R. Franta, and K. D. Levin, BRAM: The Broadcast Recognizing Access Mode, *IEEE Trans. Commun.,* **COM-27**: 1183–1189, 1979.
16. N. Abramson, The ALOHA System—Another Alternative for Computer Communications, *Proc. Fall Joint Comput. Conf.*, pp. 281–285, 1970.
17. G. Fayolle et al., The stability problem of broadcast packet switching computer networks, *Acta Informatica,* **4** (1): 49–53, 1974.
18. L. Kleinrock and F. A. Tobagi, Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput delay characteristics, *IEEE Trans. Commun.,* **23**: 1400–1416, 1975.
19. F. A. Tobagi and L. Kleinrock, Packet switching in radio channels: Part II—The hidden terminal problem in carrier sense multiple-access and the busy tone solution, *IEEE Trans. Commun.,* **23**: 1417–1433, 1975.
20. F. A. Tobagi and L. Kleinrock, Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple-access, *IEEE Trans. Commun.,* **25**: 1103–1119, 1977.
21. D. Bertsekas and R. Gallager, *Data Networks,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall International Editions, 1992.
22. J. L. Hammond and P. J. P. O'Reilly, *Performance Analysis of Local Computer Networks,* Reading, MA: Addison-Wesley, 1986.
23. R. Rom and M. Sidi, *Multiple Access Protocols; Performance and Analysis,* New York: Springer-Verlag, 1990.
24. N. Abramson, The throughput of packet broadcasting channels, *IEEE Trans. Commun.,* **25**: 117–128, 1977.
25. M. J. Ferguson, An Approximate Analysis of Delay for Fixed and Variable Length Packets in an Unslotted Aloha Channel, *IEEE Trans. Commun.,* **25**: 644–654, 1977.
26. F. A. Tobagi and V. B. Hunt, Performance analysis of carrier sense multiple access with collision detection, *Comput. Netw.,* **4** (5): 245–259, 1980.
27. J. J. Metzner, On Improving Utilization in Aloha Networks, *IEEE Trans. Commun.,* **24**: 447–448, 1976.
28. N. Shacham, Throughput-delay performance of packet-switching multiple-access channel with power capture, *Performance Evaluation,* **4** (3): 153–170, 1984.
29. R. Binder, A Dynamic Packet Switching System for Satellite Broadcast Channels, *Proc. ICC'75,* pp. 41.1–41.5, 1975.
30. W. Crowther et al., A System for broadcast communication: Reservation-Aloha, *Proc. Int. Conf. Syst. Sci.,* pp. 371–374, 1973.
31. S. S. Lam, Packet broadcast networks—A performance analysis of the R-ALOHA protocol, *IEEE Trans. Comput.,* **29**: 596–603, 1980.
32. M. J. Ferguson, On the control, stability, and waiting time in a slotted Aloha, *IEEE Trans. Commun.,* **23**: 1306–1311, 1975.
33. S. L. Beuerman and E. J. Coyle, The delay characteristics of CSMA/CD networks, *IEEE Trans. Commun.,* **36**: 553–563, 1988.
34. A. B. Carleial and M. E. Hellman, Bistable behavior of ALOHA-type systems, *IEEE Trans. Commun.,* **23**: 401–410, 1975.
35. S. S. Lam and L. Kleinrock, Packet switching in a multicast broadcast channel: Dynamic control procedures, *IEEE Trans. Commun.,* **23**: 891–904, 1975.
36. J. I. Capetanakis, Tree Algorithm for Packet Broadcast Channels, *IEEE Trans. Inf. Theory,* **25**: 505–515, 1979.
37. J. F. Hayes, An adaptive technique for local distribution, *IEEE Trans. Commun.,* **26**: 1178–1186, 1978.
38. B. S. Tsybakov and V. A. Mikhailov, Free synchronous packet access in a broadcast channel with feedback, *Prob. Inf. Trans.,* **14** (4): 259–280, 1978.
39. J. L. Massey, *Collision Resolution Algorithms and Random-Access Communications Multi-User Communications Systems,* CISM Courses and Lectures Series (G. Longo, ed.), New York: Springer-Verlag, pp. 73–137, 1981 (also in UCLA Technical Report UCLA-ENG-8016, April 1980).
40. R. G. Gallager, Conflict resolution in random access broadcast networks, *Proc. AFOSR Workshop Commun. Theory Appl.,* Provincetown, pp. 74–76, September 1978.
41. B. S. Tsybakov and V. A. Mikhailov, Random multiple packet access: Part-and-try algorithm, *Prob. Inf. Trans.,* **16**: 305–317, 1980.
42. R. G. Gallager, A perspective on multiaccess channels, *IEEE Trans. Inf. Theory,* **31**: 124–142, 1985.
43. B. S. Tsybakov, Survey of USSR contributions to multiple-access communications, *IEEE Trans. Inf. Theory,* **31**: 143—165, 1985.
44. B. S. Tsybakov and N. B. Likhanov, Upper bound on the capacity of a random multiple access system, *Prob. Inf. Trans.,* **23** (3): 224–236, 1988.
45. I. Cidon and M. Sidi, The effect of capture on collision-resolution algorithms, *IEEE Trans. Commun.,* **33**: 317–324, 1985.
46. I. Cidon and M. Sidi, Erasures and noise in multiple access algorithms, *IEEE Trans. Inf. Theory,* **33**: 132–143, 1987.

47. I. Cidon, H. Kodesh, and M. Sidi, Erasure, Capture and random power level selection in multiple-access systems, *IEEE Trans. Commun.,* **36**: 263–271, 1988.
48. M. Sidi and I. Cidon, Splitting protocols in presence of capture, *IEEE Trans. Inf. Theory,* **31**: 295–301, 1985.
49. N. D. Vvedenskaya and B. S. Tsybakov, Random multiple access of packets to a channel with errors, *Prob. Inf. Trans.,* **19** (2): 131–147, 1983.
50. I. Cidon and M. Sidi, Conflict multiplicity estimation and batch resolution algorithms, *IEEE Trans. Inf. Theory,* **34**: 101–110, 1988.
51. A. G. Greenberg, P. Flajolet, and R. E. Ladner, Estimating the multiplicities of conflicts to speed their resolution in multiple access channels, *J. ACM,* **34** (2): 289–325, 1987.
52. G. Fayolle et al., Analysis of a stack algorithm for random multiple-access communication, *IEEE Trans. Inf. Theory,* **31**: 244–254, 1985.
53. L. Georgiadis, L. F. Merakos, and P. Papantoni-Kazakos, A method for the delay analysis of random multiple-access algorithms whose delay process is regenerative, *IEEE J. Sel. Areas Comm.,* **5** (6): 1051–1062, 1987.
54. L. Georgiadis and M. Paterakis, Bounds on the Delay Distribution of Window Random-Access Algorithms, *IEEE Trans. Commun.,* **COM-41**: 1993, 683–693.
55. G. Polyzos and M. Molle, A Queuing Theoretic Approach to the Delay Analysis for the FCFS 0.487 Conflict Resolution Algorithm, *IEEE Trans. Inf. Theory,* **IT-39**: 1887–1906, 1993.

MOSHE SIDI
Technion—Israel Institute of Technology

MULTIPLIER. See ANALOG MOS MULTIPLIER.