

DISTRIBUTED MULTIPLE ACCESS PROTOCOLS AND REAL-TIME COMMUNICATION

CUCS-89-83

James F. Kurose and Yechiam Yemini
Department of Computer Science

Mischa Schwartz
Department of Electrical Engineering

Columbia University
New York, New York 10027

ABSTRACT

In the past 10 years the field of distributed multiple access communication has developed into a major area of both practical and theoretical interest within the field of computer communications. The multiple access problem arises from the necessity of sharing a single communication channel among a community of distributed users. The distributed algorithm used by the stations to share the channel is known as the multiple access protocol. This paper examines the multiple access problem and various approaches towards its resolution.

This survey first defines the multiple access problem and then presents the underlying issues and difficulties in multiple access communication. A taxonomy for multiple access protocols is then developed in order to characterize common approaches and to provide a framework in which these protocols can be compared and contrasted. Different proposed protocols are then described and discussed and aspects of their performance are examined. The use of multiple access protocols for "real-time" or "time-constrained" communication applications, such as voice transmission, is examined next. Issues in real-time communication are identified and recent work in the design of real-time multiple access protocols is surveyed.

TABLE OF CONTENTS

1. Introduction	1
2. Multiple Access Communication	3
2.1 The Multiple Access Problem	3
2.2 A Taxonomy For Multiple Access Protocols	4
3. A Survey of Multiple Access Protocols	7
3.1 Controlled Predetermined Channel Allocation Protocols	7
3.2 Controlled Demand-Adaptive Protocols	8
3.3 Contention-based Protocols	11
3.3.1 <i>Probabilistic Partitioning</i>	12
3.3.2 <i>Address Partitioning</i>	14
3.3.3 <i>Time Partitioning</i>	15
3.4 The Performance of Distributed Multiple Access Protocols	18
3.4.1 <i>Issues in Evaluating Multiple Access Protocols</i>	18
3.4.2 <i>The Capacity of Distributed Multiple Access Protocols</i>	19
3.4.3 <i>Average Time Delay and Average Throughput</i>	23
4. Time-Constrained Communication in a Distributed Environment	27
4.1 Time-Constrained Communication	27
4.2 Predetermined Channel Allocation Protocols	29
4.3 Demand-Adaptive Protocols	31
4.4 Contention-based Protocols	34
5. Conclusion	37

LIST OF FIGURES

Figure 2-1:	A Taxonomy of Multiple Access Protocols	5
Figure 3-1:	Time Division Multiple Access	8
Figure 3-2:	Two Reservation Protocols	9
Figure 3-3:	MSAP/BRAM an imaginary token passing scheme	10
Figure 3-4:	EXPRESS-NET channel connections	11
Figure 3-5:	Message vulnerability in ALOHA	12
Figure 3-6:	Stations as leaves on a binary tree	14
Figure 3-7:	The time window protocol	16
Figure 3-8:	Protocol capacity as a function of α	21
Figure 3-9:	Time delay versus throughput tradeoffs	24
Figure 3-10:	Time delay and throughput of the Urn protocol	26
Figure 4-1:	The effects of an additional initial waiting time	33

1. INTRODUCTION

A little over a decade ago, the first multiple access protocols were used in the ALOHA system [Abramson 70], a single network consisting of a centralized computer and remote terminal stations on the Hawaiian islands. Since then, their use has spread into hundreds of today's communications networks. Continuing advances in fiber optics technology and the advent of cable (CATV) systems providing communication capabilities within an entire metropolitan region indicate that these multiple access communication protocols will continue to play an important role in the communication networks of tomorrow. Existing or proposed distributed multiple access protocols provide communication capabilities within these networks for such diverse applications as interactive terminal to computer communication, computer to computer communication, inter-office communication (including data, voice, and facsimile traffic), distributed sensor network applications and satellite communication among distributed earth stations.

This paper identifies the underlying issues and problems in distributed multiple access communication and surveys the sizable amount of work which has been done in the design of distributed multiple access protocols. Specific issues arising in the context of "real-time" communication applications are also identified and the applicability of various classes of multiple access protocols in a real-time environment is examined.

In section 2, we define the multiple access problem and examine some of the inherent difficulties in achieving communication among distributed users who share a single communication channel. A taxonomy or classification system is then proposed in order to characterize common approaches taken in various proposed protocols and to provide a framework in which these protocols can be examined, compared and contrasted. Some proposed multiple access protocols are then surveyed in section 3 and the ideas and mechanisms underlying their operation are identified and discussed. Various performance issues are considered and performance measures are presented and examined for various multiple access protocols.

Recently, there has been considerable interest in the use of multiple access protocols for real-time or time-constrained communication applications such as voice transmission and distributed sensor networks. The problem of time-constrained communication in a multiple access environment is examined in section 4. The various characteristics and performance issues for time-constrained communication are first presented and their impact on protocol design issues is then discussed. The various general approaches towards achieving multiple

access communication are then re-examined in the context of time-constrained applications and recent work in this area is surveyed.

2. MULTIPLE ACCESS COMMUNICATION

2.1 THE MULTIPLE ACCESS PROBLEM

Let us consider a situation in which geographically distributed stations (users) wish to communicate over a single communication channel. This channel provides the only means of communication among the stations and its properties are such that only a single message can be successfully transmitted over it at any one time. If two or more messages are simultaneously transmitted on the channel, then these messages interfere with each other and none of them will be correctly received by the station(s) for which they were destined. Such an environment is known as a *multiple access environment*. Since all stations can monitor the single communications channel, a message sent by one station can be detected or "heard" by all the other stations; for this reason the multiple access environment is also known as one type of *broadcast environment*.

There are numerous examples of multiple access environments. An everyday example is a group of conversants and the air between them. The air provides the single physical medium through which the people must communicate. As everyone knows, if two or more people talk at once, the usual result is that no one understands what anyone has said. (Actually, the human hearing system can often filter out all but one of the simultaneous conversations, so the analogy here is not exact.) A satellite channel and geographically distributed earth stations (e.g., the ALOHA system [Abramson 73]) also constitute a multiple access environment. In a satellite network, the earth stations transmit messages up to a satellite transponder which then relays the messages down to the earth stations. Simultaneous transmissions by the earth stations or the satellite transponder result in message interference and the reception of unintelligible messages at the earth stations. Another type of multiple access environment is a ground packet radio network [Kahn 78] in which (possibly mobile) distributed stations communicate over a single radio channel. Radio waves propagate through the media between the stations and interfering radio waves (i.e., simultaneous transmissions by two or more stations) again result in unintelligible message reception at the destination stations. Perhaps the most frequently cited example of multiple access environments are local area networks [Clark 78] such as Ethernet [Metcalfe 76], in which distributed stations share a single coaxial cable or optical fiber as the sole communication medium.

Since only a single station can successfully transmit a message at any given time, the

distributed stations must somehow coordinate their access to the channel in order to share the channel among themselves. A distributed algorithm by which the stations share the channel is known as a *multiple access protocol*. Several issues complicate the problem of distributed channel sharing. First, since the stations are distributed, they must either explicitly or implicitly communicate information to each other if they wish to coordinate channel sharing. However, since there is only a single communication channel, coordination among the users about sharing the channel necessarily requires use of the channel itself! Thus, there is a circular or recursive aspect to the problem. Secondly, since the stations are distributed, they can never instantaneously know the present status of other stations in the environment; information about other stations is always at least as old as the message propagation delay between stations. This is analogous to the situation in which the light we see from the stars actually originated at the stars millions of years ago.

Over the past ten years, numerous approaches have been proposed to resolve the multiple access problem. In the following section we will provide a broad overview of these approaches and develop a taxonomy or classification system for multiple access protocols.

2.2 A TAXONOMY FOR MULTIPLE ACCESS PROTOCOLS

As previously discussed, the multiple access problem is to grant channel transmission rights to a single station which requires the use of the channel at that time. Multiple access protocols, which determine these channel transmission rights or equivalently, allocate the channel (a resource) among the stations, divide broadly into two classes: *controlled-access* and *contention-based* protocols. These two classes and their further subdivisions are indicated in the multiple access protocol taxonomy shown in figure 2-1.

Controlled-access protocols are characterized by collision-free access to the channel. That is, the distributed stations are coordinated in such a way that two or more stations never attempt to transmit messages simultaneously. This coordination is typically achieved by imposing an ordering on the allocation of channel access (transmission) rights to the stations. Controlled-access protocols can be further characterized by whether the allocation of channel transmission rights varies in response to the changing transmission requirements of the stations. *Predetermined channel allocation* (PCA) protocols allocate the channel to the stations in a static manner and thus are not responsive to changing transmission requirements of stations. *Demand-adaptive* protocols attempt to allocate the

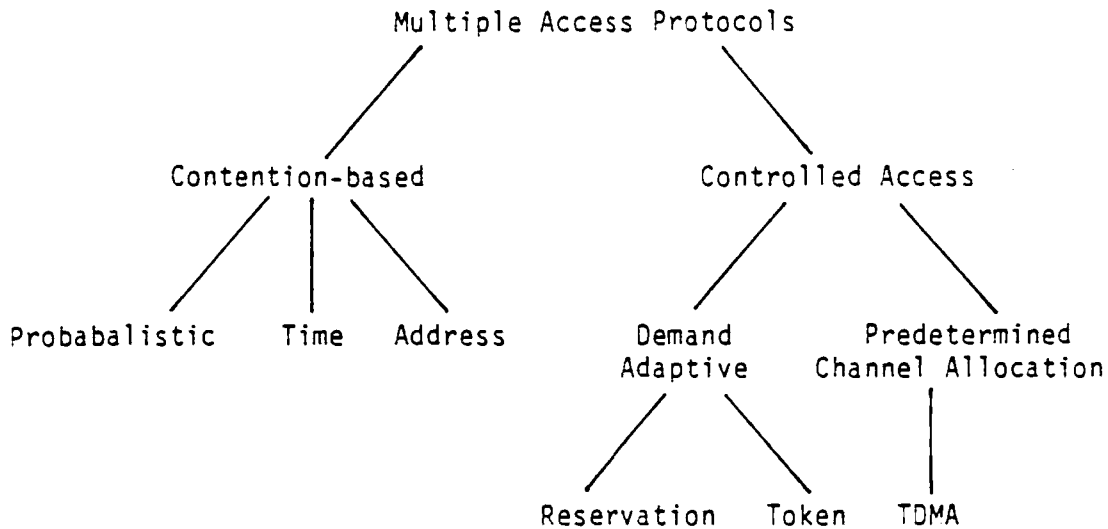


Figure 2-1: A Taxonomy of Multiple Access Protocols

channel in a manner more consistent with the immediate requirements or demands of the stations. In subsequent sections, it will be shown that both PCA protocols and demand-adaptive protocols are inefficient if there are a large number of stations with *bursty* message transmissions. When message transmissions are bursty, each station is usually idle (i.e., has nothing to transmit) but occasionally does have a large amount of data to send. The inefficiency in controlled-access protocols results from granting transmission rights (in PCA protocols) or the opportunity to claim transmission rights (in demand-adaptive protocols), in order, to *all* stations, regardless of whether or not they have messages to transmit. Thus, idle stations receive (unneeded) transmission rights while stations with messages must wait their turn before transmitting.

The second broad class of multiple access protocols, known as contention-based protocols, attempt to overcome this inefficiency by simultaneously offering transmission rights to a group of stations in the hope that exactly one of the stations has a message to send. Contention-based protocols thus operate by partitioning the stations in the network into a set of *enabled* stations (those with transmission rights) and a set of *disabled* stations (those without transmission rights); station addresses, arrival times of messages and explicit probabilistic mechanisms have been proposed as criteria for determining whether a station is enabled or disabled. If none of the enabled stations are ready (i.e., have a message to send), then the channel remains unused and a new partitioning of the stations can then eventually be determined. If exactly one enabled station is ready, then that station

transmits its message. Typically, at the end of this transmission, a new partitioning is then determined. Finally, if two or more ready stations are in the enabled set, then they transmit colliding messages. If stations can detect and abort collided transmissions, then the enabled set is often further divided in an attempt to isolate a single ready station in the enabled set. If collided transmissions cannot be detected and aborted, then a new enabled set is typically determined at the end of the colliding transmissions.

Before beginning a discussion of the various multiple access protocols, let us conclude this section by suggesting an alternate (and potentially valuable) classification of these protocols. There is a spectrum of information ranging from no information to perfect information about the state of all stations in the network. All protocols operate somewhere along this spectrum and each protocol operates using different information. Thus, multiple access protocols can potentially be characterized in terms of the information they use. But exactly what information is used? Three types can be readily identified. First, there is "hard-wired" information (e.g. a predetermined polling) known to each station when it begins operation. There is also global information that is obtained from the channel. Finally, there is local information known only to a single station (e.g., the arrival time of a message at the station). Local information can be transformed into global information when it is transmitted over the channel. Note that the use of local information may result in the lack of perfect coordination among the stations. For example, in contention-based protocols, if two stations use local information (e.g., a message arrival time or the value of a local random number) to determine whether or not to access the channel, then they may transmit colliding messages. The perfect coordination (i.e. absence of collisions) in controlled-access protocols results from the use of hard-wired information (e.g., a predetermined transmission order) known to all stations, as well as global information. Note, however, that there is a price paid for this global information since idle channel time is used to indicate that a station has no message to send.

Determining the nature and extent of information used by a protocol is surely a difficult task, but potentially a valuable one. An understanding of exactly *what* information is used could potentially lead to an understanding of its *value*. It would then be possible to determine what information is important in determining protocol performance for a given application and what additional information, if any, would be useful. Such an understanding could provide for a qualitative evaluation of the performance of protocols (e.g., ordering their performance for a given application) based on the information they possess without resorting to a quantitative (and potentially difficult) performance analysis.

3. A SURVEY OF MULTIPLE ACCESS PROTOCOLS

In this section, various proposed multiple access protocols will be briefly described and discussed. These protocols have been logically grouped according to the taxonomy presented in the previous section. Due to space limitations, the descriptions are necessarily brief and some refinements to the basic mechanisms have been omitted.

3.1 CONTROLLED PREDETERMINED CHANNEL ALLOCATION PROTOCOLS

Predetermined channel allocation (PCA) protocols provide collision-free access to the communication channel and determine the channel transmission rights of stations in a static, predetermined manner. The most prevalent PCA protocols are time division multiple access (TDMA) protocols.

TDMA protocols provide collision-free multiple access broadcast communication by permitting each station to periodically utilize the full bandwidth of the single communication channel for some fixed amount of time. In this fashion, the channel is shared in time among the stations. Time is divided into fixed length intervals or frames; each frame is further subdivided into slots as shown in figure 3-1. In the simplest version of TDMA, the number of slots per frame is the same as the number of stations in the network and each station is granted use of the channel for the duration of one time slot per frame.

It is important to note that TDMA protocols are potentially very inefficient. One inefficiency arises when the number of stations in the network changes in time, as in ground radio environments with mobile stations. Since TDMA performs *a priori* slot assignment, slots are assigned to stations independent of whether or not they are currently in the network. When a station is not in the network, its slot remains idle. A second inefficiency arises even when the number of stations remains constant. When a station has nothing to send, its time slot is unused even though other stations could potentially utilize its time slot. This problem can be particularly acute when message arrivals are bursty and the number of stations is very large.

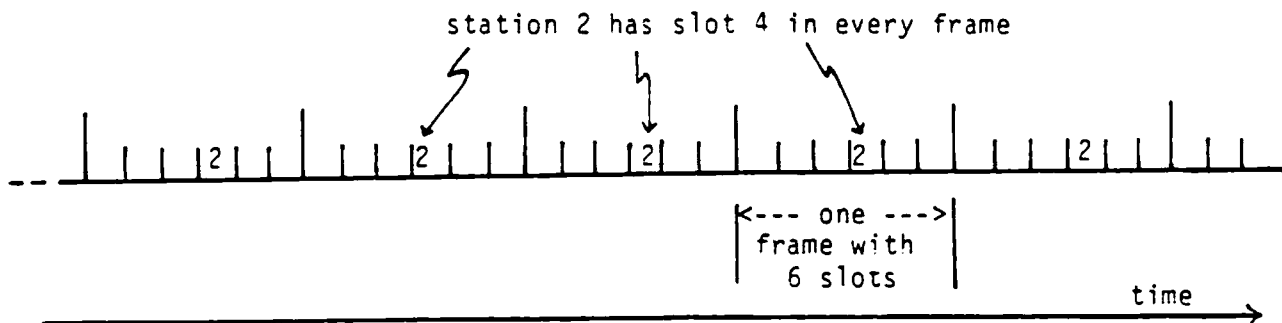


Figure 3-1: Time Division Multiple Access

3.2 CONTROLLED DEMAND-ADAPTIVE PROTOCOLS

Controlled demand-adaptive protocols were developed to overcome the previously noted inefficiencies of PCA protocols. They are based on the use of *hub polling* techniques [Schwartz 77] previously developed for communication networks with a centralized control. In demand-adaptive protocols, as in PCA protocols, channel access (transmission) rights are offered to stations according to some access order. This order may be determined *a priori* or can be dynamically determined by the stations. The inefficiencies of PCA protocols are overcome by requiring stations to immediately forfeit their access rights if they have no messages to send when they receive access rights. At the end of a station's transmission, access rights are passed to the next station in the order. Two mechanisms have been developed to maintain the access order: reservation schemes and token passing schemes.

The most straightforward reservation scheme is the basic bit-mapped protocol [Kleinrock 80]. Let us assume there are N stations and that each has a unique address between 1 and N . The bit-mapped reservation protocol consists of alternating periods of reservation posting and message transmission, as shown in figure 3-2 for the case of 6 stations. Each reservation slot is of length τ , the end-to-end propagation delay of the channel.



Figure 3-2a



Figure 3-2b

time →

Channel Activity as a Function of Time

Figure 3-2: Two Reservation Protocols

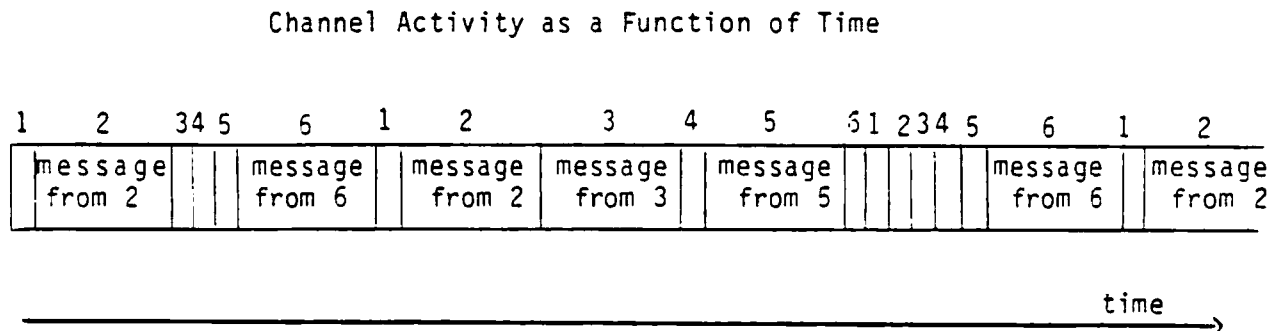
During the reservation period a station transmits a burst of noise (shown as a 1 in figure 3-2) during its reservation slot to indicate to the other stations that it is ready to transmit a message during the following message transmission period. All stations monitor the reservation process and thus each station knows the identities of the other stations that are ready to transmit.

In the protocol shown in figure 3-2a, a single message transmission follows the reservation period and then another reservation/transmission cycle begins. The one station to be granted transmission rights is selected from those stations having posted reservations according to some priority rule known to all stations. This protocol was proposed and analyzed in [Kleinrock 80] and several different priority rules were examined. A slight modification of this protocol is shown in figure 3-2b. In this protocol, all stations which post reservations are permitted to transmit their messages (once again, according to some known priority rule) before another reservation/transmission cycle begins. Note that while this scheme is more efficient (less channel time is used for reservations per message transmission), the first protocol permits higher priority stations to exercise their priority more often.

The second major class of controlled demand-adaptive protocols, called token passing

protocols. circulate a real or imaginary token message among the stations in such a manner that only a single station possesses the token message at any one time. By definition, the station possessing the token message possesses channel access rights. In [Bux 81], an explicit token message is circulated and a station cannot transmit until another station first sends it the token message. Once a station receives the token message, it transmits any messages it has and then sends the token on to another station.

BRAM (Broadcast Recognition Access Method) [Chlamtac 79] and MSAP (Minislotted Alternating Priorities) [Kleinrock 80] can be viewed as using an imaginary token to implement a form of distributed channel sharing. The order in which the stations are granted transmission rights (receive the token) is determined by the numerical order of their station addresses. Station 1 initially has transmission rights. If station 1 has a message to send, then it does so; otherwise it remains silent. The presence or absence of a transmission after τ units of time indicates to the other stations whether or not station 1 intends to send a message. If station 1 does not begin sending a message, transmission rights are implicitly passed to station 2, which repeats the same procedure as station 1 while the other stations monitor its activity. If station 1 decides to send a message, then transmission rights are passed to station 2 as soon as station 1 completes its transmission. This process is shown in figure 3-3 below.



Empty Channel Slots Indicate Idle Channel Time

Figure 3-3: MSAP/BRAM: an imaginary token passing scheme

In MSAP/BRAM, τ units of time are required to establish the absence of a message transmission. In EXPRESS-NET [Fratta 81] and in FASNET [Limb 82] the use of a folded *unidirectional* channel and a similar access protocol removes the necessity of waiting this

τ units of time. These schemes work as follows. Each station maintains an inbound connection and an outbound connection to the folded channel as shown in figure 3-4.

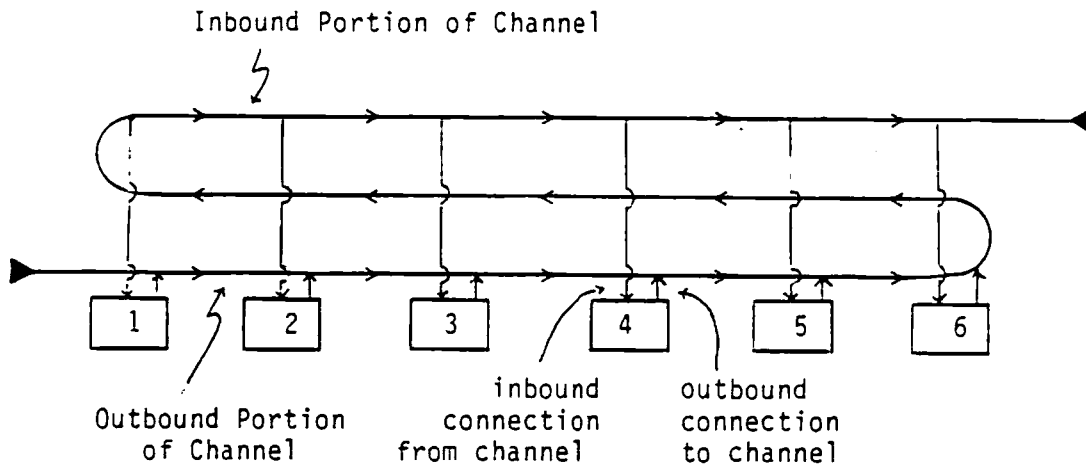


Figure 3-4: EXPRESS-NET channel connections

The underlying idea is that the station at the head of the channel (i.e. station 1 in figure 3-4) begins a "train" of messages which propagates down the unidirectional channel. When a station detects the end or "caboose" of the train passing its *outbound* connection, the station appends its message (if any) onto the message train. A station reads messages from the train as the train passes its *inbound* connection. Note that the spacing between messages on the train is only the amount of time required for a station to detect the end of a passing train and begin transmission of its message. Thus, unlike most other demand-adaptive protocols, the spacing between message transmission is independent of, and typically less than, τ . EXPRESSNET and FASNET will be further discussed in the second half of this paper.

3.3 CONTENTION-BASED PROTOCOLS

The second major class of multiple access protocols are known as contention-based or random access protocols and are characterized by the possibility that channel *contention* may result from two or more stations attempting to transmit messages simultaneously. We have previously characterized the operation of contention-based protocols as a partitioning process in which the set of all stations in the network is divided into an enabled set of

stations (stations with transmission rights) and a disabled set of stations. Three classes of mechanisms have been proposed and developed to perform this partitioning process: probabilistic mechanisms, time-based mechanisms and address-based mechanisms

3.3.1 Probabilistic Partitioning

The first contention-based solution to the multiple access problem is also the simplest. A station simply transmits a message when the message arrives. This solution, known as pure ALOHA, was first developed by Abramson [Abramson 70] and involves no coordination among the distributed stations. If two stations happen to transmit messages at the same time, then their messages interfere or "collide" and thus require later retransmission. Note that a station must schedule a retransmission to occur after a *random* amount of time. Otherwise, if two or more stations were to interfere and always schedule a retransmission to occur after the same amount of time, then these stations would interfere forever. Thus, the probabilistic element in the partitioning process permits interfering stations eventually to become ready at different times in the future.

In pure ALOHA, messages can interfere even if only the first bit of a message beginning transmission overlaps the very last bit of a message ending transmission. Thus, if all messages require t' time units to be transmitted, a station beginning transmission of a message at time t_0 is *vulnerable* to message collisions due to transmissions from other stations that began after $t_0 - t'$ or before $t_0 + t'$. The total amount of time the message is vulnerable is thus $2t'$. This situation is shown in figure 3-5 [Tannenbaum 81].

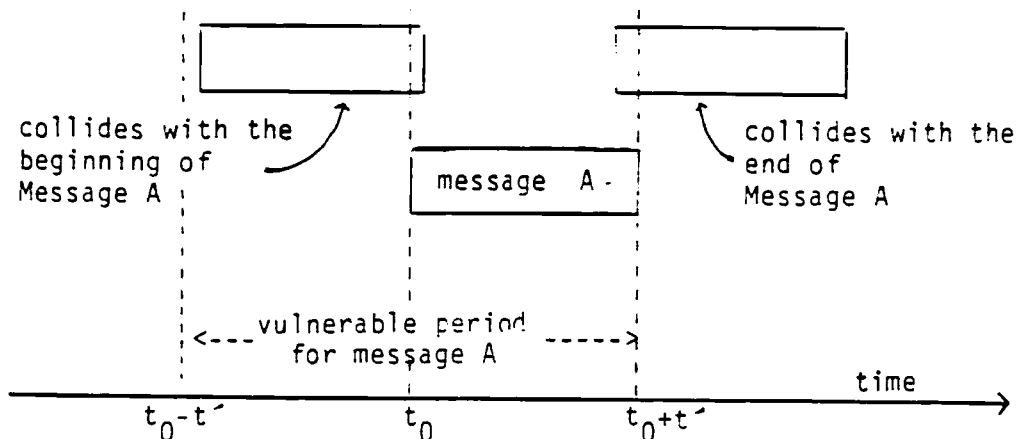


Figure 3-5: Message vulnerability in ALOHA

A modification of pure ALOHA, known as slotted ALOHA, was subsequently introduced. In slotted ALOHA, time is divided into intervals or *s/ots* of the same duration as a single message transmission time. A station can begin sending a message only at the beginning of one of these time slots. A message which arrives during a time slot cannot be sent until the beginning of the following time slot. Thus, only those messages which have arrived in the previous time slot can interfere with each other. The effect of this synchronization then is to cut the vulnerability period in half, from 2τ in pure ALOHA to τ in slotted ALOHA. Note that messages either collide completely for the duration of their transmission or do not collide at all.

Neither pure ALOHA nor slotted ALOHA use information about the state of the channel (e.g., whether or not a message is currently being transmitted) in determining the enabled set of stations. The class of protocols which have the ability to sense or "listen" to the channel and use this information in determining the enabled set are known as *carrier sense multiple access* (CSMA) protocols. Two of the earliest CSMA protocols are known as nonpersistent and p-persistent CSMA [Kleinrock 75a]. In nonpersistent CSMA, when a message arrives at a station, the station senses the channel in order to determine if another station is currently sending a message on the channel. If the channel is sensed empty, then its message is sent immediately. If the channel is sensed busy, then its message transmission is rescheduled for a later time according to some retransmission time distribution. A rescheduled message can be considered to "re-arrive" at this later time. Both slotted and unslotted versions of nonpersistent CSMA have been investigated.

In p-persistent CSMA, when a message arrives at a station, the station begins sensing the channel. When the channel is sensed to be empty, the station transmits its message with probability p . With probability $1-p$, the station waits some fixed amount of time and then senses the channel again. If the channel is again detected to be empty at this new point in time, the above procedure is repeated. However, if the channel is sensed busy (indicating that another station has begun transmission), then the transmission of the message is rescheduled for a later time as in nonpersistent CSMA. The impact of the value of p on the performance of p-persistent CSMA will be examined in section 3.4. A well known example of a persistent CSMA protocol is Ethernet [Metcalfe 76]. One proposed probabilistic retransmission mechanism for Ethernet is known as *binary backoff*. Once the channel is sensed empty, any station with a message to transmit attempts to do so. If a collision occurs, then all stations terminate their transmissions and randomly reschedule their transmissions over some period of time. The time period over which a given station reschedules its message transmission doubles each time the message experiences a collision.

Kleinrock and Yemini [Kleinrock 78] have described a slotted multiple access protocol known as the Urn protocol, which uses a probabilistic mechanism in a much different fashion. Suppose there are N stations and it is further known that some n of these stations are ready (i.e., have a message to transmit). The partitioning process which determines the set of enabled stations corresponds to selecting some k stations from an imaginary urn containing the N stations. If the k enabled stations contain exactly one ready station, then the partitioning has been successful; if the k stations contain either none or more than one ready station, then the partitioning process must be repeated. It can be shown that the value of k which maximizes the probability that *exactly* one of the k stations is ready is given by the integer part of N/n . Under the heavy traffic assumption that every station always has a message to send, then k always equals one. Thus the Urn scheme operates in a random TDMA-like fashion in the heavy traffic situation. That is, channel transmission rights are randomly passed among the stations and exactly one station has the channel transmission rights at any give time.

3.3.2 Address Partitioning

Hayes and Grami [Hayes 78] [Grami 82] and Capetanakis [Capetanakis 79] have proposed contention-based protocols which use station addresses to determine the enabled stations. In the simplest case, address partitioning works as follows. We can think of each of the N stations as a leaf in a binary tree as shown in figure 3-6.

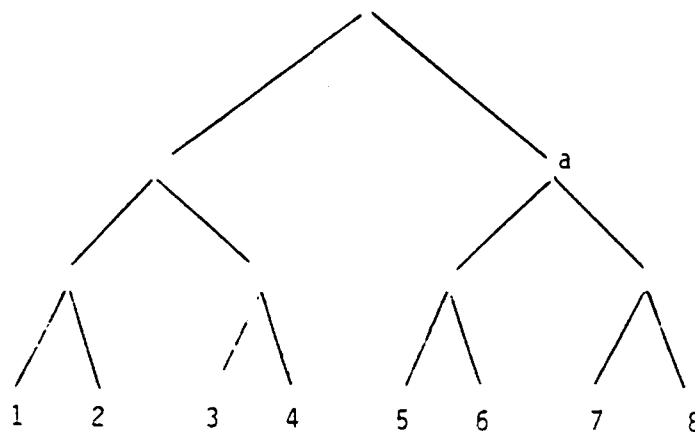


Figure 3-6: Stations as leaves on a binary tree

Time is slotted and after a successful transmission on the channel, all stations with a message to send initially attempt to do so. If a collision occurs, the set of stations is partitioned such that only stations in one of the halves of the binary tree are enabled (e.g., the subtree rooted at a in figure 3-6) in the subsequent time slot. If further collisions occur, then the enabled set is continually halved until the enabled set eventually has only one ready station. If at some point the tree is halved and there are no ready stations in the enabled half of the tree, then an empty slot occurs and the other half of the tree becomes enabled.

Several improvements exist on the basic approach. If the number of ready stations can be estimated, then the tree can be partitioned in such a way as to maximize the probability that the enabled set contains exactly one ready station. In the heavy traffic case, in which all stations have a message to send, the initially enabled set would be chosen to contain only a single station. In the light traffic case, the entire tree would initially be enabled. Note that this approach has much the same flavor as the Urn protocol. It should also be noted that if trees are deterministically split by the partitioning process, then certain ready stations will always transmit their messages before other ready stations. This *de facto* priority can be avoided by introducing randomization into the tree splitting.

3.3.3 Time Partitioning

The final group of contention-based protocols to be discussed uses the arrival times of messages at a station to determine the set of enabled stations.

Gallager [Gallager 78] and Tsybakov [Tsybakov 79] independently developed a protocol in which the enabled set of stations are those stations with message arrivals occurring during some interval of time ("window") in the past. The basic operation of the protocol is shown in figure 3-7. Message arrival times at all stations in the network are shown below the time axes in figure 3-7.

The protocol maintains a value, t_0 , such that all messages which have arrived before t_0 have either been transmitted or have been rescheduled for transmission at some time later than t_0 . Time is slotted and all stations know the initial size, w , of the time window. When the channel is sensed empty at the beginning of a time slot, any station having a message with an arrival time between t_0 and $t_0 + w$ transmits the message. In figure 3-7 stations 1 and 4 both have such a message. If a collision results, then the size of the

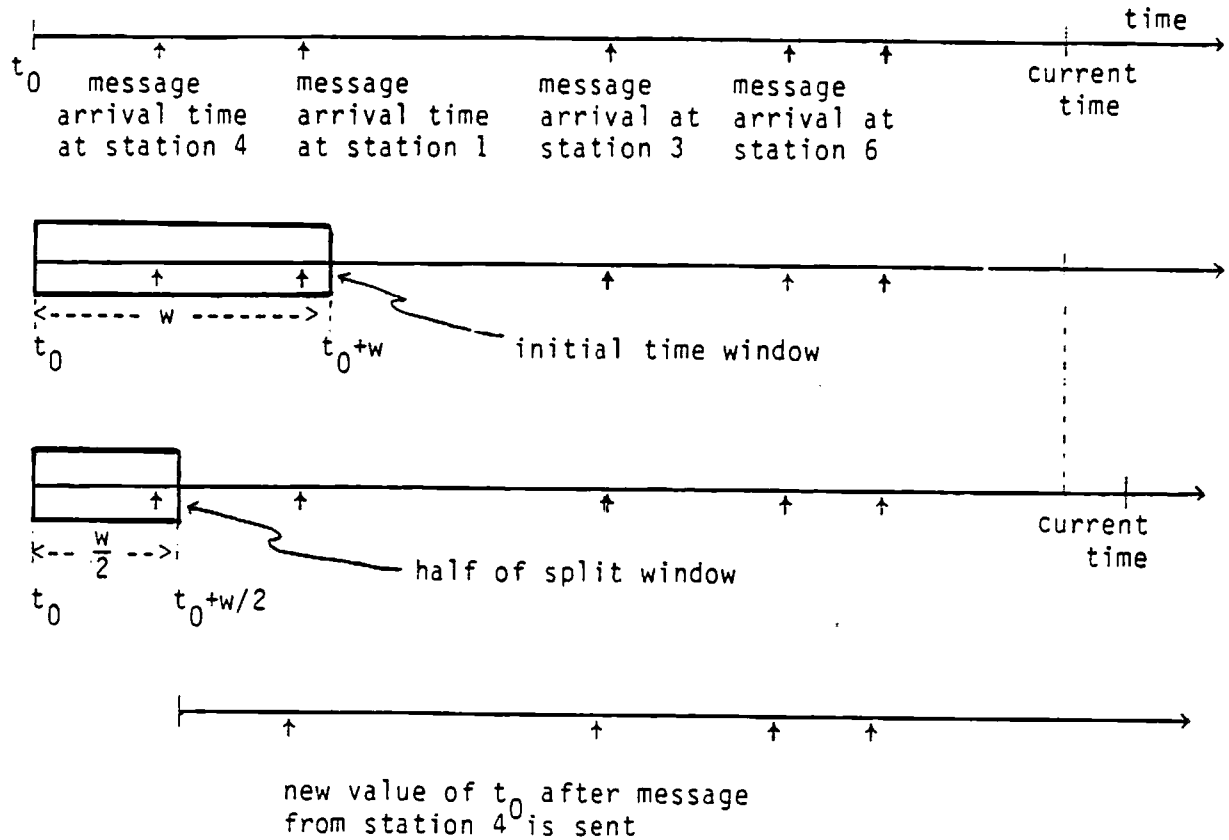


Figure 3-7: The time window protocol

window is cut in half and the older half of the cut window (i.e., the window of length $w/2$ beginning at t_0) becomes the new window. The above procedure is then repeated using this new window. If a window is split in half and no arrivals are in the selected half, an idle slot appears on the channel and the stations can continue the partitioning process on the other half of the split window. Once a window of length w' has been found to contain no message arrivals (i.e., after a single message in the window has been transmitted or if the window contained no arrivals in the first place), the value of t_0 can be assigned a new value of $t_0 + w'$; this new value of t_0 is shown on the fourth time axis in figure 3-7. Eslanadidi [Eslanadidi 82] and Towsley [Towsley 82] have studied different methods for choosing an optimal initial window size. Kurose [Kurose 83] has extended the window

method to permit the window to be arbitrarily placed between t_0 and the current time; this extension was shown to be important when the protocol is used in time-constrained communication applications.

Molle [Molle 81] has proposed a related multiple access protocol in which each station maintains two clocks: a normal clock and a virtual clock. The virtual clock is essentially used to define the value of t_0 as in the protocols of Gallager and Tsybakov. The virtual clocks can run in either slotted or continuous time, at a possibly variable speed and all stations run their virtual clocks in the same manner. When a station's virtual clock time equals the arrival time of a message at the station, that station stops its virtual clock and sends the message. If messages collide, then their transmissions are rescheduled for a later time, as in the nonpersistent CSMA protocol. When other stations detect a transmission on the channel, they too stop their virtual clocks and restart them only when channel activity ceases. In this manner the virtual clock sweeps out time such that when it reads t_0 , all messages which have arrived before t_0 have either been transmitted or have been rescheduled for later transmission.

3.4 THE PERFORMANCE OF DISTRIBUTED MULTIPLE ACCESS PROTOCOLS

3.4.1 Issues in Evaluating Multiple Access Protocols

Traditionally, the "performance" of a distributed multiple access has been characterized by the maximum number of messages that it can deliver per unit time and by its time delay/throughput tradeoff. This tradeoff reflects the effect of an increasing arrival rate on the average message delay, defined as the amount of time between the message's arrival at a sending station and its successful reception at a destination station. These performance results will be surveyed in sections 3.4.2 and 3.4.3 respectively. However, while they are certainly the most frequently cited performance measures in the literature, other issues (which have typically received less attention) must also be considered in the evaluation of an access protocol.

The *reliability* of the protocol and its ability to operate in spite of station failures is a critical issue for any multiple access protocol. A related issue is the *robustness* of the protocol, or its insensitivity to errors, channel noise and misinformation. The *stability* of the protocol, or its ability to operate in spite of varying traffic demands and short term overloading of the network is also an important issue. The stability issue has been addressed by [Tobagi 77], [Fayolle 77], [Mittal 81] and others. The ability of the protocol to support different classes of traffic and different priority levels is currently of great interest due to the possibility of developing integrated services data networks [Gitman 77] [Skrzypczak 81] based around multiple access channels; this topic will be further investigated in section 4. Finally, the engineering aspects of protocol implementation [Saltzer 81], the complexity of the hardware and software and the ease of maintenance are also important issues if a protocol is ever to be actually implemented.

A complete discussion of these issues is beyond the scope of this paper; the references cited above provide a more detailed discussion of these issues. In the following two sections, the performance issues of protocol capacity and the time delay/throughput tradeoff will be examined.

3.4.2 The Capacity of Distributed Multiple Access Protocols

Since stations in a multiple access environment are geographically distributed, the distributed protocols almost always require the exchange of control information in order to achieve some form of coordination. This information can either be explicitly communicated (e.g., reservations) or implicitly communicated (e.g., by a known ordering or by channel activity or silence). Since some protocols require the use of the channel for explicit control information as well as for successful message transmission, the channel will not always be used for "useful" work, i.e., the successful transmission of messages. The fraction of channel time actually used by successfully transmitted messages, as opposed to control information, is known as the *effective utilization* of the channel. The maximum value of the effective utilization over all possible traffic loads is known as the *capacity* of the protocol. Note that if control information is exchanged over the channel, the capacity of the protocol will be less than unity.

Several aspects of the multiple access environment and the behavior of the protocol itself influence protocol capacity. Capacity is perhaps most greatly affected by the value of the *normalized* end-to-end channel propagation delay of the channel, α , defined as:

$$\alpha = \tau/M$$

where τ is the end-to-end propagation delay of the channel and M is the average message length (in units of time). The importance of τ is clear: it represents the maximum amount of time into the past for which a station has no information about other stations. In collision-based protocols, a station that senses an empty channel and decides to transmit a message may still interfere with another station that decided to begin transmission within the past τ . Both stations, having sensed an empty channel, will nonetheless transmit colliding messages, resulting in the "non-productive" use of the channel. In reservation schemes, τ represents the minimum reservation slot length needed to insure that all stations know the content of one reservation slot before the next reservation slot begins. The fact that protocol capacity also depends on M reflects a scaling effect. For example, if the reservation or contention period preceding a successful transmission is $n\tau$ (for some value n) and the message is of length M , then the effective channel utilization is the same as if the length of the contention/reservation period was $kn\tau$ and the message length was kM . The equivalence of these two situations is reflected by their identical values of α .

The ability of stations to detect message collisions and subsequently abort transmission of their messages also influences protocol capacity. Collision detection assures that the channel is not wasted by the transmission of the entire length of a colliding message

(which will require retransmission in any case). Note that the maximum amount of time needed for a station to determine that no other stations have interfered with its transmission is 2τ , the maximum round trip end-to-end propagation delay. A third aspect of protocol operation affecting capacity is whether or not a protocol operates in slotted time. If time is divided into slots, stations transmit only when an empty channel is detected and stations begin their transmissions only at the beginning of a slot, then messages either interfere completely or do not interfere at all. Slotted time thus helps minimize the waste of channel time for the transmission of colliding messages.

In figure 3-8 we show a comparison of the capacities of some of the previously described protocols as a function of α . These results are from various sources and in some cases have been obtained under differing assumptions. Their reproduction here is meant to permit a qualitative comparison of protocol capacities and to provide insight into the operation of the protocols.

The protocol with generally the lowest capacity is pure ALOHA. The complete lack of coordination of stations in ALOHA results in a maximum of only 18% of the full communication capacity of the channel ever being utilized for useful communication! Slotted ALOHA provides a twofold increase in capacity over pure ALOHA. This is due to the previously noted twofold decrease in a message's vulnerable period under slotted ALOHA. It should be noted that the ALOHA capacity results shown in figure 3-8 are not those most frequently cited in the literature [Abramson 70] [Abramson 73], in which the end-to-end propagation delay of the channel was not considered. In that analysis, once a station terminated a message transmission, the message was no longer considered to occupy the channel. The more realistic model used here reflects the fact that a message must propagate the entire length of the channel before the channel becomes truly free. In this case, a message continues to occupy the channel for τ units of time after transmission by the sending station terminates. This consideration is reflected in the ALOHA capacity results in figure 3-8; similar capacity results are cited in [Molle 81].

Figure 3-8 indicates that pure ALOHA does not always have the lowest capacity. For a large number of stations (e.g., 50) the cost of coordination can require so much overhead in reservation slots that the capacity of some bit-mapped protocols is lower than if there was no coordination in the first place [Kleinrock 80]! As expected, figure 3-8 shows that the larger the number of stations, the larger the number of reservation slots and the lower the capacity of the bit-mapped protocol.

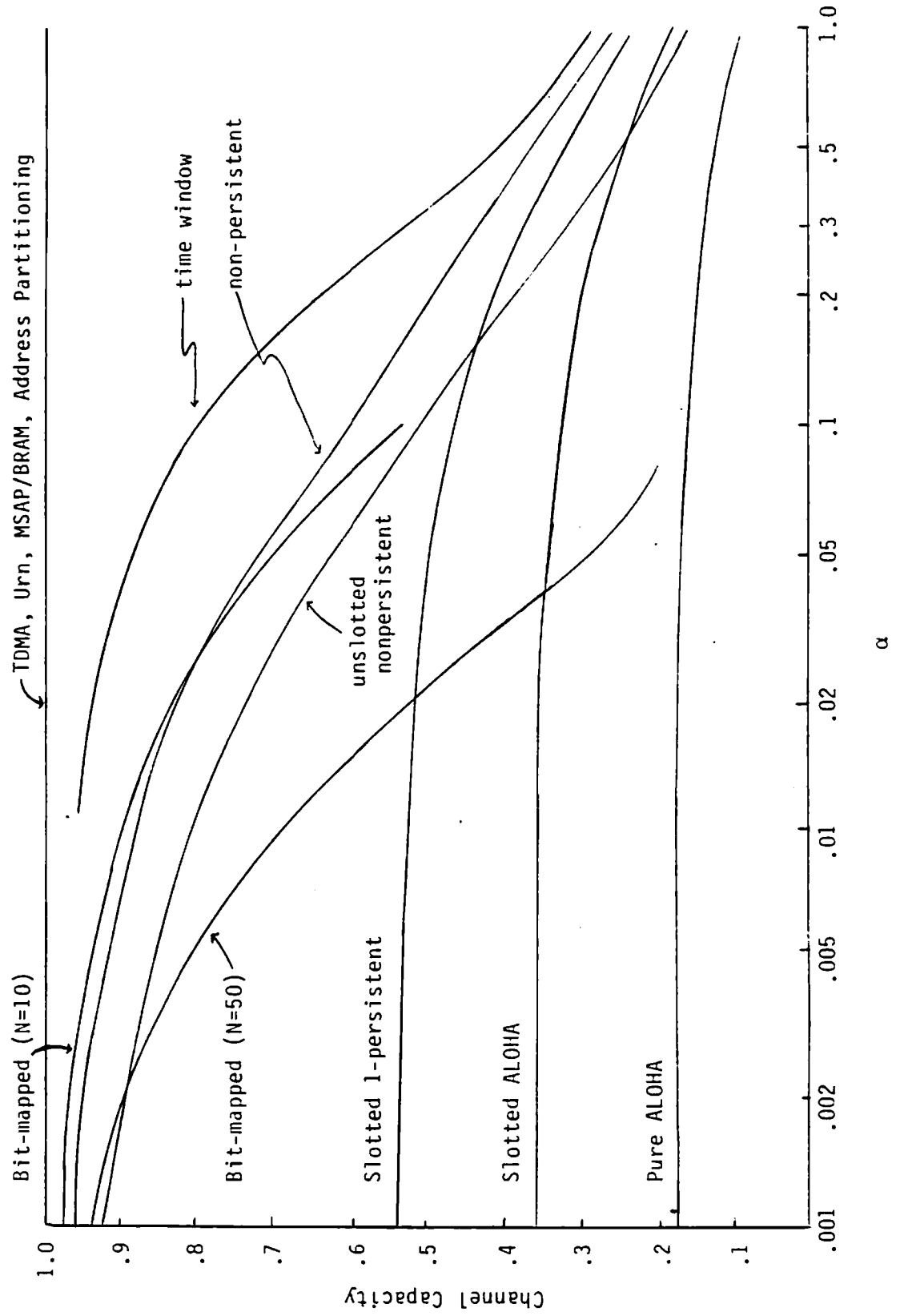
Figure 3-8: Protocol capacity as a function of α 

Figure 3-8 also indicates that those protocols with carrier sense capabilities generally have a higher capacity than ALOHA type protocols. The slotted 1-persistent and slotted nonpersistent capacity results are from [Kleinrock 75a]. Note that while the slotted 1-persistent capacity is approximately 50% at best, unslotted nonpersistent CSMA can achieve channel utilizations better than 90% for some values of α . The capacity results for slotted-nonpersistent CSMA are from [Molle 81] and correct previous results from [Kleinrock 75a]. The capacity of Molle's virtual time CSMA protocol is similar to that of slotted nonpersistent CSMA. The capacity results for window protocols is from [Eslanadidi 82] for the case in which a protocol can determine not only whether or not a collision occurred but also the number of messages in a collision. Georgiadis [Georgiadis 81] has proposed a similar protocol which uses channel energy detectors to determine the number of messages in a collision. For some values of α this additional information can result in a substantial capacity increase over CSMA schemes.

Perhaps the most interesting result in figure 3-8 is the 100% capacity of TDMA, implicit token passing (FASNET, EXPRESSNET), the Urn protocol and MSAP/BRAM. The significant underlying common characteristics of these protocols are that no explicit control information is exchanged and that, under heavy traffic conditions, each station in turn is given access to the channel according to some predetermined order. Thus, for a finite number of stations and under the heavy traffic assumption that every station always has a message to send, the channel is constantly used for useful message transmission. A station waits for its turn in the transmission sequence and then simply transmits its message.

The apparent superiority of the protocols with 100% capacity should not be overestimated. Protocol capacity is only one measure of protocol performance. It is an important performance measure in that it bounds the maximum amount of useful communication that a protocol can possibly provide. Thus, it provides a yes/no answer as to whether a given protocol can support a given traffic load. On the other hand, the theoretical capacity limit may only be achievable under circumstances which, in practice, are unrealistic. For example, the 100% capacity of the TDMA-like protocols is only achievable under heavy traffic loads, in which case the message delays may be intolerably large. Furthermore, the capacity of a protocol only represents a static measure of optimal protocol performance. It reflects protocol performance for only a *single* traffic arrival rate (i.e., that arrival rate which maximizes the effective utilization) and provides *no* information about protocol behavior for other traffic arrival rates.

Thus, capacity can at best only partially characterize protocol performance. In the

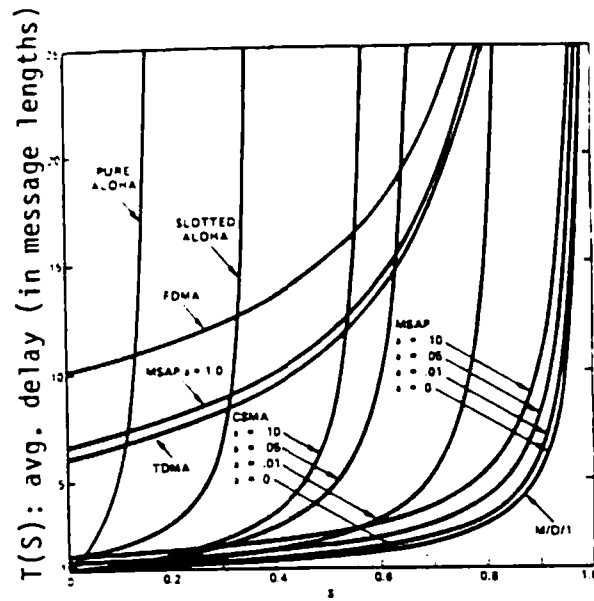
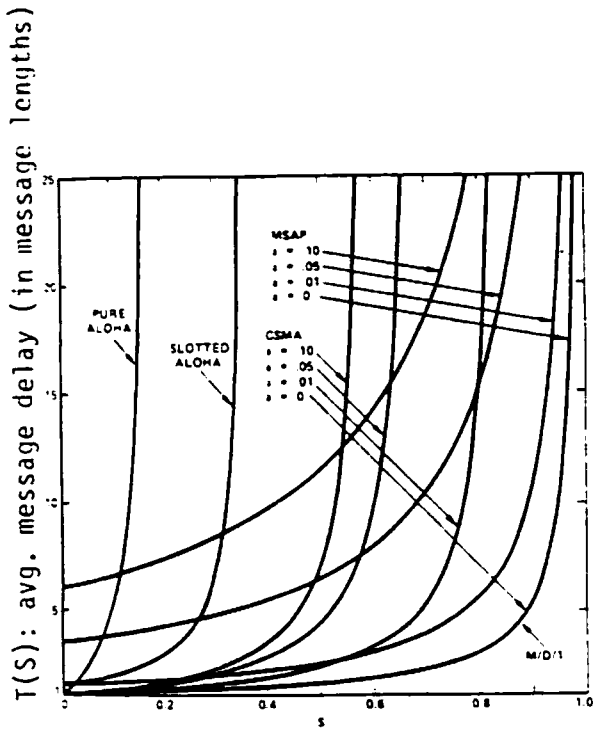
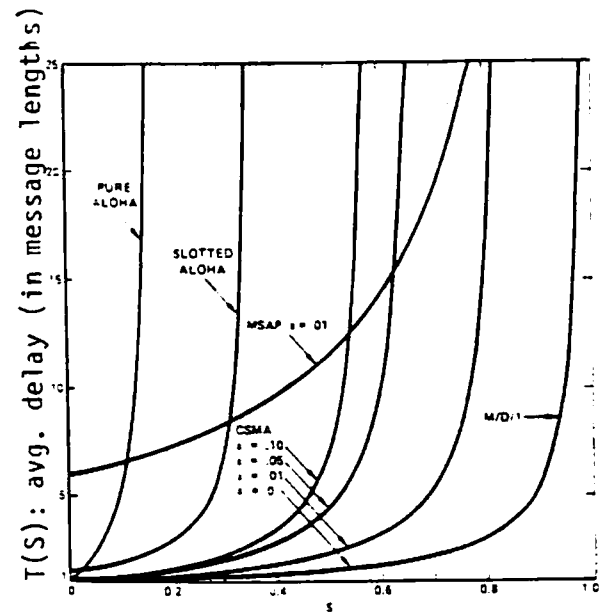
following section we examine a second performance measure of more practical interest the relationship between the arrival rate of messages to the network and the average message time delay.

3.4.3 Average Time Delay and Average Throughput

Before presenting a quantitative comparison of the time delay versus throughput performance of various multiple access protocols, let us first discuss two system parameters which greatly affect protocol performance and then discuss an upper bound on the performance of *any* distributed multiple access protocol. The first system parameter of interest is N , the number of stations in the network. As previously indicated, the number of stations in the network influences the performance of both PCA and demand-adaptive protocols. The performance of contention-based protocols is influenced by the traffic generated by the stations rather than the specific number of stations in the network. The parameter α , gives the end-to-end propagation delay of the channel in units of message transmission time and reflects the channel propagation time required to communicate from one station to another. Since the station access order is determined *a priori* in PCA protocols, no information is explicitly exchanged to coordinate channel sharing. The performance of PCA protocols is thus independent of the value of α . However, most demand-adaptive and contention-based protocols involve some form of explicit communication to achieve coordination; thus, these protocols will be affected by the value of α . Finally, we note that in a centralized environment in which all messages arrive at a single central location, messages can be selected for transmission without contention or overhead. Such a centralized system, which can be modeled as an M/D/1 queue [Kleinrock 75b], thus provides a bound for the optimal time delay versus throughput tradeoff for *any* multiple access distributed message transmission system.

Figures 3-9a through 3-9c are taken from [Kleinrock 77] and provide quantitative delay versus throughput tradeoffs for various protocols representative of the previously identified major classes of multiple access protocols (PCA, demand adaptive and contention-based). We have chosen not to include performance curves for all the previously described protocols in an attempt to characterize the relative performance of the *classes* of protocols and to emphasize how the mechanisms underlying each class affects their performance. Such an approach overlooks (often subtle) performance differences that exist between protocols within a given class. A discussion of these

Figure 3-9: Time delay versus throughput tradeoffs

Figure 3-9a: $N=10$ stationsFigure 3-9b: $N=100$ stationsFigure 3-9c: $N = 1000$ stations

differences can be found in the cited references. MSAP is the only demand-adaptive protocol shown in figure 3-9. Since MSAP uses implicit reservations while other demand-adaptive protocols require channel time for explicit reservations or token passing, MSAP offers better throughput and time delay characteristics than other demand adaptive schemes. Two ALOHA protocols and CSMA demonstrate performance characteristics of many of the contention-based protocols.

In figure 3-9, fixed message lengths are assumed and the time scale has been normalized to a message transmission time. S represents the combined arrival rate (or throughput) of messages to all stations in the network. $T(S)$ is the average message delay, i.e., the time between a message's arrival at a station and its successful reception at a destination station. The value of a corresponds to α , the normalized end-to-end propagation delay of the network. Note that the time delay versus throughput tradeoff shown in figure 3-9 is typical of most shared resource systems: as the number of jobs (messages) contending for the resource (the channel) increases, there is a concomitant increase in the average time needed to acquire the resource (i.e., successfully transmit the message). It should also be noted that the performance curves exhibit asymptotic behavior at some throughput value; this throughput value is exactly the previously examined capacity of the protocol.

Let us first examine figure 3-9a. As expected, the performance of TDMA is independent of a . Both ALOHA protocols are also shown to be independent of a ; this is not strictly valid (since a message is vulnerable to collisions as it propagates down the channel after the sending station has finished transmission) but is a reasonable first approximation. Both CSMA and MSAP are dependent on a , as previously discussed. Note that as a approaches 0, the performance of CSMA and MSAP converge to the optimal M/D/1 performance. For a relatively small number (10) of stations, MSAP offers the best performance. For small and intermediate throughput values, CSMA performance is also fairly good, especially for small values of a . For high throughput values (i.e., the heavy traffic case), PCA protocols perform as well or better than most of the other protocols; this corroborates our earlier observation that PCA protocols would perform well in heavy traffic situations. However, for small throughput values, their performance is clearly inferior.

In figure 3-9b, the number of stations has been increased to 100. Note that PCA performance curves are not shown in this figure. In TDMA, even as the throughput approaches 0, the average time delay of a randomly arriving message is still half the length of a time frame, in this case 50 time units. Thus the performance of TDMA (and MSAP

with $a=1.0$) is literally "off the scale". The performance of CSMA and ALOHA is the same in figure 3-9b as in figure 3-9a since it is independent of the number of stations in the network. The performance of MSAP has degraded considerably with the increased number of stations and has shifted away from the optimal M/D/1 curve. In figure 3-9c, N has been further increased to 1000 stations and the performance of the contention-based protocols remains constant while that of MSAP deteriorates further.

Figure 3-9 indicates that no protocol or class of protocols performs well for all values of N and S . Thus, if a protocol operates under wide and varying values for these parameters, it should ideally adapt its operation to the current values of N and S . The protocols of Hayes and Capetanakis and the Urn protocol adapt well to changing values of the throughput. In the light traffic case, they operate in a random access fashion. Under heavy traffic loads, they operate in a TDMA-like fashion. These performance characteristics are shown in the time delay versus throughput tradeoff for the Urn protocol shown in figure 3-10 [Kleinrock 78]. The constant time delay for TDMA results from the assumption of random assignment of TDMA frames and the assumption that a station can buffer at most one message.

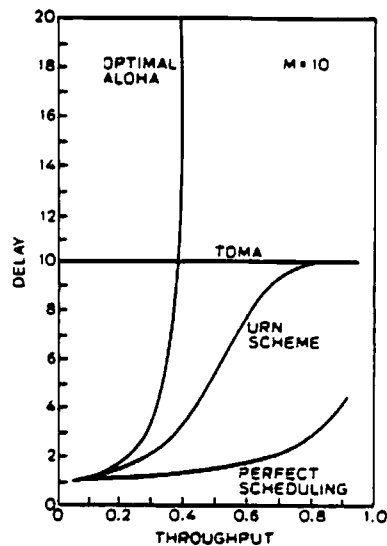


Figure 3-10: Time delay and throughput of the Urn protocol

4. TIME-CONSTRAINED COMMUNICATION IN A DISTRIBUTED ENVIRONMENT

In the previous sections of this paper we examined various aspects of the multiple access problem, namely, how a population of distributed stations can efficiently share a resource (the communication channel) when coordination among the users requires use of the resource itself. We identified general approaches towards achieving multiple access communication and examined different protocols employing these approaches. In this section, these general approaches will be re-examined in the context of a time-constrained multiple access environment. First, we present the salient characteristics of a time-constrained multiple access environment and the protocol design and performance issues which arise in such an environment. In sections 4.2, 4.3 and 4.4 we then discuss the applicability of PCA, demand-adaptive and contention-based protocols in this environment. The relative advantages and disadvantages of these three classes of protocols will be examined and current work in the design and analysis of time-constrained communication protocols will be surveyed.

4.1 TIME-CONSTRAINED COMMUNICATION

In a time-constrained communication environment, message packets generated at a source station must arrive at the destination station *within a given amount of time*. If a message's *delay* (defined as the time between the arrival of the message at the source station and the arrival of the message at the destination station) exceeds this time constraint then the message is considered *lost*, whether or not the message is ever received at the destination station. One of the principle challenges in the area of time-constrained communication is the design of robust and reliable protocols which deliver a maximum number of messages within their given time constraint.

There are several important applications of time-constrained communication in a multiple access environment. Perhaps the most important of these applications is packetized voice [Bially 80a] [Coviello 79], in which human voice is digitized, packetized at the source station, transmitted over the multiple access channel, and reconstructed and played out at the destination station. Since excessive delays can have seriously disruptive effects on human conversation, voice packets are usually constrained to arrive at the destination station

within a given amount of time after their generation at the sending station. Those packets which do not arrive within the time bound are considered lost, a small number of lost packets has been shown to have little, if any, effect on human speech intelligibility.

A second application requiring time-constrained communication is distributed sensor networks (DSN) [DSN 82], in which distributed stations attempt to track a moving object using their local observations and the communicated observations of the other stations. Since the position of the object is continually changing, only a small amount of time is available to fix its current location. The distributed observations necessary to determine the current location must thus be communicated within this small amount of time. Message loss due to excessive message delays results in uncertainty in the object's calculated position. A third class of time-constrained applications is real-time control applications in which stations must initiate some action at remote devices within a specified amount of time.

In previous sections, it was noted that numerous issues must be considered in the evaluation of a protocol. Many of these issues arise independently of the particular application and hence will be important concerns for time-constrained communication applications as well. For example, the *reliability* of the protocol is always a critical issue; a distributed protocol should be able to operate (perhaps in a degraded mode) in spite of individual station failures or incorrect operation. *Robustness*, or insensitivity to errors and misinformation (e.g., unsynchronized clocks, poor estimation of system parameters such as channel utilization) is also important. Finally, the ease of implementation and the hardware and software required are also important concerns.

Although many issues arise independently of the particular application of the protocol, the nature of time-constrained communication introduces new concerns which must be addressed. For example, we previously assumed that the message arrival process is bursty; for some time-constrained communication applications such as packetized voice, the arrival of messages to be transmitted is *not* bursty and thus a protocol's sensitivity to the arrival statistics must be examined. Also, the desire for integrated multi-media communications (e.g., integration of data, voice, video and facsimile) [Gitman 77] requires that a time-constrained protocol be capable of supporting other classes of traffic as well, or at least coexist with the protocols which handle the other classes of traffic. The most important new issues, however, arise from performance considerations which are fundamentally different from those of standard data communication.

The principal performance considerations for the standard multiple access protocols are capacity and the average time delay/throughput tradeoff. In time-constrained communication, the principal performance consideration is not average time delay but rather *the percentage of messages which are received at the destination with a message delay below the given constraint*. Alternatively, the percentage of messages with delays exceeding this time constraint can be characterized as the *loss* of the protocol for the given time bound and traffic arrival rate.

Thus *message loss* will be an important performance measure for any time-constrained communication protocol. Since message loss results from message delays exceeding some time constraint, the *time constraint* which results in a given loss will be an important performance parameter. The *offered load*, or arrival intensity of messages to the network, is also a parameter which influences the performance of the protocol. The three primary performance measures for time-constrained communication are thus loss, the imposed time-constraint, and the offered load. As might be expected, tradeoffs exist among these three measures. For example, for a fixed value of any one, a tradeoff exists among the other two:

- for a fixed offered load, the larger the imposed time constraint, the smaller the loss.
- for a fixed loss, the larger the offered traffic load, the larger the time constraint needed to realize this fixed loss.
- for a fixed time constraint, the larger the offered traffic load, the larger the loss.

Many of the protocols discussed in section 3 were designed with the primary performance measures of average time delay and capacity in mind. *This fact suggests that these protocols may not be well suited for the performance measures and three-way tradeoffs of time-constrained communication*. To date there has been relatively little work examining the design issues for time-constrained multiple access protocols. Let us now re-examine the three general approaches towards achieving multiple access communication in the context of a time-constrained communication environment.

4.2 PREDETERMINED CHANNEL ALLOCATION PROTOCOLS

Due to the previously noted inefficiencies of pure TDMA protocols, pure TDMA techniques have not received much consideration for use in time-constrained communication applications. However, a slight variation on pure TDMA techniques, known as *slot-switched*

TDMA, has received considerable attention. This technique has been examined primarily in a centralized environment in which numerous message sources converge at a central location and are then time-multiplexed onto a single outbound channel [Gruber 81] [Arthurs 79] [Mowafi 80] [Fischer 76]. Maglaris [Maglaris 81] has also investigated the use of slot switched TDMA in a distributed environment with centralized control.

In slot switched TDMA (as in pure TDMA), time is divided into fixed length frames and frames are again further divided into slots. However, there are now fewer slots per frame than there are stations. The purpose of reducing the number of slots per frame is to decrease idle channel time and message waiting time due to empty slots being held by stations with nothing to send. Since there are fewer slots than stations, a station must first *claim* a fixed slot number within each frame before using it. Slot allocation is typically determined by beginning each time frame with a slot request period as in demand-adaptive reservation schemes. Allocation is then performed either by a centralized station [Maglaris 81] or by the distributed stations according to some policy known to all stations.

Once a station is assigned the i th slot in a frame, it then has sole access to the communication channel during the i th slot of every subsequent frame until it explicitly releases the slot. Typically, a slot is held for numerous frame periods. Note that, as in pure circuit switching, once a station is assigned a slot, it is guaranteed a fixed channel bandwidth (i.e., the full channel bandwidth for one time slot each frame). However, the channel is also time multiplexed among all stations which have been assigned time slots. For this reason, PCA is also known as a "virtual circuit" approach and the time during which a station holds the channel is known as a "virtual connection".

An interesting characteristic of PCA protocols is that since a station transmits in a synchronous fashion (i.e., only during its time slot), data is received at the destination station in a synchronous fashion. Thus all messages have a deterministic waiting and transmission time and 100% of the messages are delivered with a fixed time delay. However, a severe price is paid for this 100% reliability and fixed delay. In pure TDMA and slot-switched TDMA, the time delay may be intolerably large. Moreover, since the maximum number of stations which can be multiplexed is fixed by the number of slots in a frame in slot-switched TDMA, once all the frame slots have been assigned, all other stations are blocked from using the channel. This blocking phenomenon thus trades the 100% reliability of messages in a virtual connection with the 0% reliability of messages in a blocked connection (i.e., messages at a station which is denied a frame slot). This problem will be

further examined in the following sections on demand-adaptive and contention-based protocols.

As we will see, PCA has both advantages and disadvantages with respect to other strategies. In addition to the fixed time delay and 100% reliability for assigned connections, the synchronous delivery of data can be an important advantage for applications such as packetized voice, which require synchronous playout of the received messages. Also, since the receiving station knows that data will be received synchronously, (after reception of the first packet), there is no need for control information between the first and last slots used by the connection; this contributes towards better utilization of the channel bandwidth.

As previously discussed, two factors contribute to possible substantial waste of channel bandwidth by PCA protocols. If relatively few slots within a frame are claimed, a station is still only permitted to transmit during its assigned slot, even if it could potentially utilize the remaining empty slots within the frame. Even if most of the slots within a frame are utilized, "silence" periods within a connection can result in wasted channel bandwidth. For example, in packetized voice applications, an average 60% [Bially 80a] of a connection is spent in silence. In the case of numerous connections emanating from a central location, it may be possible to multiplex other connections into the silent periods of a connection [Tasi 59]. However, in a distributed environment, multiplexing connections from geographically distributed stations into a time slot becomes much more difficult. One other limitation of PCA is that while it may be well suited for message traffic that is relatively synchronous over a long period of time, it may not be as well suited to bursty traffic such as interactive data. Thus, the integration of both stream-like and bursty sources at a station would probably require separate access mechanisms for the two types of traffic.

In summary, PCA techniques are most suitable for environments requiring long, synchronous, stream-like data transmissions. However, these PCA techniques may preclude guaranteed connections, suffer from large time delays, be potentially wasteful of channel bandwidth and may not be suitable for integration of bursty traffic sources with stream-like sources.

4.3 DEMAND-ADAPTIVE PROTOCOLS

Two characteristics of most demand-adaptive protocols make them more attractive than PCA protocols for time-constrained communication applications. First, they provide every

station with a *guaranteed* amount of communication capacity. Second, if the time required for token passing or reservation posting is small compared with the message transmission time, then unlike PCA protocols, channel bandwidth is not wasted by stations with nothing to transmit.

As in PCA protocols, most demand-adaptive protocols implement some form of round robin sharing of the channel. This assures that each station will receive a guaranteed bandwidth of one slot per round. However, unlike slot switched TDMA, *a//* stations are guaranteed this bandwidth and thus all connections are guaranteed. The time delay between transmissions by a station is bounded by the maximum length of a polling round or reservation/transmission cycle. Since not all stations may choose to transmit during a round, the length of a round can be less than this maximum value. Even though the time delay is bounded, however, if the time constraint imposed on message delays is less than the maximum time bound, then some fraction of the messages may still be lost due to excessive time delays.

If an increasing number of stations begin to transmit messages, then the increasing system load is translated into longer delays and thus possible message loss. In PCA an increasing traffic load results in the blocking of connections and the loss of all messages with these connections. Thus, demand-adaptive protocols might be considered "fairer" than predetermined channel allocation protocols, since they provide all stations with the same delay and loss characteristics rather than providing one set of stations with a high grade of service (i.e., a guaranteed connection with fixed delays) and another set of stations with a lower grade of service (i.e., no connection).

The variable message delay has an important consequence for time-constrained applications such as packetized voice, which require synchronous message playout at the receiving station. In order to "smooth out" the variability of the delays, it is necessary to buffer received messages and introduce an additional initial delay before playout begins. The effect of adding the initial additional delay is shown below in figure 4-1 [Cohen 77].

Figure 4-1 shows a station which begins synchronous generation of messages at time t_0 . Let

- $G(t)$ be the number of packets generated at the source by time t . $G(t)$ increases by unit steps.
- $A(t)$ be the number of packets received at the destination by time t .

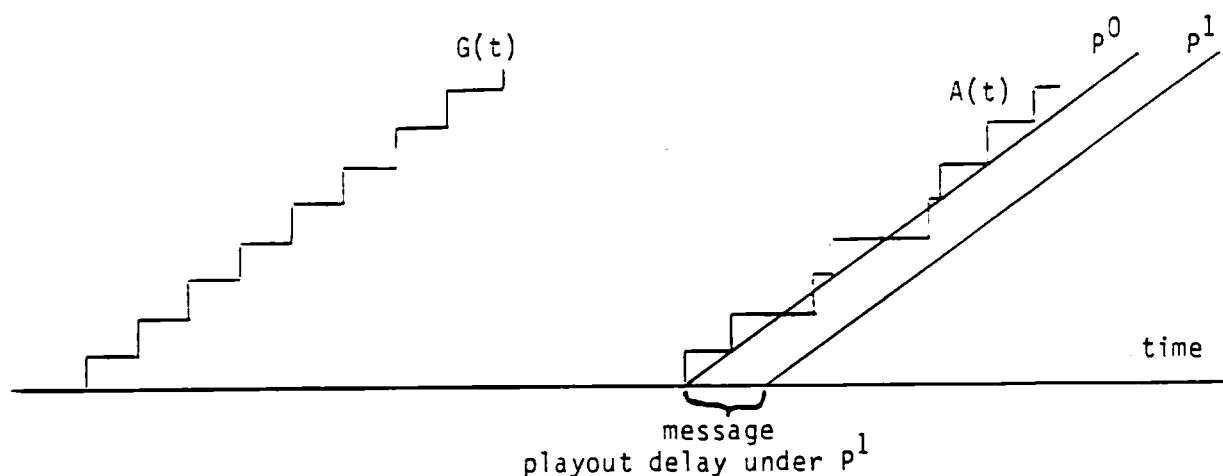


Figure 4-1: The effects of an additional initial waiting time

- $P(t)$ be defined as follows. If $P(t) > i$, then playout of message i should begin at the destination station by time t . Since playout is synchronous, $P(t)$ is a straight line. Let $P^0(t)$ be the playout strategy with no initial additional delay. Let $P^1(t)$ be the playout strategy with an initial additional delay introduced.

Figure 4-1 shows that if synchronous message playout begins immediately upon receipt of the first message (shown by the line P^0), then messages 3 and 5 will be lost. However, under playout strategy P^1 , in which an additional initial delay has been added, neither message 3 nor message 5 will be lost. Thus, the effect of adding the initial delay is to "smooth out" the differences in message delays.

In figure 4-1 messages are lost because they have not yet arrived at the receiving station when their playout is scheduled to begin. Such lost messages introduce a "glitch" into the synchronous playout. The effects of such delays and glitches on human voice communication has been extensively studied; [Forge 75] contains an excellent summary of this work. Buffering and delay strategies have also been examined in [Cohen 77] and [Gopal 81].

A critical factor in determining the applicability of demand-adaptive protocols in a time-

constrained environment is the number of stations accessing the network. If there are a large number of stations and a large fraction of those stations choose to use their transmission slots in a polling round, then the length of a polling round can grow quite large. This may result in a bounded but excessive time delay for time-constrained communication. Furthermore, if there are a large number of stations using the channel infrequently, then a large portion of the channel bandwidth will be wasted on the polling or reservation overhead. This results from the fact that each station must use the channel (if only to broadcast silence) to inform the other stations that it will not use a time slot during the current polling round.

In summary then, the properties of guaranteed connections, bounded time delay, and the ability to take full advantage of the three-way trade off between time constraint, loss and offered traffic load make demand adaptive protocols attractive for at least some time-constrained applications. The applicability of demand adaptive protocols, however, will be strongly dependent on the number of stations in the environment.

4.4 CONTENTION-BASED PROTOCOLS

The third class of protocols which have been proposed for time-constrained communication use a collision resolution process to determine channel transmission rights.

In contention-based protocols, stations access the channel on a message by message basis and thus there is no concept of a "connection". Since there are no connections to be blocked, an increasing traffic load on the channel results in increasingly longer delays and consequently higher message loss. Flow control techniques for limiting the congestion during these periods of excessive traffic have been investigated in the literature; [Bially 80b] and [Forgie 77] discuss general flow control techniques for packetized voice systems.

Contention-based protocols (like demand-adaptive protocols) introduce a variable message delay. Thus, a smoothing buffer and additional delay may also be required for applications requiring synchronous playout at the destination station. A subtle, yet important difference exists between the delays in contention-based and demand adaptive protocols. In the latter case, the time between successful transmissions by a station is always bounded above by one frame length. This is not true, however of contention-based protocols and thus there may be radically different time delay distributions at different stations. Maxemchuk

[Maxemchuk 82] has noted that while it may be valid to assume that packets are lost at random when aperiodic sources generate traffic, this assumption does not necessarily hold if a majority of the sources are periodic. In this case, the random loss assumption must be carefully examined given the particular random access mechanism employed.

In summary, the advantages of contention-based protocols for time-constrained communication are similar to those for demand adaptive protocols: guaranteed connectivity, the transfer of temporary overloading into packet delay rather than blocked connections and the ability to trade message loss for message delay. In addition, message delays depend only on the traffic load accessing the network and are independent of the *number* of stations accessing the channel. The major disadvantage of contention-based protocols for time-constrained applications is the variable message delay. For stations requiring a synchronous playout of received messages, this necessitates additional buffering capacity and the introduction of an additional time delay before synchronous playout begins.

Recently, some specific contention-based protocols have been proposed and studied in the context of time-constrained communication. Nutt [Nutt 82] has shown that the Ethernet protocol (CSMA/CD with backoff) can support up to 50 simultaneous voice calls, delivering over 99% of the messages with less than a 5 ms delay. Melvin [Melvin 81] has found that Ethernet can support up to 130 simultaneous voice sources, delivering 95% of the transmitted messages with less than 200 ms delay. The loss and delay values observed by both Nutt and Melvin are within the limits for good human speech quality, thus indicating the feasibility of voice communication in a large scale distributed multiple access environment. The performance of voice communication in a contention-based multiple access environment has also been studied by Maxemchuk [Maxemchuk 82]. He observed that the deterministic, synchronous nature of voice communication can be exploited to reduce channel contention and proposed a time-constrained protocol offering performance advantages over the Ethernet protocol.

Finally, in [Kurose 83] it was noted that, in addition to its traditional role as an arbiter of channel sharing, a multiple access protocol also serves as a *distributed scheduling mechanism* by imposing an implicit or explicit transmission order on the messages distributed among the stations in the network. This scheduling function was shown to critically affect the *distribution* of message delays and thus the time-constrained performance of the protocol as well. A random access protocol, based on a generalization of time window protocols [Gallager 78] [Towsley 82] was proposed which provided a family of scheduling disciplines based on message arrival times. The performance of this

protocol was examined for the cases in which the protocol transmitted all the messages throughout the network in FCFS, LCFS and RANDOM order.

5. CONCLUSION

In this paper we have examined various aspects of the multiple access problem. We first defined the problem and identified underlying issues and problems in achieving communication among distributed stations which must share a single communication channel. Various approaches towards resolving the multiple access problem were characterized, specific protocols embodying these approaches were presented, and performance aspects of these protocols were then examined. Issues in time-constrained multiple access communication were then discussed and multiple access protocols were then re-examined in the context of a time-constrained environment.

Current emphasis on developing multiple access multi-media networks requires the development of protocols capable of supporting not only time-constrained communication but also such diverse applications as standard data transmission, facsimile traffic and low speed video traffic in a unified manner. Past research provides much insight into the problems and inherent difficulties of multiple access communication. While this past work also suggests possible approaches towards achieving integrated multi-media communication, it is apparent that new work is required to push the technology forward towards the realization of the integrated multiple access multi-media communication networks of the future.

ACKNOWLEDGMENTS

The authors are grateful to Bruce Hillyer, Jonathan Gross and other colleagues for many helpful comments and suggestions on earlier versions of this paper.

REFERENCES

- [Abramson 70] N. Abramson.
The ALOHA System - Another Alternative for Computer Communications
In *AFIPS Conference Proceedings, FJCC*, pages 281-285. AFIPS, 1970.
- [Abramson 73] N. Abramson.
Packet Switching with Satellites
In *AFIPS Conference Proceedings, NCC*, pages 695-703. AFIPS, 1973.
- [Arthurs 79] E. Arthurs and B.W. Stuck.
A Theoretical Traffic Performance Analysis of an Integrated Voice-Data Virtual Circuit Packet Switch.
IEEE Transactions on Communications COM-27(7):1104-1111, July, 1979.
- [Bially 80a] T. Bially, A.J. McLaughlin and C.J. Weinstein.
Voice Communications in Integrated Digital Voice and Data Networks.
IEEE Transactions on Communications COM-28(9):1478-1490, September, 1980.
- [Bially 80b] T. Bially, B. Gold and S. Seneff.
A Technique for Adaptive Voice Control in Integrated Packet Networks.
IEEE Transactions on Communications COM-28(3):325-333, March, 1980.
- [Bux 81] W. Bux et al.
A Reliable Token Ring System for Local Area Communication.
In *NTC Conference Proceedings*, pages A2.2.1 to A2.2.6. 1981.
- [Capetanakis 79] J. I. Capetanakis.
Generalized TDMA: The Multi-Accessing Tree Protocol.
IEEE Transactions on Communications COM-27:1476-1484, October, 1979.
- [Chlamtac 79] I. Chlamtac, W.R. Franta and K.D. Levin.
BRAM: The Broadcast Recognizing Access Method.
IEEE Transactions on Communications COM-27:1183-1190, August, 1979.
- [Clark 78] D. Clark, K. Progran and D. Reed.
An Introduction to Local Area Networks.
Proc. IEEE 66:1497-1516, November, 1978.
- [Cohen 77] D. Cohen.
Issues in Transnet Packetized Voice Communication.
In *Conference Report. Proceedings of the Fifth Data Communications Symposium*, Snowbird, Utah, 1977.
- [Coviello 79] G.J. Coviello.
Comparative Discussion of Circuit vs Packet Switched Voice.
IEEE Transactions on Communications COM-27(8):1153-1160, August, 1979.

- (DSN 82) MIT Lincoln Laboratories.
Workshop on Distributed Sensor Networks, MIT Lincoln Laboratories,
Lexington, Mass., 1982.
- [Eslanadidi 82] M. Eslanadidi and W. Chu.
An Analysis of a Time Window Multiaccess Protocol with Collision Size
Feedback.
In *Proceedings*, pages 112-117. ACM Computer Networks Symp.,
April, 1982.
- [Fayolle 77] G. Fayolle, E. Gelenbe and J. Labetoulle.
Stability and Optimal Control of the Packet Switching Broadcast Channel.
Journal of the Association for Computing Machinery 24(3):375-386.
July, 1977.
- [Fischer 76] M.J. Fischer and T.C. Harris.
A Model for Evaluating the Performance of an Integrated Circuit and
Packet Switched Multiplex Structure.
IEEE Transactions on Communications COM-24(2):95-202. February,
1976.
- [Forgie 75] J.W. Forgie.
Speech Transmission in Packet Switched Store and Forward Networks.
In *Conference Record*, pages 137-142. Proceedings of the National
Computer Conference, 1975.
- [Forgie 77] J.W. Forgie and A.G. Nemeth.
An Efficient Packetized Voice/Data Network Using Statistical Flow
Control.
In *Conference Record*, pages 38.2-44 to 38.2-48. ICC, 1977.
- [Fratta 81] L. Fratta, F. Borgonovo and F. Tobagi.
The EXPRESS-Net: A Local Area Communication Network Integrating
Voice and Data.
In G. Pujolle (editor), *Performance of Data Communication Systems*,
pages 77-88. North-Holland, 1981.
- [Gallager 78] R. G. Gallager.
Conflict Resolution in Random Access Broadcast Networks.
In *Proc. of the AFOSR Workshop in Communication Theory and
Applications*. AFOSR, 1978.
- [Georgiadis 81] L. Georgiadis and P. Papantoni-Kazakos.
A Collision Resolution Process Using Energy Detectors.
In *1977 National Telecommunications Conference Proceedings*, pages
E3.6.1 - E3.6.5. IEEE, 1981.
- [Gitman 77] I. Gitman, H. Frank, B. Occhiogrosso and W. Hsieh.
Issues in Integrated Network Design.
In *Conference Record*, pages 38.1-36 to 38.1-43. ICC, 1977.
- [Gopal 81] P.M. Gopal, J.W. Wong and J.C. Majithia.
An Evaluation of Payout Strategies for Voice Transmission in Packet
Networks.
In *Proceedings*, pages 33-38. Computer Networking Symposium,
December, 1981.

- [Grami 82] A. Grami, K. Sohraby, and J. Hayes.
Further Results on Probing.
In *Proceeding of ICC 1982*, pages 1C.3.1 - 1C.3.3 IEEE, 1982.
- [Gruber 81] J.G. Gruber.
Delay Related Issues in Integrated Voice and Data Networks.
IEEE Transactions on Communications COM-29, June, 1981.
- [Hayes 78] J. F. Hayes
An Adaptive Technique for Local Distribution.
IEEE Transactions on Communications COM-26:1178-1186, August, 1978.
- [Kahn 78] R.Kahn et al.
Advances in Packet Radio Technology.
Proc. IEEE 66:1468-1496, November, 1978.
- [Kleinrock 75a] L. Kleinrock and F. A. Tobagi.
Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics.
IEEE Transactions on Communications COM-23:1400-1416, December, 1975.
- [Kleinrock 75b] L. Kleinrock.
Queueing Systems Volume I: Theory.
J. Wiley and Sons, N.Y., N.Y., 1975.
- [Kleinrock 77] L. Kleinrock.
Performance of Distributed Multi-access Computer Communication Systems.
In *1977 IFIP Conference Proceedings*, pages 547-552. IFIP, 1977.
- [Kleinrock 78] L. Kleinrock and Y. Yemini.
An Optimal Adaptive Scheme for Multiple Access Broadcast Communication.
In *Conference Record*, pages 7.2.1-7.2.5. International Conference on Communications, 1978.
- [Kleinrock 80] L. Kleinrock and M.O. Scholl.
Packet Switching in Radio Channels: New Conflict-Free Multiple Access Schemes.
IEEE Transactions on Communications COM-28(7):1015-1029, July, 1980.
- [Kurose 83] J.F. Kurose and M. Schwartz.
A Family of Window Protocols for Time-Constrained Applications in CSMA Networks (to appear).
In *Proceedings*. IEEE INFOCOMM '83, 1983.
- [Limb 82] J. Limb and C. Flores.
Description of FASNET - A Unidirectional Local-area Communications Network.
Bell System Technical Journal 61(7):1413-1440, Sept. 1982.
- [Maglaris 81] B. Maglaris and T. Lissack.
A Priority TDMA Protocol for Satellite Data Communications.
In *Conference Record*, pages 73.3.1 to 73.3.5. ICC, 1981.

- [Maxemchuk 82] N. Maxemchuk.
A Variation on CSMA/CD That Yields Movable TDM Slots in Integrated Voice/Data Local Networks.
Bell System Technical Journal 61(7):1527-1550, September, 1982.
- [Melvin 81] D.K. Melvin.
Voice on Ethernet - Now!
In *Conference Record*. National Telecommunications Conference, 1981.
- [Metcalfe 76] R.M. Metcalfe and D.R. Boggs.
Ethernet: Distributed Packet Switching for Local Computer Networks.
Communications of the ACM 19, July, 1976.
- [Mittal 81] K.K. Mittal and A. N. Venetsanopoulos.
On the Dynamic Control of the Urn Scheme for Multiple Access Broadcast Communication Systems.
IEEE Transactions on Communications COM-29:962-970, July, 1981.
- [Molle 81] M. Molle.
Extensions and Unifications of the Multiple Access Communication Problem.
Technical Report CSD-810730, UCLA Computer Science Dept., July, 1981.
- [Mowafi 80] O.A. Mowafi and W.J. Kelly.
Integrated Voice/Data Packet Switching Techniques for Future Military Networks.
IEEE Transactions on Communications COM-28(9):1655-1662, September, 1980.
- [Nutt 82] G.J. Nutt and D.L. Bayer.
Performance of CSMA/CD Networks Under Combined Voice and Data Loads.
IEEE Transactions on Communications COM-30(1):6-11, January, 1982.
- [Saltzer 81] J. Saltzer and D. Clark.
Why a Ring?
In *Proceedings 7th Data Communication Sym.*, pages 211-223. IEEE Computer Society, October, 1981.
- [Schwartz 77] M. Schwartz.
Computer Communication Network Design and Analysis.
Prentice Hall, 1977.
- [Skrzypczak 81] C. Skrzypczak, J. Weber and W. Falconer.
Bell System Planning of ISDN.
In *Conference Record*, pages 19.6.1 - 19.6.6. International Conference on Communications, 1981.
- [Tannenbaum 81] A. S. Tannenbaum.
Computer Networks.
Prentice Hall, Englewood Cliffs, N.J., 1981.
- [Tasi 59] K. Bullington and J. Fraser.
Engineering Aspects of TASI.
Bell System Technical Journal 38, March, 1959.

- [Tobagi 77] F. Tobagi and L. Kleinrock
Packet Switching in Radio Channels: Part IV - Stability Considerations
and Dynamic Control in Carrier Sense Multiple Access
IEEE Transactions on Communications COM-25(10) 1103-1119.
October, 1977.
- [Towsley 82] D. Towsley and G. Venkatesh.
Window Random Access Protocols for Local Computer Networks
IEEE Transactions on Computers COM-31(8), August, 1982.
- [Tsybakov 79] B.S. Tsybakov et al.
Methods of Random Multiple Access.
In *Conference Record* Fifth International Symposium on Information
Theory, July, 1979.