COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY · STANFORD, CA 94305-2192

REAL-TIME COMMUNICATION SYSTEMS: DESIGN, ANALYSIS AND IMPLEMENTATION

Final Technical Report

DARPA Contract: MDA-903-79-C-0201

DARPA Order No. A03717

Contract Period: January 1, 1979 – June 30, 1984

Fouad A. Tobagi

Principal Investigator

July 31, 1984

Prepared For

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Stanford Electronics Laboratories
Stanford University
Stanford, CA 94305
(415) 497-1708

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) REAL-TIME COMMUNICATION SYSTEMS: DESIGN, ANALYSIS AND IMPLEMENTATION		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1/1/79 - 6/30/84 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Fouad A. Tobagi		8. CONTRACT OR GRANT NUMBER(s) MDA 903-79-C-0201 DARPA Order No. A03717
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford Electronics Laboratories Stanford University Stanford, CA 94305-2192		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Information Processing Techniques Office 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE July 31, 1984 13. NUMBER OF PAGES 228
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Resident Representative Office of Naval Research Durand 165 Stanford University, Stanford, CA 94305-2192		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This final technical report covers the period from January 1, 1979 to June 30, 1984 for DARPA Contract MDA 903-79-C-0201. The main purpose of this research was to study Real-Time Communication Systems; i.e., communications systems destined to support applications with real-time constraints. The research effort included design and performance evaluation of architectures and protocols for real-time communication systems, and in some instances such as implementation and experimental evaluation. The research tasks undertaken during the contract period and the corresponding accomplishments are:

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Task A. Real-Time Protocol Performance Analysis

Accomplishments under this task are:

a) A tutorial on multiaccess protocols in packet communication systems.

b) Analysis of Carrier Sense Multiple Access with Collision Detection (CSMA-CD).

c) Design and Analysis of Message-Based Priority Functions in Multiaccess Communications Systems in general, and CSMA in particular.

d) Derivation of the distribution of Packet Delay and Packet Interdeparture Times in Slotted ALOHA and CSMA Schemes.

e) Investigation of the Performance of CSMA Local Networks When Supporting Voice Applications.

Task B. Design and Analysis of Local Networks Suitable for Real-Time Applications

Accomplishments under this task are:

a) Design of a Round Robin Scheme for Unidirectional Broadcast System Architecture.

b) Conceptual Design and Analysis of EXPRESSNET, a High-Performance Integrated-Services Local Area Network.

c) Analysis of Round Robin Schemes in Unidirectional Broadcast Local Area Networks.

d) A tutorial on Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks

Task C. Performance Evaluation of Multihop Packet Radio Networks

Accomplishments in this task are:

a) Analysis of Two-Hop Centralized Packet Radio Networks, Under Slotted ALOHA and CSMA Access Schemes

b) Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes

c) Theoretical Results in the Throughput Analysis of Multihop Packet Radio Networks

d) Simulation of Multihop Packet Radio Networks

Task D. Multinetwork Environments

a) Performance of Gateway-to-Gateway and End-to-End Flow Control Procedures in Internet Environments

Task E. Architectural Design and VLSI Implementation of Local Area Networks

Initiation of an effort aimed at combining VLSI technology and the Expressnet concept to support a multitude of local communications applications requiring high speed networking.

The research performed during this period has been reported upon in our Semi-Annual Technical Reports, in our Stanford Electronics Laboratories Technical Reports, and in the published professional literature. This report includes appendices which contain reprints of the related articles which have been published in the professional literature.

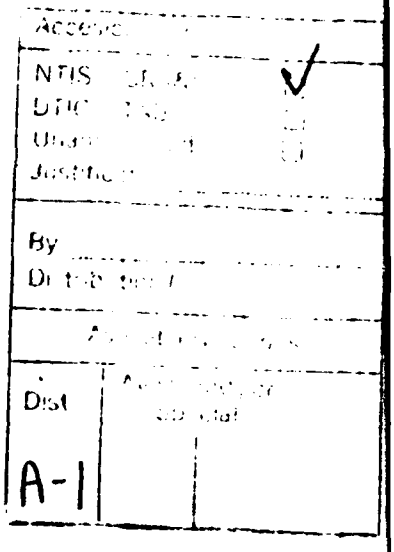
REAL-TIME COMMUNICATION SYSTEMS: DESIGN, ANALYSIS AND IMPLEMENTATION

Final Technical Report

DARPA Contract: MDA-903-79-C-0201

DARPA Order No. A03717

Contract Period: January 1, 1979 – June 30, 1984



Fouad A. Tobagi

Principal Investigator

July 31, 1984

Prepared For

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Stanford Electronics Laboratories
Stanford University
Stanford, CA 94305
(415) 497-1708

REAL-TIME COMMUNICATION SYSTEMS: DESIGN, ANALYSIS AND IMPLEMENTATION

Final Technical Report
DARPA Contract: MDA-903-79-C-0201
DARPA Order No. A03717
Contract Period: January 1, 1979 - June 30, 1984

Prepared For

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

July 31, 1984

Abstract

This final technical report covers the period from January 1, 1979 to June 30, 1984 for DARPA Contract MDA 903-79-C-0201. The main purpose of this research was to study Real-Time Communication Systems; i.e., communications systems destined to support applications with real-time constraints. The research effort included design and performance evaluation of architectures and protocols for real-time communication systems, and in some instances such as implementation and experimental evaluation. The research tasks undertaken during the contract period and the corresponding accomplishments are:

Task A. Real-Time Protocol Performance Analysis

Accomplishments under this task are:

a) A tutorial on multiaccess protocols in packet communication systems.

b) Analysis of Carrier Sense Multiple Access with Collision Detection (CSMA-CD).

c) Design and Analysis of Message-Based Priority Functions in Multiaccess Communications Systems in general, and CSMA in particular.

d) Derivation of the distribution of Packet Delay and Packet Interdeparture Times in Slotted ALOHA and CSMA Schemes.

e) Investigation of the Performance of CSMA Local Networks When Supporting Voice Applications.

Task B. Design and Analysis of Local Networks Suitable for Real-Time Applications

Accomplishments under this task are:

a) Design of a Round-Robin Scheme for Unidirectional Broadcast System Architecture.

b) Conceptual Design and Analysis of EXPRESSNET, a High-Performance Integrated-Services Local Area Network.

c) Analysis of Round-Robin Schemes in Unidirectional Broadcast Local Area Networks.

d) A tutorial on Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks

Task C. Performance Evaluation of Multihop Packet Radio Networks

Accomplishments in this task are:

a) Analysis of Two–Hop Centralized Packet Radio Networks, Under Slotted ALOHA and CSMA Access Schemes

b) Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes

c) Theoretical Results in the Throughput Analysis of Multihop Packet Radio Networks

d) Simulation of Multihop Packet Radio Networks

Task D. Multinetwork Environments

a) Performance of Gateway-to-Gateway and End-to-End Flow Control Procedures in Internet Environments

Task E. Architectural Design and VLSI Implementation of Local Area Networks

Initiation of an effort aimed at combining VLSI technology and the Expressnet concept to support a multitude of local communications applications requiring high speed networking.

The research performed during this period has been reported upon in our Semi-Annual Technical Reports, in our Stanford Electronics Laboratories Technical Reports, and in the published professional literature. This report includes appendices which contain reprints of the related articles which have been published in the professional literature.

I. Introduction

This final report covers the period from January 1, 1979 to June 30, 1984 for DARPA Contract number MDA 903–79–C–0201. The research performed during this period has been reported upon in our Semi-Annual Technical Reports, in our Stanford Electronics Laboratories Technical Reports, and in the published professional literature. Accordingly, this final report consists of a brief section describing the scope of research and its underlying tasks, a section summarizing the accomplishments attained during the contract period, a section listing all publications which appeared under this contract, and appendices which contain reprints of the related articles which have been published in the professional literature.

II. Scope of Research and Underlying Tasks

The main purpose of this research was to study Real–Time Communication Systems; i.e., communications systems destined to support applications with real–time constraints. Examples of such applications are: digitized speech, video sensor and tracking systems, seismic data, weather report, fire control, etc.

The systems considered in our studies are mainly of the packet-switched type. There are several reasons for such a choice. First, a number of successful experiments have shown that packet-switching technology is feasible under real-time constraints; in particular, reference is made here to the real-time speech experiments on the ARPANET, SATNET and PRNET. Second, packet switching has the ability to provide mixed communication services for real-time data and computer-to-computer traffic and to dynamically adapt itself to the changing requirements of each mode, thus achieving efficient use of spectral resources. Third, encryption is readily feasible in packet-switched digital systems, and thus renders these systems advantageous, particularly when we are concerned with (military) speech.

Although the ultimate objective of all types of communication networks is usually the efficient and reliable transport of data, there are advantages in using one type over the other depending on the application and the environment. For example, some advantages in using broadcast ground radio communications are: collection and dissemination of data over distributed geographical areas independent of the availability of preexisting (telephone) wire networks; the suitability of wireless connections for communication with and among mobile users, etc. Satellite-radio communications, on the other hand, are best suited to long-haul communication among distant sites. Networks of the ETHERNET-type are suitable for local, in-building communications.

The support of most real-time applications will involve the interconnection of several packet-switched networks, often of different types. Indeed, these various types have been conceived to complement each other and support given applications in geographical environments of different characteristics; it is very typical that DoD communication needs are among geographically dispersed tactical and strategic forces and thus naturally span over geographical settings of different characteristics. An example here is digitised speech in

naval tactical and strategic operations where a network could provide (local) ship-to-ship and ship-to-shore communication, and finally an ARPANET-like network could provide communication between satellite ground stations and various DoD headquarters.

The basic distinction between real-time data and regular computer-to-computer traffic are the following requirements and properties.

a. With real-time applications, a small network delay is required: indeed, with sensors old data is obsolete; with digitized speech, the feasibility of real-time interactive communication is only possible if small response time can be achieved; small response time also allows speech to be "played" out without breaking up; and finally, it allows buffering in view of smoothing the flow and playing back continuously without introducing too large a total delay.

b. With real-time applications, timing information is an integral part of each message.

c. With real-time applications, the information transmitted often is redundant (as, for example, in target tracking systems with multiple sensors, weather data collection systems and seismic data collection systems, etc.)

d. With real-time applications, a low level of information loss is often tolerable.

e. The input traffic pattern in real-time applications is different from computer-to-computer traffic and interactive traffic, and may not always be approximated by Poisson processes.

Given these observations, it is all too evident that the characteristics and requirements of real-time traffic are very different from those of conventional computer-to-computer traffic. The purpose of our research has been first to evaluate the performance and asses the behavior of existing transmission protocols with real-time constraints in packet-switched internetwork environments. Such studies are important in order (i) to determine the conditions under which the real-time constraints are satisfied and (ii) to provide initial guidance in the design and implementation of real-time transmission protocols. Another major purpose has been to conceive new architectures and protocols particularly suitable for real-time communication systems, and in some instances such as Expressnet, to undertake implementation and experimental evaluation.

The research tasks undertaken during the contract period are:

Task A. Real–Time Protocol Performance Analysis
Analytical evaluation of network protocols under real-time constraints; derivation of packet delay distributions in various network configurations and for various transmission protocols; the handling of traffic with special characteristics such as those encountered with voice applications; and the investigation of priority functions in multiaccess protocols.

Task B. Design and Analysis of Local Networks Suitable for Real–Time Applications
Studies into efficient conflict–free round–robin schemes, for local networks and their suitability to real–time applications.

Task C. Performance Evaluation of Multihop Packet Radio Networks
Analytical modeling and simulation of multihop packet radio networks operating under various channel access schemes.

Task D. Multinetwork Environments
Multinetwork description methodology; simple characterization of single networks in view of integrating such characterization in multinetwork models; study of internetwork protocol performance with respect to real–time constraints.

Task E. Architectural Design and VLSI Implementation of Local Area Networks
Design of an interface of Express–Net operating under the Express Access Protocol, and supporting a multitude of applications with different requirements; VLSI implementation of such interfaces and functions.

III. Summary of Accomplishments

Considerable progress has been made under each task. In this section we summarize these accomplishments grouped by task.

Task A. Real–Time Protocol Performance Analysis

Accomplishments under this task are:

a) A tutorial on multiaccess protocols in packet communication systems.

b) Analysis of Carrier Sense Multiple Access with Collision Detection (CSMA–CD).

c) Design and analysis of message–based priority functions in multiaccess communications systems in general, and CSMA in particular.

d) Derivation of the distribution of packet delay and packet interdeparture times in slotted ALOHA and CSMA schemes.

e) Investigation of the performance of CSMA local networks when supporting voice applications.

a) A Tutorial on multiaccess protocols in packet communications systems:

The need for multiaccess protocols arises whenever a resource is shared by many independent contending users. Two major factors contribute to such a situation: the need to share expensive resources in order to achieve their *efficient utilization*, or the need to provide a *high degree of connectivity* for communication among independent subscribers (or both). In data transmission systems, the communication bandwidth is often the prime resource, and it is with respect to this resource that we view multiaccess protocols. In this tutorial, we gave a unified presentation of the various multiaccess techniques which we group into five categories: 1) fixed assignment techniques, 2) random access techniques, 3) centrally controlled demand assignment techniques, 4) demand assignment techniques with distributed control, and 5) mixed strategies. We discussed their applicability to different environments, namely, satellite channels, local area communication networks and

multihop store–and–forward broadcast networks, and their applicability to different types of data traffic, namely stream traffic and bursty traffic. We also presented the performance of many of the multiaccess protocols in terms of bandwidth utilization and message delay. This paper appeared as an invited paper in the *IEEE Transactions on Communications, Special Issue on Computer Network Architecture and Protocols,* April 1980 (see Appendix I).

b) Performance Analysis of CSMA–CD

Packet broadcasting in computer communication is attractive in that it combines the advantages of both packet–switching and broadcast communication. All stations share a common channel which is multi–accessed in some random fashion. Among the various random access schemes known, carrier sense multiple access (CSMA) has been shown to be highly efficient for environments with relatively short propagation delay. Packet broadcasting (and in particular CSMA) has been successfully applied to coaxial cables thus providing an efficient means for communication in local environments. In addition in such environments the possibility of detecting collisions on the coaxial cable enhances the performance of CSMA by aborting conflicting transmissions, thus giving rise to the carrier sense multiple access schemes with collision detection (CSMA–CD).

We extended the models used in the analysis of CSMA to cover the cases of collision detection and variable size packets. It was shown that the throughput–delay characteristics of CSMA–CD are better than the already highly efficient CSMA scheme. We characterized the improvement in terms of the achievable channel capacity and of the packet delay at a given channel utilization as a function of the collision detection time. Furthermore, we established the fact that in uncontrolled channels, (i.e., with a fixed average retransmission delay), CSMA–CD is more stable than CSMA, in that with CSMA–CD both channel capacity and packet delay are less sensitive to variations in the average retransmission delay.

We then studied the performance of the scheme in presence of variable size packets. Numerical results have been obtained for the interesting case of dual packet size. It was shown that a small fraction of long packets is sufficient to recover a channel capacity close to the (higher) capacity achieved with only long packets. However, the improvement experienced by the introduction of long packets is in favor of the latter and to the detriment of the short packets, establishing the necessity to design and implement priority schemes.

This analysis constituted a prelude to the analysis of priority schemes in systems employing CSMA–CD; these priority schemes are of particular importance when traffic includes data with real–time constraints. A paper entitled, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection", has been presented at the Local Area Communication Symposium, Boston, May 7–9, 1979, and appeared in Computer Networks, Oct./Nov. 1980. (See Appendix I.)

c) Message–Based Priority Functions in Multiaccess/Broadcast Systems

The proliferation of computer networks has brought about a wealth of applications that impose disparate requirements upon the communication channels they use. In particular

the traffic requirements differ to such a degree that optimization of access schemes for one pattern is often detrimental to all the rest. Message priority offers a solution to the problem. It provides a means of administering channel usage to meet these requirements while maintaining high total utilization. We proposed priority schemes appropriate for introduction into different architectures of local multiaccess communication systems to achieve these desired results. The architectures examined are the bidirectional broadcast system (BBS) architecture exemplified by Ethernet, the unidirectional ring architecture, and the unidirectional broadcast system (UBS) architecture described under Task B below. A paper entitled "Message–Based Priority Functions in Local Multiaccess Communications Systems", appeared in Computer Networks, Vol. 5, No. 4, July 1981. (See Appendix I.)

We designed a simple distributed algorithm which can support message based priority functions using carrier sensing. The scheme is based on the principle that access right to the channel is exclusively granted to ready messages of the current highest priority level. It is suitable for fully–connected broadcast networks with or without the collision detection feature, and can be made preemptive or nonpreemptive. This scheme is referred to as Prioritized–CSMA or P–CSMA.

The difficulty in analyzing multiaccess schemes such as CSMA arises from the fact that the system's service time is at all times dependent on the system's state and its evolution in time. The same difficulty arises in P–CSMA and prevents us from using conventional priority queueing results to derive its performance. We show, however, that by adopting the "linear feedback model" previously used to analyze CSMA, it is possible to derive the performance of p–persistent P–CSMA. The analysis relies on properties of semi–Markov processes, regenerative processes, and delay–cycle analysis. We completed the work in this area. Analysis and simulation results have been compiled in Stanford Electronics Laboratories Technical Report #200 entitled "Carrier Sense Multiple Access with Message-Based Priority Functions," dated December 1, 1980. (The simulation work has been performed separately under a contract with the U.S. Army Center for Communications.) A conference paper has been presented at the IEEE National Telecommunications Conference entitled, "Performance Analysis of Carrier Sense Multiple Access with Message–Based Priority Functions", Houston, December 2–4, 1980. A journal paper entitled, "Carrier Sense Multiple Access with Message–Based Priority Functions", appeared in the IEEE Transactions on Communications, Vol. COM–30, No. 1, January 1982. (See Appendix I.)

d) Distribution of Packet Delay and Interdeparture Time in Packet Radio Systems

The analysis of real–time protocols differs from the more conventional network analysis in that with real–time constraints the analysis has to be extended to include the determination of delay distributions.

Existing analysis of slotted ALOHA and CSMA has led to the determination of the average packet delay. This was achieved by formulating a Markovian model for these channels with finite populations of users, each with a single packet buffer. We derived, using the same Markovian model, the distributions of packet delay and interdeparture times, and gave simple expressions for their moments. This has been the subject of Stanford

Electronics Laboratories Technical Report #186 entitled, "Distributions of Packet Delay and Interdeparture Time in Slotted ALOHA Channels", dated April 1, 1980, and Technical Report #187 entitled, "Distribution of Packet Delay and Interdeparture Time in Carrier Sense Multiple Access", dated April 1, 1980. A paper entitled "Distribution of Packet Delay and Interdeparture Time in Slotted ALOHA and CSMA Channels," appeared in JACM, Vol. 29, No. 3, October 1982. (See Appendix I.)

e) Investigation of the Performance of CSMA Local Networks When Supporting Voice Applications

In this effort we considered local networks of the broadcast bus type, exemplified by Ethernet, and investigated the performance of such systems when supporting voice communication. In particular we studied the effect on performance of various system parameters, such as channel bandwidth, vocoder rate, delay requirement, allowable packet loss rate, etc. For comparison purposes, we also considered an *ideal* conflict-free TDMA case which is undoubtedly the most suitable for voice traffic exhibiting a deterministic generation process, and thus provides the ultimate performance one can achieve.

In the above mentioned modeling effort undertaken to evaluate the average stationary performance of CSMA and P-CSMA, it was assumed that for each user the packet inter-generation time is a random variable with a memoryless distribution. When dealing with voice applications, such an assumption is not adequate as the packet generation process is to a first approximation deterministic. Moreover, due to the real-time constraints encountered in voice communication, average performance is not sufficient, and one has to derive the distribution of delay or delay percentiles. This renders stochastic analysis rather difficult, and therefore we resort to simulation techniques for this study. The version of the simulator used in this investigation is that corresponding to P-CSMA. This was done with the intent that if voice and data were to be integrated on the same network, then, due to the strict end-to-end delay requirement in voice applications, one suspects that the prioritized scheme would be more appropriate. Indeed, by giving priority to voice packets over data packets, the scheme will help guarantee to a certain extent the delay constraint for voice packets even in the presence of data traffic. In fact, analysis and simulation of P-CSMA with two classes of traffic have already provided indication to that effect. Note, however, that in the present study we considered that there exists only one class of traffic, namely voice, and that it is given the highest priority. The only difference between P-CSMA and CSMA in this case is that with the former there is an additional overhead incurred in the implementation of the priority function which degrades the performance slightly as compared to CSMA.

When supporting voice communication, we define performance as the maximum number of voice sources accommodated for a given maximum delay requirement and a tolerable packet loss rate. We studied the effect on this performance of various system parameters such as channel bandwidth, vocoder rate, delay requirement and packet loss rate. We compared these results to an ideal TDMA system which provides the ultimate best achievable performance. The results show that for a given delay constraint D_n and a given tolerable loss rate L, there is an optimum packet size B_v which provides the maximum number of

voice sources. As long as the delay requirement D_s is not too severe ($\simeq$ 200 msec.) and the channel bandwidth W is not too large (1 MBPS), then the performance of P–CSMA is comparable to that of ideal TDMA. However, if either D_s is small ($\leq$ 20 msec.) or W is large ($\geq$ 10 MBPS), or both, then P–CSMA becomes inferior to the ideal case regardless of the vocoder rate. This is basically due to the relatively small transmission time of a packet for which P–CSMA is known to have a poor performance. As a result, we note that, when the delay requirement is low, an increase in channel bandwidth with the expectation of increasing the maximum number of voice sources is rewarded by smaller than proportional improvement.

Detailed discussion of these issues appeared in Stanford Electronics Laboratories Technical Report #213 entitled, "Simulation of Message–Based Priority Functions in Carrier Sense Multiaccess/Broadcast Systems", June 1981, and in a conference paper entitled, "On CSMA-CD Local Networks and Voice Communication," INFOCOM '82, Las Vegas, April 1982. (See Appendix I.)

Task B. Design and Analysis of Local Networks Suitable for Real–Time Applications

Accomplishments under this task are:

a) Design of a Round–Robin Scheme for Unidirectional Broadcast System Architecture.

b) Conceptual Design and Analysis of Expressnet, a High-Performance Integrated-Services Local Area Network.

c) Analysis of Round–Robin Schemes in Unidirectional Broadcast Local Area Networks.

d) A tutorial on Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks.

a) Design of a Round–Robin Scheme for Unidirectional Broadcast System Architecture

In a unidirectional broadcast system architecture, signal propagation is forced to be in only one direction. Broadcast communication is then achieved by various means, such as folding the cable or repeating all signals on a separate frequency in the reverse direction, so that signals transmitted by any user reach all other users on the reverse path.

Because of the unidirectional signalling property, an inherent ordering of the subscribers can be achieved which allows an efficient conflict–free round–robin scheme to be implemented. Details of the scheme, which we shall refer to as UBS–RR, have been presented in the paper entitled, "Efficient Round–Robin and Priority Schemes for Unidirectional Broadcast Systems", IBM and IFIP 6.4 International Workshop on Local Area Networks, Zurich, Switzerland, August 27–29, 1980. (See Appendix II.)

The importance of this scheme lies mainly in the fact that it is the predecessor to the more efficient scheme, referred to as Expressnet and discussed hereafter.

b) Conceptual Design and Analysis of Expressnet, a High-Performance Integrated-Services Local Area Network.

Local Area communication networks have registered significant advances in recent years. Currently networks operating in the 1-10Mb/s range and spanning a couple of kilometers are commercially available. Although they are adequately satisfying current needs for computer communications, it appears that, in the future, there will be an increasing demand for communication resources as new system architectures (such as distributed processing) evolve and as other services such as voice, graphics and video are integrated onto the same networks.

Multiaccess broadcast bus systems have been popular since, by combining the advantages of packet switching with broadcast communication, they offer efficient solutions to the communication needs both in simplicity of topology and flexibility in satisfying growth and variability. These systems have largely used random access contention schemes such as Carrier Sense Multiple Access (CSMA). A prominent example is Ethernet. Although they have proven to perform well in the environments for which they were designed, these schemes do exhibit performance limitations particularly when the channel bandwidth is high or the geographical area to be spaned is large. For example, it has been shown that the performance of CSMA-CD degrades significantly as the ratio $\tau W/B$ increases, where τ is the end-to-end propagation delay, W is the channel bandwidth and B is the number of bits per packet (including the preamble needed for synchronisation).

In order to overcome these limitations we proposed a new approach, also based on packet broadcasting. This type of network, called Expressnet, is a Unidirectional Broadcast System (UBS) type, in that it uses a unidirectional transmission medium on which the users contend according to some distributed conflict free round-robin algorithm. (Another recent proposal of this type is Fasnet as in "Fasnet: A proposal for a High Speed Local Network", by J. O. Limb, Proceedings of Office Information Systems Workshop, St. Maximin, France, October 1981 and "Description of Fasnet, a unidirectional Local Area Communications Network", by J. O. Limb and C. Flores, Bell Systems Technical Journal, September 1982.) Expressnet was conceived jointly by our group and by Fratta and Borgonovo of the Polytechnic of Milano. In this system the access overhead between consecutive packets in a round is independent of both the end to end propagation delay and the number of users connected to the network. Due to this feature this system overcomes some of the performance limitations of the random access schemes as well as earlier round-robin schemes such as the UBS-RR discussed in a) above.

Moreover, some features of Expressnet make it particularly suitable for voice applications. In view of integrating voice and data, a simple VOICE/DATA EXPRESS access protocol was described which satisfies the bandwidth requirement and maximum packet delay constraint for voice communication at all times, while guaranteeing a minimum bandwidth requirement for data traffic. Finally, it is noted that the VOICE/DATA EXPRESS access protocol constitutes a highly adaptive allocation scheme of channel bandwidth, which allows data users to steal the bandwidth unused by the voice application.

Expressnet was discussed first in a paper entitled, "The EXPRESS-NET: A Local Area Communication Network Integrating Voice and Data", presented at the International Conference on Performance of Data Communication Systems and Their Applications, Paris, France, September 14–16, 1981; A Stanford Electronics Laboratories Technical Report #220 by the same title was published in December 1980. A journal paper entitled, "EXPRESS-NET: A High-Performance Integrated-Services Local Area Network", appeared in IEEE Journal on Selected Areas in Communications, Special Issue on Local Area Networks, November 1983. (See Appendix II.)

A patent covering Expressnet will be issued in the U.S.A., Europe and Japan.

c) Analysis of Round-Robin Schemes in Unidirectional Broadcast Local Area Networks.

Various service disciplines can be achieved in Unidirectional Broadcast networks. Expressnet achieves a "conventional" round-robin discipline where users are serviced in a predescribed order determined by their physical location on the network. If a user has no message when its turn comes up, it declines to transmit and then must wait for the next round before getting another turn. We referred to this type of discipline as the Non-Gated Sequential Service discipline (NGSS). Both Fasnet and Expressnet can be operated in a gated mode. In this mode only those users who are ready at the beginning of a given round are serviced in that round. We refer to this discipline as the Gated Sequential Service discipline (GSS). In non-gated Fasnet users are also ordered according to their position on the bus; however, following a transmission, the next user to transmit is always the most upstream user who has a packet and has not yet transmitted in the current round. This discipline is referred to as the Most Upstream First Service discipline (MUFS). UBS-RR and Fasnet can support MUFS.

Using a model consisting of M users each with a single packet buffer, fixed size packets, and a Poisson arrival process to each of the M users, the above mentioned disciplines have been analyzed. The results have been reported upon in "Performance of Uni-Directional Broadcast Local Area Networks: Express-Net and Fasnet", IEEE Journal on Selected Areas in Communications, Special Issue on Local Area Networks, November 1983. (See Appendix II.) We showed that these systems, unlike random access techniques, can achieve a channel utilization close to 100% even when the channel bandwidth is high or the propagation delay of the signal over the network is large. In addition, the network remains stable as the load increases to infinity without the need for any dynamic control of the access protocol. The throughput delay characteristics are excellent and the maximum delay is bounded from above by a finite value which is easily computed. As the throughput approaches the network capacity the variance of delay reaches a peak and then drops to zero. At network capacity the system becomes deterministic with all users transmitting in every round. Finally, we noted that all three service disciplines exhibit similar performance characteristics. However, in GSS and MUFS there is an element of unfairness which favors upstream users over downstream users, while for NGSS the access protocol is completely fair with all users achieving the same performance.

d) A tutorial on Scheduling-Delay Multiple Access Schemes For Broadcast Local Area Networks

Although various ring systems and CSMA contention bus systems have been in operation for several years, more recently a number of distributed *demand assignment multiple access* (DAMA) schemes suitable for broadcast bus networks have emerged which provide conflict-free broadcast communications by means of various scheduling techniques. Among these schemes, the Token-Passing Bus Access method uses *explicit tokens*, i.e., control messages, to provide the required scheduling. Others use *implicit tokens*, whereby stations in the network rely on information deduced from the activity on the bus to schedule their transmissions. In the paper entitled "Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks," presented at AFRICOM'84 (see Appendix II), we identified three basic access mechanisms according to which these implicit-token DAMA schemes can be classified. These are the *scheduling-delay access mechanism*, the *reservation access mechanism* and the *attempt-and-defer access mechanism*. Then we presented in a unified manner those schemes using the scheduling delay access mechanism and compare them in terms of performance and other important attributes. This class is suitable for the Bidirectional Broadcast System configuration (BBS) where the only means for coordinating the access of the various users following the end of a transmission is by staggering the potential starting times of these users.

Task C. Performance Evaluation of Multihop Packet Radio Networks

Accomplishments in this task are:

a) Analysis of Two–Hop Centralized Packet Radio Networks, Under Slotted ALOHA and CSMA Access Schemes

b) Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes

c) Theoretical Results in the Throughput Analysis of Multihop Packet Radio Networks

d) Simulation of Multihop Packet Radio Networks

a) Analysis of Two–Hop Centralized Packet Radio Networks, Under Slotted ALOHA and CSMA Access Schemes

Until recently, the work done on the performance of multiaccess schemes focused mainly on the single-hop case, leading to a good understanding of the behavior of one-hop networks. Several access schemes designed specifically for single-hop networks or shown to perform particularly well in single hop networks may suffer severe degradation in performance in the multihop environment. One such example is carrier sense multiple access (CSMA) with its "hidden terminal" problem.

The analysis of multihop packet radio networks has proven to be a complex task. In order to gain some insight into the behavior of these networks, one alternative was

to analyze accurately simple but typical configurations. This we did in our study of two-hop centralized packet radio networks (slotted ALOHA and CSMA). This work was mostly done by the principal investigator (F. Tobagi) while at UCLA, and completed at Stanford University. The study was published as papers entitled, "Analysis of a Two-Hop Centralized Packet Radio Network—Part I: Slotted ALOHA," and "Analysis of a Two-Hop Centralized Packet Radio Network—Part II: Carrier Sense Multiple Access," which appeared in the IEEE Transactions on Communications, February 1980. (See Appendix III.)

b) Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes

In addition to CSMA, several schemes have been proposed in the past for multihop networks in view of providing improved performance, but no analysis had yet been performed to evaluate these schemes. Recently, a model has been developed by Boorstyn and Kershenbaum to analyze CSMA in a multihop environment. We extended the model used for CSMA to evaluate other multiaccess schemes and compare their performance.

Among the many multihop access schemes which can be conceived today using such features as carrier sensing, code division, etc., we have selected a few for consideration so far in our work. They are such that they lend themselves to simple solutions (particularly product form solutions). Although other schemes can be handled by the model described above, the analysis becomes more complex, and in the early stage of this research, we restricted ourselves to those listed hereafter. The access schemes define the conditions under which a scheduling point results in a transmission. We divided them into two major groups: the carrier sense type schemes, and the ALOHA-type schemes. In addition, we defined capture as the ability for a receiver to correctly receive a packet despite the presence of other time-overlapping transmissions. Perfect capture refers to the ability of receiving correctly the first message to reach the receiver regardless of the number of future overlapping messages; zero capture refers to the situation where any overlap in transmission results in complete destruction of all overlapping transmissions.

The Carrier Sense Type Schemes considered are:

(i) Carrier Sense Multiple Access (CSMA)

(ii) Busy Tone Multiple Access (BTMA)

(iii) A Directional CSMA (D-CSMA)

In the ALOHA type scheme, a node does not sense the channel before transmitting. Two protocols were considered for analysis:

(i) Pure ALOHA

(ii) CDMA-ALOHA

The analysis of the above schemes was performed and reported upon in the paper entitled, "Throughput Analysis of Multihop Packet Radio Networks under Various Channel Access Schemes", presented at INFOCOM'83, San Diego, April 19-21, 1983. (See Appendix

III.) A few topologies were considered to derive numerical results which show their relative performance.

c) Theoretical Results in the Throughput Analysis of Multihop Packet Radio Networks

The work reported upon in the above mentioned paper was based on the assumptions of perfect capture and zero propagation delay. Moreover, only schemes which lend themselves to reversible Markov chains (and thus product form solutions) were considered. Since then, considerable progress has been made on this subject. A more formal formulation of the problem under consideration was done, which allowed some deeper insight to be gained. In particular, a relation between the existence of a product form and reversibility of the associated Markov chain was found, and a simple criterion for the determination of the existence of a product form solution for a given protocol and topology was stated and proved. The criterion is the following: *"A channel access scheme on a given topology leads to a product form solution if and only if: for all pairs of transmissions i and j, transmission i blocks transmission j if and only if transmission j blocks transmission i."* Moreover, the equations expressing the throughput relationships previously derived heuristically and only for the case of perfect capture, have been justified on theoretical grounds and extended to the case of zero capture. These theoretical results have appeared in "Theoretical Results in Throughput Analysis of Multihop Packet Radio Networks," , presented at the International Conference on Communications, ICC'84, Amsterdam, May 1984. (See Appendix III.) The analysis of schemes for which a product form solution does not exist will require the direct (numerical) solution of the balance equation of the corresponding Markov chain. The writing of a computer program for the analysis of general networks and general protocols is currently in progress. Also currently under investigation, is the analysis of the non-zero propagation delay case, a case of indeed great importance.

d) Simulation of Multihop Packet Radio Networks

As the analysis of delay in multihop packet radio networks has proved to be extremely complex and intractable, we have undertaken a simulation effort as well. In such a simulation, not only were we able to relax some of the assumptions made in the analysis (Poisson scheduling process, zero propagation delay, etc.), but we were also able to study other networking issues such as deadlocks, the effects of rescheduling delays and buffer allocation schemes at the repeaters.

In an original version of the simulation program, called MULTIHOP, packets arriving to a repeater are stored at the end of a repeater's queue (one queue per repeater) and served on a first-come-first-served basis; the repeater always attempts transmission of the head of its queue. If the transmission is not successful, due to a collision or to buffer shortage at the next node, (an event assumed to be known instantaneously), then the packet is rescheduled according to a rescheduling delay distribution. New arrivals at a node are denied permission to enter the network whenever a packet is already stored in the buffer of that node, (thus applying an input buffer limit scheme). The problem with that implementation was that the scheme was prone to deadlocks. It should be noted here that deadlocks occur because a packet is never dropped once it has been accepted into the network. Deadlocks would

not occur if packets were dropped following a certain maximum number of unsuccessful retries. But in that case dropping packets due to a shortage of buffer space is indicative of bad management of network resources, leading to inefficiencies. The types of deadlocks encountered were the two common ones, namely, direct and indirect store-and-forward deadlocks. A solution to the deadlock problem was the implementation of the structured buffer pool technique in conjunction with channel queue limit flow control. Although these techniques have been traditionally described in the context of point-to-point store-and-forward networks, they have been easily adapted to this kind of broadcast networks by a simple modification. With these modifications implemented, the throughput-delay performance curves for the various schemes have been generated (so far for a ring network). A major result of this simulation was the indication that the buffer size at repeaters and the allocation of the buffer space among the various classes in the structured buffer pool technique could have an enormous effect on the performance of the schemes, and need to be taken into account when comparing these schemes. Additional work is currently under way.

Task D. Multinetwork Environments

The accomplishment in this task is: "Performance of Gateway-to-Gateway and End-to-End Flow Control Procedures in Internet Environments".

As computer communication networks multiply in number, it becomes more desirable to interconnect these networks in order to broaden their user services. The interconnection of networks is implemented through entities called gateways, which are interfaced to the individual networks as hosts. As in the case of a single network, a reliable delivery of packets between the end hosts must be provided. When there is some probability of packet loss, the reliable delivery can be insured through a flow control mechanism such as windowing which incorporates an automatic-repeat-request (ARQ) feature. In order to study flow control in multihop networks, we first introduced a new technique for computing the average delivery delay across a network. The packet delivery delay is defined as the time elapsed from when a packet arrives to the source host until its first correct copy is delivered to the destination host. The delay undergone by each copy across the network is called the end-to-end delay and its distribution is assumed to be given. Moreover, the loss of copies is assumed to be independent from each other and to have a given fixed probability.

Since, in general, the delivery delay distribution of a packet depends on the end-to-end delay of every copy of that packet, the exact analysis required the knowledge of the joint distribution of the one-way delays. However, realising that this joint distribution is often not known, we developed a simple model to characterize the dependencies between the end-to-end delays. To motivate this model, we considered two extreme cases. The first case is that of fixed routing — all the copies take the same route through the network. Given that first-in-first-out scheduling is utilized at the nodes, the order of copies at arrival to the network is the same as that upon their departure. The other extreme case occurs when the one-way delays are independent. This situation is realised when every copy takes

a different path across the network from every other copy — i.e., there exists "complete alternate routing".

Based on the above observation we modeled the network as consisting of a number of identical and disjoint paths. Every copy may be transmitted over any of these paths with equal probability. Furthermore the copies that are transmitted over the same path, although their ordering is preserved, have the same marginal end-to-end delay distribution. In fact in a network where there exists a large mixing of different traffic at every node we expect that all the copies in a stream experience approximately the same delay distribution. On the other hand, the copies taking different routes to the destination source undergo independent but again identically distributed end-to-end delays.

Based on some approximations, we could express the average delivery delay only in terms of the mean and coefficient of variation of the one-way delay in addition to the time-out period and the probability of packet loss. (Note that, for most communication systems, analysis can only provide the mean and coefficient of variation of the one-way delay. Also, results based on measurements or computer simulation are usually more accurate for the first few moments than for the complete distribution of the delay.) As expected we observed that, as long as the one-way delay is independent of the load, reducing the time-out period always decreases the average delivery delay. However, we know that a shorter timeout period results in a larger retransmission traffic which in turn should increase the end-to-end delay. Therefore to account for this effect, we next assumed that the mean end-to-end delay is given as a function of the total load on the network. Then we obtained a more realistic behavior of the average delivery delay versus the time-out period. It was observed that the average delivery delay is minimized for some optimum time-out period.

In an internet environment the flow control may be implemented between the source and destination hosts, or it may be implemented across every network on the communication path, i.e., between the gateways as well as the gateways and end hosts. In this study, we referred to the former case as end-to-end flow control (EEFC) and to the latter one as gateway-to-gateway flow control (GGFC). Our objective was to make a performance comparison between EEFC and GGFC. Furthermore, we considered the use of routing and flow control algorithms to enhance the performance. The performance is measured in terms of packet delivery delay, i.e., the time elapsed since the packet arrives at the source host until the first correct copy of it is delivered to the destination host. This performance is a function of the end-to-end delay and probability of loss across the network as well as the input rate of retransmission frequency. We observed that, there is an optimum retransmission frequency, or alternatively an optimum timeout, which minimizes the average delivery delay. We also observed that when the networks are in tandem, GGFC offers better performance than that of EEFC. However, when there is a high degree of traffic bifurcation between the networks, only under adaptive routing does GGFC result in a lower average delivery delay than that of EEFC. When GGFC is employed, the optimum timeouts may be computed at gateways and hosts using numerical methods. Then any routing algorithm which minimizes the average delay in a network can be used to minimize a cost function of the average delivery delays across the internet. This work

has been reported upon in "Performance of End-To-End and Gateway-To-Gateway Flow Control Procedures in Internet Environments", Proceedings of CDC'82, Orlando, Florida, December 1982. (See Appendix IV.)

Task E. Architectural Design and VLSI Implementation of Local Area Networks

On one hand the Expressnet has proven to be an interesting concept for local networking which is simple, efficient, and amenable to a multitude of applications with real-time constraints. On the other hand the advent of VLSI technology has brought about many new prospects for implementing network functions. We have initiated an effort to combine these two aspects and implement in VLSI an Expressnet to support a multitude of applications. This effort is still in its early stages. The contribution of such a task is two-fold: (1) to demonstrate the feasibility of the networking concept and prove its capabilities; and (2) to demonstrate that a complete integration of network and communication functions is feasible. In this report we merely present some exercises performed and our basic thoughts on the subject.

We designed and built a prototype VLSI chip implementing the Expressnet link control algorithm. The main purpose of this exercise was to learn more about how to use VLSI design tools. This Expressnet IC implemented the following functions: serial-to-parallel and parallel-to-serial data conversion, link access control functions for as many as eight different data types, and address and train recognition functions. The functions such as error correction and buffer allocation were not included in order to restrict the complexity of the IC being built. Later versions of this IC may include such functions. The IC was only partially debugged.

In an effort to expand on the work done in building the Expressnet chip, we perceived the need to conduct a requirement analysis with respect to capabilities desired of an Expressnet interface chip. The requirement analysis was first done to identify various data traffic characteristics and the number of trains required on Expressnet. The important result that emerged out of this study is the need to support high end-to-end user throughputs for some of the future applications. This was one of the major impetus for widening the scope of the project to include VLSI implementation of higher level protocols.

Having identified various projected future data rate demands, we next turned our attention to the measured performance of current implementations. All of the current implementations of TCP/IP and Ethernet software had end-to-end throughputs which were far below what the future applications demanded. We also studied the best possible performance of a TCP type transport protocol on a sequential execution dedicated machine. This convinced us that the major improvements needed in packet processing rates at network interfaces require a radically different approach. Our approach has been to study and utilise the parallelism that exists in communication algorithms.

In order to expose the inherent sequentiality of packet communication functions, we designed a simple notation. The notation allows one to express precedence relationships among various communication functions. Utilising this notation, we expressed a standard

communications architecture (namely, ISO's OSI reference model). The knowledge of this structure is useful since it provides a limit on the parallel execution that is possible in any implementation.

Due to limited resources, one may choose not to implement the maximum degree of parallelism possible but to share computational and storage resources among many of the functions. Thus, the actual parallelism used in any implementation may differ from the one that is shown to be possible using the above mentioned notation. In order to compare various possible implementations, we needed a notation which will allow one to express the real parallelism used in any particular implementation. Such an enhanced notation was designed next by extending the earlier notation to include indications of resource conflicts and actual flow of control which may be encountered in any implementation.

Using the extended notation, we next briefly examined many of the current implementations. These include a TCP/IP implementation on the V system, TCP/IP implementations on BSD 4.1 Unix, 3 MBPS Ethernet software on Altos, and an Internal I/O protocol of the V system kernel.

Currently we are engaged in identifying various parallel execution architectures suitable for implementing packet communication functions in VLSI. The implementation architectures are classified into four classes on the basis of the number of sites of executions used as well as the types of execution units used. The *single-CPU* class consists of implementations which use a single general-purpose Von Neumann machine. This leads to sequential execution of communication algorithms, and hence is not of interest. The *Multi-CPU* class includes many diverse types of implementations. All of these have more than one execution unit and the execution units are of general-purpose Von Neumann type machines. The *General-purpose Parallel Machine* class also includes more than one execution unit, but the execution units are no longer general-purpose Von Neumann type machines. The emphasis in this class is to utilize parallelism at the instruction level for any algorithm. The utilisation of parallelism in most cases is possible only when the algorithms have a significant amount of vector or matrix calculations. The fourth class is that of *functional architectures*. In this style of implementation, there is a direct correspondence between the functions to be executed and the hardware units to be utilised. In other words hardware units are specialised for the functions they are to perform. The number of such hardware units to be provided then directly depends on the structure of the algorithm being mapped onto the hardware. We feel that it is this class of implementations which is more relevant to our needs of VLSI implementations of packet communications. We are currently studying this type of architectures in more detail.

IV. List of Publications Under Contract MDA-79-C-0201, January 1, 1979 to June 30, 1984.

Publications are grouped by task. First, we list all conference and journal papers. Following that, we list separately all Stanford Electronics Laboratory technical reports and semiannual reports published under this contract.

CONFERENCE AND JOURNAL PAPERS

Task A. Real-Time Protocol Performance Analysis

F. A. Tobagi, "Multiaccess Protocols in Packet Communication Systems," *IEEE Transactions on Communications, Vol. COM-28,* pp. 468–488, April 1980 (invited paper in the special issue on computer network architecture and protocols).

F. A. Tobagi and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access With Collision Detection," *Computer Networks,* Vol. 4, No. 5, pp. 245–259, October/November 1980.

R. Rom and F. A. Tobagi, "Message-based Priority Functions in Local Multiaccess Communication Systems," *Computer Networks,* Vol. 5, No. 4, pp. 273–286, July 1981.

F. A. Tobagi, "Carrier Sense Multiple Access With Message-Based Priority Functions," *IEEE Transactions on Communications,* Vol. COM-30, pp. 185–200, January 1982.

F. A. Tobagi, "Distribution of Packet Delay and Interdeparture Time in Slotted ALOHA and CSMA Channels," *JACM, Vol. 29, No. 3,* October 1982.

F. A. Tobagi and N. Gonzalez-Cawley, "On CSMA-CD Local Networks and Voice Communication," INFOCOM '82, Las Vegas, April 1982.

Task B. Design and Analysis of Local Networks Suitable for Real-Time Applications

F. A. Tobagi and R. Rom, "Efficient Round-Robin and Priority Schemes for Unidirectional Broadcast Systems," *Proceedings of the IFIP-IBM Zurich Workshop on Local Area Networks,* Zurich, Switzerland, August 27–29, 1980.

L. Fratta, F. Borgonovo, and F. A. Tobagi, "The EXPRESS-NET: A Local Area Communication Network Integrating Voice and Data," *Proceedings of the International Conference on Performance of Data Communication Systems and Their Applications,* Paris, France, September 14–16, 1981; also in the *Proceedings of CompCon, Spring 82, San Francisco,* February 1982.

M. Fine and F. A. Tobagi, "Performance Analysis of a Conflict-Free Round-Robin Scheme in a Unidirectional Broadcast System," *Proceedings of the IEEE International Conference on Communications, ICC '82,* Philadelphia, June 1982.

F. A. Tobagi, F. Borgonovo, and L. Fratta, "EXPRESS-NET: A High-Performance Integrated-Services Local Area Network," *IEEE Journal on Selected Areas in Communications, Special Issue on Local Area Networks, Vol. SAC-1, No. 5,* pp. 898–913, November 1983.

F. A. Tobagi and M. Fine, "Performance of Unidirectional Broadcast Local Area Networks: Express-Net and Fastnet," *IEEE Journal on Selected Areas in Communications, Special Issue on Local Area Networks, Vol. SAC-1, No. 5,* pp. 913–926 November 1983.

M. Fine and F. A. Tobagi, "Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks," in *Proceedings of the 1st African Conference on Computer Communications, AFRICOM'84,* Tunis, Tunisia, May 1984.

Task C. Performance Evaluation of Multihop Packet Radio Networks

F. A. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network—Part I: Slotted ALOHA," *IEEE Transactions on Communications,* Vol. COM-28, pp. 196–207, February 1980.

F. A. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network—Part II: Carrier Sense Multiple Access," *IEEE Transactions on Communications,* Vol. COM-28, pp. 208–216, February 1980.

F. A. Tobagi and J. Brásio, "Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes," *Proceedings of INFOCOM '83,* San Diego, April 1983.

J. M. Brásio and F. A. Tobagi, "Theoretical Results in Throughput Analysis of Multihop Packet Radio Networks," in *Proceedings of the International Conference on Communications, ICC'84,* Amsterdam, The Netherlands, May 1984.

Task D. Multinetwork Environments

M. Nassehi and F. A. Tobagi, "Performance of Gateway-to-Gateway and End-to-End Flow Control Procedures in Internet Environments" in *Proceeding of the 21st IEEE Conference on Decision and Control,* Orlando, Florida, December 1982.

STANFORD ELECTRONICS LABORATORIES TECHNICAL REPORTS AND SEMI-ANNUAL REPORTS

F. A. Tobagi and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," Stanford Electronics Laboratories Technical Report No. 173, June 30, 1979.

F. A. Tobagi, "Message-Based Priority Functions in Multiaccess/Broadcast Communication Systems with a Carrier Sense Capability," Stanford Electronics Laboratories Technical Report No. 181, October 1979.

F. A. Tobagi, "Distributions of Packet Delay and Interdeparture Time in Slotted ALOHA Channels", Stanford Electronics Laboratories Technical Report No. 186, April 1, 1980.

F. A. Tobagi, "Distribution of Packet Delay and Interdeparture Time in Carrier Sense Multiple Access," Stanford Electronics Laboratories Technical Report No. 187, April 1, 1980. (This report was supported by the U. S. Army, under Army Research Office Contract No. DAAG 29–79–C–0138.)

F. A. Tobagi, "Carrier Sense Multiple Access with Message-Based Priority Functions", Stanford Electronics Laboratories Technical Report No. 200, December 1, 1980.

F. A. Tobagi, F. Borgonovo and L. Fratta, "The EXPRESS–NET: A Local Area Communication Network Integrating Voice and Data", Stanford Electronics Laboratories Technical Report No. 220, December 4, 1980.

N. Gonsales-Cawley and F. A. Tobagi, "Simulation of Message-Based Priority Functions in Carrier Sense Multiaccess/Broadcast Systems," Stanford Electronics Laboratories Technical Report No. 213, June 1, 1981. (This work was supported by the U. S. Army, Army Research Office Contract No. DAAG 29–79–C–0138.)

F. A. Tobagi, "Discrete-Time Queueing Systems With Priority and State-Dependent Arrival and Departure Processes," Stanford Electronics Laboratories, *Technical Report No. 217,* August 1981.

F. A. Tobagi and D. H. Shur, "Simulation of Busy Tone Multiple Access Modes in Multihop Packet Radio Networks," Stanford Electronics Laboratories Technical Report No. 234, November 1982. (This report was supported by the U. S. Army, Army Research Office Contract No. DAAG 29–79–C–0138.)

M. Fine and F. A. Tobagi, "Performance of Unidirectional Broadcast Local Area Networks: Express-net and Fasnet," Stanford Electronics Laboratories Technical Report No. 83-252, December 1983.

"Analysis of Real–Time Protocol Performance", Semiannual Technical Report, June 30, 1979.

"Analysis of Real–Time Protocol Performance", Annual Technical Report, January 31, 1980.

"Analysis of Real–Time Protocol Performance", Semiannual Technical Report, June 30, 1980.

"Analysis of Real–Time Protocol Performance", Annual Technical Report, January 31, 1981.

"Analysis of Real–Time Protocol Performance", Semiannual Technical Report, June 30, 1981.

"Analysis of Real–Time Protocol Performance", Annual Technical Report, December 31, 1981.

APPENDIX I.

F. A. Tobagi, "Multiaccess Protocols in Packet Communication Systems," *IEEE Transactions on Communications*, Vol. COM-28, pp. 468–488, April 1980 (invited paper in the special issue on computer network architectures and protocols).

F. A. Tobagi and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access With Collision Detection," *Computer Networks*, Vol. 4, No. 5, pp. 245–259, October/ November 1980.

R. Rom and F. A. Tobagi, "Message-based Priority Functions in Local Multiaccess Communication Systems," *Computer Networks*, Vol. 5, No. 4, pp. 273–286, July 1981.

F. A. Tobagi, "Carrier Sense Multiple Access With Message-Based Priority Functions," *IEEE Transactions on Communications*, Vol. COM–30, pp. 185–200, January 1982.

F. A. Tobagi, "Distribution of Packet Delay and Interdeparture Time in Slotted ALOHA and CSMA Channels," *JACM*, Vol. 29, No. 3, October 1982.

F. A. Tobagi and N. Gonzales-Cawley, "On CSMA-CD Local Networks and Voice Communication," INFOCOM '82, Las Vegas, April 1982.

Multiaccess Protocols in Packet Communication Systems

FOUAD A. TOBAGI, MEMBER, IEEE

(Invited Paper)

Abstract—**The need for multiaccess protocols arises whenever a resource is shared by many independent contending users. Two major factors contribute to such a situation: the need to share expensive resources in order to achieve their *efficient utilization*, or the need to provide a *high degree of connectivity* for communication among independent subscribers (or both). In data transmission systems, the communication bandwidth is often the prime resource, and it is with respect to this resource that we view multiaccess protocols here. We give in this paper a unified presentation of the various multiaccess techniques which we group into five categories: 1) fixed assignment techniques, 2) random access techniques, 3) centrally controlled demand assignment techniques, 4) demand assignment techniques with distributed control, and 5) mixed strategies. We discuss their applicability to different environments, namely, satellite channels, local area communication networks and multihop store-and-forward broadcast networks, and their applicability to different types of data traffic, namely stream traffic and bursty traffic. We also present the performance of many of the multiaccess protocols in terms of bandwidth utilization and message delay.**

I. INTRODUCTION

THE need for multiaccess protocols arises whenever a resource is shared (and thus accessed) by a number of independent users. One main reason contributing to such a situation is the need to *share scarce and expensive resources*. An excellent example is typified by time-sharing systems. Time-sharing was developed in the 1960's to make the powerful processing capability of a large computer system available to a large population of users, each of whom has relatively small or infrequent demands so that a dedicated system cannot be economically justified. Two advantages are gained: the smoothing effect of large populations on the demand resulting from the law of large numbers and a lower cost per unit of service resulting from the (almost always existing) economy of scale.

A second major reason contributing to the multiaccess of a common resource by many independent entities is the need for communication among the entities; we refer to this as the *connectivity requirement*. An excellent example today is the telephone system, the main purpose of which is to provide a high degree of connectivity among its subscribers. The multiaccess protocol used in the telephone system is conceptually simple; it merely consists of placing a request for connection to one or several parties, a request which gets honored by the system if all the required resources are available.

Manuscript received September 6, 1979; revised January 7, 1980. This work was supported by the Defense Advanced Research Projects Agency under Contract MDA 903-79-C-0201, Order A03717, monitored by the Office of Naval Research.

The author is with the Computer Systems Laboratory, Stanford University, Stanford, CA 94305.

Packet Communication

Let us now consider data communication systems, the subject of interest in this paper. Communications engineers have long recognized the need to multiplex expensive transmission facilities and switching equipment. The earliest techniques for doing this were synchronous time-division multiplexing and frequency-division multiplexing. These methods assign a fixed subset of the time-bandwidth space to each of several subscribers and are very successful for stream-type traffic such as voice. With computer traffic however, usually characterized as *bursty*, fixed assignment techniques are not nearly so successful, and to solve this problem, *packet communication systems* have been developed over the past decade [1]-[7]. Packet communication is based on the idea that part or all of the available resources are allocated to one user at a time but for just a short period of time. Here each component of the system is itself a resource which is multiaccessed and shared by the many contending users. To achieve sharing at the component level, customers are required to divide their messages into small units called packets which carry information regarding the source and the intended recipient.

One type of packet communication network, known as the *point-to-point store-and-forward* network, is one where packet switches are interconnected by point-to-point data circuits according to some topological structure. Packets are transmitted independently and pass asynchronously from one switch to another until they reach their destination. The multiplexing of packets on a channel is done by queueing them at each switch until the outgoing channel is free. Typical examples are the ARPANET [7], the Cigale subnetwork [8], TELENET [9], and DATAPAC [10].

Another type of packet transmission network is the (single-hop) *multiaccess/broadcast* network typified by the ALOHA network [11], SATNET [12], and ETHERNET [5]. Here a *single* transmission medium is shared by all subscribers; the medium is allocated to each subscriber for the time required to transmit a single packet. The inherent single-hop broadcast nature of these systems achieves full connectivity at small additional cost. Each subscriber is connected to the common channel through a smart interface which listens to all transmissions and absorbs packets addressed to it.

Yet a third type of packet network can be identified. It is the (multihop) *store-and-forward multiaccess/broadcast* type which combines the features exhibited (and problems encountered) in the two types just mentioned. The best and perhaps only example of this type is the packet radio network (PRNET) sponsored by the Advanced Research Projects Agency [13], [14]. The PRNET is an extension of the

0090-6778/80/0400-0468$00.75 © 1980 IEEE

ALOHA network in that it includes many added features such as direct communication by a ground radio network between *mobile* users over *wide* geographical areas, coexistence with possibly different systems in the same frequency band, antijam protection, etc. The key requirement of direct communication over wide geographical areas renders store-and-forward switches, called repeaters, integral components of the system. Furthermore, for easy communication among mobile users and for rapid deployment in military applications, all devices employ omnidirectional antennas and share a high-speed radio channel; hence the multiaccess/broadcast nature of the system.

The main issue of concern here is how to control access to a common channel to efficiently allocate the available communication bandwidth to the many contending users. The solutions to this problem form the set of protocols known as *multiaccess protocols*. These protocols and their performance differ according to the environment in question and the system requirements to be satisfied. We devote the next few paragraphs to summarizing the basic relevant characteristics underlying these environments.

Consider first *satellite channels*. A satellite transponder in a geostationary orbit above the earth provides long-haul communication capabilities. It can receive signals from any earth station in its coverage pattern and can transmit signals to all such earth stations (unless the satellite uses spot beams). Full connectivity and multidestination addressing can both be readily accommodated. The many characteristics regarding data rates, error rates, satellite coverage, channelization, and design of earth stations have been fully discussed in a recent paper by Jacobs *et al.* [12]. Perhaps the most important characteristic relevant to this discussion is the inherent long propagation delay of approximately 0.25 s for a single hop. This delay which is usually long compared to the transmission time of a packet, has a major impact on the bandwidth allocation techniques and on the error and flow control protocols.

In *ground radio* environments, the propagation delay is relatively short compared to the transmission time of a packet, and as we shall see in the sequel, this can be of great advantage in controlling access to a common channel. It is important however to distinguish single-hop environments where direct full connectivity is assumed to prevail, and more complex user environments where, due to geographical distance and/or obstacles opaque to UHF signals, limited direct connectivity is achieved. Clearly, the latter situation is significantly more complex as it gives rise to a multihop system where global control of system operation and resource allocation (whether centralized or distributed) is much harder to accomplish. Another dimension of complexity results from the fact that, unlike satellite environments where earth stations are stationary, ground radio systems must also support mobile users. With mobile users, not only does demand on the system exhibit relatively fast dynamic changes, but the radio propagation characteristics are subject to important variations in received signal strength so that system connectivity is at all times difficult to predict; with these considerations it is important to devise access schemes and system control mechanisms that allow the system to adapt itself to these changes. Furthermore, multipath effects in urban environments can be so disastrous that special signaling schemes, such as spread spectrum, may be in order [14]. Finally, another point of growing concern today is RF spectrum utilization. This is becoming an increasingly predominant factor in determining the structure of radio systems, both in satellite and ground environments. A packet radio system which allows the dynamic allocation of the spectrum to a large population of bursty mobile user needs flexible high performance multiaccess schemes which can take advantage of the law of large numbers, and which permit coexistence of the system with other (possibly different) systems in the same frequency band.

Finally, we consider *local area communication* systems. These span short distances (ranging from a few meters up to a few kilometers) and usually involve high data rates. The transmission medium can be privately owned and inexpensive, such as twisted pair or coaxial cable. Local area environments are characterized by a large and often variable number of devices requiring interconnection, and these are often inexpensive. These situations call for communication networks with simple topologies and simple and inexpensive connection interfaces that can provide great flexibility in accommodating the variability in the environment and that achieve the desired level of reliability. With these constraints, we again face the situation in which a high bandwidth channel is to be shared by independent users. Short propagation delays and high data rates are the main characteristics that are exploited in devising multiaccess schemes appropriate to local area environments.

Multiaccess schemes are evaluated according to various criteria. The performance characteristics that are desirable are, first of all, high bandwidth utilization and low message delays. But a number of other attributes are just as important. The ability for an access protocol to simultaneously support traffic of different types, different priorities, with variable message lengths, and differing delay constraints is essential as higher bandwidth utilization is achieved by the multiplexing of all traffic types. Also, to guarantee proper operation of schemes with distributed control, robustness, defined here as the insensitivity to errors resulting in misinformation, is also most desirable.

Having so far discussed briefly the basic characteristics and system requirements underlying the various communication environments, we now proceed with a discussion of the multiaccess protocols appropriate to these environments.

II. MULTIACCESS PROTOCOLS

Multiaccess protocols differ by the static or dynamic nature of the bandwidth allocation algorithm, the centralized or distributed nature of the decision-making process, and the degree of adaptivity of the algorithm to changing needs. Accordingly, these protocols can be grouped into five classes. The first class, labeled *fixed assignment techniques*, consists of those techniques which allocate the channel bandwidth to the users in a static fashion, independently of their activity. The second class is that of *random access techniques*. In this class the entire bandwidth is provided to the users as a single channel to be accessed randomly; since collisions may result which degrade the performance of the channel, improved

performance can be achieved by either synchronizing users so that their transmissions coincide with the boundaries of time slots, by sensing carrier prior to transmission, or both. The third and fourth classes correspond to *demand assignment* techniques. Demand assignment techniques require that explicit control information regarding the users' need for the communication resource be exchanged. A distinction is made between those techniques in which the decision-making is centralized (constituting the third class in question), and those techniques in which all users individually execute a distributed algorithm based on control information exchanged among them. The latter constitute the fourth class. The fifth class, labeled *adaptive strategies and mixed modes*, includes those techniques which consist of a mixture of several distinct modes, and those strategies in which the choice of an access scheme is itself adaptive to the varying need, in the hope that near-optimum performance will be achieved at all times.

We describe here the various protocols known today, either implemented or proposed, and discuss their performance and applicability to the different environments introduced in Section I. For this we consider the (conceptually) simplest situation consisting of M users wishing to communicate over a channel. This situation arises typically in a satellite communication environment or in a single-hop ground radio environment.

A. Fixed Assignment Techniques

Fixed assignment techniques consist of allocating the channel to the user, independently of their activity, by partitioning the time-bandwidth space into slots which are assigned in a static predetermined fashion. These techniques take two common forms: *orthogonal*, such as frequency division multiple access (FDMA) or synchronous time division multiple access (TDMA), and "*quasi-orthogonal*" such as code division multiple access (CDMA).

1) *FDMA and TDMA*: FDMA consists of assigning to each user a fraction of the bandwidth and confining its access to the allocated subband. Orthogonality is achieved in the frequency domain. FDMA is relatively simple to implement and requires no real time coordination among the users.

TDMA consists of assigning fixed predetermined channel time slots to each user; the user has access to the entire channel bandwith, but only during its allocated slots. Here, signaling waveforms are orthogonal in time.

In the author's opinion, a number of disadvantages exist for FDMA when compared to TDMA. FDMA wastes a fraction of the bandwidth to achieve adequate frequency separation. FDMA is also characterized by a lack of flexibility in performing changes in the allocation of the bandwidth and certainly the lack of broadcast operation. The major disadvantages in TDMA are the need to provide A/D converters for overlap traffic such as voice, and rapid burst synchronization and sufficient burst separation to avoid time overlap. However, it has been shown that guard bands of less than 200 ns are achievable (as in INTELSAT's MAT-1 TDMA system, for example) and many operational systems are moving towards the use of TDMA [16]. Timing at an earth station is provided by a global time reference established either explicitly by a reference station, or implicitly by measurement of the propagation delay from the earth station to the transponder. In order to allow the TDMA modems to acquire frequency, phase, bit timing and bit framing synchronization for each received burst, a preamble is included in front of each burst requiring typically from 100 to 200 bit times. Thus clearly, TDMA is more complex to implement than FDMA, but an important advantage is the connectivity which results from the fact that all receivers listen to the same channel while senders transmit on the same common channel at different times. Accordingly, many network realizations, both in ground and satellite environments, are easier to accomplish [12], [14].

From the performance standpoint it has also been established that TDMA is superior to FDMA in many cases of practical interest. I. Rubin has shown that the random variable representing packet delay is always larger in FDMA than in TDMA [17] for comparable systems. Lam derived the average message delay for a TDMA system with multipacket messages and a nonpreemptive priority queue discipline [18]. There, too, it was shown that TDMA is superior to FDMA.

For both FDMA and TDMA, the fixed preallocation of the frequency or time resource does not have to be equal for all users, but can be tailored to fit their needs (assumed constant). Kosovych studied two TDMA implementations [19]. In the first, called *contiguous assignment*, the users are cyclically ordered in the time sequence in which they have access to the channel. Each user is periodically assigned its *own* fixed time duration. In the second implementation, called *distributed allocation*, all access periods are of equal time duration, but the frequency of accesses can be different from one user to the other. It was shown that for situations in which the transmission overhead (defined as guard time and synchronization preamble time) is large, the contiguous fixed assignment implementation is better suited and provides substantially better performance than distributed fixed assignments, while when the transmission overhead is small, distributed fixed assignments provide slightly better performance.

Finally we note that, even though the allocation can be tailored to the relative need of each user, fixed allocation can be wasteful if the users' demand is highly bursty, as we shall explicitly see in the sequel. Given these limitations, one may increase the channel utilization beyond FDMA and TDMA by using asynchronous time division multiple access (ATDMA), also known as statistical multiplexing [70]. Basically the technique consists of switching the allocation of the channel from one user to another only when the former is idle and the latter is ready to transmit data. Thus the channel is *dynamically* allocated to the various users according to their need. The performance of ATDMA in packet communication systems corresponds to that of a work-conserving single server queueing system, and is the best we can achieve under unpredictable demand. Unfortunately, it is not always possible to accomplish the necessary coordination among the users. This mode of multiplexing is possible only when several colocated users (such as at the same earth station) are sharing a single point-to-point channel.

2) *CDMA*: Unlike FDMA and TDMA, code division multi-

ple access allows overlap in transmission both in the frequency and time coordinates. It achieves orthogonality by the use of different signaling codes in conjunction with matched filters (or equivalently, correlation detection) at the intended receivers. Multiple orthogonal codes are obtained at the expense of increased bandwidth requirements (in order to spread the waveforms); this also results in a lack of flexibility in interconnecting all users (unless, of course, matched filters corresponding to all codes are provided at all receivers). However, CDMA has the advantage of allowing the coexistence of several systems in the same band, as long as different codes are used for different systems. Moreover, it is also possible to separate, by "capture," time overlapping signaling waveforms with the same code, thus achieving connectivity and efficient spectrum utilization. This interesting possibility falls into the class of random access techniques and is addressed in the following subsection.

B. Random Access Techniques

In computer communication, much data traffic is characterized as bursty e.g., interactive terminal traffic. Burstiness is a result of the high degree of randomness seen in the message generation time and size, and of the relatively low-delay constraint required by the user. If one were to observe the user's behavior over a period of time, one would see that the user requires the communications resources rather infrequently; but when he does, he requires a rapid response. That is, there is an inherently large peak-to-average ratio in the required data transmission rate. If fixed subchannel allocation schemes are used, then one must assign enough capacity to each subscriber to meet his peak transmission rates with the consequence that the resulting channel utilization is low. A more advantageous approach is to provide a single sharable high-speed channel to the large number of users. The strong law of large numbers then guarantees that, with a very high probability, the demand at any instant will be approximately equal to the sum of the average demands of that population. As stated in the introduction, packet communication is a natural means to achieve sharing of the common channel. When dealing with shared channels in a packet-switched mode, one must be prepared to resolve conflicts which arise when more than one demand is placed upon the channel. For example, in packet-switched radio channels, whenever a portion of one user's transmission overlaps with another user's transmission, the two collide and "destroy" each other (unless a code division multiple-access scheme is used). The existence of some positive acknowledgment scheme permits the transmitter to determine if his transmission is successful or not. The problem is how to control the access to the common channel in a fashion which produces, under the physical constraints of simplicity and hardware implementation, an acceptable level of performance. The difficulty in controlling a channel which must carry its own control information has given rise to the so-called random-access protocols, among others. We describe these here by considering again single-hop environments.

1) *ALOHA [20]-[22]*: Historically, the *pure ALOHA* protocol was first used in the ALOHA system, a single-hop terminal access network developed in 1970 at the University of Hawaii, employing packet-switching on a radio channel [11], [20]. The simplest of its kind, pure ALOHA permits a user to transmit any time it desires. If they do so, and within some appropriate time-out period it receives an acknowledgment from the destination (the central computer), then it knows that no conflict occurred. Otherwise it assumes that a collision occurred and it must retransmit. To avoid continuously repeated conflicts, the retransmission delay is randomized across the transmitting devices, thus spreading the retry packets over time. A slotted version, referred to as *slotted ALOHA*, is obtained by dividing time into slots of duration equal to the transmission time of a single packet (assuming constant-length packets)[21], [22]. Each user is required to synchronize the start of transmission of its packets to coincide with the slot boundary. When two packets conflict, they will overlap completely rather than partially, providing an increase in channel efficiency over pure ALOHA. Due to conflicts and idle channel time, the maximum channel efficiency available using ALOHA is less than 100 percent, 18 percent for pure ALOHA and 36 percent for sloted ALOHA. Both schemes are theoretically applicable to satellite, ground radio and local bus environments. The slotted version has the advantage of efficiency, but in multihop ground radio, it has the disadvantage that synchronization may be hard to achieve.

Although the maximum achievable channel utilization is low, the ALOHA schemes are superior to fixed assignment schemes when there is a large population of bursty users. This point is illustrated in comparing the performance of FDMA with that of slotted ALOHA when M users, each of which generates packets at a rate of λ packets per second, share a radio channel of W·Hz [23]. Figs. 1 and 2 display the constant delay contours in the (M, λ) and (W, λ) planes, respectively, showing the important improvement gained in terms of bandwidth required, population size supported, and delay achieved when the users are bursty.

2) *Carrier Sense Multiple Access (CSMA) [24], [25]*: In ground radio environments the channel can be characterized as wideband with a propagation delay between any source-destination pair that is small compared to the packet transmission time. In such an environment one may attempt to avoid collisions by listening to the carrier due to another user's transmission before transmitting, and inhibiting transmission if the channel is sensed busy. This feature gives rise to a random access scheme known as carrier sense multiple access (CSMA) [24], [25]. While in the ALOHA scheme only one action could be taken by the terminals, namely, to transmit, here many strategies are possible so that many CSMA protocols exist differing according to action that a terminal takes to transmit a packet after sensing the channel. In all cases, however, when a terminal learns that its transmission had incurred a collision, it reschedules the transmission of the packet according to the randomly distributed delay. At this new point in time, the transmitter senses the channel again and repeats the algorithm dictated by the protocol. There are two main CSMA protocols known as *nonpersistent* and *p-persistent* CSMA depending on whether the transmission by a station which finds the channel busy

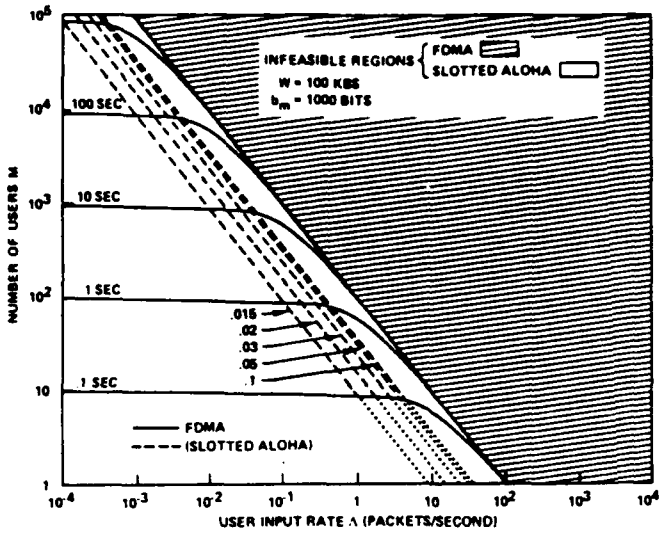


Fig. 1. FDMA and slotted ALOHA access: performance with 100 kbits/s bandwidth and 1000 bit packets [23].

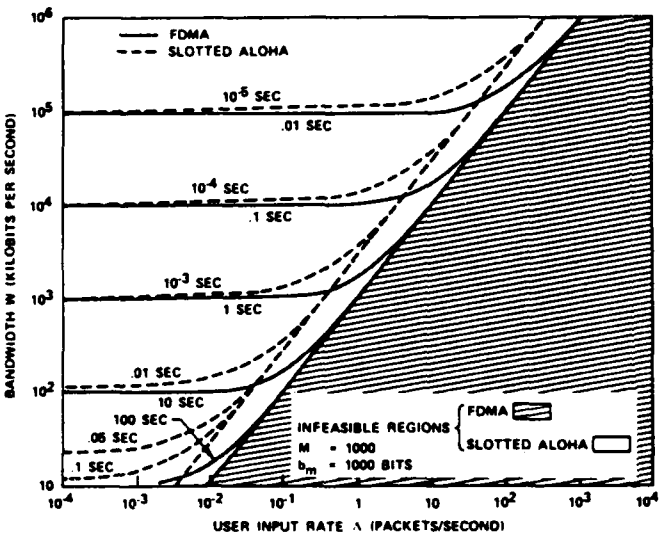


Fig. 2. FDMA and slotted ALOHA random access: bandwidth requirements for 1000 terminals. Contours are for constant delay [23].

is to occur later or immediately following the current one with probability p. Many variants and modifications of these two schemes have also been proposed. Thus, in nonpersistent CSMA, a ready terminal senses the channel and operates as follows:

1) If the channel is sensed idle, it transmits the packet.

2) If the channel is sensed busy, then the terminal schedules the retransmission of the packet to some later time according to the retransmission delay distribution. At this new point in time, it senses the channel and repeats the algorithm described.

The 1-persistent CSMA protocol, a special case of p-persistent CSMA, was devised in order to (presumably) achieve acceptable throughput by never letting the channel go idle if some ready terminal is available. More precisely, a ready terminal senses the channel and operates as follows:

1) If the channel is sensed idle, it transmits the packet with probability one.

2) If the channel is sensed busy, it waits until the channel goes idle and then immediately transmits the packet with probability one (i.e., persisting on transmitting with $p = 1$).

A slotted version of these CSMA protocols can also be considered in which the time axis is slotted and the slot size is τ s where τ is the maximum propagation delay among all pairs. Note that this definition of a slot is different from that used in the description of slotted ALOHA. Here a packet transmission time is equivalent to several slots. We make this distinction by referring to a slot of size τ s as a "minislot." All terminals are synchronized and are forced to start transmission only at the beginning of a minislot. When a packet's arrival occurs in a minislot, the terminal waits until the next minislot boundary and operates according to the protocols described above.

In the case of a 1-persistent CSMA, we note that whenever two or more terminals become ready during a packet transmission period, they wait for the channel to become idle (at the end of that transmission) and then they all transmit with probability one. A conflict will also occur with probability one. The idea of randomizing the starting time of transmission of packets accumulating at the end of a transmission period seems reasonable for interference reduction and throughput improvement. Thus we have the p-persistent scheme which involves including an additional parameter p, the probability that a ready packet persists ($1 - p$ being the probability of delaying transmission by τ seconds, the propagation delay). The parameter p is chosen to reduce the level of interference while keeping the idle periods between any two consecutive nonoverlapped transmission as small as possible.

More precisely, the p-persistent CSMA protocol consists of the following: the time axis is minislotted and the system is synchronized such that all terminals begin their transmission at the beginning of a minislot. If a ready terminal senses the channel idle, then with probability p, the terminal transmits the packet; and with probability $1 - p$, the terminal delays the transmission of the packet by τ seconds (i.e., one minislot). If at this new point in time, the channel is still detected idle, the same process is repeated. Otherwise some packet must have started transmission, and the terminal in question schedules the retransmission of the packet according to the retransmission delay distribution (i.e., acts as if it had conflicted and learned about the conflict). If the ready terminal senses the channel busy, it waits until it becomes idle (at the end of the current transmission) and then operates as above.

Packet broadcasting technology has also been shown to be very effective in satisfying many local area in-building communication requirements. A prominent example is ETHERNET, a local communication network which uses CSMA on a tapped coaxial cable to which all the communicating devices are connected [5]. The device connection interface is a passive cable tap so that failure of an interface does not prevent communication among the remaining devices. The use of a single coaxial cable achieves broadcast communication. The only difference between this and the single-hop radio is that, in addition to sensing carrier, it is possible for the transceivers, when they detect interference among several transmissions (including their own), to abort the transmission

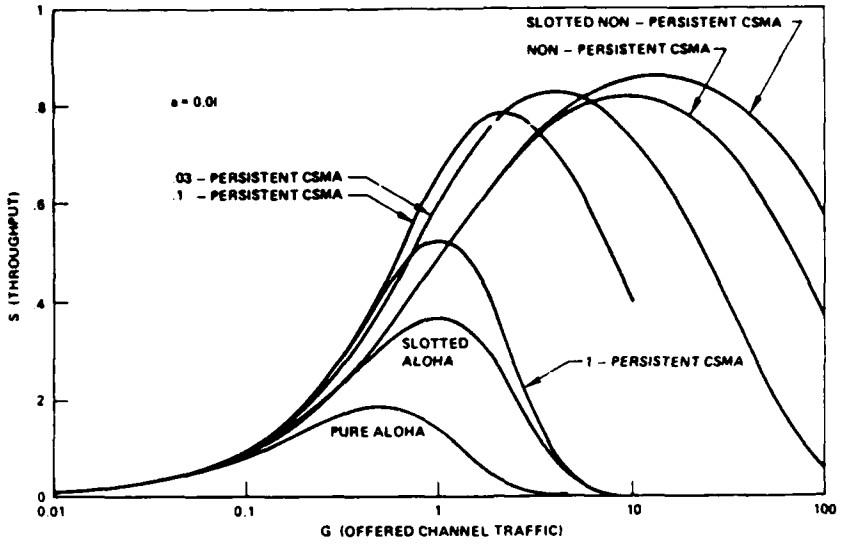


Fig. 3. Throughput for the various random access modes (propagation delay $a = 0.01$) [24].

of colliding packets. This is achieved by having each transmitting device compare the bit stream it is transmitting to the bit stream it sees on the channel. This variation of CSMA is referred to as carrier sense multiple access with collision detection (CSMA-CD) [26].

3) *Performance of Random Access:* Many theoretical studies have been carried out to determine the performance of these random access schemes [20]-[22], [24]-[29]. We summarize here the most important results. Let S denote the aggregate rate of packet generation from the entire population of users, G the rate of packet transmissions (new and repeated, hence $G \geq S$), and D the packet delay (defined as the time elapsed between the time that the packet is originated and the time it is successfully received at the destination), all normalized to the (fixed) packet transmission time T. Analytic and simulation models provide us, for each random access scheme, with a relationship between S and G (displayed in Fig. 3), and the throughput delay tradeoff (displayed in Fig. 4) for a normalized propagation delay $a = \tau/T = 0.01$. We note that the behavior of these schemes is typical of contention systems, namely that the throughput increases as the offered channel traffic increases from zero, but reaches a maximum value for some optimum value of G, and then constantly decreases as G increases beyond that optimal value. Maximizing S with respect to the channel traffic rate G for each of the access modes leads to the channel capacity for that mode. From Fig. 4 we clearly note that D increases as the throughput increases, and reaches infinite values as the throughput approaches the channel capacity. These results show the evident superiority of CSMA over the ALOHA scheme. The CSMA channel capacity in some cases may be as high as 90 percent of the available bandwidth. It is clear however that, as expected, the channel capacity and the throughput-delay tradeoff for the CSMA schemes degrade as the normalized propagation delay ($a = \tau/T$) increases. Fig. 5 illustrates the sensitivity of the channel capacity to a.

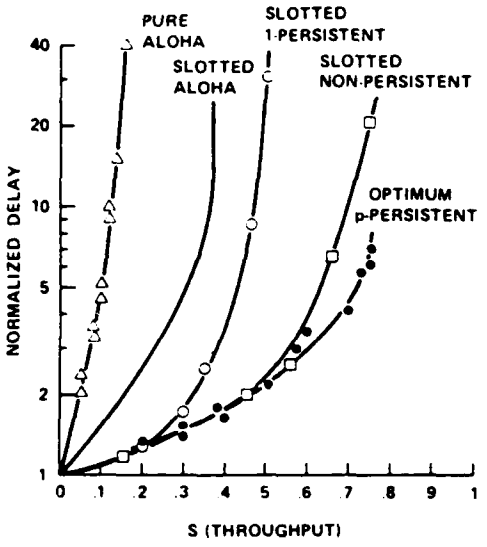


Fig. 4. CSMA and ALOHA: throughput-delay tradeoffs from simulation (propagation delay $a = 0.01$) [24].

CSMA-CD offers even more improvement. A system parameter affecting this improvement is the time required to detect collisions and abort ongoing colliding transmissions. We denote this time by ($\gamma \leq T$). The smaller γ, the better the improvement is. This is illustrated in Fig. 6 where we plot the channel capacity for nonpersistent CSMA versus γ for various packet lengths (both expressed in units of τ, the propagation delay). For larger T, CSMA provides relatively high channel capacity and thus leaves little margin for improvement; but for small T (e.g., $T = 10$ times the maximum propagation delay on the broadcast bus), the relative improvement is more important (it is about 16 percent when γ is just equal to the round trip delay). We also illustrate the improvement due to collision detection by showing packet delay versus γ for fixed channel throughput S in Fig. 7. Here, the higher the throughput is, the more significant is the improvement.

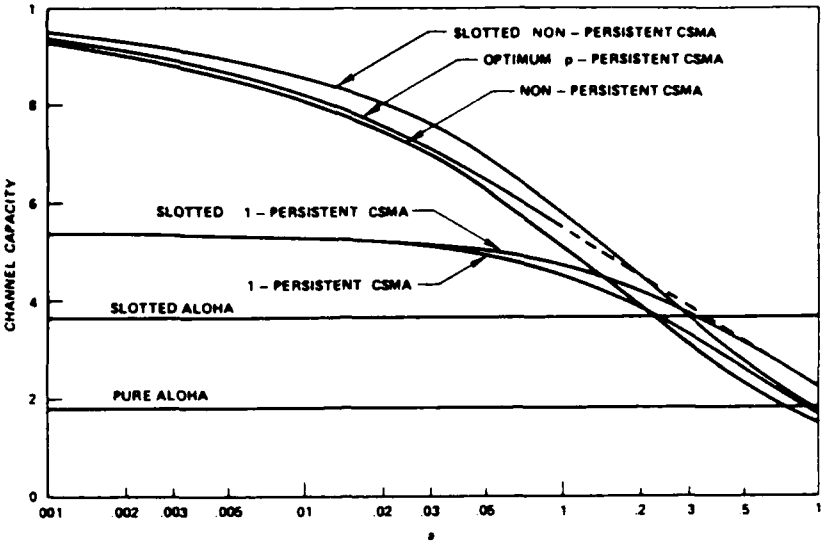


Fig. 5. CSMA and ALOHA: effect of propagation delay a on channel capacity [24].

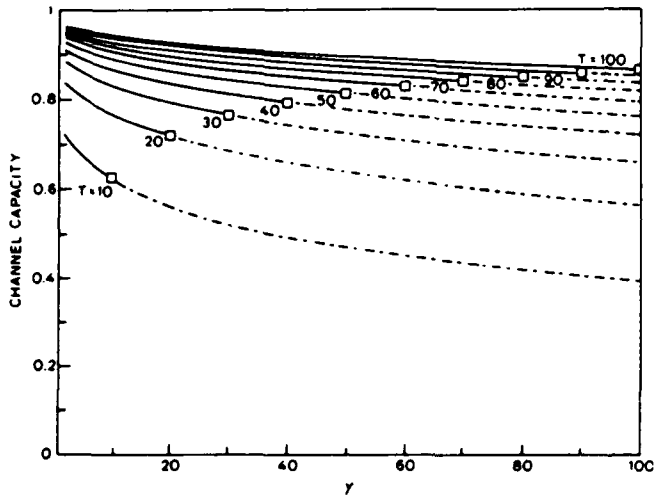


Fig. 6. Channel capacity versus γ in nonpersistent CSMA-CD. T = packet transmission time [26].

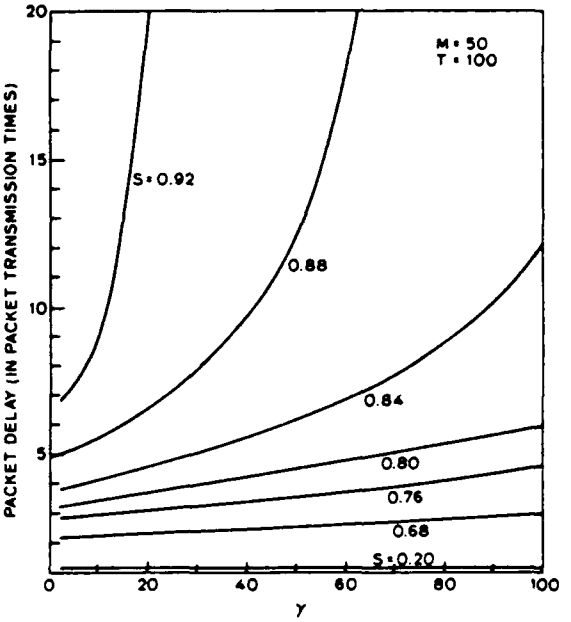


Fig. 7. Packet delay in nonpersistent CSMA-CD at fixed throughput versus collision recovery time γ [26].

The results displayed in the above figures have two important assumed conditions, namely 1) acknowledgments are instantaneous, always received correctly and for free (i.e., do not occupy any channel time), and 2) all devices are within range and in line-of-sight of each other so that sensing of all transmissions on the channel is perfect. While Condition 1) is relevant to both ALOHA and CSMA, Condition 2) is mostly relevant to CSMA. We discuss these issues in the following.

4) *Acknowledgment Procedures and Their Effect*: Basically, errors in multiaccess radio channels are due to two major causes: 1) random noise on the radio channel and 2) multiuse interference in the form of overlapping packets. A very reliable method ensuring the integrity of the transmitted data, is the use of an error detecting (e.g., cyclic) block code in conjunction with a positive acknowledgment of each correctly received message. Each packet contains a field for the cyclic checksum in its header. Each receiver responds to a complete packet addressed to it with a correct checksum by having the destination device transmit an acknowledgment packet back to the originating terminal. This acknowledgment contains (among other things) the unique identification of the originating terminal along with a checksum to ensure the integrity of the acknowledgment packet itself.

It is all too evident that acknowledgments will use part of the total available bandwidth (our limited resource). The amount of overhead introduced, as well as the degradation in delay incurred, varies with the mode of operation. When the available bandwidth is provided as a single channel to be shared by both information and acknowledgment packets, then the channel performance will further suffer from interference between information packets and acknowledgment packets unless some kind of priority scheme is provided.

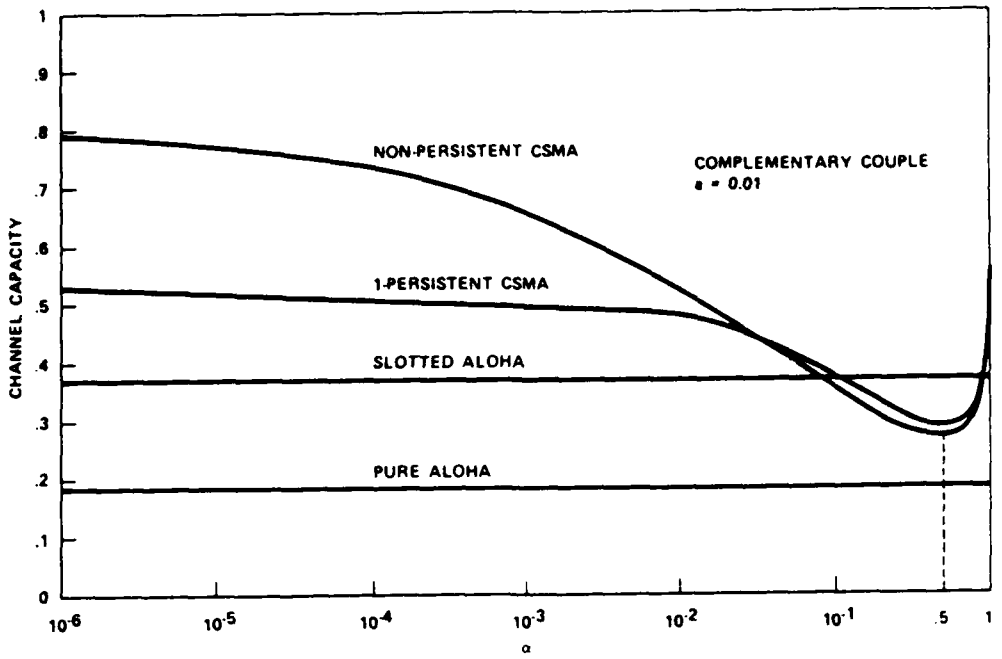


Fig. 8. Complementary couple configuration: channel capacity versus α, the relative sizes of the two decoupled populations [28].

The degradation in channel capacity due to the overhead created by the error control traffic has been studied in [30]. It has been shown that, in a common-channel configuration with nonpriority acknowledgment traffic, the channel capacity of slotted ALOHA drops to 14 percent of the channel bandwidth. However, if by some means acknowledgment traffic can be given priority so as to guarantee its transmission free-of-conflict, then the channel capacity for slotted ALOHA can be maintained at around 26 percent (assuming here that an acknowledgment packet uses an entire slot). The effect of acknowledgment traffic on CSMA channels need not be as dramatic since it is very simple to implement schemes which give priority to acknowledgment packets. One mode of operation is as follows [30]:

1) If a terminal, with a packet ready for transmission, senses the channel idle, then the terminal transmits its packet τ seconds (the propagation delay) later if and only if the channel is still sensed idle.

2) If such a terminal senses the channel busy, then it follows the protocol in question (nonpersistent, 1-persistent,...) repeating step 1) whenever the channel is sensed idle.

3) All acknowledgment packets are transmitted immediately, without incurring the τ seconds delay.

The capacity of the nonpersistent CSMA protocol with priority acknowledgment and $a = 0.01$ drops gradually from 0.85 to about 0.45 as the acknowledgment packet size increases from 0 to a full packet size.

5) *The Hidden Terminal Problem in CSMA and the Busy-Tone Multiple Access (BTMA) [28]*: We now relax the assumption that all users are in line-of-sight and within range of each other. Typically, two terminals can be within range of the intended receiver, but out-of-range of each other or separated by some physical obstacle opaque to radio signals. The existence of hidden terminals in a radio environment significantly degrades the performance of CSMA. To illustrate this effect, consider a population of users, each of which is communicating with a central station. This station is in line of sight communication with the entire population, but this population is divided into two groups (of relative sizes α and $1 - \alpha$) such that the radio connectivity exists only between users in the same group. Fig. 8 displays the CSMA channel capacity versus α, showing that the channel capacity drops drastically as α increases from 0 and reaches a minimum at $\alpha = 0.5$ [28].

Fortunately, in environments where all users communicate with a single central station such as in the ALOHA system, the hidden-terminal problem can be eliminated by frequency dividing the available bandwidth into two separate channels: a busy-tone channel and a message channel, thus giving rise to so called *busy-tone multiple access* (BTMA). The operation of BTMA rests on the fact that, by definition, there exists a central station which is within range and in line of sight of all users. As long as the central station senses carrier on the message channel it transmits a (sine wave) busy-tone signal on the busy-tone channel. It is by sensing carrier on the busy-tone channel that the users' terminals determine the state of the message channel. The action that a terminal takes pertaining to the transmission of the packet is again prescribed by the particular protocol being used.

In CSMA, the difficulty of detecting the presence of a signal on the message channel when this message occupies the entire bandwidth is minor and is therefore neglected. This is not realistic when we are concerned with the (statistical) detection of the (sine wave) busy tone signal on a narrow-band channel. In BTMA, the system's design involves a more complex set of system variables, namely the window detection time, the false alarm probability F, and the fraction of bandwidth devoted to the busy-tone signal. For a detailed analysis

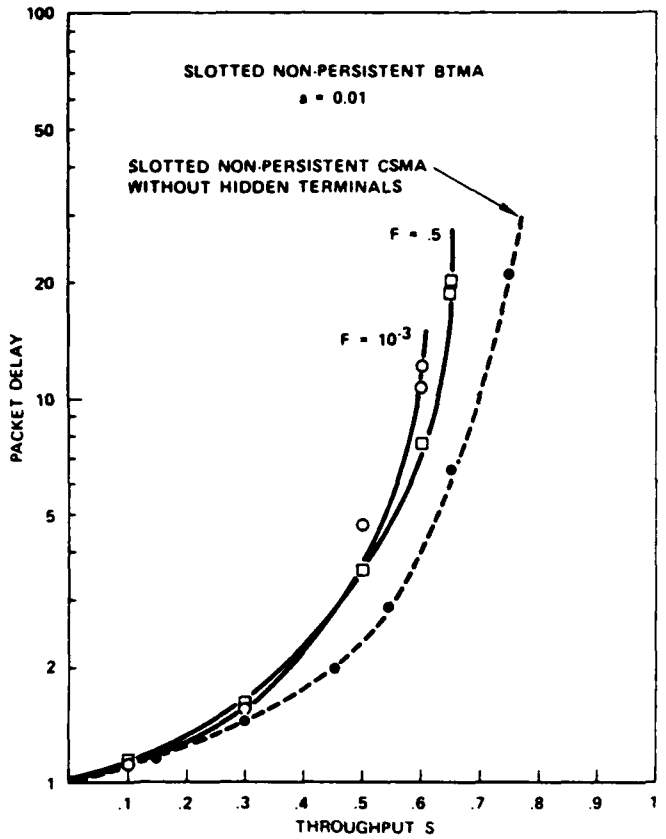


Fig. 9. BTMA: throughput-delay tradeoffs (propagation delay a = 0.01, F = false alarm probability) [28].

of this scheme, the reader is referred to [28]. Fig. 9 displays the throughput-delay tradeoff for BTMA in comparison to CSMA with no hidden terminals, showing the relatively good performance of BTMA.

6) *Dynamic Behavior and Dynamic Control of Random Access Schemes*: The performance results reported upon above were based on renewal theory and probabilistic arguments, assuming that steady-state conditions exist. If one examines in more detail the (S, G) relationships displayed above, one can see that the steady-state may not exist because of an inherent instability of these random-access techniques. This instability is simply explained by the fact that statistical fluctuations in the offered traffic increase the level of mutual interference among transmissions which in turn increases total G which increases the frequency of collisions, and so forth. Such positive feedback causes the throughput to decrease to very low values. Extensive simulation runs performed on a slotted ALOHA channel with an *infinite population* of users have indeed shown that the assumption of channel equilibrium is not strictly speaking valid; in fact after some finite period of quasi-stationary conditions, the channel will drift into saturation with probability one [31]. Thus a more accurate measure of channel performance must reflect the tradeoffs among stability, throughput, and delay. To that effect, Markov models have been formulated to analyze slotted ALOHA and CSMA when M interactive users contend for the channel [31]-[34]. These models permit one not only to derive analytic expressions for the average throughput-delay performance, but also to understand the dynamic behavior of these systems. In particular, it was observed that even in a finite population environment, if the retransmission delay is not sufficiently large, then the stationary performance attained is significantly degraded (low throughput, very high delay), so that, for all practical purposes, the channel is said to have failed; it is then called an unstable channel. With an infinite population, stationary conditions just do not exist; the channel is always unstable, thus confirming the results obtained from simulation, as just discussed. For unstable channels, Kleinrock and Lam [32] defined a stability measure which consists of the average time the system takes, starting from an empty channel, to reach a state determined to be critical. In fact, this critical state partitions the state space into two regions: a safe region, and an unsafe region in which the tendency is towards degraded performance. The stability measure is the average first exit time (FET) into the unsafe region. As long as the system operates in the safe region, the channel performance is acceptable; but then, of course, it is only usable over a finite period of time with an average equal to FET. For more details concerning the determination of FET and the numerical results, the reader is referred to [32], [33].

In the above discussion, it was furthermore assumed that the system parameters were all fixed, time invariant, and state-independent. These systems are referred to as *static*. It is often advantageous to design systems that dynamically adapt to time-varying input and to system state changes, thus providing improved performance. Dynamic adaptability is achieved via dynamic control consisting of time and state dependent parameters. The basic problem then is to find the control functions which provide the best system performance. Markov decision theory has successfully been applied by Lam and Kleinrock to the design and analysis of control procedures suitable to slotted ALOHA in particular and random-access techniques in general [35]. Two main types of control are proposed: an *input* control procedure (ICP) consisting of either accepting or rejecting all new packets generated in the current slot, and a *retransmission* control procedure (RCP) consisting of selecting a retransmission delay; in both cases the action taken is a function of the current system state, defined as the number of active users with outstanding packets. In order to implement such control schemes, each channel user must individually estimate the channel state by observing the channel outcome over some period of time. The control is of a distributed nature, as there is no central station monitoring and broadcasting state information or control actions. In the context of slotted ALOHA, Lam and Kleinrock give some heuristic control-estimation algorithms which prove to be very satisfactory [35]. With appropriate modification and extensions, these algorithms can be applied to CSMA channels as well. These algorithms are best suited to fully connected single-hop type environments. The dynamic control problem in multihop environments is more complex and little progress has yet been made in this area.

7) *Capture*: In the preceding discussions it was assumed that whenever two packet transmissions overlap in time, these packets destroy each other. This assumption is pessi-

mistic as it neglects *capture* effects in radio channels. Capture can be defined as the ability for a receiver to successfully receive a packet (with nonzero probability) although it is partially or totally overlapped by another packet transmission. Capture is mainly due to a discrepancy in receive power between two signals allowing the receiver to correctly receive the stronger; both distance and transmit power contribute to this discrepancy. Clearly capture improves the overall network performance, and, by the means of adaptive transmit power control, it allows one to achieve either fairness to all users, or intentional discrimination. Some of these effects have been addressed in [27], [36].

8) *Spread Spectrum Multiple Access (SSMA)*: Spread spectrum multiple access (SSMA) is the most common form of CDMA whereby each user is assigned a particular code sequence which is modulated on the carrier with the digital data modulated on top of that. Two common forms exist: the frequency-hopped SSMA and the phase-coded SSMA. In the former, as its name indicates, the frequency is periodically changing according to some known pattern; in the latter the carrier is phase modulated by the digital data sequence and the code sequence. SSMA has many applications: it is useful in satellite communications, mobile ground-radio, and computer communication networks [37]. In [14], Kahn *et al.* addressed many of the issues concerning the use of SSMA in packet radio systems. Security, coexistence with other systems, and ability to counteract the effects of multipath are key factors contributing to the choice of SSMA in the PRNET; however one main point of interest in this presentation is the benefit of capture in asynchronous SSMA. Even when several users employ the same code, the effect of interference is minimized by the "capture effect," defined here as the ability of the receiver to "lock on" one packet while all other overlapping packets appear as noise. The receiver locks on a packet by correctly receiving the preamble appended in the front of the transmitted packet. As long as the preamble of different packets do not overlap in time, and the signal strength of the late packets is not too high, capture of the earliest packet can be guaranteed with a high probability. In essence SSMA allows a packet to be captured at the receiver, while CSMA allows a user to capture the channel. CSMA can still be used in conjunction with SSMA. This mode will have the benefit of keeping away all users within hearing distance of the transmitter and thus help keep the capture effect and antijamming capability of the system at the desired level. For a complete discussion of all these issues, the reader is referred to [14].

C Centrally Controlled Demand Assignment

We have so far discussed the two extremes in the bandwidth allocation spectrum as far as control over the user's access right is concerned: the tight fixed assignment which has the most rigid control, is nonadaptive to dynamically varying demand, and can be wasteful of capacity if small-delay constraints are to be met; and random access which involves no control, is simple to implement, is adaptive to varying demand, but which, in some situations, can be wasteful of capacity due to collisions. In this and the following subsections, we examine demand assignment techniques which require that explicit information regarding the need for the communication resource be exchanged. We distinguish those demand assignments which are controlled by a central scheduler from those which employ a distributed algorithm executed by all users. We address centrally controlled assignments in the present subsection.

1) *Circuit Oriented Systems*: In these systems, the bandwidth is divided into FDMA or TDMA subchannels which are assigned on demand. The satellite SPADE system, for example, has a pool of FDMA subchannels which get allocated on request [38]. It uses one subchannel operated in a TDMA fashion with one slot per frame permanently assigned to each user to handle the requests and releases of FDMA circuits. Intelsat's MAT-1 system uses the TDMA approach [39]. TDMA subchannels are periodically reallocated to meet the varying needs of earth stations.

The Advanced Mobile Phone Service (AMPS), recently introduced by Bell Laboratories, is yet another example of a centrally controlled FDMA system [40]. The uniqueness of this system, however, lies in an efficient management of the spectrum based on space division multiple access (SDMA). That is, each subchannel in the pool of FDMA channels is allocated to different users in separate geographical areas, thus considerably increasing the spectrum utilization. To accomplish space division, the AMPS system has a cellular structure and uses a centralized handoff procedure (executed by a central office) which reroutes the telephone connections to other available subchannels as the mobile users move from one cell to another.

Given the significant setup times required in allocating subchannels, the above systems are attractive only when applications have stream-type traffic. When traffic is bursty, we again turn to packet-oriented systems, such as in the following.

2) *Polling Systems*: In packet oriented systems, polling is one of two modes used to centrally control access to the communication bandwidth, again provided as a single high-speed channel. A central controller sends polling messages to the terminals, one-by-one, asking the polled terminal to transmit. For this the station may have a polling list giving the order in which the terminals are polled. If the polled terminal has something to transmit, it goes ahead; if not, a negative reply (or absence of reply) is received by the controller, which then polls the next terminal in sequence. Polling requires this constant exchange of control messages between the controller and the terminals, and is efficient only if 1) the round-trip propagation delay is small, 2) the overhead due to polling messages is low, and 3) the user population is not a large bursty one. Polling has been analyzed by Konheim and Meister [41], and their analysis has been applied to the environment of M users sharing a radio channel in [23]. Denoting by L the ratio of the data message length to the polling message length, and by a the ratio of propagation delay to message transmission time, Fig. 10 displays numerical results corresponding to some typical values of L and a. These curves show that indeed as the population size increases, thus containing more and more bursty users, the performance of polling degrades significantly. Channel utilization can reach

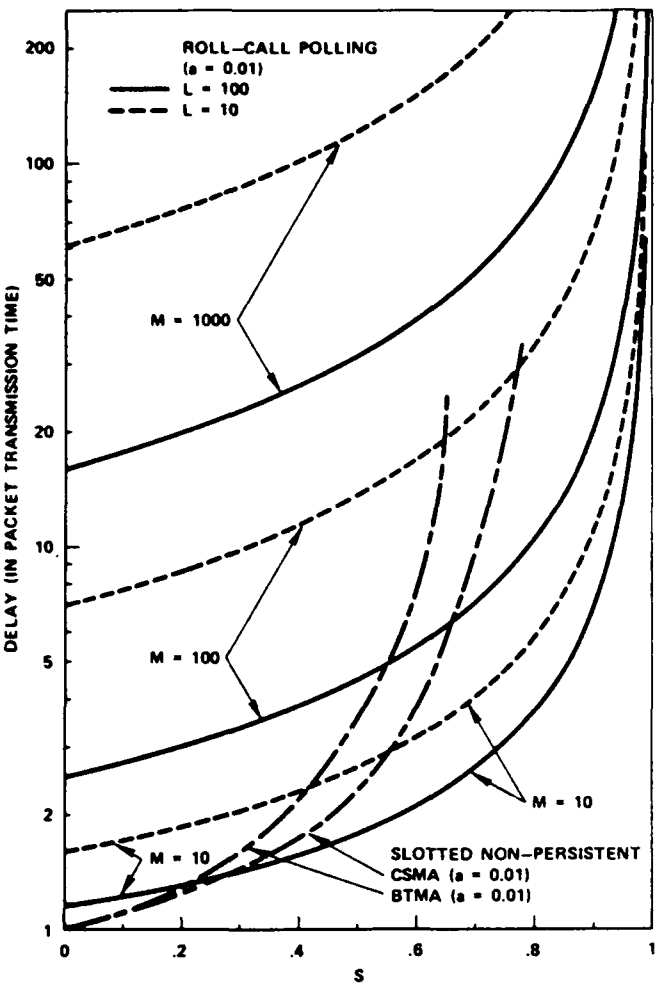


Fig. 10. Packet delay in roll-call polling. L = ratio of data message length to polling message length, a = normalized propagation delay, M = number of stations [23].

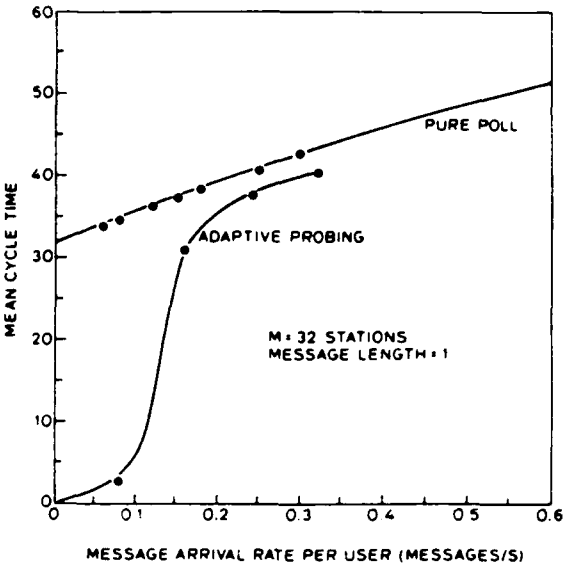


Fig. 11. Polling and adaptive probing: mean cycle time versus message arrival rate (simulation results--32 stations) [42].

100 percent of the channel bandwidth if the terminals are allowed to empty their buffers when they are polled. But as a result, the variance of packet delay can become intolerably large.

3) *Adaptive Polling or Probing [42]*: The primary limitation of polling in lightly loaded systems is the high overhead incurred in determining which of the terminals have messages. In order to decrease this overhead, a modified polling technique, based on a tree searching algorithm, and referred to as *probing*, has been proposed [42]. This technique assumes that the central controller can *broadcast* signals to all terminals. First the controller interrogates all terminals asking if any of them has a message to transmit, and repeats this question until some terminals respond by putting a signal on the line. When a positive response is received, the central station breaks down the population into subsets (according to some tree structure) and repeats the question to each of the subsets. This can be performed simply, for example by using binary addresses for the terminals and by transmitting as probing signal the common prefix of the addresses of a group of terminals. The process is continued until the terminals having messages are identified. When a single terminal is interrogated, it transmits its message.

Assume that the number of terminals is a power of 2, say $M = 2^n$. Let a cycle be recursively defined as the time required for the polling and transmission of all messages that were generated in the preceding cycle. If a single terminal has a message to transmit, probing requires $2n + 1$ inquiries per cycle as opposed to 2^n for conventional polling; but if all terminals have messages, probing requires $2^{n+1} - 1$ inquiries as opposed to 2^n for conventional pollings. To avoid incurring such a penalty when the system is heavily loaded, the probing technique can be made adaptive whereby the controller starts a cycle by probing smaller groups as the probability of terminals having messages increases. In particular, the group size may be considered a function of the duration of the immediately preceding polling cycle. Simulation of the adaptive probing technique has shown that this scheme is always superior to polling in that its mean cycle time is always smaller than that of polling. Fig. 11 displays the mean cycle time (obtained from simulation) as a function of the message arrival rate for both polling and probing [42]. Reference [42] did not provide any results concerning message delay, but it is intuitively clear that the smaller the mean cycle time is, the lower is the average delay.

4) *Split-Channel Reservation Multiple Access (SRMA) [23]*: An attractive alternative to polling is the use of explicit reservation techniques. In dynamic reservation systems, it is the terminal which makes a request for service on some channel whenever it has a message to transmit. The central scheduler manages a queue of requests and informs the terminal of its allocated time.

Since the channel is the only means of communication among terminals, the main problem here is, once again, how to communicate the request to the central scheduler. The contention on the channel of these request packets is of exactly the same nature as the contention of the data packets themselves. Fixed assignment and random access techniques suggest themselves, but it is clear from previous results that random access modes for multiplexing the requests on the channel would be

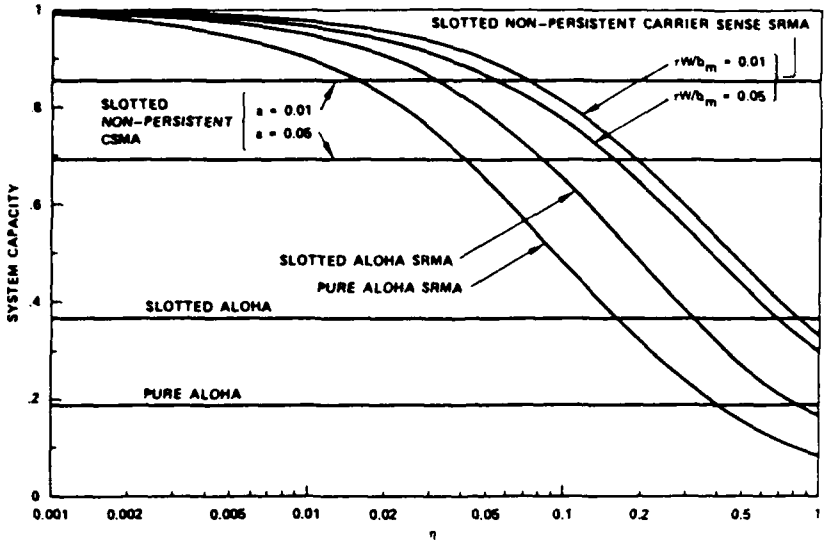


Fig. 12. SRMA: channel capacity versus η, ratio of request packet length to data packet length (normalized propagation delay of 0.01 and 0.05) [23].

more efficient. Furthermore, in order to prevent collisions between the requests and the actual message packets, the available bandwidth is either time divided or frequency divided between the two types of data. In the split-channel reservation multiple access (SRMA) scheme, frequency division of a ground radio channel is considered [23]. The available bandwidth is divided into two channels: one used to transmit control information, the second used for the data messages themselves. With this configuration, there are many operational modes. In the request/answer-to-request/message scheme (RAM), the bandwidth allocated for control is further divided into two channels: the request channel and the answer-to-request channel. The request channel is operated in a random access mode (ALOHA or CSMA). Upon correct reception of the request packet, the scheduling station computes the time at which the backlog on the message channel will empty and transmits an answer packet back to the terminal, on the answer-to-request channel, containing the address of the terminal and the time at which it can start transmission. Another version of SRMA, called the RM scheme, consists of having only two channels: the request channel and the message channel. When correctly received by the scheduling station, the request packet joins the request queue. Requests may be serviced on a "first-come first-served" basis (or any other scheduling algorithm). When the message channel is available, an answer packet (containing the ID of a queued terminal scheduled for transmission) is transmitted by the station on the message channel. After hearing its own ID repeated by the station, the terminal starts transmitting its message on the message channel. If a terminal does not hear its own ID repeated by the scheduling station within a certain appropriate time after the request is sent, the original transmission of the request packet is assumed to be unsuccessful. The request packet is then retransmitted.

We now examine the performance of SRMA. Let η denote the ratio of request packet length to data packet length, this representing a measure of the overhead due to control information. In Fig. 12 we plot the (RAM) SRMA system capacity versus η for the following access modes: pure ALOHA SRMA, slotted ALOHA SRMA, and slotted nonpersistent carrier sense SRMA. In addition, we show the system capacity for both ALOHA and CSMA. We note that the system capacity in SRMA reaches 1 for very small η. Typical values for η fall in the range (0.01, 0.1). Fig. 12 shows that a high improvement is gained when the request channel is operated in slotted nonpersistent CSMA as compared to ALOHA. The delay for ALOHA-SRMA and slotted nonpersistent carrier sense SRMA (normalized to b_m/W, where W denotes again the total channel bandwidth, and b_m is the number of bits per packet) is shown in Fig. 13 as a function of S for various values of η. We again note an important improvement in using CSMA for the request channel. Finally, in Fig. 14 we compare carrier sense SRMA with the random access modes ALOHA, CSMA, BTMA, and $M/D/1$, the perfect scheduling with fixed size packets and Poisson sources. We note that unless η is large (0.1 and above), there is a value of S below which CSMA or BTMA performs better than SRMA and above which the opposite is true.

5) *Global Scheduling Multiple Access (GSMA) [43]*: GSMA is a conflict-free reservation multiaccess scheme suitable for a high-speed data bus, which is based on the time-division concept for reservation. Here too a scheduler oversees all scheduling tasks. The users, all connected to the same line, listen for scheduling assignments and transmit in accordance with the slot allocation initiated by the scheduler. The channel time is divided into frames (of variable lengths). A frame is partitioned into two subframes: a subframe of status slots statically assigned to the users (in a fixed TDMA mode) to request data slot allocation, and a subframe of data slots, each sufficient to transmit a data packet of P bits. The fixed assignment of the status slots removes the need to transmit users' ID's and thus reduces the size of these slots. In each frame, a user can be allocated a number of data slots which does not exceed the number of packets generated at the user during the preceding frame or a maximum number specified, whichever is smaller. As a consequence

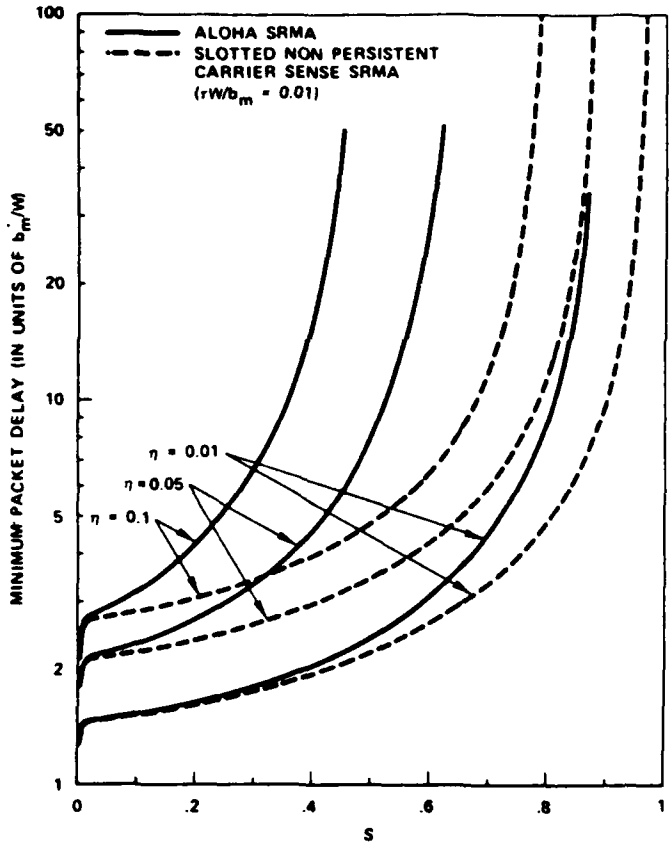


Fig. 13. Packet delay in SRMA (normalized propagation delay = 0.01) [23].

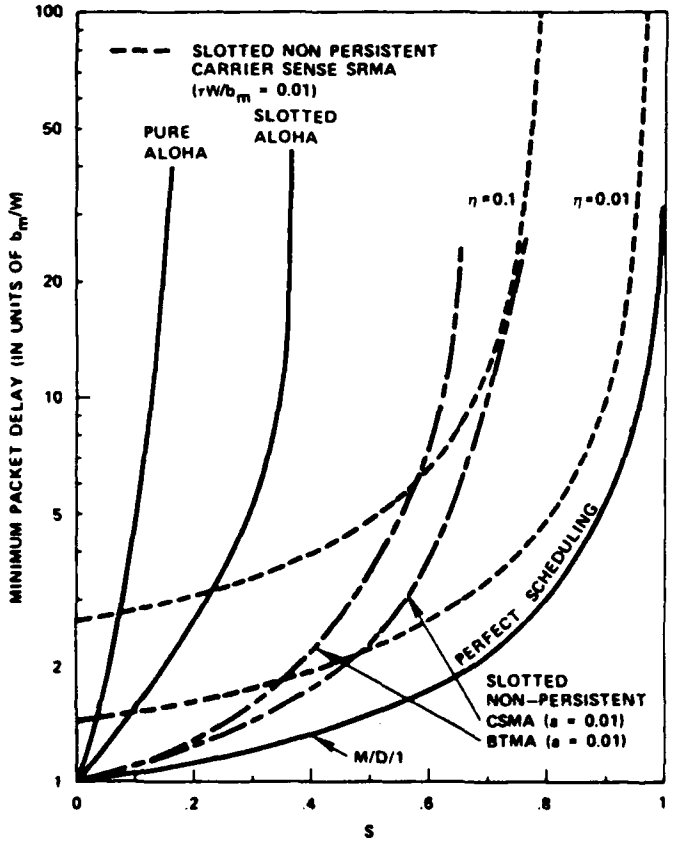


Fig. 14. Comparison of various schemes (parameters defined as in Figs. 12 and 13) [23].

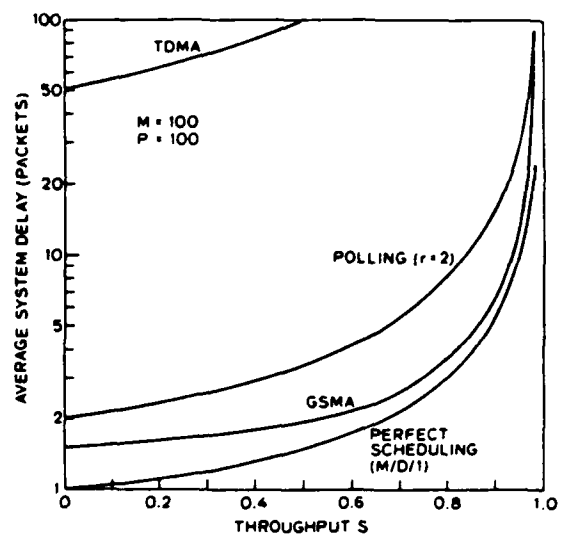


Fig. 15. GSMA: throughput delay performance (M = number of stations, P = number of bits per data packet) [43].

each active user is guaranteed at least one slot per frame. Fig. 15 displays the performance of GSMA (with $P = 100$ and number of stations $M = 100$) in comparison to polling (for some typical parameter values regarding the polling overhead r) and $M/D/1$ (the perfect scheduling). This illustrates some improvement gained in GSMA over polling [43].

D. Demand Assignment with Distributed Control

There are two reasons why distributed control is desirable. The first is *reliability*; with distributed control the system is not dependent on the proper operation of a central scheduler. The second is improved *performance*, especially when dealing with systems with long propagation delays, such as those using satellite channels. Indeed, if an earth station were to play the role of a scheduler, the minimum packet delay in a packet reservation scheme would be three times the round-trip propagation delay. (Of course, this can be decreased if on-board processing is available.) With distributed control, this minimum delay can be brought down to twice the round-trip delay or less without affecting the bandwidth utilization. Clearly, in slotted ALOHA, the best random access scheme available for satellite channels, the minimum packet delay is exactly one round-trip delay; but this is guaranteed only for a channel utilization approaching zero! In fact, the inherent long propagation delay in satellite channels is really the nasty characteristic that makes this environment "more distributed" than the single-hop ground radio or local area environments. In the latter, we have seen that efficient random access schemes, such as CSMA, are available; and the shorter the propagation delay, the better the CSMA performance. With zero propagation delay, collisions in CSMA can be completely avoided and CSMA's performance then corresponds to that of an $M/D/1$ queue,[1] the best we can achieve under random demand. In fact, as observed in [44], when the propagation

[1] This corresponsdence applies to CSMA with fixed size packets and Poisson sources.

delay is zero we no longer have a distributed environment, and the cost of creating a common queue disappears.

The basic element underlying all distributed algorithms is the need to exchange control information among the users, either explicitly or implicitly. Using this information, all users then execute independently the same algorithm resulting in some coordination in their actions. Clearly, it is essential that all users receive the same information regarding the demand placed on the channel and its usage in order to achieve a global optimum, and thus distributed algorithms are most attractive in fully connected systems. This attribute is not always present in ground radio environments, but certainly exists in satellite environments due to their inherent broadcast nature.[2] The long-delay/broadcast combination of attributes has been one of the reasons why many distributed control algorithms have been proposed in the context of satellite environments. We examine in this subsection distributed control algorithms suitable for each of our three environments (satellite, ground radio and local area), starting with satellite channels.

1) *Reservation-ALOHA [45]*: Reservation-ALOHA for a satellite channel is based on a slotted time axis, where the slots are organized into frames of equal size. The duration of a frame must be greater than the satellite propagation delay. A user who has successfully accessed a slot in a frame is guaranteed access to the same slot in the succeeding frame and this continues until the user stops using it. "Unused" slots, however, are free to be accessed by all users in a slotted ALOHA contention mode. An unused slot in the *current* frame is a slot which, in the *preceding* frame, either was idle or contained a collision. (Note again the effect of long delays on the control procedure.) Users need to simply maintain a history of the usage of each slot for just one frame duration. Since no request is explicitly issued by the user, this scheme has been referred to as an *implicit reservation* scheme. Clearly Reservation-ALOHA is effective only if the users generate stream type traffic or long multipacket messages. Its performance will degrade significantly with single packet messages, as every time a packet is successful the corresponding slot in the following frame is likely to remain empty.

2) *A First-in First-out (FIFO) Reservation Scheme [46]*: In this scheme, reservations are made explicitly. Time division is used to provide a reservation subchannel. The channel time is slotted as before, but every so often a slot is divided into V small slots which are used for the transmission of reservation packets (as well as possibly acknowledgments and small data packets); these packets contend on the V small slots in a slotted ALOHA mode. All other slots are data slots and are used on a reservation basis, free of conflict. The frequency of occurrence of reservation slots can be made adaptive to the load on the channel and the need to make new reservations. This adaptivity can be achieved as a result of the time-division of bandwidth allocation between reservations and data packets.

[2] This is valid unless the satellite uses spot beams, in which case we may lose on the connectivity requirement but gain the benefits of space division multiple access (SDMA).

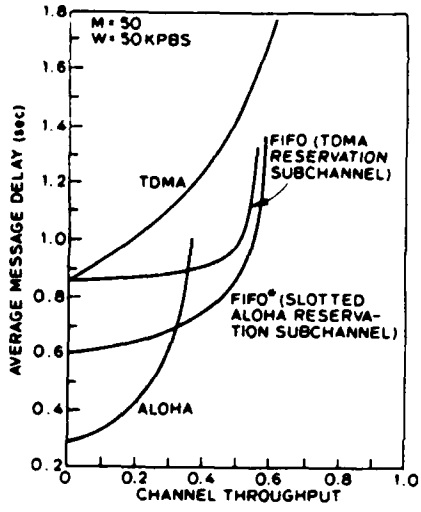


Fig. 16. Slotted ALOHA, TDMA, and FIFO reservation: delay throughput tradeoff for 50 users and single-packet messages in a satellite environment [66].

To execute the reservation mechanism properly, each station must maintain information on the number of outstanding reservations (the "queue in the sky") and the slots at which its own reservations begin. These are determined by the FIFO discipline based on the successful reservations received. Each successful reservation can accommodate up to a design maximum of, say, eight packets, thus preventing stations from acquiring exclusivity of the channel for long periods of time. To maintain synchronization of control information at the proper time, and to acquire the correct count of packets in the queue if out-of-sync conditions do occur, each station sends, in its data packet, information regarding the status of its queue. This information is also used by new stations which need to join the queue. The robustness of this system is achieved by a proper encoding of the reservation packets to increase the probability of their correct reception at *all* stations. Furthermore, to limit the effect of errors, a station reacquires synchronization if it detects a collision in one of its reserved slots or an error in a reservation packet.

Fig. 16 compares the throughput-delay tradeoff of the FIFO reservation scheme (operated with either a TDMA or a slotted ALOHA reservation subchannel) to that of TDMA and slotted ALOHA [66]. FIFO-Reservation offers delay improvements over TDMA. When compared to ALOHA, we note that higher system capacity is achieved but at the expense of a higher delay at low channel throughputs (due to a higher overhead).

3) *A Round-Robin (RR) Reservation Scheme [47]*: The basis of this scheme is fixed TDMA assignment, but with the major difference that "unused" slots are assigned to the active stations on a round-robin basis. This is accomplished by organizing packet slots into equal size frames of duration greater than the propagation delay and such that the number of slots in a frame is larger than the number of stations. One slot in each frame is permanently assigned to each station. To allow other stations to know the current state (used or unused) of its own slot, each station is required to transmit

information regarding its own queue of packets piggybacked in the data packet header (transmitted in the previous frame.) A zero count indicates that the slot in question is free. All stations maintain a table of all stations' queue lengths, allowing them to allocate among themselves free unassigned slots in the current frame. Round-robin is the discipline proposed by Binder [47], but other scheduling disciplines can be used as well. A station recovers its slot by deliberately causing a conflict in that slot which other users detect. For a station which was previously idle, initial acquisition of queue information is required and is achieved by having one of the stations transmit its table at various times. However, it is interesting to note that in this scheme, while acquiring queue synchronization, a station can always reclaim and use its own assigned slot.

The above three schemes have been proposed for satellite channels. All assumed fixed size slots, and thus can be implemented in systems which have been built for synchronous TDMA. The effect of large propagation delay is important. Framing is used in two of the schemes to deal with it, with the frame duration being equal to or longer than the propagation delay. Due to their dynamic nature, these protocols perform better than synchronous TDMA. However, when compared to random access (namely ALOHA here), they offer higher capacity, but also higher delay at low throughput. If used in systems with small propagation delay, such as ground radio, then they will perform significantly better, and are expected to have a performance comparable to SRMA. In fact, due to the inherent small propagation delay in ground radio environments, other access modes with distributed control are also possible if all devices are in line-of-sight and within range of each other. We describe these in the following.

4) *Minislotted Alternating Priorities (MSAP) [48]*: MSAP is a conflict-free multiple access scheme suitable for a *small* number of data users. In essence, MSAP is a "carrier-sense" version of polling with distributed control. The time axis is slotted with the minislot size again equal to the maximum propagation delay. All users are synchronized and may start transmission only at the beginning of a minislot. Users are considered to be ordered from 1 to M. When a packet transmission ends, the alternating priorities (AP) rule assigns the channel to the same user who transmitted the last packet (say user i) if he is still busy; otherwise the channel is assigned to the next user in sequence (i.e., user (i, mod M + 1)). The latter (and all other users) detects the end of transmission of user i by sensing the absence of carrier over one minislot. At this new point in time, either user (i mod M + 1) starts transmission of a packet (which will be detected by all other users) or he is idle in which case a minislot is lost and control of the channel is handed to the next user in sequence. The overhead at each poll in this scheme is simply one minislot.

Scheduling rules other than AP are also possible, namely round-robin (RR) or random order (RO). MSAP, however, exhibits the least overhead incurred in switching control between users. On the other hand, MSRR may be more suitable to environments with unbalanced traffic since then smaller users will be guaranteed more frequent access than with MSAP. These scheduling rules have also appeared in the literature as BRAM, the broadcast recognizing access method. For details, see [72].

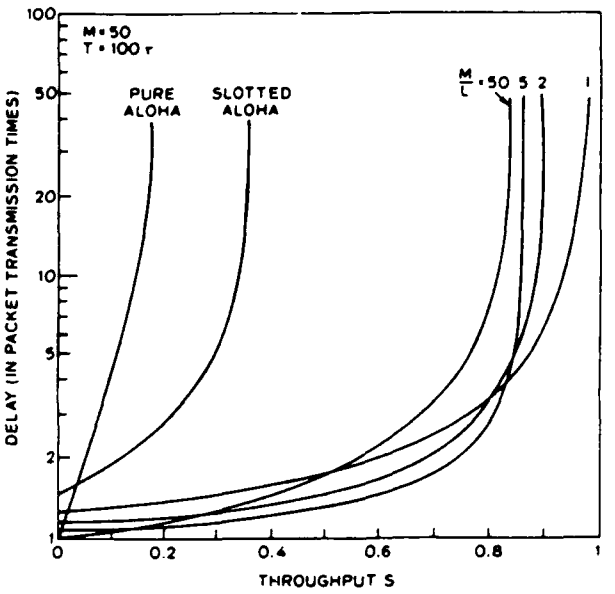


Fig. 17. Assigned-slot listen-before transmission protocol: throughput delay tradeoff for 50 users and $T = 100$ (propagation time $a = 0.01$) [49].

5) *The Assigned-Slot Listen-Before-Transmission Protocol [49]*: MSAP, being a "carrier sense" version of polling, behaves like polling. In particular, as the system load decreases, the overhead incurred in locating a nonidle user increases, and so does the delay. The assigned-slot listen-before-transmission protocol has been proposed to improve on MSAP by allowing several users to share common minislots. In such a case, there exists a tradeoff between the time wasted in collisions, and the time wasted in control overhead. Time is divided into frames, each containing an equal number of minislots (say, L). To each minislot of a frame is assigned a given subset of M/L users. A user with a packet ready for transmission in a frame can sense the channel only in his assigned minislot. If the channel is sensed idle, transmission takes place; if not, the packet is rescheduled for transmission in a future frame. A packet transmission spans T slots. The parameter M/L is adjusted according to the load placed on the channel. For high throughput, $M/L = 1$ is found to be optimum. In fact, with $M/L = 1$, the scheme becomes a conflict-free one which approaches MSAP and gives nearly identical results [49]. For very low throughput, $M/L = M$ (i.e., $L = 1$) is found to be optimum; this corresponds to pure CSMA. In between the two extreme cases intermediate values of M/L are optimum. Fig. 17 displays the throughput-delay performance of this scheme for various values of M/L when $M = 50$ and $T = 100$. It also shows how this scheme (and thus, MSAP) compare to CSMA.

6) *Distributed Tree Retransmission Algorithms in Packet Broadcast Channels [71]*: In many of the multiaccess protocols examined above, conflict resolution is achieved by retransmitting randomly in the future. Such a rescheduling discipline in slotted ALOHA achieves a 36 percent bandwidth utilization, but exhibits some sort of instability unless the rescheduling is controlled, as discussed in Section II-B. Tree algorithms

are based on the observation that a contention among several active sources is completely resolved if and only if all the sources are somehow subdivided into groups such that each group contains at most one active source. (Such observation is similar to that made in the probing technique discussed in Section II-C3). In its simplest form, the tree algorithm consists of the following. Each source corresponds to a leaf on a *binary* tree. The channel time axis is slotted and the slots are grouped into pairs. Each slot in a pair corresponds to one of the two subtrees of the node being visited. Starting with the root node of the tree, we let all terminals in each of the two subtrees of the root transmit in their corresponding slot. If any of the two slots contains a collision, then the algorithm proceeds to the root of the subtree corresponding to the collision and repeats itself. This continues until all the leaves are separated into sets such that each of them contains at most one packet. This is known to all users, as the outcome of the channel is either a successful transmission or an idle slot. Collisions caused by the left subtree (1st slot of a pair) are resolved prior to resolving collisions in the right subtree. This scheme provides a maximum throughput of 0.347 packets/slot, and all moments of the delay are finite if the aggregate packet arrival rate is less than 1/3 packets/slot [71].

Clearly, a binary tree is not always optimum. If, each time we return to the root node, we allow the tree to be reconfigured according to the current traffic conditions, it can be shown that the optimum tree is binary everywhere except for the root node whose optimum degree depends on traffic conditions [71]. The dynamic scheme achieves a throughput of 0.430 packets/slot, and all the moments of the delay are finite for $\lambda < 0.430$ packets/slot. Tree algorithms are implementable in both ground radio and satellite channels as long as the broadcast capability is available.

7) *Distributed Control Algorithms in Local Area Networks*: In addition to the random access schemes described previously, the above two algorithms are also applicable to local area (broadcast) *bus* networks as these exhibit the required characteristics of small propagation delay and full connectivity. But in local area communication, a slightly different topology, namely the *ring* (or loop), has also been widely considered. In the ring topology, messages are not broadcast but rather passed from node to node along unidirectional links, until they return to the originating node. A simple scheme suitable for a ring consists of passing the access right sequentially from node to node around the ring. (Note that in a ring, the physical locations of the nodes define a natural ordering among them.) One implementation of this scheme is exemplified by the Distributed Computing System's network where an 8-bit *control token* is passed sequentially around the ring [50]. Any node with a ready message may, upon receiving the control token, remove the token from the ring, send the message, and then pass on the control token. Another implementation consists of providing a number of *message slots* which are continuously transmitted around the ring. A message slot may be empty or full; a node with a ready message waits to see an empty slot pass by, marks it as full, and uses it to send its message [51]-[53]. A still different strategy is known as the *register insertion* technique [3], [54], [55]. Here a message to be transmitted is first loaded into a shift register. If the ring is idle, the shift register is just transmitted. If not, the register is inserted into the network loop at the next point separating two adjacent messages: the message to be sent is shifted out onto the ring while an incoming message is shifted into the register. The shift register can be removed from the network loop when the transmitted message has returned to it. The insertion of a register has the effect of increasing the transport delay of messages on the ring.

E. Adaptive Strategies and Mixed Modes

We have so far examined quite a large number of multiaccess schemes and compared their performance. One thing is clear: each of these schemes has its advantages and limitations. No one scheme performs better than all others over the entire range of system throughput (except, of course, the hypothetical perfect scheduling, which is clearly unachievable in a distributed environment). If a scheme performs nearly as well as perfect scheduling at low input rates, then it is plagued by a limited achievable channel capacity. Conversely, if a scheme is efficient when the system utilization is high, the overhead accompanying the access control mechanism becomes prohibitively large at low utilization. Although some characteristics of a system (propagation delay, channel speed, etc.) are unlikely to vary during operation, it is certain that the load placed upon the system will be time varying. In the case of a single subscriber type (say with periodic traffic, stream-type traffic, or bursty traffic) the volume of the traffic may be varying; if several subscriber types are simultaneously present, the volume of traffic introduced by each, and therefore the proportional mix of traffic types, may also be time varying.

We have discussed at several points in this paper the dynamic control of a specific access scheme which improved its performance to a certain extent; but such an adaptive control did not change the nature of the access scheme nor the nature of its limitation. Dynamically controlled random access schemes provide improved packet delay over uncontrolled versions, but still exhibit channel capacity less than 1. The adaptive polling technique decreased the overhead at low throughput but only to a certain extent. Actually, what one really needs is a strategy for choosing an access mode which is itself adaptive to the varying need so that optimality is maintained at all times. Clearly, in order to accomplish adaptivity, a certain amount of information is needed by the distributed decision makers. The type and amount of information required by an adaptive strategy, as well as the implementation of the information acquisition mechanism are among the most crucial factors in determining the performance and robustness of the strategy. A great deal of effort has been spent in recent years on such adaptive strategies. We devote this subsection to schemes which fall into this category.

1) *The URN Scheme [56]*: We start with this more recent scheme because of its simplicity, elegance, and the smoothness by which it adapts to varying loads. It has been proposed in the context of a ground radio fully connected environment in which the time axis is divided into packet slots, and all

users are synchronized. Assuming that all users know the exact number n of busy users, the scheme consists of giving full access right (i.e., the right to transmit with probability 1) to some number k of users. A successful transmission will result if there is exactly one busy user among these k. The probability of such an event is maximized when $k = [M/n]$, where $[M/n]$ denotes the integer part of M/n. This is in contrast to the controlled slotted ALOHA scheme where all users are given the same partial access right: the right to transmit with probability $p = 1/n$. Assume the system is lightly loaded (for instance $n = 1$); a large number of users are given access right (in the example $n = 1$, the number is $k = M$), but only a few and hopefully only one will make use of it (in the example $n = 1$, a successful transmission takes place). As the load increases, k decreases and the access right is gradually restricted. For the extreme case of $n = M$, $k = 1$ and the scheme converges to TDMA. If the sampling of k is random, the urn scheme converges to random TDMA; if the sampling of k is without repetitions from slot to slot until all users have been sampled once, the urn scheme converges to round-robin TDMA.

Two important questions remain: how to estimate n, and how to reach a consensus on who the k users are. In [56], Kleinrock and Yemini offer a few alternatives. One possible scheme for estimating n with good accuracy is to include a single reservation minislot at the beginning of each data slot. An idle user who turns busy sends a standard reservation message of few bits. All users are able to detect the following three events: no new busy users, one new busy user, and more than one new busy user (termed an erasure). As it is impossible with this minimal overhead to estimate the exact number of new busy users when the latter is greater than one, errors in estimation result; however analysis and simulation have shown that this error is negligible, and furthermore that the scheme is insensitive to small perturbations in n. This last statement is even more important with respect to the robustness of the scheme since it means that all users need not have exactly the same estimate for n. As for coordinating the selection of the k users, an effective mechanism is the use of synchronized pseudorandom generators at all users which allow them to draw the same k pseudorandom numbers. Another mechanism, referred to as a round-robin slot sharing window mechanism, consists of having a window of size k move over the population space. When a collision occurs, the window stops and decreases in size. When there is no collision, the tail of the window is advanced to the head of the previous window, and the size is again set to k as determined by n.

The improvement obtained by this scheme over slotted ALOHA and TDMA can be seen in Fig. 18 where the throughput-delay performance of all these schemes is displayed for a pupulation size $M = 10$ [56].

2) *Another Adaptive Strategy for the Dynamic Management of Packet Radio Slots [57]*: Another way to achieve adaptivity is as follows. The time axis is again slotted with the slot size equal to a packet transmission time. Slots are grouped into k equivalence classes or subchannels. Slots are

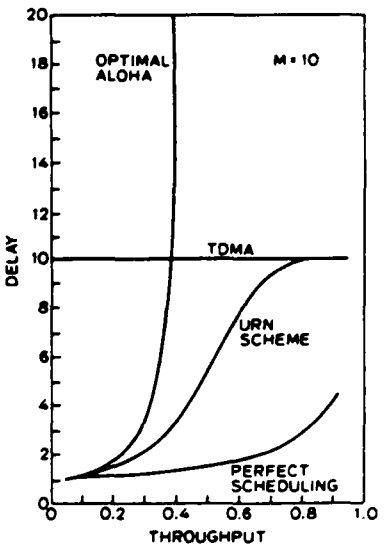


Fig. 18. Throughput-delay performance for the URN scheme (example for 10 users) [56].

furthermore grouped into frames of m slots, $m \geq k$, each containing at least one slot for every equivalence class. Let M be again the number of users. Each user is at any one time assigned to one of the k equivalence classes. All stations in a given class use a random access mode to access slots assigned to their class. If CSMA is used as the contention scheme, then time slots are minislots of size τ, assigned to the k equivalence classes just as before. By dynamically varying the size of the frame and the assignment of slots within the frame to classes of users, one can vary the access mode to best fit the situation. At low load, for example, choosing $k = m = 1$ with all users in the same class leads to a pure random access mode of low delay. Choosing $k = m = M$ with each user constituting a separate class leads to TDMA. Increasing the parameter k has the effect of decreasing the rate of collisions among users of the same class. The frame size m can be used to allow a smooth changeover between the schemes. By partitioning the frame into two subframes, both contention and pure TDMA can coexist simultaneously. The information used in adapting to the situation is the collision rate and the rate of empty slots (or minislots) for the randomly accessed slots, and the rate of empty slots for the TDMA assigned slots. For example, when one minislot of a TDMA slot goes empty, the remainder of the TDMA slot may be cancelled and reassigned to some other groups (then to be used via CSMA).

Schemes other than CSMA and TDMA can be combined by this adaptive strategy. One may, for example, mix CSMA with MSRR. In [57], Ricart and Agrawala studied, via simulation, some typical adaptation algorithms of this type. Some of their simulation results for a CSMA/TDMA combination are shown in Fig. 19. These results exhibit clearly the improvement gained over the entire throughput range by using the adaptive strategy.

3) *The Reservation upon Collision Schemes (RUC) [58]*: In the Reservation upon Collision schemes, the channel time is divided into slots of fixed length which in turn are divided

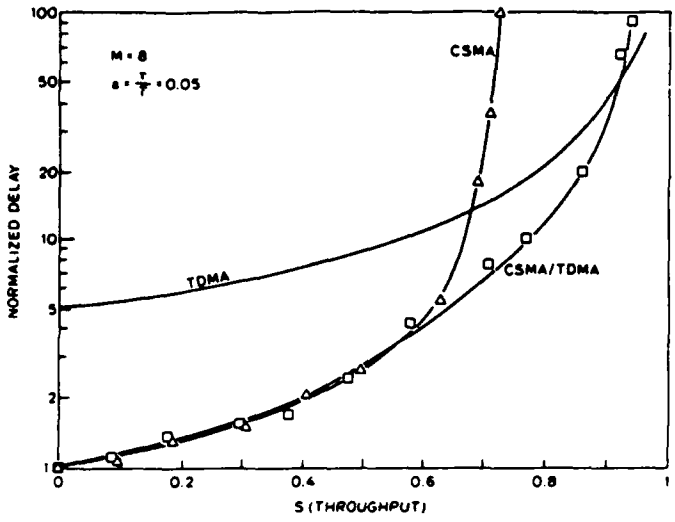


Fig. 19. Simulation results for an adaptive CSMA/TDMA strategy (eight stations, normalized propagation delay of $a = 0.05$) [57].

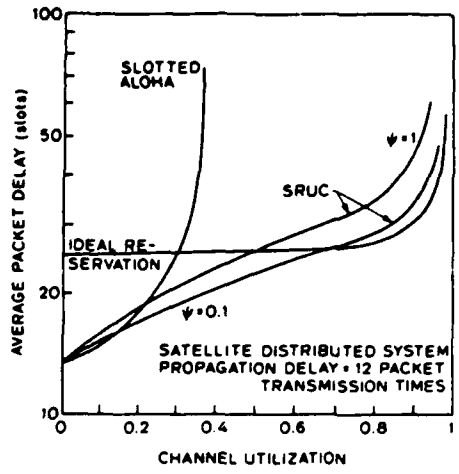


Fig. 20. Split reservation upon collision: throughput delay performance at zero overhead $\Psi = 0.1$ and $\Psi = 1$ [58].

into two parts: a data-subslot SSO for the transmission of information packets and a subslot SS1 for the transmission of (signaling) information regarding the transmitting user(s). The data subchannel can be in one of two states: the contention state or the reserved state. It is normally in the contention state and users can access the slots in a slotted ALOHA mode as long as no collisions occur. When a collision is detected, then the data subchannel switches to the reserved state and remains in that state until the queue of reservations is cleared, at which time it switches back to the contention state. That is, if a collision is detected, reservations are automatically implied for the colliding users. To accomplish this, the signaling information identifying the users must be received by all users free of interference, and thus an ingenious use of the SS1 subslots must be devised. CDMA and TDMA have been proposed in [58]. When the number of users is large, a particularly suitable approach is to consider grouping the slots into a frame of, say, L slots. Each of the L SS1 subslots is assigned to a group of size M/L users instead of M users, thus decreasing the degree of multiplexing signaling information over the SS1 subslots. TDMA or CDMA still needs to be used. In this approach, users need not transmit their identification as this is implied from the position of the SS1 subslot. However, each user has to send the number of packets transmitted in the frame, and this information requires at most $\log_2(L + 1)$ bits. This scheme is referred to as the split reservation upon collision (SRUC).

Fig. 20 shows the performance of SRUC in a satellite environment as compared to slotted ALOHA and pure reservation for two values of the overhead Ψ required per frame for the signaling information. Clearly, this performance degrades as Ψ increases. More detailed results can be found in [58].

Since slotted ALOHA and reservations are both suitable for satellite channels, RUC schemes are also particularly suitable for these as well as ground radio channels.

4) *Priority-Oriented Demand Assignment (PODA) [12]*: In the context of a satellite channel, PODA has been proposed as the ultimate scheme which attempts to incorporate all the properties and advantages seen in many of the previous schemes. It has provision for both implicit and explicit reservations, thus accommodating both stream and packet-type traffic. It may also integrate the use of both centralized and distributed control techniques thus achieving a high level of robustness.

Channel time is divided into two basic subframes, an information subframe and a control subframe. The information subframe contains scheduled packets and packet streams, which also contain, piggybacked, control information such as reservations and acknowledgments. The control subframe is used exclusively to send reservations that cannot be sent in the information subframe in a timely manner. In order to achieve integration of centralized and distributed assignments, the information subframe is further divided into two sections, one for each type.

Access to the control subframe (which is divided into slots accommodating fixed size control packets) can take any form that is suitable to the environment. It can be by *fixed assignment* (TDMA) if the number of stations is small (giving rise to the so-called FPODA), or by *contention* as in ALOHA if the stations have a low-duty cycle (giving rise to CPODA), or a combination of both. The boundary between the control subframe and the information subframe is not fixed, but varies with the demand placed on the channel. As in the FIFO and RR reservation schemes, distributed control is achieved by having all stations involved in this type of control keep track of their queue length information. Priority scheduling can thus be achieved. For stream traffic, a reservation is made only once, and is retained by each station in a stream queue. Centralized assignment may be used when delay is not the crucial element. This scheme has been proposed in the context of a satellite channel but may be applied to other environments as well.

5) *More on Mixed Modes*: Other studies have appeared in the literature that also deal with integrating several different access modes into the same system.

The *Mixed ALOHA Carrier Sense* (MACS) scheme consists of allowing a large user to steal, by carrier sensing, slots which are unused by a large population of small users accessing the channel in a slotted ALOHA mode [59]. Analysis has shown that the total channel utilization is significantly increased with MACS, and the throughput-delay performance of both the large user and the background ALOHA users is better with MACS than with a split-channel configuration in which the large user and the ALOHA users are each permanently assigned a portion of the channel [59].

Group Random Access (GRA) procedures consist of using only certain channel time-periods to allow some network terminals to transmit their information-bearing packets on a random access basis. The channel can then be utilized at other times to grant access to other terminals or other message types, by applying, as appropriate, group random access, reservation procedure or fixed assignment. The idea is simply a fixed time-division assignment among groups utilizing different access schemes. For analysis of GRA, the reader is referred to [60], [61].

Finally, we consider satellite systems with on-board processing capability. These have recently received increased attention and are being considered as a means to increase the capacity of packet satellite channels [62]-[65]. One example is typified by the integration of slotted ALOHA on several uplink channels, with TDMA on one or several downlink channels. The on-board processing capability is used to filter out all collisions and thus improve the utilization of the downlink channels. The overall spectrum efficiency is also improved especially if the ratio of uplink channels to downlink channels is properly chosen. Analysis of these disciplines is given in [62], [63]. Additional improvement over these disciplines is possible by providing buffering capability on board the satellite to smooth the input and more completely fill the downlink channels.

III. CONCLUSION

Tremendous advances have been made in recent years in devising multiaccess schemes suitable to a variety of data communication environments. In this paper, we have briefly reviewed a large number of these protocols which we have grouped into five categories according to: 1) the degree of control exercised over the users' access; 2) the (centralized or distributed) nature of the decision-making process; and 3) the degree of adaptivity of the algorithm to the changing need. We have seen that these link level protocols have a great impact on the utilization of the communication resource in particular and the overall system performance in general. We have also briefly discussed their suitability to various traffic characteristics.

Although an attempt has been made to render the presentation complete, it is by no means exhaustive of all existing schemes, and the field is still so wide open that new schemes are constantly being introduced. Throughout the paper, an emphasis was placed on that class of packet communications that service very many bursty users, since this has been a major concern for many years. It is important, however, to note that there is a growing interest in the support of applications which lend themselves to stream-type traffic (such as packetized voice, facsimile, video data for remote conferencing, etc.) and which may also require real-time communications service on the part of the network. Moreover, with an even greater interest in integrating the many different applications onto the same network structure, it is becoming important to devise multiaccess protocols which can provide all the capabilities and features required for this integration. The adaptive strategies discussed in the paper provide an attempt at solving this problem but it is still far from being completely resolved.

Another point of great importance is the impact that these link level protocols have on the design of higher level protocols. Indeed, due to the basically different nature and behavior of some of these multiaccess schemes, one is faced with the necessity to find new ways to deal with many of the higher level functions. The routing problem in store-and-forward multiaccess/broadcast systems, for example, is significantly different from the well-known routing algorithms devised for point-to-point store-and-forward networks; here the transmitted packet should carry, at each transmission, the next node's address, and each *receiving* node has to decide as to whether to relay or ignore the packet. A discussion of routing schemes appropriate to these systems can be found in [14]. Clearly, in single-hop broadcast systems, and in local area ring architectures, the routing problem is absent.

Acknowledgment procedures may also have to be handled differently in broadcast networks. In the PRNET, for example, hop-by-hop acknowledgments can be passive, in the sense that, due to the broadcast nature of transmission, the relaying of a packet over a hop constitutes the acknowledgment for the transmission over the previous hop. Acknowledgments may also be active in the sense that an acknowledgment packet is actually created and transmitted. If acknowledgment packets are given priority, the active acknowledgment procedure has the benefit of minimizing buffering requirements at the repeaters since the acknowledgments are sent at the earliest opportunity, and possibly minimizing channel overhead since the additional transmissions beyond success resulting from delayed acknowledgments can then be kept to a minimum [67]. (In fact, it was found that if acknowledgments were instantaneous, then a few buffers in each packet radio unit appear to be sufficient to handle the storage requirements; indicating that the system becomes more channel bound than storage bound [68], [69].) In satellite environments, PODA achieves the same objective by piggybacking, whenever possible, acknowledgments on pending reservation requests, which are heard by all users, including the sender.

To conclude, we can say that despite the many advances already accomplished, this area still presents many challenging open problems, and that to best make use of the progress already achieved in link-level protocols, one also needs to turn one's attention to the many unresolved issues concerning higher level protocols.

REFERENCES

[1] D. W. Davies, K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson, "A digital communication network for computers giving rapid response

at remote terminals," presented at ACM Symp. Operating System Principles, Gatlinburg, TN, Oct. 1–4, 1967.
[2] W. D. Farmer, and E. E. Newhall, "An experimental distributed switching system to handle bursty computer traffic," in *Proc. ACM Conf.*, Pine Mountain, GA, Oct. 1969.
[3] M. T. Liu and C. C. Reames, "Communication protocol and network operating system design for the distributed loop computer network (DLCN)," in *Proc. 4th Annu. Symp. Computer Architecture*, Mar. 1977, pp. 193–200.
[4] M. T. Liu, "Distributed loop computer networks," in *Advances in Computer Networks*, M. Rubinoff and M. C. Yovitts, Eds. New York: Academic, 1978.
[5] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Commun. Ass. Comput. Mach.*, vol. 19, pp. 395–403, 1976.
[6] L. Pouzin, "Presentation and major design aspects of the Cyclades computer network," presented at Datacom 73, ACM/IEEE, 3rd Data Commun. Symp., St. Petersburg, FL, Nov. 1973, pp. 80–87.
[7] L. G. Roberts and B. D. Wessler, "Computer network developments to achieve resource sharing," in *1970 Spring Joint Comput. Conf., Proc. AFIPS Conf.*, vol. 36, 1970, pp. 543–549.
[8] L. Pouzin, "CIGALE, The packet switching machine of the CYCLADES computer network," presented at IFIP Congress, Stockholm, Sweden, Aug. 1974, pp. 155–159.
[9] H. Opderbeck and R. B. Hovey, "Telenet—Network features and interface protocols," in *Proc. NTG-Conf. Data Networks*, Baden-Baden, West Germany, Feb. 1976.
[10] W. W. Clipshaw and F. Glave, "Datapac network review," in *Int. Comput. Commun. Conf. Proc.*, Aug. 1976, pp. 131–136.
[11] N. Abramson, "The Aloha system," in *Computer Communication Networks*, N. Abramson and F. Kuo, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1973.
[12] I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose packet satellite networks," *Proc. IEEE*, vol. 66, Nov. 1978.
[13] R. E. Kahn, "The organization of computer resources into a packet radio network," in *Nat. Comput. Conf., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1975, pp. 177–186; also in *IEEE Trans. Commun.*, vol. COM-25, Jan. 1977.
[14] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, Nov. 1978.
[15] D. Clark *et al.*, "An introduction to local area networks," *Proc. IEEE*, vol. 66, Nov. 1978.
[16] W. G. Schmidt, "Satellite time-division multiple access systems: Past, present and future," *Telecommun.*, vol. 7, pp. 21–24, Aug. 1974.
[17] I. Rubin, "Message delays in FDMA and TDMA communication channels," *IEEE Trans. Commun.*, to be published.
[18] S. Lam, "Delay analysis of time-division multiple access (TDMA) channel," *IEEE Trans. Commun.*, vol. COM-25, Dec. 1977.
[19] O. Kosovych, "Fixed assignment access technique," *IEEE Trans. Commun.*, vol. COM-26, Sept. 1978.
[20] N. Abramson, "The ALOHA system—Another alternative for computer communications," in *1970 Fall Joint Comput. Conf. AFIPS Conf. Proc.*, vol. 37. Montvale, NJ: AFIPS Press, 1970, pp. 281–285.
[21] L. G. Roberts, "ALOHA packet system with and without slots and capture," *Comput. Commun. Rev.*, vol. 5, pp. 28–42, Apr. 1975.
[22] L. Kleinrock and S. Lam, "Packet-switching in a slotted satellite channel," *Nat. Computer Conf., AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, 1973, pp. 703–710.
[23] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part III—Polling and (dynamic) split channel reservation multiple access," *IEEE Trans. Commun.* vol. COM-24, pp. 832–845, Aug. 1976.
[24] 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.*, vol. COM-23, pp. 1400–1416, Dec. 1975.
[25] F. Tobagi, "Random access techniques for data transmission over packet switched radio networks," Ph.D. dissertation, Comput. Sci. Dep., School of Eng. and Appl. Sci., Univ. California, Los Angeles, Rep. UCLA-ENG 7499, Dec. 1974.
[26] F. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," in *Proc. Local Area Commun. Network Symp.*, Boston, MA, May 1979; also Stanford Electronics Lab., Comput. Syst. Lab., Tech. Rep. 173, June 30, 1979.
[27] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Trans. Commun.*, vol. COM-25, pp. 117–128, Jan. 1977.
[28] F. 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.* vol. COM-23, pp. 1417–1433, Dec. 1975.
[29] F. A. Tobagi, M. Gerla, R. W. Peebles, and E. G. Manning, "Modeling and measurement techniques in packet communication networks," *Proc. IEEE*, vol. 66, pp. 1423–1447, Nov. 1978.
[30] F. Tobagi and L. Kleinrock, "The effect of acknowledgment traffic on the capacity of packet-switched radio channels," *IEEE Trans. Commun.*, vol. COM-26, pp. 815–826, June 1978.
[31] S. S. Lam, "Packet switching in a multi access broadcast channel with application to satellite communication in a computer network," Ph.D dissertation, Dep. Comput. Sci., Univ. California, Los Angeles, Mar 1974; also in Univ. California, Los Angeles, Tech. Rep. UCLA-ENG-7429, Apr. 1974.
[32] L. Kleinrock and S. S. Lam, "Packet switching in a multiaccess broadcast channel; Performance evaluation," *IEEE Trans. Commun.*, vol. COM-23, pp. 410–423, Apr. 1975.
[33] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-25, pp. 1103–1120, Oct. 1977.
[34] G. Fayolle, E. Gelembe, and J. Labetoulee, "Stability and optimal control of the packet-switching broadcast channels," *J. Ass. Comput. Mach.*, vol. 24, pp. 375–386, July 1977.
[35] S. S. Lam and L. Kleinrock, "Packet switching in a multiaccess broadcast channel: Dynamic control procedures," *IEEE Trans. Commun.*, vol. COM-23, pp. 891–904, Sept. 1975.
[36] J. Metzner, "On improving utilization in ALOHA networks," *IEEE Trans. Commun.*, vol. COM-24, Apr. 1976.
[37] Special Issue on Spread Spectrum Communications, *IEEE Trans. Commun.*, vol. COM-25, Aug. 1977.
[38] B. Edelson and A. Werth, "SPADE system progress and application," *COMSAT Tech. Rev.*, vol. 2, pp. 221–242, Spring 1972.
[39] W. Schmidt *et al.*, "Mat-1: INTELSAT's Experimental 700-channel TDMA/DA system," in *Proc. INTELSAT/IEEE Int. Conf. Digital Satellite Commun.*, Nov. 1969.
[40] N. Erlich, "The advanced mobile phone service," *IEEE Commun. Mag.*, vol. 17, Mar. 1979.
[41] A. G. Konheim and B. Meister, "Service in a loop system," *J. Ass. Comput. Mach.*, vol. 19, pp. 92–108, Jan. 1972.
[42] J. F. Hayes, "An adaptive technique for local distribution," *IEEE Trans. Commun.*, vol. COM-26, Aug. 1978.
[43] J. W. Mark, "Global scheduling approach to conflict-free multiaccess via a data bus," *IEEE Trans. Commun.*, vol. COM-26, Sept. 1978.
[44] L., Kleinrock, "Performance of distributed multiaccess computer communication systems," in *Proc. IFIP Congress*, 1977.
[45] W. R. Crowther, R. Rettbert, D. Walden, S. Ornstein, and F. Heart, "A system for broadcast communication: Reservation-ALOHA," in *Proc. 6th Hawaii Int. Syst. Sci. Conf.*, Jan. 1973.
[46] L. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *Proc. AFIPS Conf.*, vol. 42, June, 1973.
[47] R. Binder, "A dynamic packet switching system for satellite broadcast channels," in *Proc. ICC'75*, San Francisco, CA, June 1975.
[48] L. Kleinrock and M. Scholl, "Packet switching in radio channels: New conflict-free multiple access schemes for a small number of data users," in *ICC Conf. Proc.*, Chicago, IL, June 1977, pp. 22 1-105–22 1-111.
[49] L. W. Hansen, and M. Schwartz, "An assigned-slot listen-before-transmission protocol for a multiaccess data channel," *IEEE Trans. Commun.*, vol. COM-27, pp. 846–857, June 1979.
[50] D. C. Loomis, "Ring communication protocols," Univ. California, Dep. Inform. and Comput. Sci., Irvine, CA, Tech. Rep. 26, Jan. 1973.
[51] J. R. Pierce, "Network for block switching of data," *Bell Syst. Tech. J.*, vol. 51, pp. 1133–1143, July/Aug. 1972.
[52] A. Hopper, "Data ring at computer laboratory, University of Cambridge," *Computer Science and Technology: Local Area Networking*. Washington DC: Nat. Bur. Stand., NBS Special Publ. 500-31, Aug. 22-23, 1977, pp. 11-16.
[53] P. Zafiropoulo and E. H. Rothauser, "Signalling and frame structures in highly decentralized loop systems," *Proc. Int. Conf. on Comput. Commun.* (Washington, DC), IMB Res. Lab., Zurich, Switzerland, pp. 309–315.
[54] E. R. Hafner *et al.*, "A digital loop communication system," *IEEE Trans. Commun.*, p. 877, June 1974.
[55] M. V. Wilkes, "Communication using a digital ring," in *Proc. PACNET Conf.*, Sendai, Japan, Aug. 1975, pp. 217–255.

[56] L. Kleinrock and Y. Yemini, "An optimal adaptive scheme for multiple access broadcast communication," *ICC Conf. Proc.*, Chicago, IL, June 1977.

[57] G. Ricart and A. Agrawala, "Dynamic management of packet radio slots," presented at *Third Berkeley Workshop on Distributed Data Management and Comput. Networks*, Aug. 1978.

[58] F. Borgonovo and L. Fratta, "SRUC: A technique for packet transmission on multiple access channels," in *Proc. Int. Conf. Comput. Commun.*, Kyoto, Japan, 1978.

[59] M. Scholl and L. Kleinrock, "On a mixed mode multiple access scheme for packet-switched radio channels," *IEEE Trans Commun.*, vol. COM-27, pp. 906–911, June 1979.

[60] I. Rubin, "A group random-access procedure for multi-access communication channels," in *NTC'77 Conf. Rec. Nat. Telecommun. Conf.*, Los Angeles, CA, Dec. 1977, pp. 12:5-1–12:5-7.

[61] ——, "Integrated random-access reservation schemes for multi-access communication channels," School Eng. Appl. Sci., Univ. California, Los Angeles, Tech. Rep. UCLA-ENG-7752, July 1977.

[62] J. K. DeRosa, and L. H. Ozarow, "Packet switching in a processing satellite," *Proc. IEEE*, vol. 66, pp. 100–102, Jan. 1978.

[63] R. E. Eaves, "ALOHA/TDM systems with multiple downlink capacities," *IEEE Trans. Commun.*, vol. COM-27, pp. 537–541, Mar. 1979.

[64] S. F. W. Ng and J. W. Mark, "A multiaccess model for packet switching with a satellite having some processing capability," *IEEE Trans. Commun.*, vol. COM-25, pp. 128–135, Jan. 1977.

[65] ——, "Multiaccess model for packet switching with a satellite having processing capability: Delay analysis," *IEEE Trans. Commun.*, vol. COM-26, pp. 283-290, Feb. 1978.

[66] S. S. Lam, "Satellite multi-access schemes for data traffic," in *Proc. Int. Conf. Commun.*, Chicago, IL, 1977, pp. 37.1-19–37.1-24.

[67] F. Tobagi *et al.*, "On measurement facilities in packet radio systems," in *Nat. Comput. Conf. Proc.*, New York, NY, June 1976.

[68] F. Tobagi, "Analysis of a two-hop centralized packet radio network: Part I—Slotted ALOHA," *IEEE Trans. Commun.*, vol. COM-28, pp. 196–207, Feb. 1980.

[69] ——, "Analysis of a two-hop centralized packet radio network: Part II—Carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-28, pp. 208–216, Feb. 1980.

[70] W. W. Chu, "A study of asynchronous time division multiplexing for time-sharing computer systems," in *1969 Spring Joint Comput. Conf. AFIPS Conf. Proc.*, vol. 35, 1969, pp. 669–678.

[71] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505–515, Sept. 1979.

[72] I. Chlamtac *et al.*, "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.*, vol. COM-27, pp. 1183–1190, Aug. 1979.

★

Fouad A. Tobagi (M'77), for a photograph and biography, see page 207 of the February 1980 issue of this TRANSACTIONS

Performance Analysis of Carrier Sense Multiple Access with Collision Detection*

Fouad A. Tobagi and V. Bruce Hunt **
Computer Systems Laboratory, Stanford University, Stanford, California, USA

Packet broadcasting in computer communication is attractive in that it combines the advantages of both packet-switching and broadcast communication. All stations share a common channel which is multi-accessed in some random fashion. Among the various random access schemes known, carrier sense multiple access (CSMA) has been shown to be highly efficient for environments with relatively short propagation delay. Packet broadcasting (and in particular CSMA) has been successfully applied to coaxial cables thus providing an efficient means for communication in local environments. In addition, in such environments the possibility of detecting collisions on the coaxial cable enhances the performance of CSMA by aborting conflicting transmissions, thus giving rise to the carrier sense multiple access schemes with collision detection (CSMA-CD). In this paper we extend an analysis of CSMA to accommodate collision detection. The analysis provides the throughput-delay performance of CSMA-CD and its dependence on such key system parameters as the average retransmission delay and the collision recovery time.

* This research was supported by the Advanced Research Projects Agency, Department of Defense, Contract No. MDA903-79-C-0201, Order No. AO3717, monitored by the office of Naval Research.

** V. Bruce Hunt is now with the Telecommunications Sciences Center SRI-International, 333 Ravenswood Avenue, Menlo Park, California 94025, USA.

Computer Networks 4 (1980) 245–259

1. Introduction

There are numerous reasons why advances in local area communication networks have significantly increased in the past few years. The recent interest in the application of the (now available) inexpensive processing power to office and industrial automation, the necessity for the sharing of expensive scarce resources, the need for local collection and dissemination of information, and the rising interest in distri-

Dr. Tobagi is Assistant Professor of Electrical Engineering, School of Engineering, Stanford University. His current research interests include computer communication networks, packet switching in ground radio and satellite networks, modeling and performance evaluation of computer communications systems.

From 1971 to 1974, he was with the University of California, Los Angeles, where he participated in the ARPA Network Project as a postgraduate research engineer and did research in packet radio communication. During the summer of 1972, be was with the Communications Systems Evaluation and Synthesis Group, IBM, J. Watson Research Center, Yorktown Heights, NY. From December 1974 to June 1978, he was a research staff project manager with the ARPA project at the Computer Science Department, UCLA, and engaged in the modeling, analysis and measurements of packet radio systems. In 1978, he joined the faculty at Stanford.

Dr. Tobagi received the engineering degree from Ecole Centrale des Arts et Manufactures, Paris, France, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1971 and 1974, respectively.

V. Bruce Hunt is employed by SRI International and is a graduate student at Stanford University. Previously, he was with Zilog Corporation where he designed, implemented and analyzed ARIEL, a microprocessor-based local network based on the principles employed by Ethernet (packet broadcasting CSMA-CD). His current research interests include distributed operating system design and local network analysis and design.

From 1972 to 1978 he was with the Stanford Center for Information Processing working on the multiprocessor system at the Stanford Linear Accelerator Center and the Stanford time sharing system. He was also the group leader for WILBUR of the Stanford time sharing system.

Mr. Hunt received a bachelor of science degree in mathematics from Harvey Mudd College. He was a California State Scholar from 1966 to 1970 and a Mayer Scholar in 1966. He is a member of A.C.M. and IEEE.

buted architectures for data processing are but a few examples.

Just as in any field, the development of local area computer communication systems is subject to a number of constraints. Simplicity, flexibility and reliability usually portray these constraints. The environments in question are generally characterized by a large and often variable number of devices requiring interconnection. Such environments call for networks with simple topologies and simple inexpensive connection interfaces which provide great flexibility to accommodate the variability in the environment, and which achieve the desired level of reliability.

Several architectures have been proposed which include MITRE'S Mitrix, Bell Telephone Laboratory's Spider, and U.C. Irvine's Distributed Computing System (DCS) [1–4]. Spider and DCS use a ring topology, while Mitrix uses two one-way busses implemented by CATV technology. As for system control, Mitrix and Spider use a central mini-computer for switching and bandwidth allocation, while DCS uses distributed control.

Another network architecture, based on the packet broadcasting technology and exemplified by Ethernet [5] appears to be a very effective solution in satisfying the above mentioned constraints. Packet broadcasting is attractive in that it combines the advantages of both packet switching and broadcast communication. Packet switching offers the efficient sharing of communication resources by many contending users with unpredictable demands; broadcast communication, whenever possible, eliminates complex topological design problems. Given that computer communication traffic is bursty in nature, it has been well established that it is more efficient to provide the available communication bandwidth as a single high-speed channel to be shared by the many contending users, thus attaining the benefits of the strong law of large numbers. This clearly results in a multiaccess environment that calls for schemes to control access to the channel, referred to as random access schemes. The earliest and simplest such scheme is the so-called pure-ALOHA, first used in the ALOHA-System [6]; unfortunately, pure-ALOHA provides a maximum channel utilization which does not exceed 18%. Another such scheme, carrier sense multiple access (CSMA), has been shown to be highly efficient in environments with propagation delays which are short compared to the packet transmission time [7]. In essence, CSMA reduces the level of interference caused by overlapping packets in the random multiaccess channel by allowing devices to sense carrier due to other users' transmissions, and inhibit transmission when the channel is in use. Packets which are inhibited or suffer a collision are rescheduled for transmission at a later time according to some rescheduling policy.

Ethernet is a local communication network which uses CSMA on a tapped coaxial cable to which all the communicating devices are connected. The device connection interface is a passive cable tap so that failure of an interface does not prevent communication among the remaining devices. The use of a single coaxial cable naturally achieves broadcast communication. Moreover, given the physical characteristics of data transmission on coaxial cables, in addition to sensing carrier, it is possible for Ethernet transceivers to detect interference among several transmissions (including their own) and abort transmission of their colliding packets. This produces a variation of CSMA which we refer to as carrier sense multiple access with collision detection (CSMA-CD). It is networks of the Ethernet type that we address in this paper.

CSMA in fully connected environments has been previously analyzed and its performance derived [7–10]. We extend here the analysis of CSMA to accommodate collision detection. This analysis provides the throughput-delay performance of CSMA-CD and its dependence on such key system parameters as the average rescheduling delay and collision recovery time. We furthermore characterize the improvement gained by CSMA-CD over CSMA for fixed and variable size packets.

The CSMA-CD schemes are described in section 2, followed by the analysis in section 3. Numerical results are discussed in section 4.

2. The CSMA-CD schemes

Carrier sense schemes require that each device with a packet ready for transmission senses the channel prior to transmission. A number of protocols exist which pertain to the action taken by the terminal after observing the state of the channel. In particular, a terminal never transmits when it senses that the channel is busy. Tobagi and Kleinrock described two such protocols in the context of ground radio channels [7,11]. They are the non-persistent CSMA and the p-persistent CSMA protocols. These protocols are extended here to environments in which the collision detection capability is available.

In the non-persistent CSMA-CD scheme, a terminal with a packet ready for transmission senses the channel and proceeds as follows.

1. If the channel is sensed idle, the terminal initiates transmission of the packet.
2. If the channel is sensed busy, then the terminal schedules the retransmission of the packet to some later time and repeats the algorithm.
3. If a collision is detected during transmission, the transmission is aborted and the packet is scheduled for retransmission at some later time. The terminal then repeats the algorithm.

In the 1-persistent CSMA-CD protocol (a special case of p-persistent CSMA), a terminal which finds the channel busy persists on transmitting as soon as the channel becomes free. Thus a ready terminal senses the channel and proceeds as in nonpersistent CSMA-CD, except that, when the channel is sensed busy, it monitors the channel until it is sensed idle and then with probability one initiates transmission of the packet.

The p-persistent protocol is an enhancement of the 1-persistent protocol by allowing ready terminals to randomize the start of transmission following the instant at which the channel goes idle. Thus a ready terminal senses the channel and proceeds as in the above schemes except that when the channel is sensed busy, the terminal persists until the channel is idle, and

(i) with probability p it initiates transmission of the packet
(ii) with probability $1 - p$ it delays transmission by τ seconds (the end-to-end propagation delay); if, at this new point in time, the channel is sensed idle, then the terminal repeats this process [steps (i) and (ii)], otherwise, it schedules retransmission of the packet to some later time.

Note that the p-persistent and non-persistent protocols become identical if the rescheduling delays are chosen for both protocols as an integer number of τ delay units geometrically distributed, with parameter p (the parameter in the p-persistent protocol). This follows because of the memoryless property of the geometric distribution. In this paper we analyze only the non-persistent and 1-persistent protocols.

In all CSMA-CD protocols, given that a transmission is initiated on an *empty* channel, it is clear that it takes at most one end-to-end propagation delay, τ, for the packet transmission to reach all devices, as depicted in fig. 1; beyond this time the channel is guaranteed to be sensed busy for as long as data transmission is in process [1]. A collision can occur only if another transmission is initiated before the current one is sensed, and it will then take, at most, one additional end-to-end delay before interference reaches all devices. (See fig. 1.) Let ξ denote the time it takes a device to detect interference once the latter has reached it. ξ depends on the implementation and can be as small as 1 bit transmission time, as is the case with Ethernet [5]. Furthermore, Ethernet has a collision consensus reenforcement mechanism by which a device, experiencing interference, jams the channel to ensure that all other interfering devices detect the collision. We denote by ζ the period used for collision consensus reenforcement. Given that a collision occurs, the time until all devices stop transmission, γ, is thus given by [2]

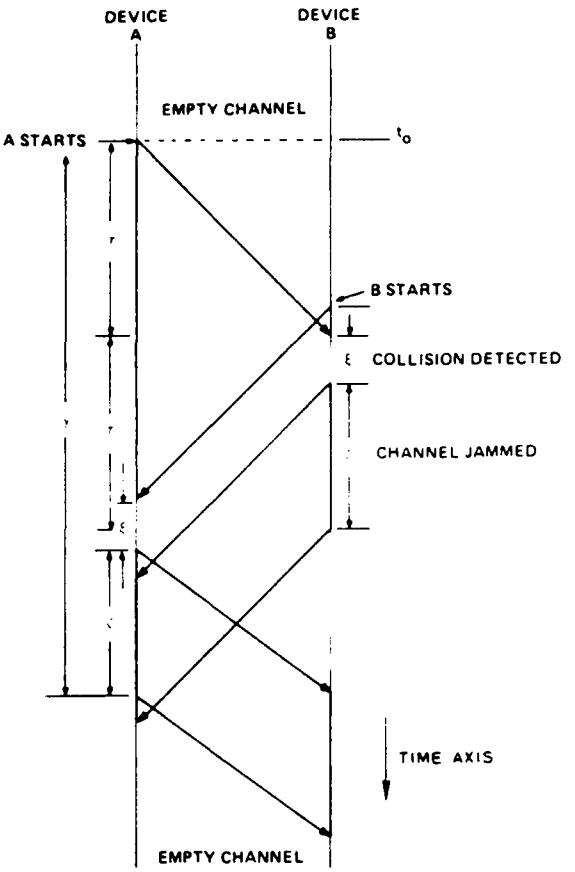


Fig. 1. Collision Detection and Recovery Time in CSMA-CD.

[1] We assume that the sensing operation is instantaneous on this (high-bandwidth) channel.

[2] This assumes that all interfering devices undertake the collision consensus reenforcement.

$\gamma = 2\tau + \xi + \zeta$.

The time until the channel is again sensed idle by all devices is clearly $\gamma + \tau$.

3. Analysis

We assume that the time axis is slotted where the slot size is the end-to-end propagation delay. For simplicity in analysis, we consider all devices to be synchronized and forced to start packet transmission only at the beginning of a slot. When a device becomes ready in some slot, it senses the channel during the slot and then operates according to the CSMA-CD protocols described above.

3.1. Analysis of CSMA-CD with fixed size packets

3.1.1. Channel capacity

As in previous analysis of random access schemes, channel capacity is obtained by considering an infinite population model which assumes that all devices collectively form an independent Poisson source, and that the average retransmission delay is arbitrarily large [7,10].

Consider first the non-persistent CSMA-CD protocol. We observe on the time axis an alternate sequence of transmission periods (successful or unsuccessful) and idle periods. A transmission period followed by an idle period is called a cycle (see fig. 2). With the infinite population assumption, all cycles are statistically identical. Let g denote the rate of devices becoming ready during a slot. Let T denote the transmission time (in slots) of a packet. A successful transmission period is of length $T + 1$ slots. In case of a collision, the length of a transmission period is $\gamma + 1$ slots. Given that the source is Poisson, the probability that a transmission is successful is $P_s = ge^{-g}/(1 - e^{-g})$; the average idle period is $\overline{I} = e^{-g}/(1 - e^{-g})$; the average transmission period is $\overline{TP} = P_s T + (1 - P_s)\gamma + 1$; and the throughput is given by

$$S = \frac{P_s T}{\overline{TP} + \overline{I}}$$

$$= \frac{Tg\,e^{-g}}{Tge^{-g} + (1 - e^{-g} - ge^{-g})\gamma + 1} \qquad (1)$$

The channel capacity is obtained by maximizing S with repect to g.

Consider now the 1-persistent CSMA-CD protocol. We observe on the time axis an alternate sequence of busy and idle periods, whereby a busy period is any collection of juxtaposed transmission periods surrounded by idle periods. A busy period followed by an idle period constitutes a cycle (see fig. 3). Again, all cycles are statistically identical. Let $\overline{B}$ denote the average duration of a busy period, $\overline{I}$ the average duration of an idle period, and $\overline{U}$ the average time during a cycle that the channel is carrying successful transmissions. The throughput is given by $S = \overline{U}/(\overline{B} + \overline{I})$. In this infinite population model, the success or failure of a transmission period in the busy period is only dependent on the preceding transmission period (and thus its length), except for the first transmission period of the busy period, which depends on arrivals in the preceding slot. Accordingly, given that a transmission period in the busy period is of length X($X = T + 1$ or $\gamma + 1$), the length of the remainder of the busy period is a function of X, and we let $B(X)$ denote its average. In the same manner we define $U(X)$. Let $q_i(X)$ be the probability that there are i arrivals in X slots. Under the Poisson assumption,

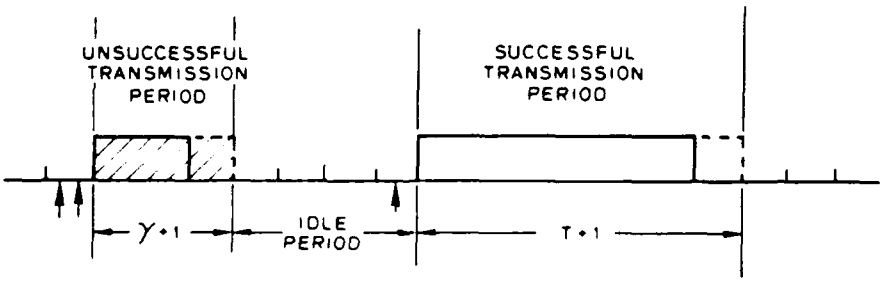


Fig. 2. Transmission and Idle Periods in Slotted Nonpersistent CSMA-CD. (Vertical arrows represent users becoming ready to transmit).

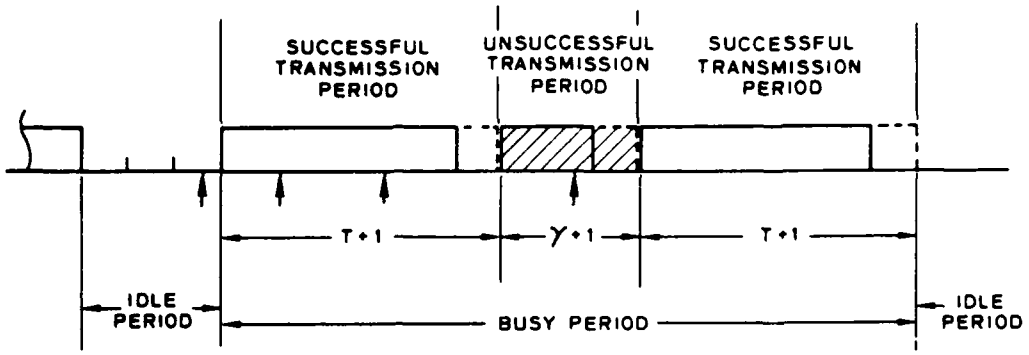


Fig. 3. Busy and Idle Periods in Slotted 1-persistent CSMA-CD. (Vertical arrows represent users becoming ready to transmit).

$q_i(X) = (gX)^i e^{-gX}/i!$. $B(X)$ is given by

$$B(X) = \frac{q_1(X)}{1 - q_0(X)} [T + 1 + (1 - q_0(T + 1)) B(T + 1)] + \left[1 - \frac{q_1(X)}{1 - q_0(X)}\right] \times [\gamma + 1 + (1 - q_0(\gamma + 1)) B(\gamma + 1)] . \quad (2)$$

Writing eq. (2) with $X = T + 1$ and $X = \gamma + 1$, we obtain two equations in the two unknowns, $B(T + 1)$ and $B(\gamma + 1)$. The average busy period $\bar{B}$ is then given by $B(1)$, expressed in terms of $B(T + 1)$ and $B(\gamma + 1)$.

Similarly, $U(X)$ is given by

$$U(X) = \frac{q_1(X)}{1 - q_0(X)} [T + (1 - q_0(T + 1)) U(T + 1)] + \left[1 - \frac{q_1(X)}{1 - q_0(X)}\right][(1 - q_0(\gamma + 1)) U(\gamma + 1)] . \quad (3)$$

By taking $X = T + 1$ and $X = \gamma + 1$, we obtain two equations in the two unknowns $U(T + 1)$ and $U(\gamma + 1)$. As above, $\bar{U} = U(1)$ given in terms of $U(T + 1)$ and $U(\gamma + 1)$. $\bar{I}$ is simply equal to $1/(1 - e^{-g})$. Note that when $\gamma = T$, the expression for the throughput of 1-persistent CSMA-CD reduces to that of 1-persistent CSMA as given in [7].

3.1.2. Delay Analysis

We consider here the non-persistent protocol. To analyze packet dealy, we adopt the same *"linear feedback model"* used for the analysis of CSMA in [9,10]. The model consists of a finite population of M devices in which each device can be in one of two states: backlogged or thinking. In the thinking state, a device generates and transmits (provided that the channel is sensed idle) a new packet in a slot with probability σ. A device is said to be backlogged if its packet either had a channel collision or was blocked because of a busy channel. A backlogged device remains in that state until it completes successful transmission of the packet, at which time it switches to the thinking state. The rescheduling delay of a backlogged packet is assumed to be geometrically distributed with a mean of $1/\nu$ slots; this in effect is identical to considering that each backlogged user senses the channel in the current slot with a probability ν.

In this study, we assume M, σ and ν to be time invariant. We consider τ (the slot size) to be the unit of time. We again denote by S the average stationary channel throughput defined as the fraction of channel time occupied by valid transmissions. We denote by C the channel capacity defined as the maximum achievable channel throughput. We finally denote by D the average packet delay defined as the time lapse from when the packet is first generated until it is successfully received by the destination device.

Let N^t be a random variable representing the number of backlogged devices at the beginning of slot t. We follow the approach used in [9], and consider the embedded Markov chain identified by the first slot of each idle period (see fig. 4). We then use properties resulting from the theory of regenerative processes to derive the stationary channel performance under CSMA-CD, as outlined in [9,10].

We seek the transition probability matrix $\boldsymbol{P}$ between consecutive embedded points. $\boldsymbol{P}$ is the product of several single-slot transition matrices which we now define. N^t is invariant over the entire idle period except over slot $t_e + I - 1$. We denote by $\boldsymbol{R}$ the transition matrix for slot $t_e + I - 1$ and $\boldsymbol{Q}$ for all

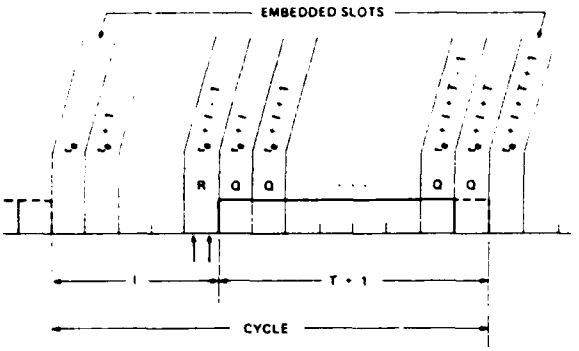


Fig. 4. The Embedded Slots in Nonpersistent CSMA Schemes.

remaining slots of the busy period. Since the length of the busy period depends on the number of devices which become ready in slot $t_e + I - 1$, we write $\boldsymbol{R}$ as $\boldsymbol{R} = \boldsymbol{S} + \boldsymbol{F}$, where the (i, k)th elements of $\boldsymbol{S}$ and $\boldsymbol{F}$ are defined as

$$s_{ik} = \Pr\{N^{t_e+I} = k \text{ and transmission is successful } | N^{t_e+I-1} = i\}, \quad (4)$$

$$f_{ik} = \Pr\{N^{t_e+I} = k \text{ and transmission is unsuccessful } | N^{t_e+I-1} = i\}. \quad (5)$$

For any slot t in the busy period, $\boldsymbol{Q}$ simply reflects the addition to the backlog from the $M - N^t$ thinking devices. If the transmission is successful, the transmission period has length $T + 1$; if it is unsuccessful, its length is $\gamma + 1$. The transition matrix $\boldsymbol{P}$ is therefore expressed as

$$\boldsymbol{P} = \boldsymbol{S}\boldsymbol{Q}^{T+1}\boldsymbol{J} + \boldsymbol{F}\boldsymbol{Q}^{\gamma+1}, \quad (6)$$

where $\boldsymbol{S}$, $\boldsymbol{F}$, and $\boldsymbol{Q}$ are given by

$$s_{ik} = \begin{cases} 0 & \\ \text{for } k < i & \\ \dfrac{(1-\sigma)^{M-i}[i\nu(1-\nu)^{i-1}]}{1-(1-\nu)^i(1-\sigma)^{M-i}} & \\ \text{for } k = i & \\ \dfrac{(M-i)\,\sigma(1-\sigma)^{M-i-1}[1-\nu]^i}{1-(1-\nu)^i(1-\sigma)^{M-i}} & \\ \text{for } k = i+1 & \\ 0 & \\ \text{for } k > i+1 & \end{cases} \quad (7)$$

$$f_{ik} = \begin{cases} 0 & \\ \text{for } k < i & \\ \dfrac{(1-\sigma)^{M-i}[1-(1-\nu)^i - i\nu(1-\nu)^{i-1}]}{1-(1-\nu)^i(1-\sigma)^{M-i}} & \\ \text{for } k = i & \\ \dfrac{(M-i)\,\sigma(1-\sigma)^{M-i-1}[1-(1-\nu)^i]}{1-(1-\nu)^i(1-\sigma)^{M-i}} & \\ \text{for } k = i+1 & \\ \dfrac{\binom{M-i}{k-i}(1-\sigma)^{M-k}\sigma^{k-i}}{1-(1-\nu)^i(1-\sigma)^{M-i}} & \\ \text{for } k > i+1 & \end{cases} \quad (8)$$

$$q_{ik} = \begin{cases} 0 & \\ \text{for } k < i & \\ \binom{M-i}{k-i}(1-\sigma)^{M-k}\sigma^{k-i} & \\ \text{for } k \geq i & \end{cases} \quad (9)$$

and where $\boldsymbol{J}$ represents the fact that a successful transmission decreases the backlog by 1, its (i, k)th elements being defined as

$$j_{ik} = \begin{cases} 1 & \\ \text{for } k = i-1 & \\ 0 & \\ \text{otherwise} & \end{cases} \quad (10)$$

It is clear that with $\gamma = T$, the above expression for P then corresponds to CSMA without CD. Let $\Pi = [\pi_0, \pi_1, \ldots, \pi_M]$ denote the stationary probability distribution of N^t at the embedded points. Π is obtained by the recursive solution of $\Pi = \Pi\boldsymbol{P}$.

Since N^{t_e} is a regenerative process, the average stationary channel throughput is computed as the ratio of time the channel is carrying successful transmission during a cycle (an idle period followed by a busy period) averaged over all cycles, to the average cycle length [9,10]. Therefore we have

$$S = \frac{\sum_{i=0}^{M} \pi_i P_s(i)\, T}{\sum_{i=0}^{M} \pi_i \left\{ \dfrac{1}{1-\delta_i} + 1 + P_s(i)\, T + [1 - P_s(i)]\, \gamma \right\}}, \quad (11)$$

$P_s(i)$ is the probability of a successful transmission during a cycle with $N^te = i$, and is given by

$$P_s(i) = ((M-i)\,\sigma(1-\sigma)^{M-i-1}(1-\nu)^i + i\nu(1-\nu)^{i-1}(1-\sigma)^{M-i})/(1-(1-\nu)^i(1-\sigma)^{M-i}) \quad (12)$$

$(1-\delta_i)^{-1}$, where $\delta_i = (1-\nu)^i(1-\sigma)^{M-i}$, is the average idle period given $N^te = i$.

Similarly, the average channel backlog is computed as the ratio of the expected sum of backlogs over all slots in a cycle (averaged over all cycles). to the average cycle length [9,10]. Therefore we have

$$\bar{N} = \frac{\sum_{i=0}^{M} \pi_i \left[\frac{i}{1-\delta_i} + A(i)\right]}{\sum_{i=0}^{M} \pi_i \left\{\frac{1}{1-\delta_i} + 1 + P_s(i)\,T + [1-P_s(i)]\,\gamma\right\}}, \quad (13)$$

where $A(i)$ is the expected sum of backlogs over all slots in the busy period with $N^te = i$, and is given by [3]

$$A(i) = \sum_{l=0}^{T}\sum_{j=i}^{M} j[SQ^l]_{ij} + \sum_{l=0}^{\gamma}\sum_{j=i}^{M} j[FQ^l]_{ij} = \sum_{j=i}^{M} j\left[S\sum_{l=0}^{T} Q^l + F\sum_{l=0}^{\gamma} Q^l\right]_{ij} \quad (14)$$

By Little's result [12], the average packet delay (normalized to T] is simply expressed as

$$D = \bar{N}/S \quad (15)$$

3.2. Analysis of CSMA-CD with variable size packets

T is now a discrete random variable. Let

$$G_T(z) \triangleq \sum_{t=1}^{\infty} z^t \Pr\{T = t\} \quad (16)$$

be the generating function of the distribution of T. In case of collison, regardless of the number of colliding packets and their lengths, the length of the busy period is $\gamma + 1$. In case of success, the length of the busy period is now random and has the same distribution as $T + 1$. The reason this is true, despite the fact that the length of a packet remains constant during its entire lifetime, is simply explained by the fact that the successful or unsuccessful outcome of a transmission period is solely dependent on the number of devices becoming ready at the beginning of that transmission period, and is independent of the lengths of the contending packets. Since the length of the busy period in case of a collision is constant (equal to $\gamma + 1$), the evolution of the channel over time is statistically identical to that in which the length of a packet is drawn from the packet length distribution only when its transmission is successful.

However, in the case of CSMA *without* collision detection, the collision period is a function of the lengths of the contending packets so the evolution of the channel over time is not statistically identical to that in which the length of a packet is drawn from the length distribution upon success. We include in appendix A an approximate analysis of CSMA in which the packet length at each transmission is independently redrawn from the packet length distribution.

The performance of nonpersistent CSMA-CD can thus be obtained from the previous analysis with the following simple modifications. The matrix P is now rewritten as

$$\boldsymbol{P} = \boldsymbol{S}G_T(\boldsymbol{Q})\,\boldsymbol{QJ} + \boldsymbol{FQ}^{\gamma+1} \quad (17)$$

and T is replaced by

$$\bar{T} \triangleq \sum_t t \Pr\{T = t\} \quad (18)$$

the average packet size, in all of equations (1), (11), (13) and (14).

In this case, the average packet delay given by Eq. (15) is normalized with respect to $\bar{T}$. For the same reason stated above, the average channel acquisition time (i.e., the time from when a packet is generated until it starts its successful transmission), denoted by W (in slots), is given by

$$W = D\bar{T} - \bar{T} \quad (19)$$

Accordingly, the delay incurred by packets of length t is expressed as

$$D_t = W + t \text{ (in slots)}. \quad (20)$$

It is interesting to note that for any throughput S, the difference in the delay incurred by packets of two different sizes is just the difference in transmission time of these packets. Smaller packets incur a smaller delay. The throughput contributed by packets of

[3] For an arbitrary matrix $\underline{B}$, we adopt the notation $[\underline{B}]_{ij}$ to represent the (i,j)th element of $\underline{B}$.

size t, denoted by S_t is expressed as

$$S_t = \frac{t\Pr\{T = t\}\, S}{\bar{T}} . \tag{21}$$

4. Numerical Results and Discussion

4.1. Fixed Packet Size

The behavior of CSMA-CD for fixed γ is, as expected, similar to that of CSMA [9], namely its throughput-delay performance is sensitive to ν, and therefore to the average retransmission delay. Figures 5 and 6 display the throughput-delay curves for nonpersistent CSMA and CSMA-CD respectively, with $M = 50$, $T = 100$, $\gamma = 2$, and various values of ν. For a fixed value of ν, the channel exhibits a maximum achievable throughput which depends on that value, hereafter referred to as the ν-capacity. We observe that, for a given ν, CSMA-CD always achieves, again as expected, lower delay for a given throughput and a higher ν-capacity [4]. The optimum throughput-delay performance is obtained by taking the lower envelope of all fixed-ν curves. Overall, CSMA-CD provides an improvement both in terms of channel capacity and throughput-delay characteristics.

We discuss now the sensitivity of this improvement to the collision detect time, γ, and the packet length T. Just as with CSMA, the larger T is, the better is the CSMA-CD performance for fixed γ. In fig. 7, we plot the channel capacity for the nonpersistent CSMA-CD versus γ for various packet lengths. The capacity at $\gamma = T$ is that of CSMA. The relative improvement in channel capacity obtained by CSMA-CD becomes more important as T decreases, that is, as the performance of CSMA degrades. We note for example that at best (i.e., when $\gamma = 2$) this relative improvement is about 16% (0.62 to 0.76) for $T = 10$ and about 11% (0.86 to 0.96) for $T = 100$. Clearly, for larger T ($T \geq 100$), nonpersistent CSMA provides relatively high channel capacity, and thus leaves little margin for improvement. With the 1-persistent protocol, however, the improvement can be more substantial. Channel capacity increases from about 0.53 for

[4] Note that for all values of ν used in plotting figs. 5 and 6, the ν-capacity with CSMA-CD approached the channel capacity (maximized over ν); there are values of ν (higher than $\nu = 0.15$) for which the ν-capacity is much lower than the CSMA-CD channel capacity similarly to what is seen in fig. 5 for CSMA and $\nu = 0.10$.

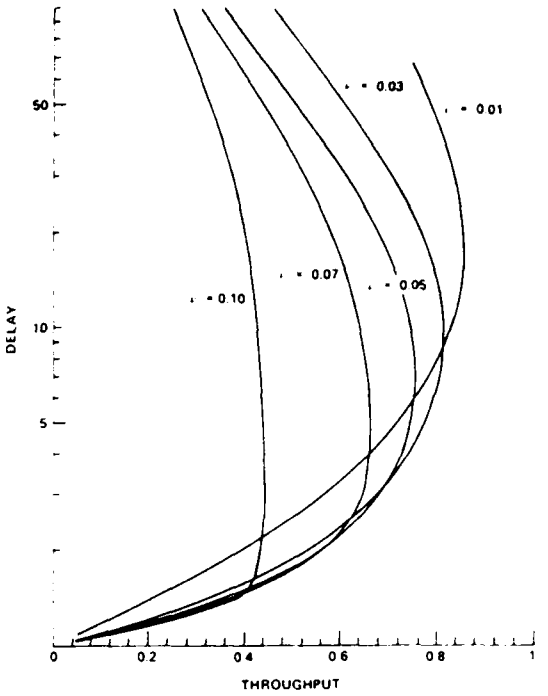


Fig. 5. The Throughput-Delay Tradeoff in CSMA at Fixed ν.

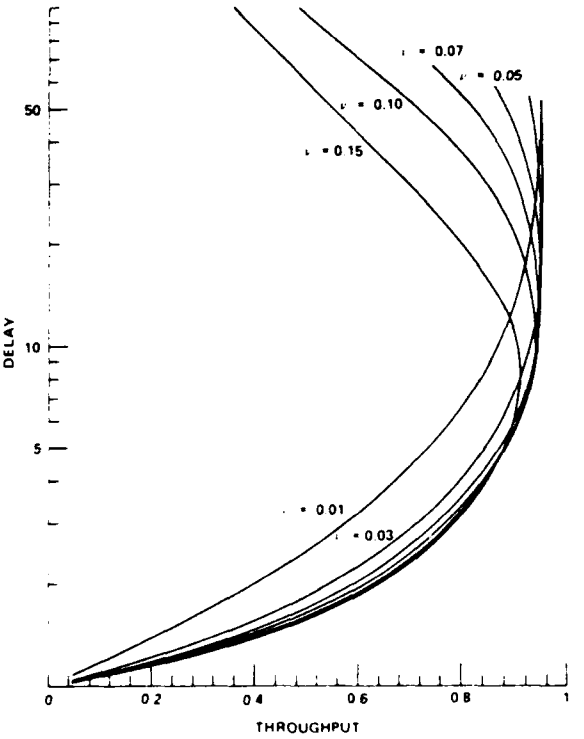


Fig. 6. The Throughput-Delay Tradeoff in CSMA-CD at Fixed ν.

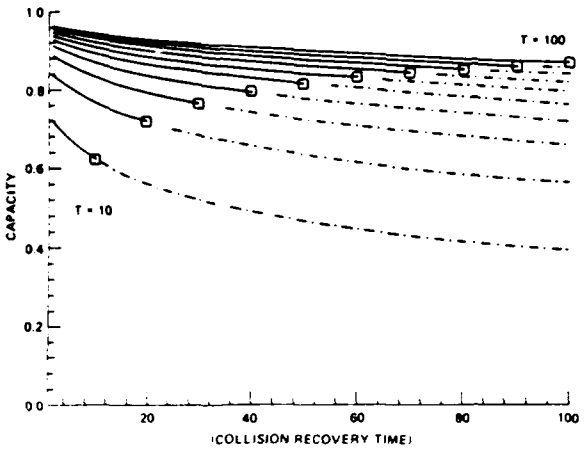


Fig. 7. Channel Capacity Versus γ in CSMA-CD.

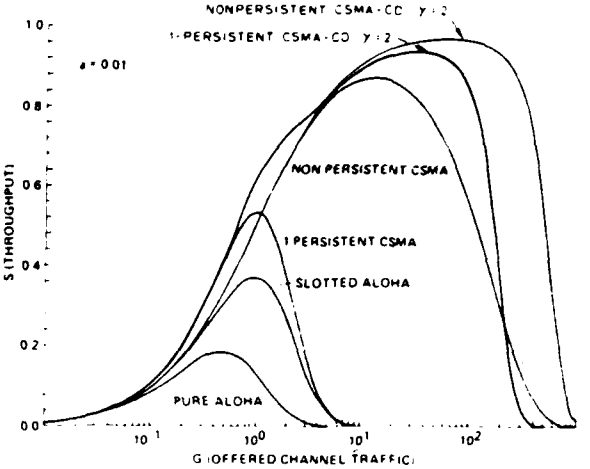


Fig. 9. Throughput Versus Channel Traffic (Infinite Population Model).

1-persistent CSMA to about 0.93 for 1-persistent CSMA-CD with $\gamma = 2$.

The effect of CD on the minimum delay (optimized with respect to ν) for a fixed channel throughput is seen in fig. 8, where we plot this minimum delay versus γ for the nonpersistent case with $M = 50$ and $T = 100$. We note that the higher the throughput is the better is the improvement. At low throughput (e.g., $S = 0.20$), the delay is insensitive to γ. With moderately high throughputs (e.g., $S = 0.68$), the delay with CSMA-CD (at $\gamma = 2$) is 70% that of CSMA.

As the throughput approaches the CSMA channel capacity (e.g. $S = 0.84$) the ratio in delay can be as low as 1/3 in favor of collision detection ($\gamma = 2$). Of course, for even higher throughputs, CSMA-CD achieves a finite delay as long as γ is sufficiently small.

The (S, G) relationship for CSMA-CD is displayed in fig. 9 along with the curves corresponding to the ALOHA and CSMA schemes. This figure exhibits again the improvement in channel capacity gained by CSMA-CD over all other schemes. For random access

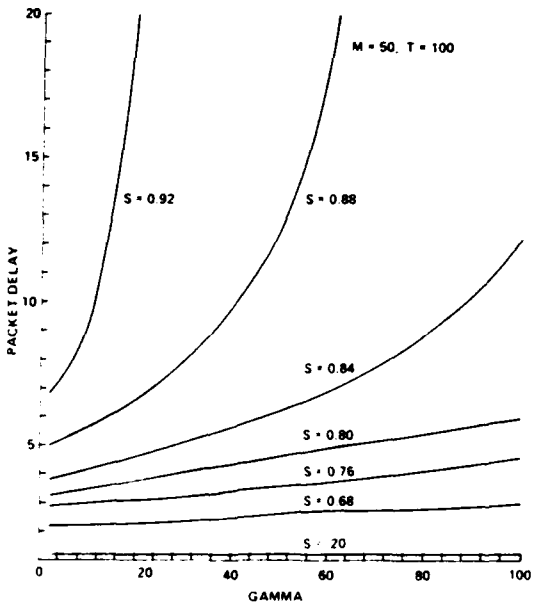


Fig. 8. Packer Delay in CSMA-CD at Fixed Throughput Versus γ.

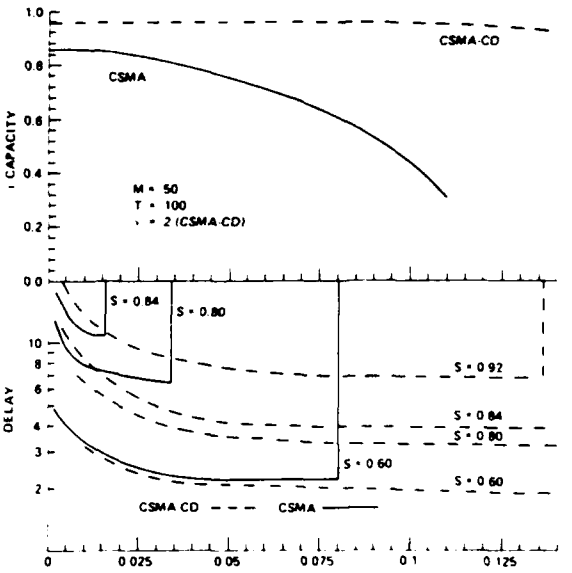


Fig. 10. Channel Capacity and Packet Delay at Fixed Throughput Versus ν for CSMA and CSMA-CD.

schemes in general, the fact that the throughput drops to zero as the offered channel traffic increases indefinitely is indicative of unstable behavior [9,14]. With CSMA-CD the ability to maintain a throughput relatively high and near capacity over a very large range of the offered channel traffic (see fig. 9) suggests that CSMA-CD may not be as unstable as the other schemes. That is, in the absence of dynamic control, CSMA-CD is capable of sustaining proper behavior when the channel load exceeds that for which the system has been tuned (i.e., optimized with respect to ν). (Note that, with respect to this stability argument, the nonpersistent CSMA-CD proves to be superior to 1-persistent CSMA-CD, in that it offers high throughput over a larger range of the offered traffic.) We further illustrate this important feature by plotting in fig. 10, as a function of ν, the ν-capacity and the packet delay at various channel throughputs for both nonpersistent CSMA and CSMA-CD (γ = 2) with T = 100 and M = 50. As ν approaches zero, the delay at fixed throughput gets arbitrarily large (due to large retransmission delays), while as ν approaches 1, the ν-capacity approaches zero (due to higher level of interference among backlogged devices). Thus there is a limited range for ν which is of practical interest. As we see in fig. 10, for T = 100 this range is about (0.005, 0.3). The ν-capacity curve for CSMA-CD is flat over a large portion of this range; with CSMA, the ν-capacity drops steadily as ν increases, and exhibits insensitivity only for smaller values of ν falling outside our range. Consider now CSMA. Given a channel throughput S, packet delay decreases as we increase ν (starting from relatively small values) and remains relatively constant, until, due to the decrease in ν-capacity, we reach a value of ν for which the ν-capacity approaches S, and thus the delay increases very sharply; this "practical" range of ν gets narrower as S increases, indicating that for high throughput, the system requires fine tuning. Let S = 0.60 be, for example, the (moderate) stationary channel throughput we expect the system to support. The channel is properly tuned (i.e., minimum delay is achieved) for ν in the range (0.04, 0.08). Consider now that the offered load on the channel is time-varying and suppose that the desired throughput exceeds 0.60 reaching values close to channel capacity (e.g., S = 0.84). This actually happens for increasing values of σ (i.e., when devices generate packets at a faster rate). If the desired load remains at such a high value for a relatively long period of time the channel saturates (i.e., the throughput drops to a low value, nearly all devices become backlogged and packet delay increases indefinitely). We can certainly support variations in offered load covering the entire range of achievable throughputs ($S \leqslant 0.84$) by setting ν at a value in the (now narrow) range corresponding to S = 0.84. This is achieved at the expense of increased average delay for S = 0.60 of 36% (from 2.2 to 3) unless, of course, dynamic control is exercised [9].

With CSMA-CD, on the contrary, there is a relatively wide range of ν for which the delay at fixed throughput is near optimum for all throughput levels up to 0.92.

Numerical results obtained for different values of the system parameters, namely M = 50 and 250, and T = 10 and 100 have shown that basically as T decreases or as M increases or both, then CSMA-CD starts exhibiting a behavior similar to that of CSMA, while always achieving improved performance.

In summary, the kind of improvement over slotted ALOHA we saw in [9] for CSMA due to carrier sensing, is now seen in CSMA-CD over CSMA.

4.2. Variable Packet Size

It is clear from the above discussion that as the packet size decreases the improvement obtained with collision detection is more important. We inquire here about the performance of the channel with collision detection when packets are of variable length. Instead

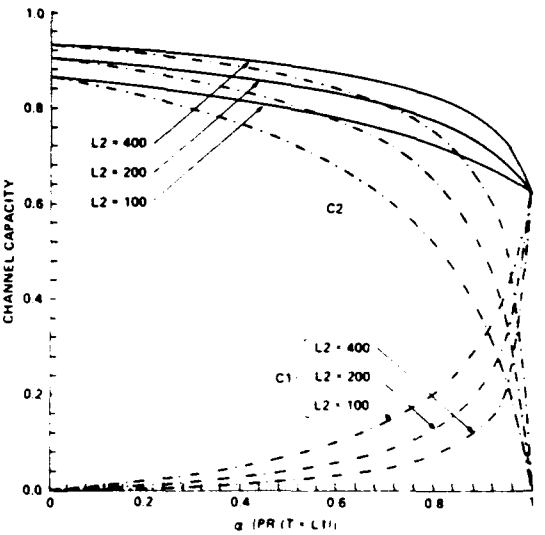


Fig. 11. CSMA Channel Capacity Versus α for Dual Packet Size.

of examining the general message length distribution case, we present here numerical results for the simpler dual packet size case; that is, traffic consists of a mixture of short and long packets. This simple distribution represents accurately many real situations, among them the important instance of the mixture of short packets resulting from interactive traffic and long packets resulting from file transfers. In fact, measurements performed on Xerox's Ethernet have clearly exhibited such a distribution [15]. Moreover, results obtained here are expected to be representative of the performance of a channel in more general packet length distributions.

We let L_1 (L_2) denote the transmission time of short (long) packets. We let α denote the fraction of packets generated which are short. Figure 11 displays the non-persistent CSMA channel capacity versus α for the case of short packets equal to 10 (slots) and three cases of long packets (100, 200, 400). The capacity of the channel decreases as α increases. With larger values of L_2 (e.g., $L_2 = 400$), this decrease is fairly slow until α is about 0.80; beyond 0.80 the capacity rapidly declines to reach the (lower) capacity of $T = L_1$. This shows that a relatively small fraction of long packets in the traffic mix can result in a channel utilization close to that obtained with only long packets. However it is important to note that, as the fraction of long packets increases, the fraction of channel capacity due to long packets, denoted by C_2, increases extremely rapidly to the detriment of that due to short packets, denoted by C_1, which decreases dramatically. This is seen in fig. 11 where we also plot C_1 and C_2 versus α. Recall that, by Equation (21) which also holds for CSMA under the independence assumption, C_1 and C_2 are given by

$$C_1 = \frac{\alpha L_1}{\alpha L_1 + (1-\alpha) L_2} C, \tag{22}$$

$$C_2 = \frac{(1-\alpha) L_2}{\alpha L_1 + (1-\alpha) L_2} C, \tag{23}$$

where C is the channel capacity.

In fig. 12 we plot the capacity versus α for CSMA-CD ($L_1 = 10, L_2 = 100$) at various values of γ, along with the corresponding CSMA capacity curve. The insensitivity of CSMA-CD capacity to variations of α over a large range of α is more apparent than with CSMA. However, the relative importance of C_1 and C_2 remains the same as in CSMA since the ratio of C_1 and C_2 is independent of the capacity.

In fig. 13 we plot the packet delay (averaged over all packets and normalized to L_1) versus throughput for various values of α for both CSMA and CSMA-CD. Packet delay includes the (successful) transmission time of the packet; thus clearly as the fraction of long packets increases, so does the average packet delay. Figure 13 exhibits the clear tradeoff between average packet delay and attainable channel capacity as the mix α varies. The improvement due to collision detection is also apparent for all values of α.

Most commonly, short packets belong to interactive users who require small delay, while long packets result from file transfer which, when intro-

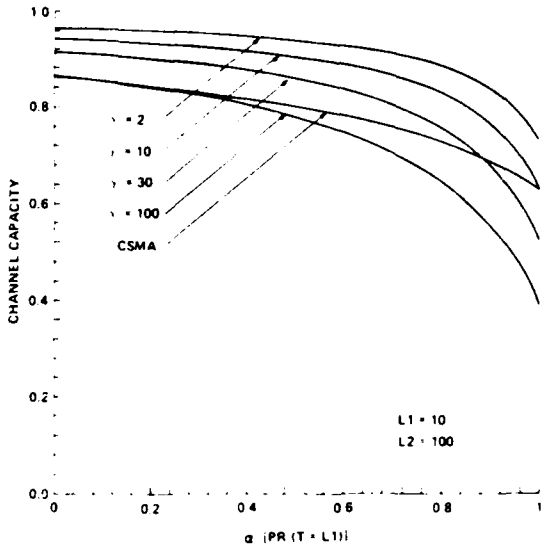


Fig. 12. CSMA-CD Channel Capacity Versus α for Dual Packet Size.

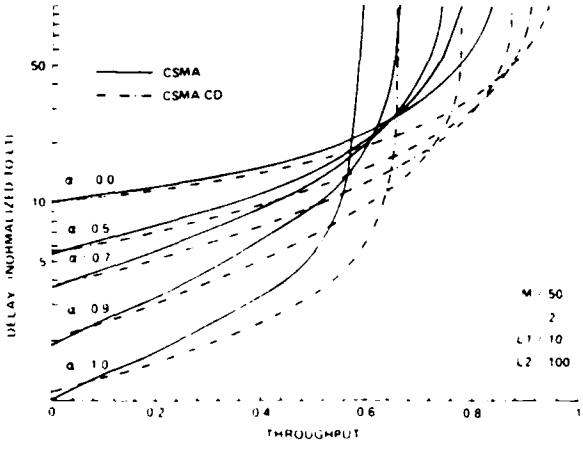


Fig. 13. Average Packet Delay Versus Channel Throughput for Various Values of α (Dual Packet Size).

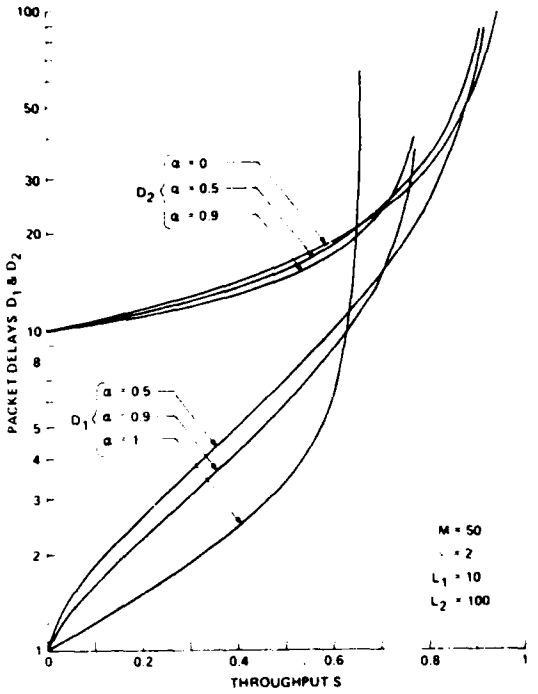


Fig. 14. Packet Delays for Short and Long Packets Versus Channel Throughput for Fixed α.

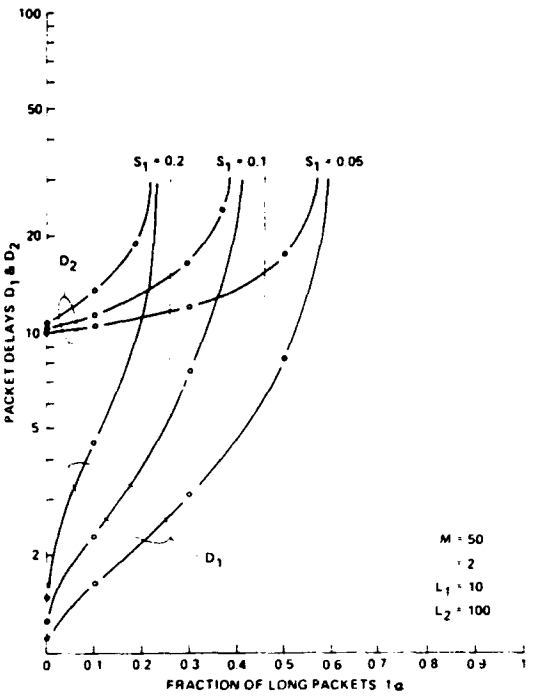


Fig. 16. D_1 and D_2 Versus $1 - \alpha$ for Constant S_1.

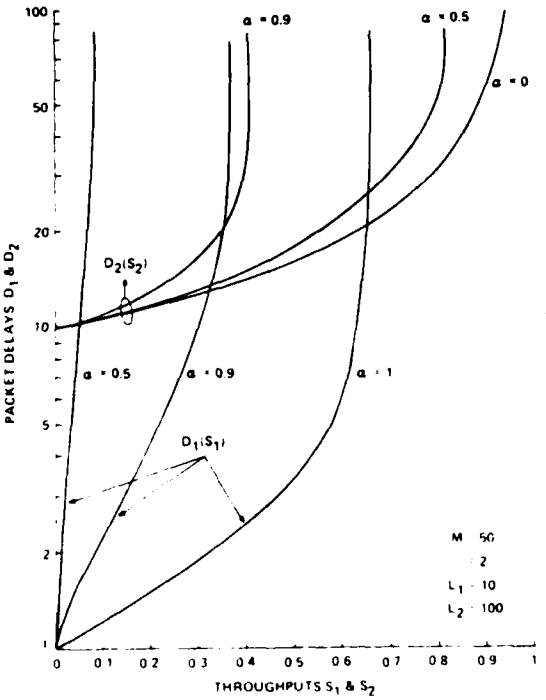


Fig. 15. Throughput-Delay Characteristics for Short and Long Packets.

duced, allow to recover an important fraction of the excess capacity. We inquire now as to the behavior and relative importance of the system performance measures with respect to each of the two packet sizes. Let S_1 (S_2) and D_1 (D_2) denote the throughput and

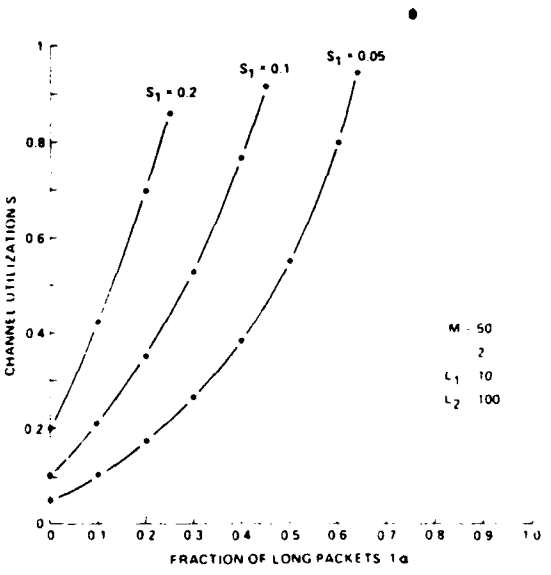


Fig. 17. Channel Throughput S Versus $1 - \alpha$ for Constant S_1.

packet delay for short (long) packets, respectively. In fig. 14 we plot for CSMA-CD, D_1 and D_2 versus S ($S_1 + S_2$) for various values of α and $M = 50; L_1 = 10; L_2 = 100; \gamma = 2$. As pointed out in the previous section the difference between D_2 and D_1 for a given value of α is always $L_2 - L_1$. For a given global achievable channel utilization S, D_1 and D_2 increase as the fraction of long packets increases in the mix; indeed the presence of long packets increases the waiting time W (the time to acquire the channel). In fig. 15, we plot D_1 and D_2 versus S_1 and S_2 respectively for various values of α, illustrating the degradation in throughput-delay tradeoff for short packets as the fraction of long packets $1 - \alpha$ increases. The throughput-delay tradeoff for long packets, however, improves.

Consider now a channel required to support interactive traffic at some level S_1. Certainly S_1 has to be lower than the channel capacity at $\alpha = 1$. Assume that S_1 is at some low level (e.g., 0.05 to 0.2). The introduction of long (file transfer) packets in view of achieving a higher channel utilization has the negative effect of significantly increasing the delay for the interactive traffic. This is illustrated in fig. 16 where we plot D_1 versus $1 - \alpha$ for fixed values of S_1. Clearly the channel utilization increases with $1 - \alpha$ as shown in fig. 17 where we plot S versus $1 - \alpha$ for fixed S_1. Thus in summary as the traffic mix includes more and more long packets, the overall channel capacity is improved in favor of long packets and to the detriment of the throughput-delay performance of short packets, indicating the need for priority schemes to maintain good performance for interactive traffic.

5. Conclusion

We extended the models used in the analysis of CSMA to cover the cases of collision detection and variable size packets. It was shown that the throughput-delay characteristics of CSMA-CD are better than the already highly efficient CSMA scheme. We characterized the improvement in terms of the achievable channel capacity and of the packet delay at a given channel utilization as a function of the collision detection time. Furthermore we established the fact that in uncontrolled channels (i.e., with a fixed average retransmission delay) CSMA-CD is more stable than CSMA, in that with CSMA-CD both channel capacity and packet delay are less sensitive to variations in the average retransmission delay.

We then studied the performance of these schemes in presence of variable size packets. Numerical results have been obtained for the interesting case of dual packet size. It was shown that a small fraction of long packets is sufficient to recover a channel capacity close to the (higher) capacity achieved with only long packets. However the improvement experienced by the introduction of long packets is in favor to the latter and to the detriment of the throughput-delay performance of short packets, establishing the necessity to design and implement priority schemes.

Acknowledgments

The authors would like to acknowledge the Stanford Linear Accelerator Center for providing the computer time used in this study.

Appendix

A. Variable Packet Size CSMA Without Collision Detection

All previous analyses of CSMA have dealt with fixed packet size [7–10]. In order to compare the performance of CSMA-CD to that of CSMA in presence of variable size packets, we undertake here the analysis of the latter. An important factor contributes to the complexity of an exact analysis. Contrary to CSMA-CD, the length of a busy period here is a function of the number of contending devices and their packet lengths. Accordingly, the backlog at an embedded point is a function of not only the backlog at the previous embedded point but also on the length of packets in the backlog. Conversely, the packet length distribution for those packets in the backlog is correlated with the number of such packets. For the sake of tractability we consider an approximate analysis based on removing this correlation by continually redrawing the lengths of packets independently from the packet length distribution [5].

Let $N^te = i$ be the state of the system at some embedded point t_e; let k denote the number of backlogged devices at the start of the corresponding transmission period (that is, $k - i$ new devices have joined the backlog in the last slot of the idle period). Let B be the random variable representing the number of devices simultaneously transmitting. If the transmission period is successful, then $B = 1$ with probability one. Given $N^te = i$ and given that the transmission period is unsuccessful, the distribution of B is given by

[5] This assumption was made by Ferguson in the analysis of pure-ALOHA which exhibits a similar correlation: the validity of the assumption in the context of pure-ALOHA was verified by simulation [13].

$P_B(b|i;\text{failure};k-i) \triangleq$

$\Pr\{B = b | N^{t_e} = i \text{ and transmission unsuccessful and } k-i \text{ new arrivals}\}$

$$= \begin{cases} \binom{i}{b-k+i} \nu^{b-(k-i)}(1-\nu)^{k-b} & k-i \geq 2;\ k-i \leq b \leq k \\ \dfrac{\binom{i}{b-1}\nu^{b-1}(1-\nu)^{i-b+1}}{1-(1-\nu)^i} & k-i=1;\ 2 \leq b \leq k \\ \dfrac{\binom{i}{b}\nu^b(1-\nu)^{i-b}}{1-i\nu(1-\nu)^{i-1}-(1-\nu)^i} & k-i=0;\ 2 \leq b \leq k \\ 0 & \text{otherwise} \end{cases} \quad \text{(A1)}$$

The length of the busy period, denoted by T_{max}, is equal to the maximum length among all B packets. Given $B = b$, the distribution of T_{max} is given by

$$P_{T_{max}}(t|b) \triangleq \Pr\{T_{max} \leq t | B = b\} = [\Pr\{T \leq t\}]^b . \quad \text{(A2)}$$

It is thus clear from the above discussion that the length of the busy period is a function of the state of the system in slot $t_e (N^{t_e} = N^{(t_e + I - 1)} = i)$ and in slot $t_e + I (N^{t_e + I} = k)$. Given the two latter conditions, and given that $T_{max} = t$, the state of the system at the next embedded point is j with probability $[Q^{t+1}]_{kj}$. Therefore, removing all conditions, the (i, j)th element of the transition matrix P is now given by

$$P_{ij} = [SG_T(Q)\,QJ]_{ij} + \sum_{k=i}^{M}\left\{\sum_{b=k-i}^{k}\left[\sum_{t=1}^{\infty} f_{ik}(Q^{t+1})_{kj} P_{T_{max}}(t|b)\right] \times P_B(b|i;\text{failure};k-i)\right\} . \quad \text{(A3)}$$

Similar considerations lead to the following expressions for the stationary channel throughput and backlog:

$$S = \frac{\sum_{i=0}^{M} \pi_i P_s(i)\,\bar{T}}{\sum_{i=0}^{M} \pi_i \left[\frac{1}{1-\delta_i} + 1 + P_s(i)\,\bar{T} + \bar{T}_{max}(i)\right]} , \quad \text{(A4)}$$

$$\bar{N} = \frac{\sum_{i=0}^{M} \pi_i \left[\frac{i}{1-\delta_i} + A(i)\right]}{\sum_{i=0}^{M} \pi_i \left[\frac{1}{1-\delta_i} + 1 + P_s(i)\,\bar{T} + \bar{T}_{max}(i)\right]} , \quad \text{(A5)}$$

where

$$\bar{T} = \sum_{t=1}^{\infty} t \Pr\{T = t\} \quad \text{(A6)}$$

$$\bar{T}_{max}(i) = \sum_{k=i}^{M}\left[\sum_{b=k-i}^{k}\left(\sum_{t=1}^{\infty} t f_{ik} P_{T_{max}}(t|b)\right) \times P_B(b|i;\text{failure};k-i)\right] \quad \text{(A7)}$$

$$A(i) = \sum_{j=i}^{M} j \left(\underline{S}\sum_{l=0}^{\bar{T}} \underline{Q}^l\right)_{ij} + \sum_{j=i}^{M} j \left\{\sum_{k=i}^{M}\left\{\sum_{b=k-i}^{k}\left[\sum_{t=1}^{\infty} f_{ik}\left(\sum_{l=0}^{t} Q^l\right)_{kj} P_{T_{max}}(t|b)\right] \times P_B(b|i;\text{failure};k-i)\right\}\right\} \quad \text{(A8)}$$

Under the independence assumption made in analyzing CSMA with variable size packets, the delay obtained by the ratio $\bar{N}/S$ is normalized with respect to $\bar{T}$. Moreover under this assumption equations (19), (20), and (21) hold here too.

B. Derivation of Channel Capacity Using the Infinite Population Model

Here all cycles are statistically identical. The average time during a cycle that the channel is carrying a valid transmission is simply $\bar{U} = \bar{T}ge^{-g}/(1 - e^{-g})$. The average idle period is, as before, $\bar{I} = e^{-g}/(1 - e^{-g})$. The distribution of B is given by $\Pr\{B = b\} = g^b e^{-g}/b!(1 - e^{-g})$. Given $B = b$, the average transmission period is

$$\bar{TP} = \sum_{t=1}^{\infty} [1 - (\Pr\{T \leq t\})^b] . \quad \text{(A9)}$$

Therefore the throughput is given by

$$S = \frac{\bar{T}g}{1 + \sum_{b=1}^{\infty}\sum_{t=1}^{\infty} (g^b/b!)[1 - (\Pr\{T \leq t\})^b]} . \quad \text{(A10)}$$

References

[1] D.G. Willard, Mitrix: A Sophisticated Digital Cable Communications System, Proceedings of the National Telecommunications Conference, November 1973.

[2] A.G. Fraser, A Virtual Channel Network, Datamation, February 1975.

[3] D.J. Farber, et al., The Distributed Computing System, Proceedings of the 7th Annual IEEE Computer Society International Conference, February 1973.

[4] D.J. Farber, A Ring Network, Datamation, February 1975.

[5] R.M. Metcalfe and D.R. Boggs, Ethernet: "Distributed Packet Switching for Local Computer Networks," Communications of the ACM, vol. 19, no. 7, pp. 395–403, 1976.

[6] N. Abramson, The ALOHA system – Another alternative for computer communications, in 1970 Fall Joint Comput. Conf. AFIPS Conf. Proc., vol. 37, Montvale, NJ: AFIPS Press, 1970, pp. 281–285.

[7] 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., vol. COM-23, pp. 1400–1416, Dec. 1975.

[8] 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., vol. COM-23, pp. 1417–1433, December 1975.

[9] F. Tobagi and L. Kleinrock, Packet switching in radio channels: Part IV – Stability considerations and dynamic control in carrier sense multiple access, IEEE Trans. Commun., vol. COM-25, pp. 1103–1120, October 1977.

[10] F. Tobagi, et al., Modeling and Measurement Techniques in Packet Communication Networks, IEEE Proceedings, pp. 1423–1447, November 1978.

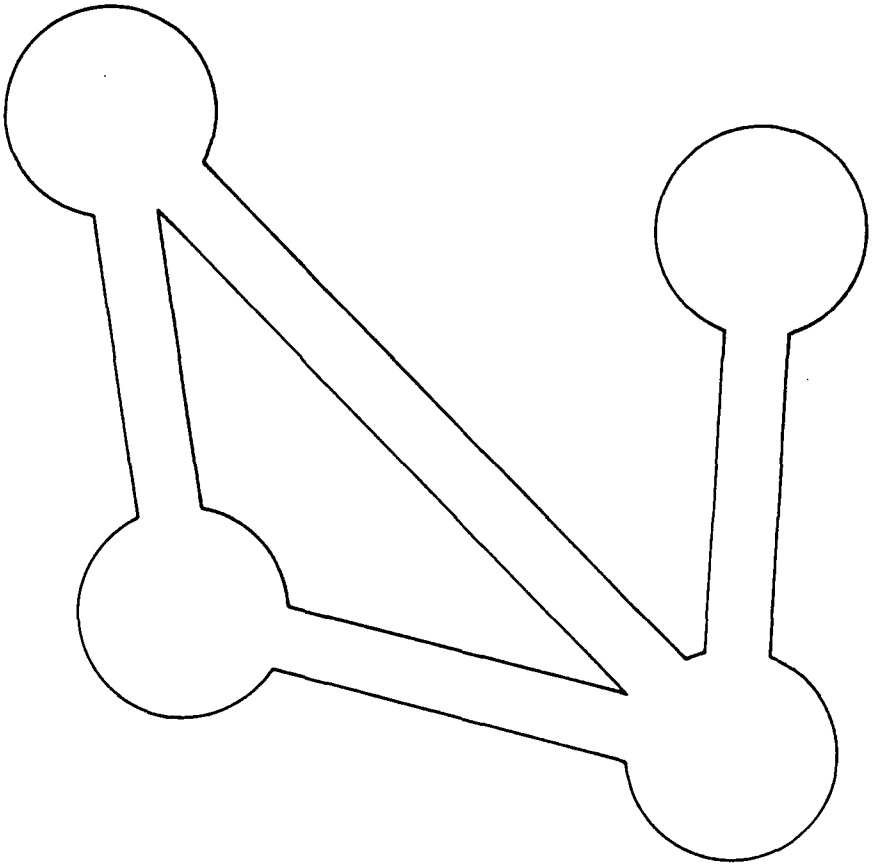
[11] F. Tobagi, Random access techniques for data transmission over packet switched radio networks. Ph.D. dissertation, Comput. Sci. Dep., School of Eng. and Appl. Sci., Univ. of California, Los Angeles, rep. UCLA-ENG 7499, December 1974.

[12] J. Little, "A Proof of the Queueing Formula L = λW," Operation Res., vol. 9, pp. 383–387, March-April 1961.

[13] M.J. Ferguson, An approximate Analysis of Delay for Fixed and Variable Length Packets in an Unslotted ALOHA Channel, IEEE Trans. Commun., pp. 644–654, July 1977.

[14] L. Kleinrock and S.S. Lam, Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation, IEEE Trans. Communications, pp. 410–423, April 1975.

[15] J.F. Shoch and J.A. Hupp, Performance of an Ethernet Local Network – A Preliminary Report, Proceedings of the Local Area Communication Network Symposium, Boston, May 1979.



Message-Based Priority Functions in Local Multiaccess Communication Systems

Raphael Rom

Telecommunications Sciences Centre, Computer Science and Technology Division, SRI International, Menlo Park, CA 94025, U.S.A.

and

Fouad A. Tobagi

Department of Electrical Engineering, Stanford University, Stanford, CA 94305, U.S.A.

The proliferation of computer networks has brought about a wealth of applications that impose disparate requirements upon the communication channels they use. In particular, the traffic requirements differ to such a degree that optimization of access schemes for one pattern is often detrimental to all rest. Message priority offers a solution to the problem. It provides a means of administering channel usage to meet these requirements while maintaining high total utilization. This paper proposes priority schemes appropriate for introduction into different architectures of local multiaccess communication systems to achieve these desired results.

Keywords: Multiaccess, Local Computer Networks, Priority, Access Protocols, Broadcast Bus, Ethernet, Ring, Carrier Sense Multiple Access, Performance

1. Introduction

There are numerous reasons why local area communication networks have registered such significant advances in the past few years. As examples, one can cite the recent interest in applying the increasingly inexpensive processing power to office and industrial automation, the necessity to share expensive scarce resources, the need for local collection and dissemination of information, and the growing interest in distributed architectures for data processing.

Just as in any field, the development of local-area computer communication systems is subject to a number of constraints. Typical of these are such basic features as simplicity, flexibility and reliability. The environments in question are generally characterized by a large number of devices, often relatively inexpensive, that require interconnection. Such environments call for networks with simple topologies and simple, low-cost interfaces, that provide considerable flexibility for accommodating the variability in the

Raphael Rom is a senior researcher in the Telecommunication Science Center at SRI international. His current areas of interest include architectures for local computer networks, computer network protocols, performance analysis, and interactive software systems. Dr. Rom joined SRI International in 1975 and has since worked and directed research on a wide variety of subjects in distributed processing.

Dr. Tobagi is Assistant Professor of Electrical Engineering, School of Engineering, Stanford University. His current research interests include computer communication networks, packet switching in ground ratio and satellite networks, modeling and performance evaluation of computer communications systems. From 1971 to 1974, he was with the University of California, where he participated in the ARPA Network Project, and did research in packet radio communication. In 1972, he was with the Communications Systems Evaluation and Synthesis Group, IBM, Yorktown Heights, NY. From 1974 to 1978, he was a research staff project manager with the ARPA project at UCLA. In 1978, he joined the faculty at Stanford. Dr. Tobagi was the winner of the IEEE 1981 Leonard G. Abraham award for the best paper in the field of Communications Systems.

North-Holland Publishing Company
Computer Networks (1981) 273–286

environment and ensure the desired reliability.

Several architectures have been proposed, including Mitre's Mitrix, UC/Irvine's Distributed Computer System (DCS) and Xerox's Ethernet [1,2,3,4]. An element common to them all is the packet broadcasting technology on which they are based. Packet broadcasting is attractive in that it combines the advantages of both packet switching and broadcast communication. Packet switching offers the efficient sharing of communication resources by many contending users with heterogeneous requirements. Broadcast communication, whenever possible, eliminates complex topological design problems. It is well understood that computer communication traffic is bursty. Consequently, it has been established that, rather than furnish individual low-speed channels to users, it is more efficient to provide the available communication bandwidth as a single high-speed channel to be shared by the contending users. This solution conveys with it the attendant benefits of the law of large numbers, which states that with very high probability the aggregate demand placed on the channel will be equal to the sum of the average individual demands.

The need for priority functions in multiaccess environments is clear. One is usually tempted to multiplex traffic from several users and different applications on the same bandwidth-limited channel in order to achieve a higher utilization of the channel. Since different applications impose different requirements on the system, it is important that multiaccess schemes be responsive to the particular exigencies of each. Priority functions offer a solution to this problem, and constitute the subject of this paper.

In Section 2 we shall describe briefly the various network architectures and their characteristics – namely, the bidirectional broadcast systems (BBS) of the Ethernet type, the unidirectional ring networks, and the unidirectional broadcast systems (UBS). In Section 3 we illustrate the need for priority functions in multiaccess environments. We then propose and discuss priority schemes appropriate to each of the three specific environments described earlier.

2. Network Architectures and their Characteristics

2.1. Bidirectional Broadcast Systems (BBS)

In bidirectional broadcast systems, such as Ethernet [4], all the communicating devices are connected to a common cable on which transmission signals propagate in both directions. The device connection interface is a passive cable tap, so that failure of an interface will not prevent the remaining devices from communicating. The interface is capable of identifying and accepting messages destined for it. The use of a single coaxial cable with bidirectional transmission naturally achieves broadcast communication.

The difficulty encountered in controlling access to the channel by users who can communicate via that channel only has given rise to what are known as random-access techniques. The best-known schemes are ALOHA [5] and carrier-sense multiple access (CSMA) [6]. In the ALOHA scheme, users transmit any time they desire; when conflicts occur, the conflicting users reschedule transmission of their packets. In the CSMA scheme the risk of a collision is decreased by having users sense the channel prior to transmission. If the channel is sensed busy, transmission is inhibited. CSMA performs well only if the propagation delay is short compared to the transmission time of a packet (a situation encountered in local-area networks and ground radio systems) and if all users can hear all transmissions on the channel (i.e., when the system is physically fully connected, as is the case with BBS).

Many CSMA protocols exist that differ as regards the action taken by a ready subscriber that finds the channel busy. In the nonpersistent CSMA, the terminal simply schedules the retransmission of the packet to some later time. In the 1-persistent CSMA, the terminal monitors the channel, waits until it goes idle and then transmits the packet with probability 1. In the p-persistent CSMA, the terminal monitors the channel as in 1-persistent but, when the channel does go idle, it transmits the packet with probability p only, and with probability $1-p$ it waits through the maximum propagation delay interval and then repeats the process as long as the channel is still sensed idle. [1]

Given the physical characteristics of data transmission on coaxial cables, in addition to sensing the carrier it is possible for transceivers to detect interference among several transmissions (including their own) and abort the transmission of their colliding packets. This produces a variation of CSMA referred to as carrier-sense multiple access with collision detection (CSMA-CD) [4,7].

[1] Maximum propagation delay is the time required for signals to propagate between the two most disparate transmitter and receiver in the system. It is therefore the elapsed time after which a transmitted signal is guaranteed to have been received by all subscribers.

2.2. Ring Networks

In a ring topology messages are passed from node to node along unidirectional links until they reach their destination or, if required by the protocol, until they return to the originating node. Each subscriber is attached to the cable by means of an active tap that allows the information to be examined before it proceeds along the cable. This, in effect, renders the ring a 'cut-through store-and-forward' architecture, in which, to avoid excessive delays, messages are not stored in their entirety but rather (re)-transmitted onto the cable as soon as possible. The delay incurred at each intermediate node can thus be limited to a small number of bit-times.

A simple access scheme suitable for a ring consists of passing the right of access sequentially from node to node around the ring. (Note that in a ring the physical locations of the nodes define a natural ordering among them). One implementation of this scheme is exemplified by the Distributed Computing System's network, in which an 8-bit control token is passed sequentially around the ring [8]. Any node with a ready message, upon receiving the control token, may remove the token from the ring, send the message, and then pass the token on. Another implementation consists of providing a number of message slots that are continuously transmitted around the ring. A message slot may be empty or full: a node with a ready message waits until an empty slot comes by, marks it as full, and then uses it to send its message [9,10,11].

Still another strategy is known as the register insertion technique [12]. Here a message to be transmitted is first loaded into a shift register. If the ring is idle, the shift register is simply transmitted. If not, the register is inserted into the network loop at the next point separating two adjacent messages; the message to be sent is shifted out onto the ring while an incoming message is shifted into the register. The shift register can be removed from the network loop when the transmitted message has returned to it. The insertion of a register has the effect of prolonging the transport delay of messages on the ring.

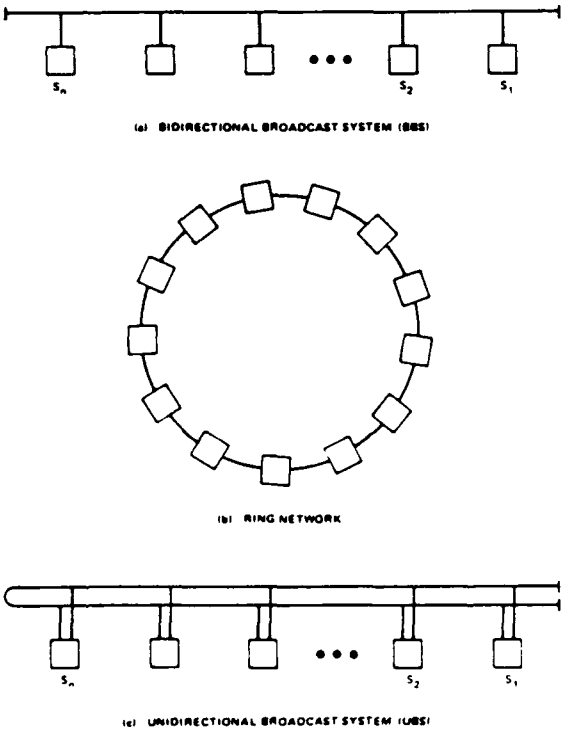


Fig. 1. Various Local Network Architectures.

2.3. Unidirectional broadcast system (UBS)

By contrast to the BBS, transmission signals in a UBS are forced to propagate in only one direction of the cable. This may be achieved, for example, by the use of taps that considerably attenuate the signals in the opposite direction (CATV technology utilizes such taps which have also been used by such local networks as MITRIX [1]).

The entire system consists of two interconnected unidirectional channels – the forward (or outbound) channel and the reverse (or inbound) channel. Subscriber devices are connected to both channels via

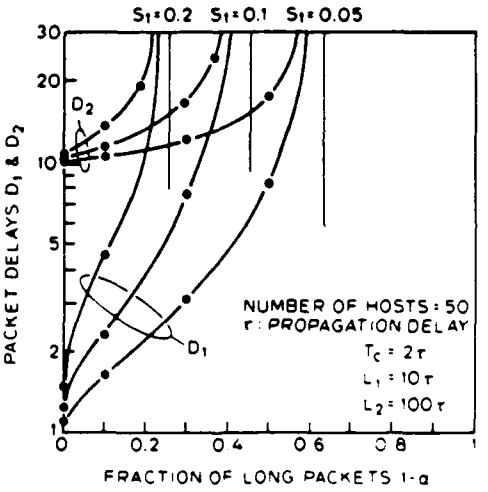


Fig. 2. Packet delays D_1 and D_2 Versus the Fraction of Long Packets for Constant Throughput S_1.

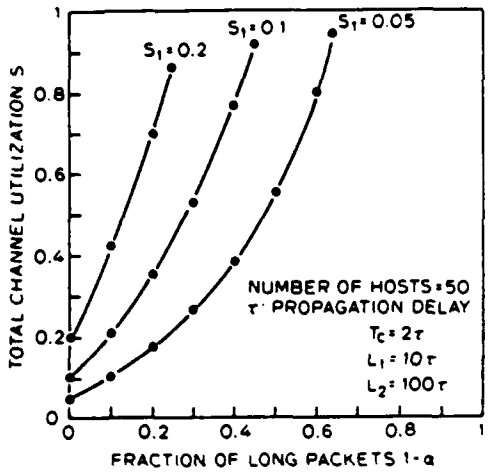


Fig. 3. Total channel utilization versus the fraction of long packets for constant throughput S_1.

passive taps. In the simplest case, the tap on the forward portion of the cable is used for message transmittal, while the tap on the reverse portion is used for message reception. The channel interconnection can be simply implemented by folding a single cable.

Broadcast communication is accomplished because all signals traverse the entire inbound channel to which all devices are attached. This configuration results in an inherent physical ordering of the subscribers, a feature we take advantage of in the sequel. Fig. 1 displays the various architectures schematically.

3. On Priority Functions in Multiaccess Environments

3.1. Illustrating the Need for Priority Functions

To illustrate the need for priority functions in multiaccess environments, we consider the following scenario on a broadcast bus used with the nonpersistent CSMA-CD protocol. Assume that the channel is required to support interactive traffic at some low throughout level S_1, (e.g., 5% to 20% of the available bandwidth). Clearly a large portion of the channel is unused, and can be recovered if we allow traffic from other applications, such as file transfer, to be transmitted on the channel simultaneously. File transfer traffic typically consists of packets that are 'long' compared with those encountered in interactive traffic. Analysis of CSMA-CD with variable packet size [7] has shown that higher channel utilization is indeed achieved by the introduction of file transfer packets, but to the detriment of the short interactive packets which consequently experience increasingly long delays.

Numerical results (taken from [7]) are displayed in Figs. 2 and 3. Denoting by $1-\alpha$ the fraction of long packets introduced into the mix, Fig. 2 shows a plot of packet delays D_1 and D_2 as incurred by short packets of size $L_1 = 10\tau$, and long packets of size $L_2 = 100\tau$ (where τ is the maximum propagation delay) versus $1-\alpha$. Fig. 3 shows a plot of the total channel utilization achieved versus $1-\alpha$. To recover the available excess capacity while maintaining an acceptable performance for interactive traffic, we need to implement a scheme that gives to all interactive messages priority over file transfer messages.

3.2. General Specifications Required of Priority Schemes

Little work has been done in attempting to incorporate priority functions into multiaccess protocols. The distributed nature of the system has been a major obstacle. Priority functions here are viewed in their most general sense; that is, priority is defined as a function of the message to be transmitted and not of the device transmitting the message. Before proceeding with a description of priority schemes, we briefly discuss here the requirements for acceptability of a priority scheme:

(1) *Hierarchical independence of performance* – The performance of the scheme as seen by messages of a given priority class should be uneffected by the load exercised on the channel by lower priority classes. Increasing loads from lower classes should not degrade the performance of higher-priority classes.

(2) *Fairness within each priority class* – Several messages of the same priority class may be present simultaneously in the system. These should be able to contend equally on the communication bandwidth.

(3) *Robustness* – A priority scheme must be robust in that its proper operation and performance should be uneffected by errors in status information.

(4) *Low overhead* – The overhead required to implement the priority scheme (including any control information to be exchanged among the contending users, as required by the scheme) must be kept minimal.

To satisfy requirement (1), a priority scheme must be based on the principle that the right of access 'at any instant' be given exclusively to ready messages of the highest current priority level. This principle is easy to achieve in a nondistributed environment, such as a single-server queueing system, where one has

knowledge of all events occurring in the system. In a distributed environment, such as the one in question there are three basic problems that we need to address in designing a multiaccess protocol with a message-based priority function:

(1) identifying the exact instants at which to assess the highest current priority class that has ready messages;

(2) design of a mechanism for assessing the highest nonempty priority class;

(3) design of a mechanism that assigns the channel to the various ready users within a class.

In the following sections we provide solutions to these problems for each of the network architectures described in Section 2.

4. A Priority Scheme for Bidirectional Broadcast Systems

4.1. Mechanism for Priority Assessment (Nonpreemptive Discipline)

Because of the broadcast nature of transmission, users can monitor activity on the channel at all times. Assessment of the highest priority class with ready messages is done, (as is the case in the nonpreemptive discipline) at least at the end of each transmission period, whether successful or not, i.e., every time the carrier on the channel goes idle. When detected at a subscriber, end-of-carrier (EOC) establishes a time reference for that subscriber. Following EOC, the channel time is considered to be slotted with the slot size equal to $2\tau + \gamma$, where τ denotes the maximum one-way propagation delay between pairs of subscribers, and γ is a sufficiently long interval for a subscriber to detect an unmodulated carrier. The priority of a subscriber at any time is the highest-priority class with messages present in its queue.

Let s denote an arbitrary subscriber and EOC(s) denote the time of end-of-carrier at subscriber s. Let $p(s)$ denote the priority level of subscriber s at time EOC(s). The priority assessment algorithm has subscriber s operate as follows:

(1) If, following EOC(s), carrier is detected in slot i, with $i < p(s)$ (thus meaning that at least one subscriber has priority i higher than $p(s)$ and that access right must be granted to class i), then subscriber s awaits the following end-of-carrier (at the end of the next transmission period) at which time it reevaluates its priority and repeats this step.

(2) If, following EOC(s), no carrier is detected prior to the jth slot, where $j = p(s)$, subscriber s transmits a short burst of unmodulated carrier of duration γ at the beginning of slot j (thus reserving channel access to priority class $p(s)$ and, immediately following this slot, operates according to the contention resolution algorithm decided upon within class $p(s)$ (such as p-persistent CSMA, for example). At the next end-of-carrier, subscriber s reevaluates its priority level and repeats the algorithm (step 1 above).

(3) If, following EOC, no reservation burst is detected for K consecutive slots, where K is the total number of priority classes available in the system, then the channel becomes free to be accessed by any subscriber, regardless of its priority, until a new EOC is detected.

Thus, by means of short-burst reservations following EOC, the highest nonempty priority class is granted exclusive access right, and messages within that class can access the channel according to any contention algorithm. If the contention algorithm is CSMA, we refer to the scheme as prioritized CSMA (P-CSMA).

Note that the above algorithm corresponds to a nonpreemptive discipline, since a subscriber that has been denied access does not reevaluate its priority until the next end-of-carrier. However, note that, by assessing the highest priority level at the end of each transmission period, whether the latter has been successful or not, the scheme allows higher-priority messages to regain the right of access without incurring substantial delays.

The scheme is robust, as no precise information regarding the demand placed upon the channel is exchanged among the users. Information regarding the existing priority classes is implied by the position of the burst of unmodulated carrier following EOC. Note also that there is no need to synchronize all users with a universal time reference. By choosing the slot size to be $2\tau + \gamma$, we guarantee that a burst emitted by any subscriber in its kth slot is received with the kth slot of all other subscribers.

We illustrate this procedure in Fig. 4 by displaying a snapshot of the activity on the channel. For the sake of simplicity and without loss of generality we consider, in this illustration that there are only two possible priority levels in the system. We denote by n_1 and n_2 the number of active subscribers in class 1 (C_1) and class 2 (C_2), respectively. We adopt the convention that C_1 has priority over C_2. We also show a reservation burst as occupying the entire slot in which

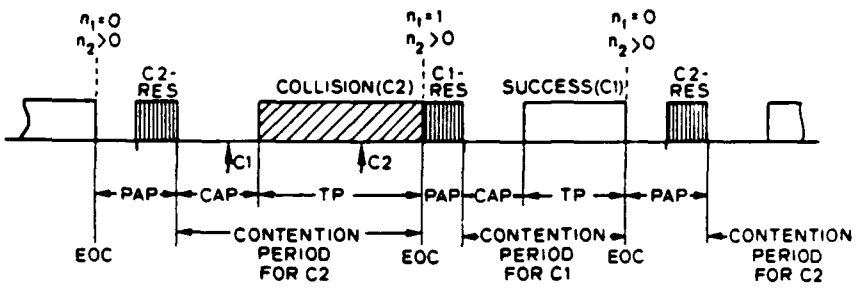


Fig. 4. Channel Activity for *P*-CSMA.

it is transmitted. Finally, we represent by a vertical upward arrow the arrival of a new message to the system; the label C_1 or C_2 indicates the priority class to which the message belongs. In Fig. 4 we assume that we have $n_1 = 0$ and $n_2 > 0$ at the first EOC.

Following EOC, a reservation burst is transmitted in the second slot. In this case the priority resolution period, also called priority assessment period (PAP), is equal to two slots. Following the reservation, we observed a channel access period (CAP) consisting of the elapsed idle time until the channel is accessed by some user(s) in class 2. Clearly CAP is a function of the channel access procedure employed by class 2. Following CAP we observe the transmission period (TP) itself, the end of which establishes the new EOC time reference. (A cross-hatched TP signifies a collision.) The time between a reservation and the following EOC, called the contention period and equals to CAP + TP, is the interval during which exclusive access right is given to the class that succeeded in reserving the channel. In this nonpreemptive case, message arrival C_1 – although of higher priority – is not granted access until the EOC following its arrival, at which time it reserves the channel.

Note that the overhead incurred in a resolution period following EOC is a function of the current highest-priority level. The higher this class, the smaller the overhead and the smaller the delay in gaining access.

4.2. 1-Persistant and p-Persistent P-CSMA

Immediately after a reservation burst for class i, the p-persistent CSMA scheme consists of having each subscriber with priority i do the following:

(1) with probability p it transmits the message;

(2) with probability $1 - p$ it delays the transmission by τ sec and repeats this procedure (provided the channel is still sensed idle).

This is equivalent to having each subscriber with priority i transmit its message following a geometrically distributed delay with mean $1/p$ propagation delays, provided that no carrier is detected prior to that time. When EOC is detected, a new time reference is established and a new reservation period undertaken.

In a 1-persistent CSMA mode, subscribers with ready messages, instead of sending a short burst to indicate a reservation, simply start transmission of their highest-priority messages in the corresponding slot following EOC – provided, of course, that no carrier is detected in previous slots. If a single subscriber is transmitting, its transmission is successful and transmission termination establishes a new EOC time reference.

On the other hand, if two or more subscribers overlap in transmission, a collision results; all users become aware of the collision and will consider it in lieu of a reservation. In other words, the end of this first transmission does not constitute a new time reference and so no new reservation period is started. All subscribers involved in the collision randomly reschedule the transmission of their respective messages according to some distribution (e.g., a geometrically distributed number of τ periods, with mean $1/p$). The subscribers transmit their messages at the scheduled time, provided that no carrier is detected at that time. The end of this new transmission period constitutes a new time reference and the procedure is repeated. (See Fig. 5.)

In general, 1-persistent CSMA is known to be inferior to its p-persistent counterpart, since $p = 1$ is certainly not optimum if the likelihood of having several subscribers with ready messages of the same priority level is high. However, if the load placed on the channel by some priority class is known to be low (as would most probably be the case for high-priority levels to guarantee their performance), then 1-persistent CSMA used within that class may present some

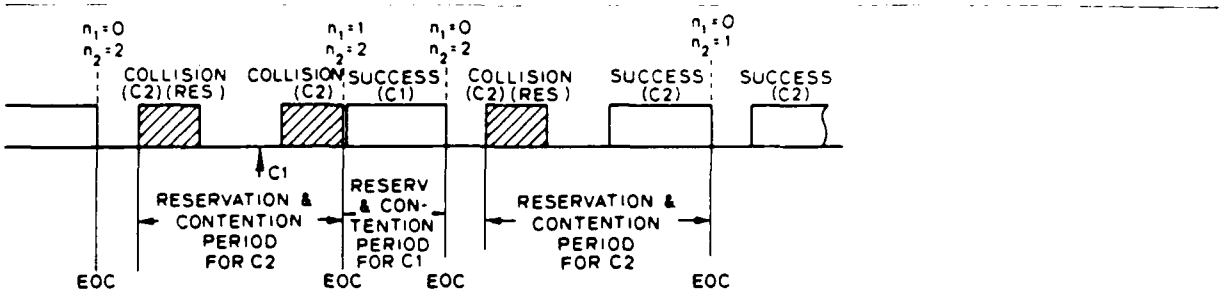


Fig. 5. Channel Activity for 1-persistent *P*-CSMA.

benefit. In environments where a collision detection feature is available and the collision detection and recovery period is small (on the order of $2\tau + \gamma$, as is the case with Ethernet), 1-persistent CSMA is clearly superior to p-persistent CSMA.

4.3. A Semipreemptive P-CSMA Scheme

Consider that, after the reservation process has taken place, the channel has been assigned to class j. Assume that, before a transmission takes place, a message of level i, $i<j$, is generated at some subscriber s. The nonpreemptive scheme dictates that subscriber s await the next time reference before it can ascertain its (higher) level i. The semipreemptive scheme allows subscriber s to preempt the right of access to class j, as long as no transmission from class j has yet taken place, by simply transmitting the message. If the generation of the level i message takes place after a transmission period is initiated, subscriber s waits until end-of-carrier is detected. Both nonpreemptive and semipreemptive schemes are applicable, whether or not collision detection is in effect.

4.4. A Preemptive P-CSMA Scheme

The difference between this scheme and the semipreemptive P-CSMA is that a subscriber with a newly generated packet may also preempt an ongoing transmission of a lower-priority level by intentionally causing a collision. Clearly this scheme is only appropriate if collision detection is in effect! It can offer some benefit if lower priority classes have long messages. One may also envision an adaptive preemption scheme whereby an ongoing transmission is preemptive only if the already elapsed transmission time is short.

5. Priority Schemes for Ring Networks

In a unidirectional transmission system, unidirectionality lends itself to a simple ordering of the subscribers. In the algorithms described in this and the next sections, we make use of this ordering to assign the channel to the highest priority class and to resolve conflicts among messages of the same priority class. In this section we consider the ring architecture and describe a nonpreemptive scheme.

The use of tokens, as described in Section 2, results in round-robin scheduling, because the subscriber immediately following the last one to transmit is the first to identify the token at the end of the message; it is therefore first to have a chance to transmit its message. Depending on the load, it is possible to observe a concatenation of messages on the bus (i.e., messages appearing in tandem on the cable) and we assume that each subscriber can identify packet boundaries in such a concatenation. This token algorithm must be modified to fit a prioritized environment, because we must ensure that no low-priority message is transmitted if a higher-priority one is ready, even if the subscriber with the lower-priority message is the first to encounter the token. One way this could be achieved is by augmenting the token algorithm with the capability to intercept messages; that is, a ready subscriber will intercept the message currently being transmitted if its own message is of a higher priority.

Let us assume that each message is preceded by a number representing its priority (followed, as usual, by source and destination identifiers and by data portions). Let s denote an arbitrary subscriber with a ready message and let BOM(s) and EOM(s) denote the time at which subscriber s identifies the beginning and end of a message respectively. Further, let $p(s)$ denote the priority of subscriber s at time BOM(s) and p(msg) the priority level of the message currently

being transmitted. Subscriber s with a ready message will operate as follows:

(1) If, following EOM(s), a token is encountered this subscriber will replace the token with its own message followed by a token.

(2) If, following EOM(s), the beginning of a message (BOM) in a concatenation of messages is encountered then:

(a) If $p(s) > p(\text{msg})$, the current message will be intercepted, i.e., the subscriber will remove the current message from the ring and replace it with its own (higher-priority) message. If the intercepted message is followed by a token, the subscriber will also follow its message with a token.

(b) If $p(s) <= p(\text{msg})$ the subscriber will defer transmission, wait until the next EOM, and then repeat this algorithm (step 1).

In many cases the round-trip propagation delay on the ring is shorter than the length of a message; hence an intercepting message may arrive at the originator of the intercepted message while it is still transmitting. In these cases, we require that a subscriber whose message has been intercepted abort its current transmission.

The critical element in this algorithm lies in the replacement of the intercepted message by the intercepting one. Such substitution implies that the intercepted message has been lost and will have to be retransmitted. Under certain circumstances, this interception can lead to a distortion of the order of messages on the ring. Assume for example, that three messages form a concatenation on the ring with corresponding priorities of 1 (highest), 3, and 4. A ready subscriber with $p(s) = 2$ will defer to the first message and will intercept the second message in the sequence. The third message of priority 4 will remain unaffected and can arrive at its destination. The net result is that a message of priority 4 (lowest) is successfully transmitted while a higher priority message waits. This reordering does not quite contradict the hierarchical-independence requirement of priority schemes (Section 3) since it was the high-priority message that intervened and caused this reordering. Nevertheless, if a stricter ordering is desired, the following is proposed.

A possible solution is to separate priority class resolution from the message transmission in a manner similar to that of the P-CSMA algorithm described in the previous section. After end-of-message (EOM), ready subscribers commence a reservation period, during which the channel behaves like in the interception algorithm described above. At the end of this period exactly one subscriber from the current highest-priority class is identified and will transmit its message.

Each ready subscriber prepares a reservation message containing only the priority level $p(s)$, without any source and destination information. At the EOM, if a token is encountered on the ring, a ready subscriber replaces it with its reservation message (with no following token). Otherwise, if a reservation message is encountered, the subscriber intercepts it or defers to it depending whether $p(s)$ is higher or lower than the current reservation message. Accordingly, at most one reservation message exists on the ring at any one time; after one round trip it contains the highest priority of all ready subscribers.

After one round trip, each ready subscriber (referred to as in the READY state) has had a chance to place its reservation; the ready subscribers can be divided into two groups – those that did not intercept a reservation message and remained in the READY state, and those that did intercept a reservation message and moved to the ALERT state because they potentially belong to the current highest-priority class. It should be noted that there can be at most one subscriber of each priority class in the ALERT state. The ALERT subscriber of the highest priority is the one that generated the current reservation message. We therefore continue to circulate the reservation message until it reaches the subscriber that generated it. The latter will then replace it with its true message, followed by a token.

In sum, therefore, a subscriber in the READY state waits until EOM and then operates as follows:

(1) If a token is identified, it is replaced by the reservation message.

(2) If a reservation message is identified and if $p(s) > p(\text{msg})$, i.e., this subscriber has a message of higher priority than the current one, the current reservation message will be replaced by a reservation for $p(s)$. The subscriber moves to the ALERT state.

(3) If $p(s) \leqslant p(\text{msg})$, the subscriber remains in the READY state and wait until EOM, at which time the algorithm is repeated (step 1 above).

And a subscriber in the ALERT state will operate as follows:

(1) If a reservation message is received with $p(s) = p(\text{msg})$ the subscriber removes that reservation message, transmits its data message followed by a token, and resets its state.

(2) If a reservation message is received with $p(s) \neq p(\text{msg})$ the subscriber moves to the READY state and repeats the algorithm.

According to the above, the reservation message must go through at least one round trip and at most two. The first round trip serves to identify the priorities of the ready subscribers. The reservation message stops after this round trip if the first subscriber is also of the highest priority, but continues for one additional full round trip if the highest priority subscriber is the last one. Note that the reservation packet is very short and that the comparison between $p(s)$ and $p(\text{msg})$ can be done in each subscriber within a delay interval of only one bit! Since reservation messages are all of equal length, message replacement as called for in step (2) of the algorithm can be accomplished easily.

In both of the above algorithms the subscriber immediately following the one currently transmitting is the first to identify an EOM condition. It thus has the opportunity to be first to transmit a message. Since both algorithms are memoryless across priority classes, this occurs regardless of the subscriber's priority class and results in unfairness. This can be illustrated by the following example. Assume subscribers are numbered S_1, S_2, S_3 ... and that S_1 has a message of priority 1 (highest), whereas the rest have a message of priority 2. After subscriber S_1 has transmitted its message, S_2 transmits. If, at this point, S_1 has another-high priority message, it will be transmitted (and rightfully so) but after that transmission S_2 will have a second chance to transmit a message before S_3 gets its first chance.

Remedying this situation requires that more information be transmitted and that state information be remembered for longer periods. We introduce a 'deferral' flag to assist in administering a fair round-robin scheme across priority classes. The flag (one per priority class) distinguishes within each priority class between ACTIVE subscribers that did not transmit a message in this round and DORMANT subscribers that did. The reservation message is composed of the pair (p, f), where p is the priority level and f the state of the flag. During the reservation period, the pair (p, ACTIVE) is considered higher than (p, DORMANT); consequently, those subscribers whose flag for the given priority indicate ACTIVEness have a chance to transmit their messages before other (DORMANT) subscribers are accorded their second chance. After transmitting a message an ACTIVE subscriber becomes DORMANT, and when the channel is granted to a reservation of the type (p, DORMANT), all subscribers reset the flag and become ACTIVE with respect to priority p.

Both of the above schemes can be made preemptive. Full preemption can be introduced in the first algorithm by allowing a subscriber to intercept other than whole messages. In the second algorithm full preemption can be achieved by jamming the channel and forcing a renewed reservation period. In both cases we assume that subscribers can distinguish between successful and intercepted messages. This can be done easily by such means as checksums or acknowledgments.

6. Priority Scheme for Unidirectional Broadcast Systems

In this section we describe a priority scheme for a UBS with a round-robin scheduling discipline within each priority class. A round robin is an inherently circular mechanism and can therefore be applied to a ring structure in a straightforward way since one can decide dynamically where the ring should start, changing that point for every new cycle, as desired. In a UBS, however, the physical ordering does not enjoy the circular symmetry of the ring and special steps must therefore be included in the algorithm to compensate for its absence.

The UBS considered here has two separate channels – the outbound channel which all subscribers access to transmit, and the inbound channel which subscribers access to read the transmitted information. In addition to transmission capability on the outbound channel, we assume that subscribers can also sense activity on that channel in a way similar to that required in other channel sensing systems, such as CSMA. In a UBS this capability results in an interesting feature. Assume subscribers are numbered sequentially S_1, S_2, S_3, etc. and that subscriber S_1 is defined as the 'farthest', i.e., has the longest round trip delay (see Fig. 1). Because of the unidirectional-signaling property, S_2 is able to sense signals generated by S_1 on both the inbound and outbound channels whereas the converse does not hold; that is, S_1 can sense signals generated by S_2 only on the inbound channel. This asymmetry will be utilized in establishing the ordering in a round-robin scheme.

We first describe the mechanism that allows implementation of a general and efficient round-robin scheduling discipline in a nonprioritized environment.

We shall then discuss the applicability of this mechanism to a prioritized environment.

6.1. An Efficient Round-Robin Algorithm

In the scheme described here, a subcriber is considered in one of three states. A subscriber is in the IDLE state if it does not have any message awaiting transmission. A non-IDLE subscriber, called a ready subscriber, can assume one of two states – ACTIVE if it has not transmitted its message in the 'current round' or DORMANT if it has transmitted and is now waiting for completion of the round. To achieve fair scheduling, DORMANT subscribers defer to all ACTIVE subscribers. Consequently, we are assured that no subscriber will transmit its second message before other ready subscribers have a chance to transmit their first ones. Eventually all ready subscribers will have transmitted their messages (i.e. all will have become DORMANT); this constitutes the end of a round, at which time all reset their state and a new round starts.

While each subscriber distinguishes between its DORMANT and ACTIVE states (with a 1-bit flag), arbitration among active subscribers must be provided by additional means. To that end each ACTIVE subscriber transmits a short burst of unmodulated carrier after the end of the previous message to indicate its ACTIVEness and, at the same time, it senses the outbound channel. All but one ACTIVE subscriber will sense the outbound channel busy (because of transmission from lower indexed subscribers – see Fig. 1) thus singling out the next subscriber to transmit. As explained earlier, we make use here of the asymmetry of the outbound channel. If a given subscriber senses the outbound channel busy, there exists at least one ready subscriber 'ahead' of it which generated that signal; a subscriber will always defer its transmission in favor of those 'ahead' of it.

Initially all subscribers reset their state, meaning that all ready subscribers are ACTIVE. An ACTIVE subscriber will operate as follows:

(1) It waits until the next end-of-carrier (EOC) is detected on the inbound channel at the end of a message.

(2) It transmits a short burst of unmodulated carrier and listens to the outbound channel for one round-trip delay.

(3) If the outbound channel is sensed idle during the entire period, the subscriber transmits its message and moves to the DORMANT state. Otherwise the subscriber repeats the algorithm.

A DORMANT subscriber will become ACTIVE if the inbound channel is sensed idle for one round trip delay or longer, and will then perform the above steps. A subscriber becoming ready after the channel has been idle for longer than one round-trip delay need not wait for an EOC but rather transmits its reservation burst immediately, i.e., starts the algorithm at step 2.

The algorithm is efficient because a conflict free scheduling is achieved with little overhead. The time separating two consecutive conflict free transmissions is between one and two round trip delays, allowing for both the EOC of the first message and the reservation burst to propagate through the system. The minimum, one round trip delay, occurs when the second of two consecutive transmissions is due to the highest index subscriber. The maximum, two round-trip delays, may occur when the second transmission within the same round is due to the subscriber with the lowest index. An extra (idle) round-trip delay is required to signal the end of a round to all subscribers – altogether a nominal overhead especially in a loaded system.

A separation of one round-trip delay between consecutive messages can be achieved at all times if following the transmission of its reservation burst a subscriber waits for a time equal to a full round trip delay minus the propagation time between its own outbound and inbound taps. The drawback of this approach is that each subscriber's parameters must be tuned according to its position on the cable.

The algorithm presented here differs slightly from a conventional round-robin algorithm. In a conventional round-robin discipline, each subscriber, in a prescribed order, is given a chance to transmit; it does so if it has a message ready and declines if it has none. This subscriber will not be given a second chance before all other subscribers have had their chance. In our algorithm, although no subscriber transmits more than once within each 'round', the order of transmission within the round may vary, depending on the specific instant a message arrives. For example, assume that subscriber S_1 has just completed transmission of its message. Assume also that at this moment S_2 does not have a message ready and therefore S_3, assumed to be ready, transmits next. While S_3 is transmitting, a message arrives at S_2; consequently S_2 will transmit when S_3 is finished. The order of transmission in this case was S_1, S_3, S_2, whereas if all the subscribers had had ready messages at the beginning of the round, the order would have been S_1, S_2, S_3.

6.2. A Prioritized Round-Robin Scheme for a UBS

In this section we adapt the round-robin algorithm described above to a prioritized environment. Hence we must modify the algorithm to ensure that fairness will be administered within each priority class and that high-priority messages will be transmitted first. To achieve this, contention among subscribers is resolved in two stages. First, ready subscribers exchange information regarding the priorities of their current messages (i.e., undertake priority class assessment) and then the round-robin algorithm described previously is used to resolve contention among subscribers of the current highest priority.

Here again we distinguish between ACTIVE and DORMANT subscribers, depending on whether they did or did not transmit a message in the current round. However, to achieve fair scheduling within each priority class, subscribers maintain separate states for each priority class; i.e., a subscriber can be ACTIVE with respect to one class and DORMANT with respect to another. Since only two states must be distinguished the total memory required is just one bit for each priority class.

All ready subscribers, ACTIVE or DORMANT, participate in priority class assessment. A mechanism similar to the one described in Section 4 may be used. When the priority assessment period is over, the current highest-priority class is established (independent of the internal state of the ready subscriber holding these messages); the channel is then considered to be operating at this priority level. Let p(channel) denote the latter. Those subscribers for which $p(s)$ differs from p(channel) refrain from proceeding while those for which these priorities are equal operate according to the round-robin algorithm described previously.

A ready subscriber will therefore wait until the next end of message and operate according to the following procedure:

(1) Participate in the priority class assessment (at which time p(channel) is established).

(2) If $p(s) \neq p$(channel), wait until the next EOC and then repeat the algorithm.

(3) If $p(s) = p$(channel) and the subscriber is ACTIVE with respect to this priority class, then:

(a) It transmits a short burst and listens to the outbound channel for one round-trip delay.

(b) If the outbound channel is sensed idle during this entire period the subscriber transmits its message and moves to the DORMANT state (with respect to this priority class). Otherwise, the subscriber repeats the algorithm (step 1).

(4) If $p(s) = p$(channel) and the subscriber is DORMANT with respect to this priority class then it senses the inbound channel for one round-trip delay and, if sensed busy, repeats the algorithm (step 1); otherwise it becomes ACTIVE with respect to p(channel) and performs step (3) above.

It is possible to achieve collision-free scheduling, even if we do not consider separate ACTIVE/DORMANT states for each priority class; i.e., a subscriber can be ACTIVE or DORMANT regardless of its priority class. The first stage, priority class assessment, still takes place to ensure that high-priority messages will be handled first. The algorithm then becomes memoryless across priority levels. This results in a slightly less fair ordering, such as the one described previously for a ring network. While for the ring architecture we do not recommend the introduction of separate states per priority class we do recommend it here for two reasons. In the algorithm presented for a ring architecture it meant a choice between requiring or not requiring memory altogether, whereas here it differs only by the amount of memory needed, which, in any case, is very small. Moreover, in a ring architecture there is no 'end of round' concept, which does exist in a UBS and entails overhead to handle. In contrast thereto, separation of states according to priority classes causes fewer end-of-round occurrences and thus reduces overhead.

The scheme presented here is nonpreemptive. A semipreemptive scheme in which a high-priority subscriber intervenes between the end of the priority class assessment period and the actual transmission is not meaningful, since the duration of the relevant time window is only one half round-trip delay which is too short an interval for any preemption activity. A full preemption scheme can be introduced by allowing a subscriber to jam the channel and force a new priority class resolution period.

7. Notes on Overhead

In general, a prioritized multiaccess scheme requires extra overhead in comparison with similar nonprioritized schemes. There are two principal reasons for this: the need to pass priority-related information among subscribers and the additional com-

plexity of the algorithm. While the extra overhead is compensated for by additional functionality of the system, minimization of its extent is still instrumental in achieving efficient communication. This section explores the factors that effect the overhead introduced by the above priority schemes.

The factors that most influence the extent of overhead are the relation between the priority assessment period (PAP) and the intraclass resolution period (ICRP), the representation of priority levels, and the degree of concurrence in the priority assessment process. Ideally one would like all subscribers to transmit at the same time the shortest code at the highest rate. Unfortunately, this cannot be done because not all of these features are independent and because some combinations place strict physical requirements on the system. A compromise is therefore mandatory – one that is based on the specific characteristic of the system being designed.

7.1. Separate versus Integrated PAP

In a distributed system, the priority level of a subscriber's message must generally be conveyed to all others (perhaps along with other scheduling information). This may be done by transmitting control data explicitly as part of a regular data message or in a dedicated control message, or by having the control information inferred from other actions taken by subscribers. This information provides basis for answering the two basic questions: what is the current highest-priority level and which of the subscribers holding such a message will transmit next? If information regarding the first of these is separated from the second, we observe a PAP followed by a regular (nonprioritized) channel access within the class. Otherwise priority assessment becomes an integral part of the channel access algorithm.

In principle, it is possible to merge the PAP and ICRP. Such an integrated assessment period is useful for schemes in which the ICRP requires explicit exchange of information, e.g., in polling or reservation-based schemes. In such schemes the addition of priority information to the rest of the information being exchanged affects performance only marginally and achieves a unified PAP and ICRP.

Merging of the PAP and the ICRP becomes impossible when random-access schemes are used. While a certain degree of fairness can be predicted for pure random-access schemes no specific ordering can be guaranteed. Using random-access schemes for priority assessment is therefore likely to cause violation of the hierarchical-independence rule (see Section 3); this can happen because portions of the scheduling are somewhat left to chance, in effect contradicting the deterministic nature of the rule. Consequently, a separate PAP is required when a random-access scheme is used for the ICRP.

7.2. Priority Representation

In general, the representation used to designate priority levels influences the trade-off between overhead incurred and the resulting performance at the various levels. One objective is to minimize delays in gaining access to the channel for subscribers with high-priority messages. This property is important if message delay for high priority classes is a critical measure of performance (this criterion was used in the P-CSMA scheme that was presented in Section 4). However, this may result in a limited overall channel utilization. To guarantee a low delay performance for high priority classes, their load on the channel must be limited – but the consequence of this is that the bulk of traffic falls into the lower-priority classes which incur high overhead during each priority assessment period.

One alternative is to have all ready subscribers start transmitting a reservation as soon as is permitted, but so that the higher the priority the longer will be the reservation duration. In this case, the current highest priority class gains access by persisting the longest.

Yet another alternative is to use a hierarchical reservation scheme (e.g., a tree priority resolution algorithm [13]), which is particularly effective if the number of priority levels is large and the total overhead incurred in such reservation processes is high). For example, in the priority assessment mechanism described for BBS (Section 4), a burst in the first slot designates that messages belonging to the highest group of priority levels are present. Following that, each level in the group is assigned its own slot for reservations, etc.

7.3. Shortening the PAP

The most decisive factors in shortening the PAP are concurrence and the rate at which priority information is exchanged among subscribers. A nonconcurrent scheme is one in which each subscriber announces its priority separately. If affords freedom to choose compact codes for the exchange of priority

information and allows high transmission rate. All in all, however, it causes long dealys – at least in proportion to the number of ready subscribers. At the opposite extreme, full concurrence means that all subscribers announce their priorities at the same time, which is clearly a desired mode of operation. However, depending on the data rate used, full concurrence introduces problems of synchronization. Indeed, precise bit-synchronization is imperative if control data are to be transmitted concurrently at the channel's normal rate of operation.

We are faced with a trade-off between full concurrence and the rate at which priority information is exchanged. Only when a high degree of synchronization can be guaranteed (a rare case indeed) can both high data rate and concurrence be achieved at the same time. One could relax the strict synchronization requirements by transmitting priority information at a lower than usual rate. If this is carried to the extreme, one would transmit at a rate of one bit per round-trip delay, at which point the synchronization problem is practically eliminated.

In a UBS, it is possible to achieve full synchronization by adjusting the clocks at all stations with staggered delays so that an exact overlap occurs when all subscribers transmit at a local time t. For example (referring back to Fig. 1), subscriber S_2 will start its transmission so that its first bit coincides exactly with the arrival of the first bit transmitted by subscriber S_1. In this case high data rates can be achieved at the (high) cost of strict synchronization.

In conjunction with synchronization and data rate, a proper choice must be made for representing priority levels in order to accommodate full concurrence. This choice depends, among other considerations, on the way subscribers access the channel since each subscriber must be able to transmit its data and retrieve relevant information from the channel's activity simultaneously. In the ring architecture, for example, where active taps are used, full concurrence (at a high data rate) can be achieved because collisions can be avoided and because each subscriber can examine and modify the message as it passes by. In such configurations one can use straight binary codes, which is a very compact such representation, to represent priority levels.

When passive taps are used and several subscribers transmit messages at the same time, a collision occurs. In most systems this would imply that none of the content of the colliding messages has been received. To allow for concurrence with passive taps, we assume that subscribers transmit at a relatively lower rate (e.g., one bit per round-trip delay). At this rate, synchronization is simple; a logical '1' is represented by a short burst of an unmodulated carrier while a logical '0' is the absence thereof. The state of the channel is the logical OR of all subscribers' transmissions from which each individual subscriber must be able to deduce the highest priority level present.

Priority information can be extracted from the ORed transmission if specific bases in the N-dimensional binary space are used to represent N priority levels. For example, one could use '1000', '0100', '0010', and '0001' to represent four priority levels. In fact, this is the code used in the P-CSMA scheme presented in Section 4 (except that trailing zeros are omitted for the sake of efficiency). Another example is '1111', '1110', '1100' and '1000' which constitutes the 'persistence' code mentioned earlier. Other such bases can of course be constructed to optimize other requirements.

8. Conclusion

This paper is concerned with the problem of introducing message-based priority functions to local networks. We have identified the major features of priority functions, the most important of which is the hierarchical independence of performance that specifies that messages of high priority must not only be delivered first but must remain unaffected by the load produced by lower priorities. In a distributed multiaccess system this becomes the central issue.

We have chosen three different local network architectures: the bidirectional broadcast system, the ring, and the unidirectional broadcast system; for each we presented a priority scheme that takes advantage of the particular architecture.

Finally, we have outlined and characterized the factors underlying the additional overhead caused by the introduction of priority schemes to a multiaccess environment. Shown as the most crucial of these factors are the separation between the priority assessment period and the intraclass resolution period, and the extent of synchronization.

References

[1] D.G. Willard, Mitrix – A sophisticated digital cable communications system, in: Proc. IEEE National Telecommunications Conf. (1977).

[2] A.G. Fraser, A virtual channel network, Datamation (2) (1975) 51–56.

[3] D.J. Farber et al., The distributed computing system, in: Proc. 7th Ann. IEEE Computer Society International Conf. (1973).

[4] R.M. Metcalfe and D.R. Boggs, ETHERNET: Distributed packet switching for local computer networks, Comm. ACM 19 (7) (1976) 395–403.

[5] N. Abramson, The ALOHA system – Another alternative for computer communications, Proc. Fall Joint Computer Conf. (AFIPS, 1970) 281–285.

[6] 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. Comm. (12) (1975) 1400–1416.

[7] F.A. Tobagi and V.B. Hunt, Performance analysis of carrier sense multiple access with collision detection, in: Proc. Local Area Communication Networks Symp. (The MITRE Corp., 1979) 217–244; and Computer Networks, Volume 4, No. 5, October/November 1980.

[8] D.C. Loomis, Ring communication protocols, Dept. of Computer Science University of California, Irvine (1973).

[9] J.R. Pierce, Network for block switching data, Bell System Tech. J. 51 (6) (1972) 1133–1145.

[10] A. Hopper, Data ring at Computer Laboratory University of Cambridge, in: Computer Science and Technology: Local Area Networking (National Bureau of Standards, Washington, DC, 1977).

[11] P. Zafiropoulo and E.H. Rothauser, Signaling and frame structures in highly decentralized loop systems, in: Proc. International Conf. on Computer Communications (Washington, DC, 1972) 309–315.

[12] M.T. Liu and C.C. Reames, Communication protocol and network operating system design for the distributed loop computer network (DLCN), in: Proc. 4th Ann. IEEE Symp. on Architecture (1977) 193–200.

[13] J.F. Hayes, An adaptive technique for local distribution, IEEE Trans. Comm. 26 (1978) 1178–1186.

Carrier Sense Multiple Access with Message-Based Priority Functions

FOUAD A. TOBAGI, MEMBER, IEEE

Abstract—**We consider packet communication systems of the multiaccess/broadcast type, exemplified by ETHERNET [1] and single-hop ground radio networks [2], in which all communicating devices share a common channel which is multiaccessed in some random fashion. Among the various random access schemes known, carrier sense multiple access (CSMA) has been shown to be highly efficient for environments where the propagation delay is short compared to the transmission time of a packet on the channel [3]–[5]. In this paper, we describe a new version of CSMA which incorporates message-based priority functions, referred to as prioritized CSMA (P-CSMA). The scheme is based on the principle that access right to the channel is exclusively granted to ready messages of the current highest priority level. It can be made preemptive or nonpreemptive, and is suitable to fully connected broadcast networks with or without the collision detection feature. We analyze the *p*-persistent protocol of P-CSMA with two priority levels and derive the throughput-delay characteristics for each priority class. Finally, we discuss numerical results obtained from the analysis and from simulation, and thus evaluate the effect of priority functions and preemption on the throughput-delay characteristics for each class.**

I. INTRODUCTION

IN multiaccess/broadcast systems, all users share a common transmission medium over which they broadcast their packets. Each subscriber is connected to the common communication medium through an interface which listens to all transmissions and absorbs packets addressed to it.

New multiaccess schemes for packet broadcasting systems have been abundant in recent years [6]. However, little work has been done to incorporate message-based priority functions to these protocols. The need for priority functions in multiaccess environments is a clear matter: having multiplexed traffic from several users and different applications on the same bandwidth-limited channel, we require that a multiaccess scheme be responsive to the particular requirements of each user and each application. For a prioritized scheme to be acceptable, we require the following:

1) The performance of the scheme as seen by messages of a given priority class should be insensitive to the load exercised on the channel by lower priority classes. Increasing loads from lower classes should not degrade the performance of higher priority classes.

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication after presentation at the National Telecommunications Conference, Houston, TX, December 1980. Manuscript received November 17, 1980; revised June 25, 1981. This work was supported by the Defense Advanced Research Projects Agency under Contract MDA903-79-C-0201, Order A0371?, monitored by the Office of Naval Research, and by the U.S. Army, CECOM, Fort Monmouth, NJ, under Army Research Office Contract DAAG 29-79-C-0138.

The author is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

2) Several messages of the same priority class may be simultaneously present in the system. These should be able to contend on the communication bandwidth with equal right (fairness within each priority class).

3) The scheme must be robust in the sense that its proper operation and performance should be insensitive to errors in status information.

4) The overhead required to implement the priority scheme, and the volume of control information to be exchanged among the contending users, as required by the scheme, must be minimal.

To implement priority functions in these distributed environments, one needs to address three basic problems: a) to identify the instants (which should be known to all users) at which to assess the highest current priority with ready messages, b) to design a mechanism by which to assess the highest nonempty priority class, and c) to design a mechanism which assigns the channel to the various ready users within a class. The scheme discussed in this paper is the *p*-persistent prioritized CSMA (P-CSMA), which consists of resolving the first two problems by the means of reservation bursts and carrier sensing, and the third by using the *p*-persistent carrier sense multiple access [3]-[6].

Two papers related to this topic have appeared in the literature. In the first by Franta and Bilodeau, the scheme consists of CSMA with different rescheduling delays assigned to the various devices; by staggering the delays, access right to the channel is prioritized across the devices and a gain in performance may be attained [8]. Unfortunately, the scheme as described does not provide priority functions which are based on the messages to be transmitted. The second by Onoe *et al.* does provide message-based priority functions via the use of different preambles for the various priority classes of messages [9], [10]. However, in case of a collision between two equal priority messages, these are rescheduled into the future, resulting in an operation which violates requirement 1) listed above.

In Section II, we give a precise description of the *p*-persistent P-CSMA protocol. In Section III, we provide an analysis of the scheme with two priority classes. The model allows us to derive the throughput-delay characteristics for each priority class. In Section IV, we discuss numerical results from the analysis and from simulation. Finally, in Section V, we make a few comments regarding variations of P-CSMA.

II. THE *p*-PERSISTENT P-CSMA PROTOCOL

A. Carrier Sense Multiple Access [3]-[6]

When dealing with multiaccess channels, one must be prepared to resolve conflicts which arise when more than one

0090-6778/82/0100-0185$00.75

demand is placed upon the channel. Whenever a portion of one user's transmission overlaps with another user's transmission, the two collide and "destroy" each other. CSMA reduces the level of interference caused by overlapping packets by having devices sense carrier due to other users' transmissions, and inhibit transmission when the channel is in use. Packets which either are inhibited or suffer a collison are rescheduled for transmission at a later time according to some rescheduling policy. There are several CSMA protocols [6]. We begin with a description of the p-persistent CSMA protocol since it forms the basis of the scheme described in this paper.

In the p-persistent CSMA protocol, a ready terminal senses the channel and operates as follows.

1) If the channel is sensed idle, then it transmits the packet.

2) If the channel is sensed busy, it waits until it becomes idle (at the end of the current transmission) and then with probability p the terminal transmits the packet, and with probability $1 - p$ the terminal delays the transmission of the packet by τ seconds, where τ is the maximum propagation delay among all pairs of terminals. If at this new point in time the channel is still detected idle, the same process is repeated. Otherwise, some packet must have started transmission. In this case we may use one of two versions: either a) the terminal in question schedules the retransmission of the packet according to the retransmission delay distribution (i.e., acts as if it had conflicted and learned about the conflict, at which later time it repeats the algorithm); or b) the terminal in question repeats step 2). [In this paper, we use version b).]

Packet broadcasting technology has also been shown to be very effective in satisfying many of the local area in-building communication requirements. A prominent example is ETHERNET, a local area communication network which uses CSMA on a tapped coaxial cable to which all the communicating devices are connected [1]. The device connection interface is a passive cable tap so that failure of an interface does not prevent communication among the remaining devices. The use of a single coaxial cable achieves broadcast communication. The only difference between a broadcast bus system and a single-hop radio system is that on a bus, in addition to sensing carrier, it is possible for the transceivers to detect interference among several transmissions (including their own), and to abort the transmission of colliding packets. This is achieved by having each transmitting device compare the bit stream it is transmitting to the bit stream it sees on the channel. This variation of CSMA is referred to as carrier sense multiple access with collision detection (CSMA-CD) [5].

In all CSMA protocols, given that a transmission is initiated on an *empty* channel, it is clear that it takes at most one end-to-end propagation delay τ for the packet transmission to reach all devices; beyond this time the channel is guaranteed to be sensed busy for as long as data transmission is in progress.[1] A collision can occur only if another transmission is initiated before the current one is sensed, and it will then take, at most, one additional end-to-end delay before interference "reaches" all devices. For CSMA-CD, we let ξ denote the time it takes a device to detect interference once the latter has reached it. ξ depends on the implementation and can be as small as 1 bit for transmission time, as is the case with ETHERNET [1]. Furthermore, ETHERNET has a collision consensus reinforcement mechanism by which a device, experiencing interference, jams the channel to ensure that all other interfering devices detect the collision. We denote by ζ the period used for collision consensus reinforcement. Given that a collision has occurred in CSMA-CD, the time until all devices stop transmission T_c is thus given by[2]

$$T_c = 2\tau + \xi + \zeta.$$

The time until the channel is again sensed idle by all devices is clearly $T_c + \tau$.

B. Basic Mechanism for Priority Assessment (a Nonpreemptive Discipline)

With the broadcast nature of transmission, users can monitor the activity on the channel at all times. The assessment of the highest priority class with ready messages is done at the end of each transmission period, whether successful or not, i.e., every time the carrier on the channel goes idle. When detected at a user, end of carrier (EOC) establishes a time reference for that user. Following EOC, the channel time is considered to be slotted with the size of a slot (referred to as *reservation-slot*) equal to $2\tau + \gamma$, where γ is the period of time of the shortest burst of unmodulated carrier which can be reliably detected. At each user, messages are ordered according to their priority. The priority of a user at *any time* is the highest priority class of messages present in its queue.

Let h denote an arbitrary user, and $t_e(h)$ denote the time of end of carrier at user h. Let $\nu(h)$ denote the priority level of user h at time $t_e(h)$. The priority resolution algorithm consists of having user h operate as follows.

1) If, following $t_e(h)$, the carrier is detected in reservation-slot i, with $i < \nu(h)$ (thus meaning that some user(s) has priority i higher than $\nu(h)$ and access right must be granted to class i), then user h awaits the following end of the carrier (at the end of the next transmission period) at which time it reevaluates its priority and repeats the algorithm.

2) If no carrier is detected prior to the jth reservation-slot, where $j = \nu(h)$, then user h transmits a short burst of unmodulated carrier of duration γ at the beginning of reservation-slot j [thus reserving channel access to priority class $\nu(h)$] and, immediately following this reservation-slot, operates according to the p-persistent CSMA protocol. That is, it senses the channel and a) if the channel is sensed idle, then with some probability p it transmits the message, and with probability $1 - p$ it delays action by τ seconds and repeats the CSMA procedure; b) if the channel is sensed busy, then the user awaits the next EOC and reevaluates its priority level and repeats the entire algorithm; c) if, during the time that channel access is granted to class $\nu(h)$, some user h' generates a (*new*) message of the same priority level, then h' transmits its message with proba-

[1] We assume that the sensing operation is instantaneous on this (high-bandwidth) channel.

[2] This assumes that all interfering devices undertake the collision consensus reinforcement.

bility one, provided that the channel is sensed idle. If, however, the channel is sensed busy at the message generation time, then h' awaits EOC, reevaluates its priority level and executes the algorithm. (Thus when a message is generated, the user undertakes *immediate first transmission* provided that the channel is idle and channel access is granted to the priority class corresponding to the newly generated message.)

3) If, following EOC, no reservation burst is detected for K consecutive reservation-slots, where K is the total number of priority classes available in the system, then the channel becomes free to be accessed by all users regardless of their priority, until a new EOC is detected.

Thus, by the means of short burst reservations following EOC, the highest nonempty priority class is granted exclusive access right, and messages within that class can access the channel according to p-persistent CSMA. Note that the above algorithm corresponds to a nonpreemptive discipline, since a user which has been denied access does not reevaluate its priority until the next EOC. However, by assessing the highest priority level at the end of each transmission period, whether successful or not, the scheme allows higher priority messages to regain the access right without incurring substantial delays.

We illustrate this procedure in Fig. 1 by displaying the activity on the channel. In this and all subsequent figures, we consider that there are only two possible priority levels in the system, and we denote by n_1 and n_2 the number of active users at EOC in class 1 (C_1) and class 2 (C_2), respectively. We adopt the convention that C_1 has priority over C_2. We also show a reservation burst as occupying the entire reservation-slot in which it is transmitted. Finally, we represent by a vertical upward arrow the arrival of a new message to the system; the label C_1 or C_2 indicates the priority class to which the message belongs. We assume in Fig. 1 that at the first EOC we have $n_1 = 0$ and $n_2 > 0$. Following EOC, a reservation burst is transmitted in the second reservation-slot. The priority resolution period, also called *priority assessment period* (PAP), is in this case equal to two reservation-slots. Following the reservation, we observe a *channel access period* (CAP) which consists of the idle time until the channel is accessed by some user(s) in class 2. Following CAP we observe the transmission period (TP) itself, the end of which establishes the new EOC time reference. (A crosshatched TP signifies a collision.) The time period between a reservation and the following EOC, called the *contention period* and equal to CAP + TP, is the time period during which exclusive access right is given to the class which succeeded in reserving the channel. In this nonpreemptive case, the message arrival labeled C_1, although of higher priority, is not granted access right until the EOC following its arrival, at which time it reserves the channel.

Note that the overhead incurred in a reservation period following EOC is a function of the currently highest priority level. The higher this class is, the smaller the overhead and the delay to gain access right.

The scheme is robust since no precise information regarding the demand placed on the channel is exchanged among the users. Information regarding the existing classes of priority is implied from the position of the burst of unmodulated carrier following EOC. Note also that there is no need to synchronize all users to a universal time reference. By choosing the reservation-slot size to be $2\tau + \gamma$ we guarantee that a burst emitted by a transceiver in its kth reservation-slot is received within the kth reservation-slot of all other users.

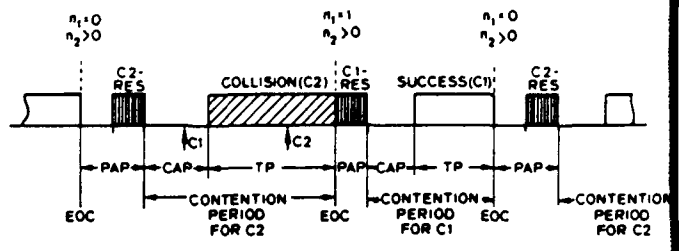


Fig. 1. Nonpreemptive p-persistent P-CSMA.

C. Preemptive P-CSMA

Consider that after the reservation process has taken place the channel has been assigned to class j. Assume that before a transmission takes place a message of level i, $i < j$, is generated at some user h. The nonpreemptive scheme dictates that user h awaits the next time reference before it can ascertain its (higher) level i. The *semipreemptive* scheme is one which allows user h to preempt access right to class j, as long as no transmission from class j has yet taken place, by simply transmitting the message (the transmission starting during the idle time representing CAP). If the generation of the message of level i takes place after a transmission period is initiated, then user h waits until end of carrier is detected. Both nonpreemptive and semipreemptive schemes are applicable whether collision detection is in effect of not.

A *fully preemptive* P-CSMA scheme is also defined in which a host with a newly generated packet may also preempt an ongoing transmission of a lower priority level by intentionally causing a collision. Clearly this scheme is only appropriate if collision detection is in effect! It can offer some benefit if lower priority classes have long messages. One may also envision a partial preemption scheme whereby an ongoing transmission is preempted only if the already elapsed transmision time has not exceeded some fraction of the total transmission time, where the packet transmision time is assumed to be known, as is the case with fixed size packets.

III. ANALYSIS OF THE NONPREEMPTIVE p-PERSISTENT P-CSMA

The difficulty in analyzing multiaccess schemes such as CSMA and P-CSMA arises from the fact that the system's service rate is at all times dependent on the system's state and its evolution in time; for example, the time required to successfully transmit a message is a function of the evolution of the number of contending users during the lifetime of the message. This prevents us from using conventional priority queueing results. To analyze P-CSMA we adopt the "feedback model" previously used to analyze CSMA [4], [5]. The analysis then relies on properties of semi-Markov processes, regenerative processes, and delay-cycle analysis. In this section, we present the analysis for the nonpreemptive case; preemptive P-CSMA can be handled in the same way.

A. The Model

Although the real operation of the scheme does not require time synchronization of all devices, it is assumed here, for simplicity in analysis, that the channel axis is slotted, with the slot size equal to τ seconds, and that all users are synchronized to the same universal time axis. In particular, they begin transmission only at slot boundaries. We furthermore neglect the effect of γ, ξ, and ζ. These, however, can be easily taken into account by redefining the slot size. With these definitions, a slot is τ seconds; a reservation-slot is 2τ seconds or two slots. Moreover, due to the extreme complexity of the analysis, we restrict ourselves here to only two classes of priority, C_1 and C_2, again with the convention that C_1 is of higher priority than C_2. We consider a population of M users such that a subset of size $M_1 \leq M$ generates messages of priority 1, and a subset of size $M_2 \leq M$ generates messages of priority 2. We allow $M_1 + M_2$ to be greater than M, meaning that some users generate both high and low priority messages. Each user is assumed to have, at any time, at most one message of each priority class. A new message of a given priority class cannot be generated at the user until the previous one has already been successfully transmitted. Thus, with respect to each priority class $C_j (j = 1, 2)$, a user can be in one of two states: backlogged or thinking. In the thinking state, a user generates (and possibly transmits, as dictated by the p-persistent P-CSMA procedure) a new message (of priority j) in a slot with probability a_j. With respect to class C_j, a user is said to be backlogged if it has a message of class C_j in transmission, or awaiting transmission. It remains in that state until it completes successful transmission of the message following the p-persistent P-CSMA procedure with parameter p_j, at which time it switches to the thinking state. For each class C_j, we let $n_j(t)$ denote the number of backlogged users in slot t. The number of users in the thinking state is then $M_j - n_j(t)$.

Let t_e again denote the time of EOC, and let $(n_1(t_e), n_2(t_e))$ denote the state of the system at t_e. As long as $n_1(t_e)$ or $n_2(t_e)$ is nonzero, EOC is followed by a priority assessment period and a contention period. The latter is for class C_1 if $n_1(t_e) \neq 0$, and for class C_2 if $n_1(t_e) = 0$ and $n_2(t_e) \neq 0$. [It is assumed here that a user does not update its priority during the priority resolution period; thus the PAP is entirely determined by $(n_1(t_e), n_2(t_e))$.] The interval of time between two consecutive EOC's is called a *subcycle*. A subcycle is referred to as *C_j-subcycle*, $j = 1, 2$, if the contention period is for C_j-messages. Examples of C_1- and C_2-subcycles are depicted in Fig. 2. When $n_1(t_e) = n_2(t_e) = 0$, then the subcycle is referred to as a *C_0-subcycle*. The various cases of C_0-subcycles are depicted in Fig. 3. Let T_j denote the length in slots (assumed fixed) of a message of class C_j. We let TP_j denote a transmission period of class C_j. If the transmission of the message is successful, then $TP_j = T_j + 1$ (where the additional slot accounts for the propagation delay since it is only one slot after the end of transmission that the channel will be sensed idle by all users); and if the transmission of the message is unsuccessful, then $TP_j = T_c^{(j)} + 1$, with $T_c^{(j)} = T_c$ in the case the collision detection feature is in effect, and $T_c^{(j)} = T_j$, otherwise.

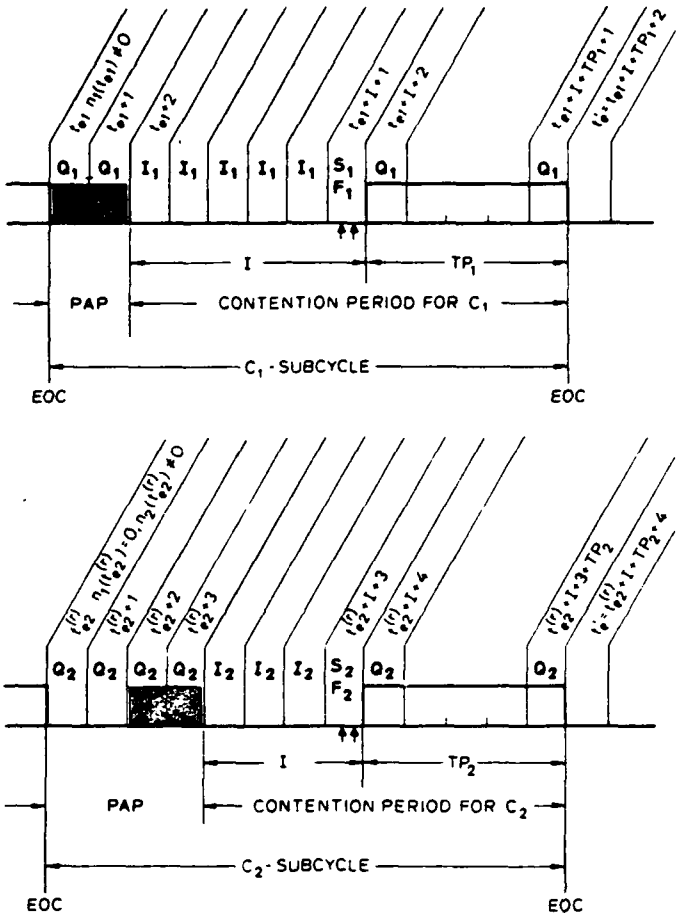


Fig. 2. C_1- and C_2-subcycles.

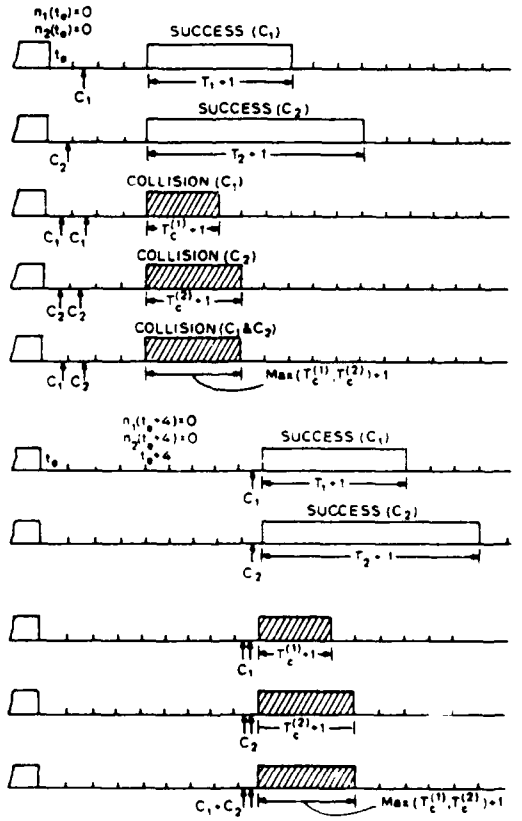


Fig. 3. The various situations arising after empty backlogs.

Let $\{t_{e1}^{(r)}\}_{r=-\infty}^{\infty}$ denote the sequence of all EOC's, also called C_1-*imbedded points*; and let $\{t_{e2}^{(r)}\}_{r=-\infty}^{\infty}$ denote the sequence of EOC's such that $n_1(t_{e2}^{(r)}) = 0$, also called C_2-*imbedded points*. Given our model, the imbedded processes $\{n_j(t_{ej}^{(r)})\}_{r=-\infty}^{\infty}$, $j = 1, 2$, are two interacting Markov chains. Let P_j denote the transition probability matrices and let $\Pi_j = \{\pi_0^{(j)}, \pi_1^{(j)}, \cdots, \pi_{M_j}^{(j)}\}$, where $\pi_k^{(j)} = \lim_{r\to\infty} \Pr\{n_j(t_{ej}^{(r)}) = k\}$, denote the stationary distributions.

B. Stationary Performance Measures

Define a C_j-*cycle* to be the time between two successive C_j-imbedded points. From the theory of regenerative processes, we can state that the average stationary channel throughput for class j, S_j, is computed as the ratio of the average time that the channel is carrrying successful C_j-transmissions during a C_j-cycle to the average duration of a C_j-cycle. Similarly, the average channel backlog N_j is computed as the ratio of the expected sum of the backlog over all slots in a C_j-cycle to the average duration of a C_j-cycle. Letting $P_s^{(j)}(k)$ denote the probability of a successful C_j-transmission during a C_j-cycle given $n_j(t_{ej}^{(r)}) = k$ (and clearly there is exactly one C_j-transmission in a C_j-cycle) and letting $E[\cdot]$ denote the expectation of the random variable following the letter E, we have

$$S_j = \frac{\sum_{i=0}^{M_j} \pi_i^{(j)} P_s^{(j)}(i) T_j}{\sum_{i=0}^{M_j} \pi_i^{(j)} E[t_{ej}^{(r+1)} - t_{ej}^{(r)} \mid n_j(t_{ej}^{(r)}) = i]} \tag{1}$$

$$N_j = \frac{\sum_{i=0}^{M_j} \pi_i^{(j)} E\left[\sum_{t=t_{ej}^{(r)}}^{t_{ej}^{(r+2)}} n_j(t) \mid n_j(t_{ej}^{(r)}) = i\right]}{\sum_{i=0}^{M_j} \pi_i^{(j)} E[t_{ej}^{(r+1)} - t_{ej}^{(r)} \mid n_j(t_{ej}^{(r)}) = i]}. \tag{2}$$

From Little's result, the average packet delay, normalized to T_j, is then simply expressed as

$$D_j = \frac{N_j}{S_j}. \tag{3}$$

In the remainder of this section, we give all the basic elements needed to evaluate (1)–(3).

C. The One-Step Transition Matrices for Processes $n_1(t)$ and $n_2(t)$

For an arbitrary matrix P, we adopt the notation $[P]_{i,k}$ to represent its (i, k)th element. For $j = 1, 2$, let I_j denote the identity matrix of dimension $(M_j + 1)$. Consider the matrices defined for $0 \leq i, k \leq M_j$ by

$$[S_j]_{i,k} = \begin{cases} \dfrac{(1-\sigma_j)^{M_j-i}[ip_j(1-p_j)^{i-1}]}{1-(1-p_j)^i(1-\sigma_j)^{M_j-i}} & k = i \\ \dfrac{(M_j-i)\sigma_j(1-\sigma_j)^{M_j-i-1}(1-p_j)^i}{1-(1-p_j)^i(1-\sigma_j)^{M_j-i}} & k = i+1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$[F_j]_{i,k} = \begin{cases} 0 & k < i \\ \dfrac{(1-\sigma_j)^{M_j-i}[1-(1-p_j)^i - ip_j(1-p_j)^{i-1}]}{1-(1-p_j)^i(1-\sigma_j)^{M_j-i}} & k = i \\ \dfrac{(M_j-i)\sigma_j(1-\sigma_j)^{M_j-i-1}[1-(1-p_j)^i]}{1-(1-p_j)^i(1-\sigma_j)^{M_j-i}} & k = i+1 \\ \dfrac{\binom{M_j-i}{k-i}(1-\sigma_j)^{M_j-k}\sigma_j^{k-i}}{1-(1-p_j)^i(1-\sigma_j)^{M_j-i}} & k > i+1 \end{cases} \tag{5}$$

$$[Q_j]_{i,k} = \begin{cases} 0 & k < i \\ \binom{M_j-i}{k-i}(1-\sigma_j)^{M_j-k}\sigma_j^{k-i} & k \geq i \end{cases} \tag{6}$$

$$[J_j]_{i,k} = \begin{cases} 1 & k = i-1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We note that these matrices are the *one-step* transition matrices for process $n_j(t)$ during a C_j-subcycle, where the correspondence to the slot is as shown in Fig. 2. Indeed, considering the slot immediately preceding the transmission period, $[S_j]_{i,k}$ is the probability of transition from state i to state k *and* the transmission period being successful (i.e., there being a single user becoming ready, given that some user became ready), and $[F_j]_{i,k}$ is the probability of transition from state i to state k *and* the transmission period being unsuccessful. During the priority assessment period and the transmission period, all new arrivals join the backlog, and thus the transition probabilities are given by Q_j. During the idle period, with the exception of its last slot, the backlog remains invariant since, according to the CSMA procedure described above, a new arrival sensing the channel idle would transmit with probability one; thus I_j is the corresponding one-step transition probability. Finally, J_j is introduced to represent the fact that a successful transmission decreases the backlog by 1. The one-step transition matrix for process $n_2(t)$ during a C_1-subcycle is simply Q_2 and vice versa. Similarly, one can establish the correspondence to slots in a C_0-subcycle. From Fig. 2, we also deduce that for $i \neq 0$ we have

$$[P_1]_{i,k} = [Q_1^2(S_1 Q_1^{T_1+1} J_1 + F_1 Q_1^{T_c^{(1)}+1})]_{i,k} \qquad i \neq 0. \tag{8}$$

D. Transition Matrix P_2

Consider first the case $n_2(t_{e2}^{(r)}) = i \neq 0$. The time separating $t_{e2}^{(r)}$ and $t_{e2}^{(r+1)}$ consists of the C_2-subcycle immediately following $t_{e2}^{(r)}$, and a succession of C_1-subcycles for as long as $n_1(t) \neq 0$. During all C_1-subcycles, new arrivals to class C_2 join the backlog. Let t_e' denote the time of the first EOC following $t_{e2}^{(r)}$, and let $L_m \triangleq t_{e2}^{(r+1)} - t_e'$ conditioned on $n_1(t_e') = m$. If $m = 0$, then $L_m = 0$ with probability 1. For

$m \neq 0$, in order to compute L_m, we consider the imbedded Markov chain $n_1(t_{e1}^{(v)})$ with the transition probabilities $\{[P_1]_{i,k}, i = 1, 2, \cdots, M_1; k = 0, 1, \cdots, M_1\}$ as expressed in (8), and the transition probability $[P_1]_{0,0} = 1$ to render state 0 an absorbing state. L_m is the time it takes process $n_1(t_{e1}^{(v)})$ to reach state 0 (for the first time), starting in state m.

The length of a C_1-subcycle is a function of the state of the semi-Markov chain before and after the transition. Let $t_{e1}^{(v)}$ and $t_{e1}^{(v+1)}$ be two consecutive imbedded points, and let $l_{i,j}$ (equal to $t_{e1}^{(v+1)} - t_{e1}^{(v)}$) denote the length of the C_1-subcycle, given that $n_1(t_{e1}^{(v)}) = i$ and $n_1(t_{e1}^{(v+1)}) = j$. The idle period in the C_1-subcycle is a function of the backlog at the end of the priority assessment period. Given that $n_1(t_{e1}^{(v)} + 2) = k$; the length of the idle period, denoted by $I_k^{(1)}$, is geometrically distributed. The z-transform $I_k^{(1)*}(z)$ of the probability mass function of $I_k^{(1)}$ is given by

$$I_k^{(1)*}(z) = \frac{(1-\delta_k^{(1)})z}{1-\delta_k^{(1)}z} \tag{9}$$

where

$$\delta_k^{(1)} = (1-p_1)^k(1-\sigma_1)^{M_1-k}. \tag{10}$$

The transmission period TP_1 equals $T_1 + 1$ or $T_c^{(1)} + 1$ depending on the success or failure of the transmission. Therefore, letting $l_{i,j}^*(z)$ denote the generating function of the probability mass function of $l_{i,j}$, we can write

$$\begin{aligned} l_{i,j}^*(z) = &\sum_{k=i}^{j+1} \frac{[Q_1^2]_{i,k}[S_1 Q_1^{T_1+1} J_1]_{k,j}}{[P_1]_{i,j}} \\ &\cdot \frac{(1-\delta_k^{(1)})z^{T_1+4}}{1-\delta_k^{(1)}z} \\ &+ \sum_{k=i}^{j} \frac{[Q_1^2]_{i,k}[F_1 Q_1^{T_c^{(1)}+1}]_{k,j}}{[P_1]_{i,j}} \\ &\cdot \frac{(1-\delta_k^{(1)})z^{T_c+4}}{1-\delta_k^{(1)}z}. \end{aligned} \tag{11}$$

Let $L_m^*(z)$ denote the generating function for L_m; due to the recursive nature of Markov chains, we can finally write

$$L_m^*(z) = \sum_{j=m-1}^{M} [P_1]_{m,j} l_{m,j}^*(z) L_j^*(z). \tag{12}$$

Equation (12) defines a system of M_1 equations in the M_1 unknowns $\{L_m^*(z)\}_{m=1}^{M_1}$. As it is difficult to solve this system symbolically for reasonable values of M, we numerically compute the distributions in question by successive iterations, starting with an arbitrary set of initial distributions. Note that the above system also allows us to compute very easily the moments of $\{L_m\}_{m=1}^{M_1}$. Indeed, the nth-order differentiation of (12) evaluated at $z = 1$ leads to a linear system relating the nth-order moments, with coefficients which are functions of the lower order moments. Given the special form of the transition matrix, namely the fact that $[P_1]_{i,k} = 0$ for $k < i - 1$, this linear system can then be solved recursively to obtain the nth-order moments once the lower order moments have been computed.

The number m of C_1-messages accumulating at t_e' is a function of the length of the C_2-subcycle. Given that $n_2(t_{e2}^{(r)} + 4) = k$, we let $I_k^{(2)}$ denote the length of the idle period. $I_k^{(2)}$ has the same distribution as $I_k^{(1)}$ in which the parameter $\delta_k^{(1)}$ is replaced by

$$\delta_k^{(2)} = (1-p_2)^k (1-\sigma_2)^{M_2-k}. \tag{13}$$

Success or failure in the transmission period is a direct function of $n_2(t_{e2}^{(r)} + 4)$. Given that $n_2(t_{e2}^{(r)} + 4) = k$ and that the transmission period TP_2 is successful, we denote by $Y_k^{(s)}$ the length of the C_2-cycle; $Y_k^{(s)}$ is equal to $4 + I_k^{(2)} + T_2 + 1$ and has a moment generating function $Y_k^{(s)*}(z)$ expressed as

$$Y_k^{(s)*}(z) = \frac{(1-\delta_k^{(2)})z^{T_2+6}}{1-\delta_k^{(2)}z}. \tag{14}$$

Similarly, we define $Y_k^{(f)}$ for the case of failure; its moment generating function is expressed as

$$Y_k^{(f)*}(z) = \frac{(1-\delta_k^{(2)})z^{T_c^{(2)}+6}}{1-\delta_k^{(2)}z}. \tag{15}$$

All C_1-messages arriving in the C_2-subcycle will accumulate at the end of the C_2-subcycle and initiate the sequence of consecutive C_1-subcycles. Given that the length of the C_2-subcycle is y slots, the probability that $n_1(t_e') = m$ is given by $[Q_1^y]_{0,m}$. Removing the condition on y, this probability is $[Y_k^{(s)*}(Q_1)]_{0,m}$ in case of success, and $[Y_k^{(f)*}(Q_1)]_{0,m}$ in case of failure. Given that $n_2(t_{e2}^{(r)}) = i$, the probability that $n_2(t_{e2}^{(r)} + 4)$ is k is simply $[Q_2^4]_{i,k}$. Given that $n_2(t_e') = m$ and $L_m = \alpha$, the transition matrix for process $n_2(t)$ over the entire sequence of C_1-subcycles is simply Q_2^{α}. Removing the condition on α, the latter becomes $L_m^*(Q_2)$. As a result, we can write the (i, j)th element of matrix P_2 for $i \neq 0$ as

$$\begin{aligned} [P_2]_{i,j} = \sum_k [Q_2^4]_{i,k} \Bigg[&\sum_{m=0}^{M_1} [Y_k^{(s)*}(Q_1)]_{0,m} \\ &\cdot S_2 Q_2^{T_2+1} J_2 L_m^*(Q_2) \\ &+ \sum_{m=0}^{M_1} [Y_k^{(f)*}(Q_1)]_{0,m} F_2 Q_2^{T_c+1} L_m^*(Q_2) \Bigg]_{k,j}. \end{aligned} \tag{16}$$

Consider now an imbedded point $t_{e2}^{(r)}$ for process $n_2(t_{e2})$ such that $n_2(t_{e2}^{(r)}) = 0$. We are seeking the elements $[P_2]_{0,k}$ of the transition matrix P_2. We distinguish several cases as shown in Fig. 3. The first five cases correspond to the situations in which some arrivals from either class C_1 or class C_2 or both occur in the priority assessment period, and thus

initiate a transmission immediately following the end of the priority assessment period; the remaining five cases correspond to the situations where no arrivals take place in the priority assessment period, and thus an idle period I_0 is observed before a transmission period is encountered. We let t_e' again denote the time of the first EOC following $t_{e2}^{(r)}$. Let $a(\alpha_1, \alpha_2)$ and $b(\alpha_1, \alpha_2)$ be defined as

$$a(\alpha_1, \alpha_2) \triangleq [Q_1^4]_{0,\alpha_1}[Q_2^4]_{0,\alpha_2}$$

$$0 \leq \alpha_1 \leq M_1; \quad 0 \leq \alpha_2 \leq M_2 \tag{17}$$

$$b(\alpha_1, \alpha_2) \triangleq a(0,0)\frac{[Q_1^2]_{0,\alpha_1}[Q_2^2]_{0,\alpha_2}}{1-[Q_1^2]_{0,0}[Q_2^2]_{0,0}}$$

$$0 \leq \alpha_1 \leq M_1; \quad 0 \leq \alpha_2 \leq M. \tag{18}$$

It is easy to see that the transition probabilities between $t_{e2}^{(r)}$ and t_e' are given by

$$\begin{aligned}
&\Pr\{n_1(t_e') = k_1, n_2(t_e') = k_2 \mid n_1(t_{e2}^{(r)}) = n_2(t_{e2}^{(r)}) = 0\} \\
&= [a(1,0) + b(1,0)][Q_1^{T_1+1}J_1]_{1,k_1}[Q_2^{T_1+1}]_{0,k_2} \\
&\quad + [a(0,1) + b(0,1)][Q_1^{T_2+1}]_{0,k_1}[Q_2^{T_2+1}J_2]_{1,k_2} \\
&\quad + \sum_{j=2}^{M_1}[a(j,0) + b(j,0)][Q_1^{T_c^{(1)}+1}]_{j,k_1} \\
&\quad \cdot [Q_2^{T_c^{(1)}+1}]_{0,k_2} \\
&\quad + \sum_{j=2}^{M_2}[a(0,j) + b(0,j)][Q_1^{T_c^{(2)}+1}]_{0,k_1} \\
&\quad \cdot [Q_2^{T_c^{(2)}+1}]_{j,k_2} + \sum_{j_1=1}^{M_1}\sum_{j_2=1}^{M_2}[a(j_1,j_2) \\
&\quad + b(j_1,j_2)][Q_1^{\max(T_c^{(1)},T_c^{(2)})+1}]_{j_1,k_1} \\
&\quad \cdot [Q_2^{\max(T_c^{(1)},T_c^{(2)})+1}]_{j_2,k_2}.
\end{aligned} \tag{19}$$

Note that if $n_1(t_e') = 0$, then $t_{e2}^{(r+1)} \equiv t_e'$, otherwise $t_{e2}^{(r+1)}$ is the first EOC following t_e' such that $n_1(t_e) = 0$. As a result we have

$$\begin{aligned}
[P_2]_{0,j} &= \Pr\{n_1(t_e') = 0, n_2(t_e') = j \\
&\quad \mid n_1(t_{e2}^{(r)}) = n_2(t_{e2}^{(r)}) = 0\} \\
&\quad + \sum_{k_1=1}^{M_1}\sum_{k_2=0}^{j}\Pr\{n_1(t_e') = k_1, n_2(t_e') = k_2 \\
&\quad \mid n_1(t_{e2}^{(r)}) = n_2(t_{e2}^{(r)}) = 0\}[L_{k_1}^*(Q_2)]_{k_2,j}.
\end{aligned} \tag{20}$$

Since $L_0^*(z) = 1$, adopting the convention $L_0^*(Q_2) = Q_2^0 = I_2$, we can express (20) as

$$\begin{aligned}
[P_2]_{0,j} &= \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{j}\Pr\{n_1(t_e') = k_1, n_2(t_e') = k_2 \\
&\quad \mid n_1(t_{e2}^{(r)}) = n_2(t_{e2}^{(r)}) = 0\}[L_{k_1}^*(Q_2)]_{k_2,j}.
\end{aligned} \tag{21}$$

E. Transition Matrix P_1

Equation (8) gives $[P_1]_{i,k}$ for $i \neq 0$. It remains to compute $[P_1]_{0,k}$. Given $t_{e1}^{(r)}$ such that $n_1(t_{e1}^{(r)}) = 0$ and $n_2(t_{e1}^{(r)}) = i$, we note that $t_{e1}^{(r)}$ corresponds also to a C_2-imbedded point, and therefore we have

$$\begin{aligned}
&\Pr\{n_1(t_{e1}^{(r+1)}) = k \mid n_1(t_{e1}^{(r)}) = 0, n_2(t_{e1}^{(r)}) = i\} \\
&= \begin{cases}
\sum_{j=i}^{M_2}[Q_2^4]_{i,j}[p_s^{(2)}(j)Y_j^{(s)*}(Q_1) \\
\quad + (1 - p_s^{(2)}(j))Y_j^{(f)*}(Q_1)]_{0,k} & i \neq 0 \\
\sum_{k_2=0}^{M_2}\Pr\{n_1(t_e') = k, n_2(t_e') = k_2 \\
\quad \mid n_1(t_{e2}) = n_2(t_{e2}) = 0\} & i = 0
\end{cases}
\end{aligned} \tag{22}$$

where $p_s^{(2)}(j)$ is the probability that the transmission period in a C_2-subcycle is successful when j backlogged users and $M_2 - j$ thinking users are contending, and is given in (28) below. We remove the condition $n_2(t_{e1}^{(r)}) = i$ by simply noting that, in steady state, the probability of this event is $\pi_i^{(2)}$. Thus, we get

$$\begin{aligned}
[P_1]_{0,k} &= \sum_{i=0}^{M_2}\pi_i^{(2)}\Pr\{n_1(t_{e1}^{(r+1)}) = k \mid n_1(t_{e1}^{(r)}) \\
&\quad = 0, n_2(t_{e1}^{(r)} = i\}.
\end{aligned} \tag{23}$$

F. Throughput-Delay Performance

To complete the evaluation of (1) and (2) we need to compute, for $j = 1, 2$, $P_s^{(j)}(i)$, the average duration of a cycle, and the expected sum of the backlog over the cycle, given that $n_j(t_{ej}^{(r)}) = i$. It is easy to see that the $P_s^{(j)}(i)$ are given by

$$P_s^{(1)}(0) = \pi_0^{(2)}[a(1,0) + b(1,0)] \tag{24}$$

$$P_s^{(2)}(0) = a(0,1) + b(0,1) \tag{25}$$

$$P_s^{(1)}(i) = \sum_{k=i}^{M_1}[Q_1^2]_{i,k}p_s^{(1)}(k) \qquad i \neq 0 \tag{26}$$

$$P_s^{(2)}(i) = \sum_{k=i}^{M_2}[Q_2^4]_{i,k}p_s^{(2)}(k) \qquad i \neq 0 \tag{27}$$

TABLE I
COMPARISON BETWEEN ANALYTIC AND SIMULATION RESULTS

P-CSMA-CD; $M_1 = M_2 = 5$; $T_1 = 10$; $T_2 = 100$; $T_c = 2$; $\sigma_1 = 0.0022$

p_1, p_2	σ_2	S_1		S_2		D_1		D_2	
		ANALYSIS	SIMUL.	ANALYSIS	SIMUL.	ANALYSIS	SIMUL.	ANALYSIS	SIMUL.
0.1	0.0002	0.106	0.099	0.098	0.100	20.41	21.94	114.0	116.56
	0.0010	0.100	0.094	0.429	0.431	48.14	49.53	165.2	165.16
	0.0016	0.097	0.096	0.587	0.573	63.56	62.25	219.7	226.57
	0.0020	0.096	0.098	0.649	0.658	69.79	68.31	259.7	272.27
0.2	0.0002	0.106	0.102	0.098	0.085	19.12	19.42	115.5	115.18
	0.0010	0.101	0.099	0.433	0.420	43.80	44.14	154.6	155.53
	0.0016	0.098	0.096	0.602	0.596	58.04	58.08	201.7	206.51
	0.0020	0.097	0.095	0.671	0.660	64.27	66.11	237.7	247.92
0.5	0.0002	0.106	0.101	0.098	0.101	18.78	19.51	110.3	113.25
	0.0010	0.101	0.096	0.434	0.429	42.95	45.41	152.3	149.17
	0.0016	0.098	0.097	0.602	0.594	56.79	58.05	201.4	203.46
	0.0020	0.098	0.096	0.668	0.663	62.70	65.03	241.6	252.17

where

$$p_s^{(j)}(k) = \frac{kp_j(1-p_j)^{k-1}(1-\sigma_j)^{M_j-k} + (M_j-k)\sigma_j(1-\sigma_j)^{M_j-k-1}(1-p_j)^k}{1-(1-p_j)^k(1-\sigma_j)^{M_j-k}}. \tag{28}$$

The expressions for the expected duration of a cycle and for the expected sum of backlogs over a cycle are straightforward and are given in the Appendix. We simply note that if $n_j(t) = k (k = 0, 1, \cdots, M_j)$ for some t, and $R_1, R_2, \cdots, R_l$ are the one-step transition matrices over l consecutive slots following t, then the expected sum of backlogs over the l slots is given by the kth element of vector $[(I_j + \sum_{v=1}^{l-1} R_1 R_2 \cdots R_v)H_j]$, where H_j is a column vector of $M_j + 1$ elements such that its transpose is $H_j^T = (0, 1, 2, \cdots, M_j)$.

IV. NUMERICAL RESULTS

We discuss in this section numerical results concerning the performance of P-CSMA. In addition to the analysis presented in Section III for the nonpreemptive case, simulation of P-CSMA has been performed [7]. The purpose of the simulation is twofold: 1) to cross-validate the results obtained from two models, and 2) to experiment with variations of the scheme, traffic patterns, and network loads which are not easily handled by the present analysis. For example, although the analysis of preemptive P-CSMA is feasible following the approach used in Section III, the effect of preemption has been studied by simulation, as the number of different situations which arise in the preemptive case pertaining to the occurrence of various events is larger than in the nonpreemptive case, and thus renders the analysis a more tedious exercise. Furthermore, the analysis presents some limitations on the size of the system, namely M_1 and M_2, and on the load offered to the channel, in particular from class C_1, for which the computations can be economically performed. The simulation is thus used to examine larger systems and to verify that the behavior of P-CSMA is the same in both small and large systems. It is also important to note that the cross-validation of results from both models is perhaps among the greatest benefits. The excellent agreement which is observed between the results obtained from both models (as shown in Table I) allows us to verify that a) both the analytic and simulation models are correct; b) the analysis is computationally feasible (and economically feasible for relatively small systems such as $M = 5$) in that the accuracy of the computations, especially in solving (12), is perfectly acceptable; and c) the length of the simulation runs and the accuracy of the simulation results are acceptable without the need to provide confidence intervals. Finally, note that since the behavior of p-persistent CSMA has been extensively studied in the past and thus is fairly well understood [4], [5], we focus in this paper on numerical results pertaining to the priority function and the effect of various system parameters on its performance.

A. Effect of the Transmission Probabilities p_1 and p_2

Typically, one is given the volume of traffic which needs to be carried for each class, that is, S_1 and S_2, and measures the performance of P-CSMA in terms of the average packet delays D_1 and D_2. Just as with CSMA without priority, for given values of S_1 and S_2 there are optimum values of p_1 and p_2 which lead to the minimum delays D_1 and D_2. The optimum throughput-delay characteristics of P-CSMA are given by the lower envelopes of all the constant (p_1, p_2) throughput-delay surfaces $D_1(S_1, S_2)$ and $D_2(S_1, S_2)$, where the latter are obtained by varying σ_1 and σ_2.

To study the sensitivity of packet delays to p_1 and p_2, we consider the nonpreemptive p-persistent P-CSMA-CD scheme on a broadcast bus with two classes of priority. We let $T_1 = 10$ slots (i.e., short C_1-packets), $T_2 = 100$ slots (i.e., long C_2-packets), and $T_c = 2$ slots (neglecting the parameters ξ and ζ). Fixing $S_1 = 0.1$, we plot in Fig. 4 D_1 and D_2 versus the total channel throughput $S_1 + S_2$ for $M_1 = M_2 = 5$ and various values of p_1 and p_2. We note that with $M_1 = M_2 = 5$, the value $p_1 = p_2 = 0.2$ gives near optimum performance over the entire range of achievable throughput. Smaller values of p_1 and p_2 may achieve slightly higher maximum total throughput $S_1 + S_2$ (also called channel capacity), but provide higher delays; larger values of p_1 and p_2 achieve decreasing values of

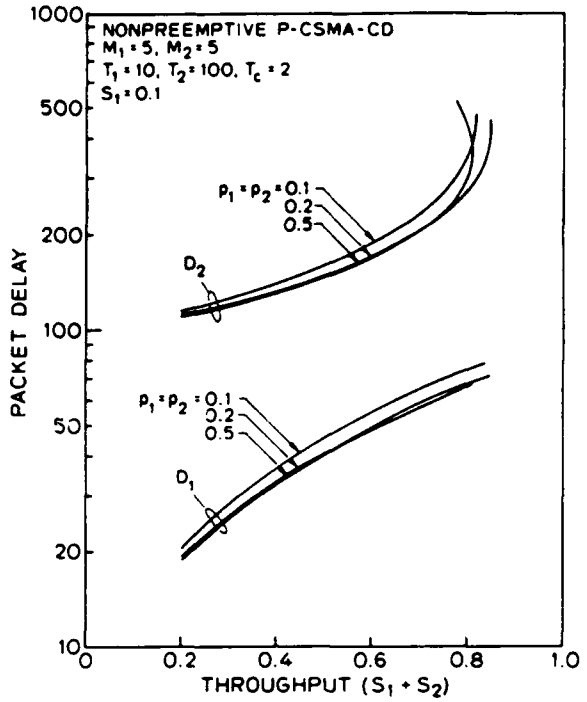


Fig. 4. Sensitivity of packet delay to p_1 and p_2 in nonpreemptive P-CSMA-CD with $M_1 = M_2 = 5$ (obtained from analysis).

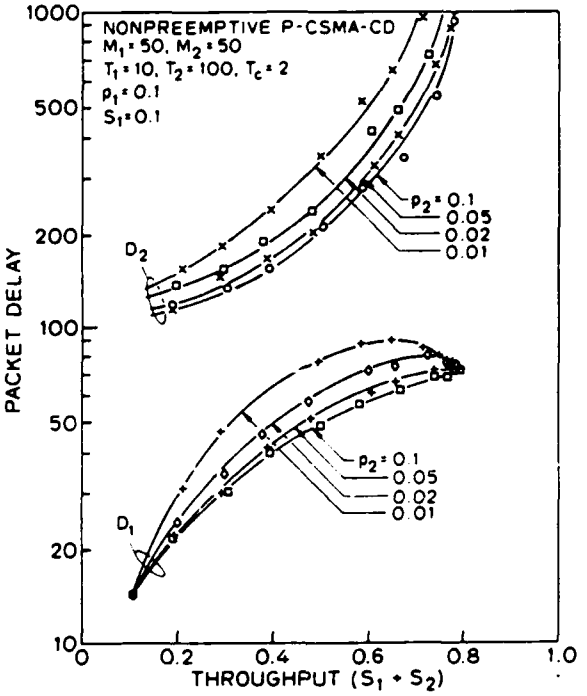


Fig. 5. Sensitivity of packet delay to p_2 in nonpreemptive P-CSMA-CD with $M_1 = M_2 = 50$ (obtained by simulation).

channel capacity without noticeable improvement in delay. Note, however, that in general for small systems ($M_1 = M_2 = 5$) D_1 and D_2 are fairly insensitive to changes in p_1 and p_2 falling in the range (0.1, 0.5). In the sequel we shall use $p_1 = p_2 = 0.2$ to plot near optimum performance of P-CSMA-CD with $M = 5$.

For larger systems such as $M_1 = M_2 = 50$ users, the choice of p_1 and p_2 becomes more critical, as can be seen in Fig. 5 which is obtained from simulation. Keeping S_1 relatively small (e.g., $S_1 = 0.1$), D_1 and D_2 are still fairly insensitive to p_1 as long as p_1 is reasonably selected, such as $p_1 = 0.1$; but the effect of p_2 is more important and the selection of an optimum p_2 is more crucial. These conclusions are not surprising and conform to the known behavior of nonprioritized CSMA [4], [5].

B. The Nonpreemptive Priority Function

We now examine the effect of the nonpreemptive priority function on the throughput-delay characteristics of each priority class. We again let $T_1 = 10$, $T_2 = 100$, and $T_c = 2$. For $M_1 = M_2 = 5$ and $p_1 = p_2 = 0.2$, we plot in Fig. 6 D_1 and D_2 versus $S_1 + S_2$ for various fixed values of S_1, namely $S_1 = 0.1$, 0.2, and 0.4. The solid curves correspond to the nonpreemptive P-CSMA-CD, while the dashed curves correspond to CSMA-CD without priority (obtained here by simulation). Since the packet delay includes the time of successful transmission of the packet, for this case where $T_2 > T_1$, D_2 is greater than D_1 for both CSMA-CD and P-CSMA-CD. The important point to make, however, is that as S_2 increases to reach saturation, D_1 increases in this nonpreemptive P-CSMA-CD but only to reach a *finite* average at saturation, while in CSMA-CD without priority, both D_1 and D_2 increase indefinitely. The increase in D_1 in nonpreemptive P-CSMA-CD

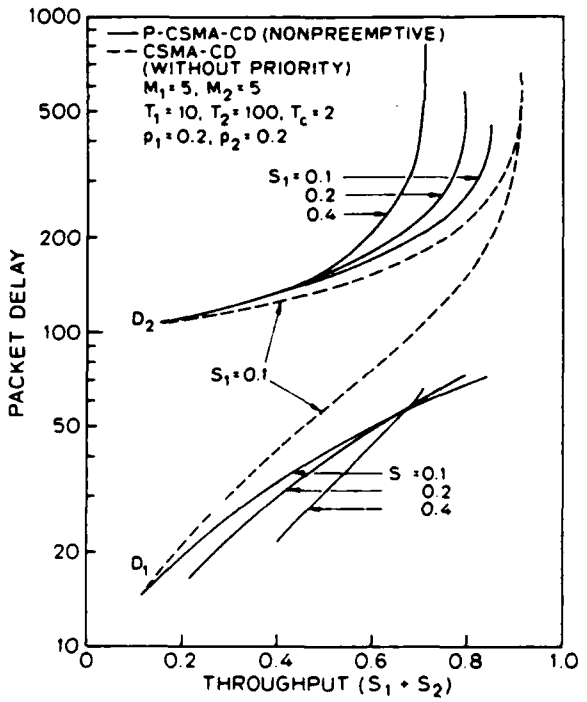


Fig. 6. Throughput-delay tradeoffs for nonpreemptive P-CSMA-CD with $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained from analysis).

depends on T_2 and is more important for larger T_2. In Fig. 7 where $T_1 = T_2 = 10$, we observe that the increase in D_1 is not as steep and the maximum delay D_1 reached is only $2.5T_1$. From Figs. 6 and 7, we also observe the effect on performance of the overhead incurred in implementing the priority function. Clearly, the price we pay for the priority function is more important with smaller packet sizes.

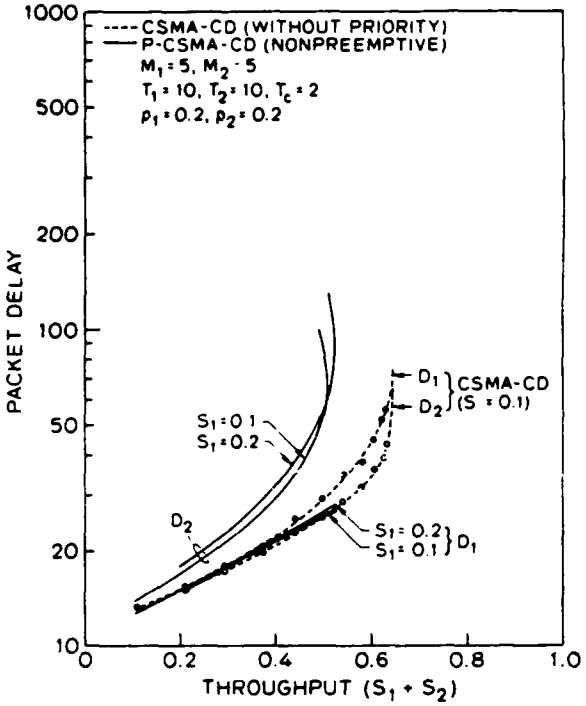


Fig. 7. Throughput-delay tradeoffs for nonpreemptive P-CSMA-CD with $M_1 = M_2 = 5$ and $T_1 = T_2 = 10$ (obtained from analysis), and CSMA-CD without priority (obtained from simulation).

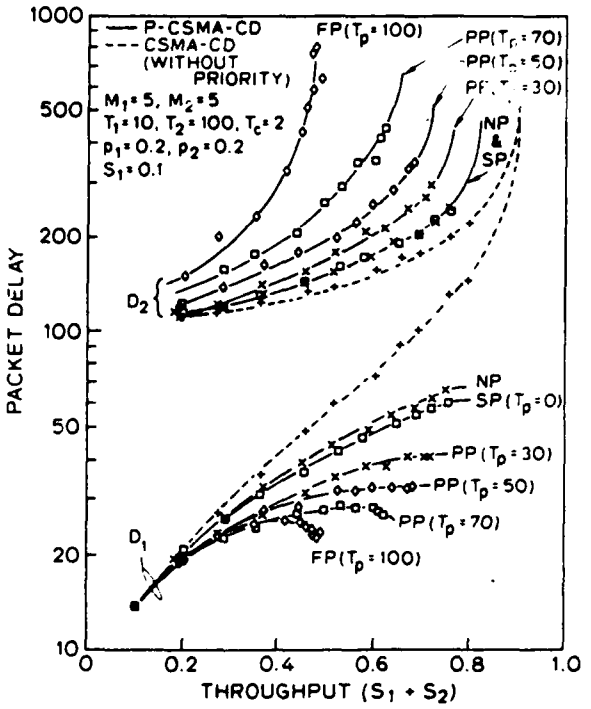


Fig. 8. Throughput-delay tradeoffs for nonpreemptive, semipreemptive, partially preemptive, and fully preemptive P-CSMA-CD with $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

C. Effect of Preemption on the Performance of P-CSMA-CD

The fact that in the nonpreemptive case the increase in D_1 as S_2 increases is greater with larger T_2 indicates that preemption will clearly improve the delay characteristics for class C_1. Denoting by T_p the period of time at the beginning of a C_2-transmission during which class C_1 is allowed to preempt, we plot in Fig. 8 the packet delays for the nonpreemptive (NP), semipreemptive (SP; $T_p = 0$), partial preemptive (PP; $0 < T_p < T_2$), and full preemptive (FP; $T_p = T_2$) cases for $M_1 = M_2 = 5$ and $T_1 = 10$, $T_2 = 100$. Fig. 9 displays the variance of packet delay for the same cases. An improvement in C_1-packet delay (for both expectation and variance) is clearly achieved, but at the expense of lower channel capacity. The degradation in channel capacity experienced as the degree of preemption gets higher is shown in Fig. 10.

Consider now the large system $M_1 = M_2 = 50$. In Fig. 11 we plot D_1 and D_2 for the NP, SP, and FP cases for $p_1 = 0.1$ and for various values of p_2 (namely, 0.01 and 0.05). It is interesting to observe that, contrary to the NP case, the average delay D_1 in the SP and FP cases is not sensitive to p_2. The same is true for the variance. This indicates that, if p_2 is not properly selected, the use of SP (and FP) safely provides the optimum performance for class C_1.

Preemption is most desirable when T_2 is larger than T_1. If T_2 is small, say $T_2 = 10$, the improvement in packet delay (and degradation in channel capacity) is more moderate than with larger T_2. This can be easily seen from Figs. 12 and 13, which display results for the two cases $T_1 = T_2 = 10$ and $T_1 = 100$, $T_2 = 10$, respectively. In fact, in the latter case where $T_2 < T_1$, D_1 remains fairly constant over the entire

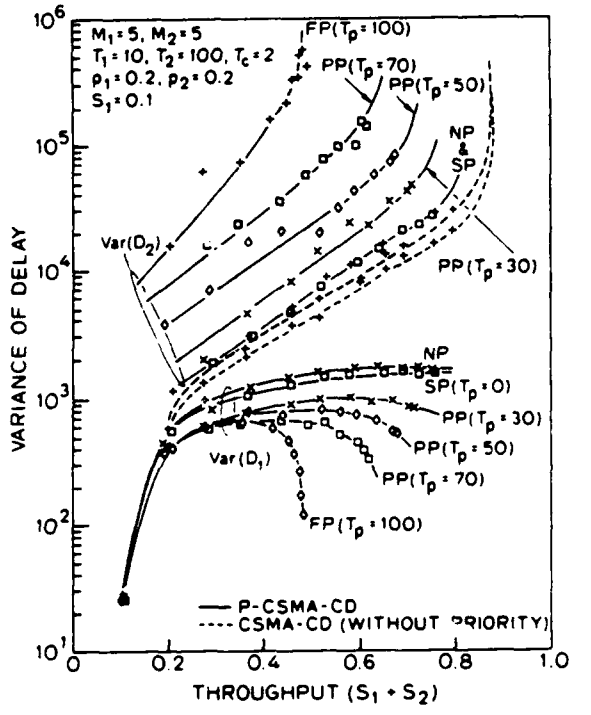


Fig. 9. Variance of packet delay for nonpreemptive, semipreemptive, partially preemptive, and fully preemptive P-CSMA-CD with $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

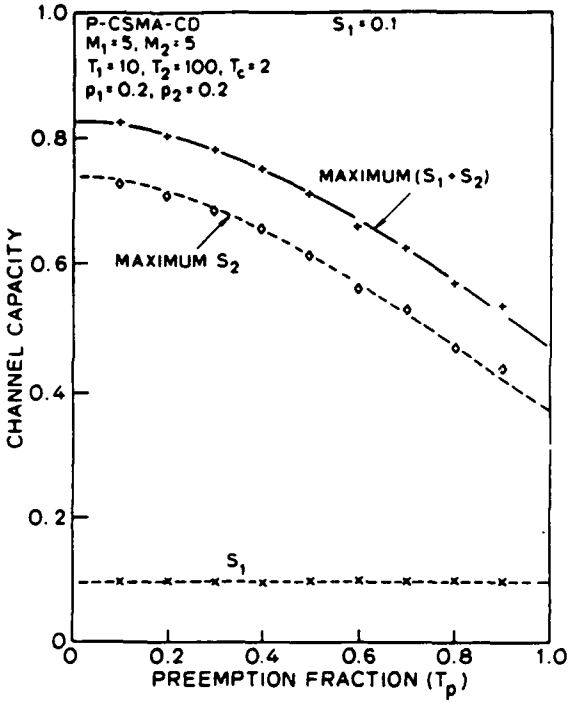


Fig. 10. Channel capacity as a function of the preemption fraction T_p for the partially preemptive P-CSMA-CD, with $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

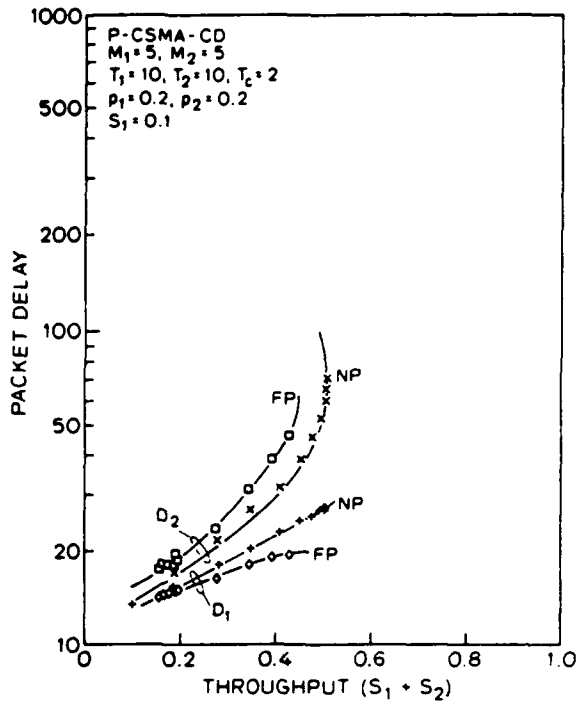


Fig. 12. Throughput-delay tradeoffs for P-CSMA-CD with $M_1 = M_2 = 5$, and $T_1 = T_2 = 10$ (obtained from analysis and simulation).

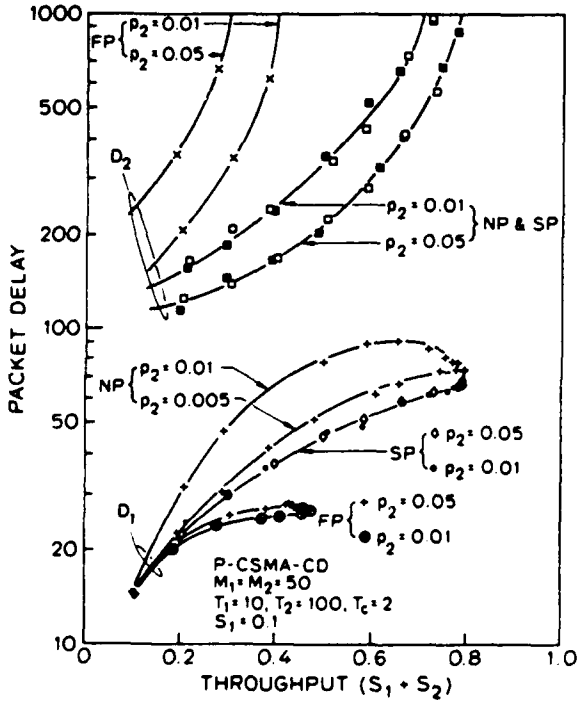


Fig. 11. Comparison of nonpreemptive, semipreemptive, and fully preemptive P-CSMA-CD in large systems; $M_1 = M_2 = 50$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

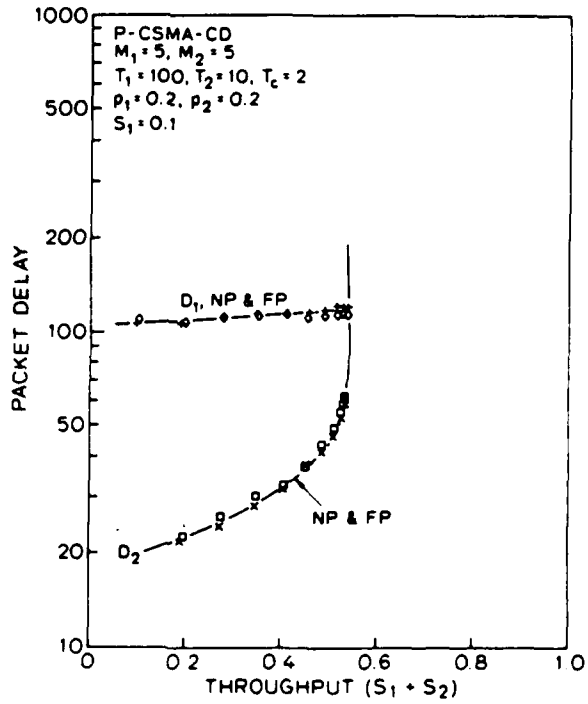


Fig. 13. Throughput-delay tradeoffs for P-CSMA-CD, with $M_1 = M_2 = 5$, $T_1 = 100$, and $T_2 = 10$ (obtained by simulation).

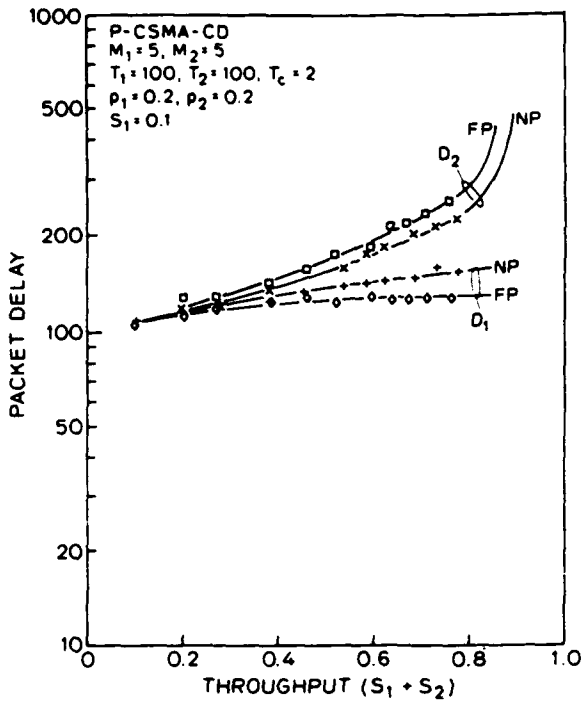


Fig. 14. Throughput-delay tradeoffs for P-CSMA-CD with $M_1 = M_2 = 5$ and $T_1 = T_2 = 100$ (obtained from simulation).

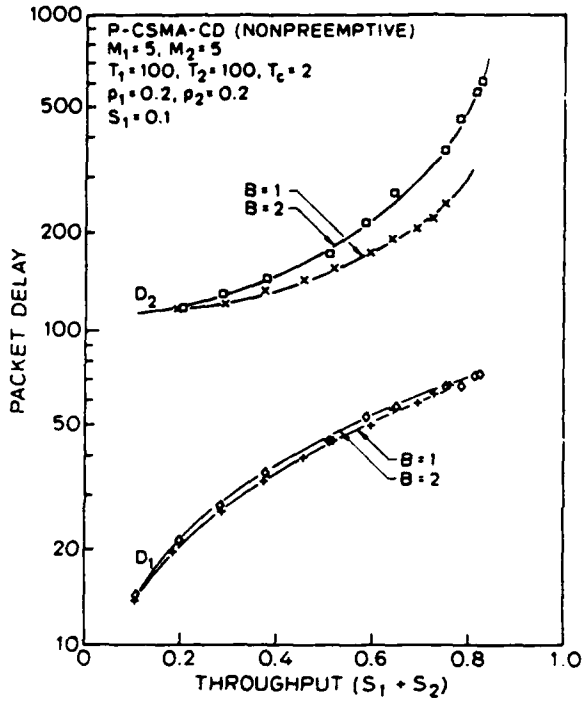


Fig. 15. Effect of buffer size on average packet delay in nonpreemptive P-CSMA-CD; $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

range of $S_1 + S_2$, and both the preemptive and nonpreemptive disciplines give almost identical results. Fig. 14 displays numerical results for the case $T_1 = T_2 = 100$. Figs. 12 and 14 show that when $T_1 = T_2$ the rate of increase in D_1 as S_2 increases is less significant than in the case $T_1 < T_2$, and that the effect of preemption is relatively moderate.

D. Effect of Buffer Size

According to the model description in Section III, each user possesses one packet buffer per priority class, and does not generate a new packet until the previous one has been successfully transmitted. Having assumed all users to be identical, this has allowed us to use a simple system state description, namely $n_j(t)$, the number of busy users at time t. Since the generation process of a thinking user is a Bernouilli one, and thus is memoryless, the model (and analysis) corresponds also to the situation where the user generates new packets according to the Bernouilli process at all times, but where new packets which find the buffer occupied are lost. In this case, the delay measure represents the delay of packets which are not rejected. The probability of a packet getting lost is $(\sigma_j - S_j)/\sigma_j$. Letting B denote the number of packet buffers per user and per priority class, we investigate, via simulation, the effect of values of B larger than one. We plot in Fig. 15 the average packet delays for nonpreemptive P-CSMA-CD when $M_1 = M_2 = 5$, $T_1 = 10$, $T_2 = 100$, $S_1 = 0.1$ and two values of B, $B = 1$ and $B = 2$. In Fig. 16 we plot the variance of packet delay, and in Fig. 17 we plot the probability of packet loss. We note that for this small system, $M = 5$, and small load, $S_1 = 0.1$, the average packet delay and its variance remain the same for class C_1, but increase for class C_2 as B is increased to two (due to queueing). Clearly, packet loss decreases for both classes; for C_1 with $B = 2$ and $S_1 =$

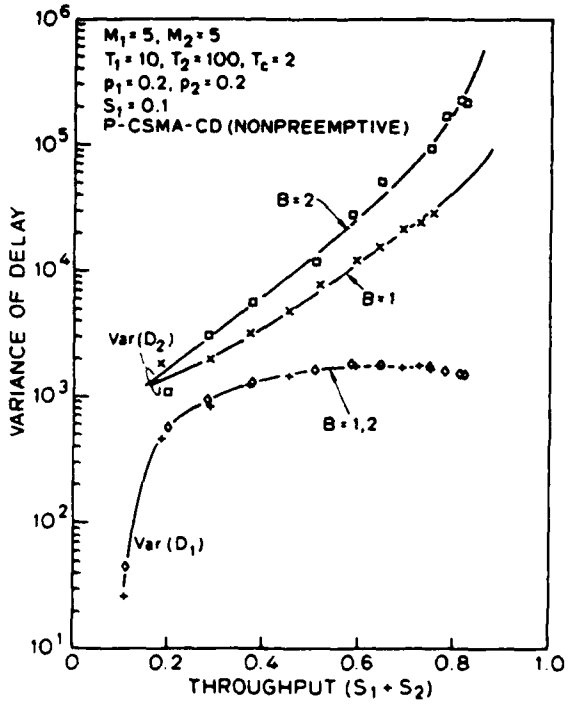


Fig. 16. Effect of buffer size on variance of packet delay in nonpreemptive P-CSMA-CD; $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

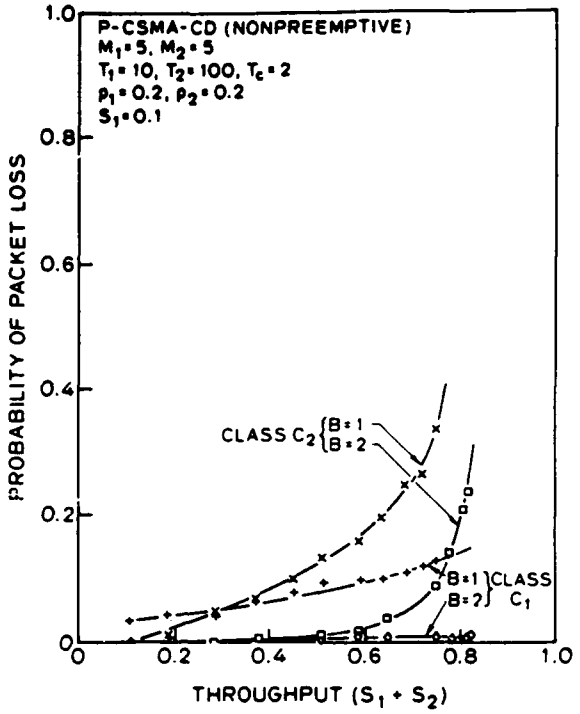


Fig. 17. Probability of packet loss for various buffer sizes in nonpreemptive P-CSMA-CD with $M_1 = M_2 = 5$, $T_1 = 10$, and $T_2 = 100$ (obtained by simulation).

0.1, packet loss decreases to almost zero. This indicates that, as longs as C_1-throughput is not too high, combining priority functions with two packet buffers guarantees excellent delay performance and negligible packet loss. Needless to say that a preemptive scheme would achieve even smaller packet loss, and that will larger M, packet loss is naturally lower for the same throughput S_1 as the throughput per user is then smaller, and therefore the storage capacity is relatively larger.

V. VARIATIONS OF P-CSMA

A. 1-Persistent Versus p-Persistent P-CSMA

Immediately following a reservation burst for class i, the p-persistent CSMA scheme consists of having each user with priority i do the following: 1) with probability p it transmits the message, 2) with probability $1 - p$ it delays the transmission by one slot and repeats the procedure if the channel is still sensed idle. This is equivalent to having each user with priority i transmit its message following a geometrically distributed delay with mean $1/p$ slots, provided that no carrier is detected prior to that time. When EOC is detected, a new time reference is established and a new reservation period is undertaken.

In a 1-persistent CSMA mode ($p = 1$), instead of sending a short burst to indicate a reservation, users with ready messages simply start transmission of their highest priority messages in the corresponding reservation-slot following EOC, of course, provided that no carrier is detected in previous reservation-slots. If a single user is transmitting, then its transmission is successful and its termination establishes a new EOC time reference. On the other hand, if two or more users overlap in transmission, a collision results; all users become aware of the collision and will consider it *in lieu* of a reservation. (That is, the end of this first transmission does not constitute a new time reference and no new reservation period is started.) All users involved in the collision reschedule the transmission of their respective messages incurring a random delay, say geometrically distributed with mean $1/p$ slots, and transmit their messages at the scheduled time provided that no carrier is detected prior to that time. The end of this new transmission period constitutes a new time reference and the procedure is repeated. (See Fig. 18.)

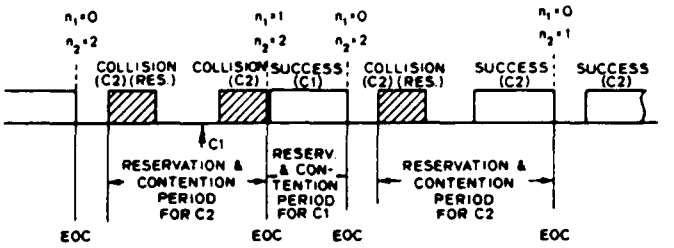


Fig. 18. Nonpreemptive 1-persistent P-CSMA-CD.

In general, 1-persistent CSMA is known to be inferior to p-persistent, since $p = 1$ is certainly not optimum, especially if the likelihood of having several users with ready messages of the same priority level is high. However, if the load placed on the channel by some priority class is known to be low (as it would most probably be the case for high priority levels in order to guarantee their performance) then 1-persistent CSMA used within that class may present some benefit.

In environments where a collision detection feature is available and the collision detection and recovery period T_c is small (on the order of 2τ, as is the case with ETHERNET), 1-persistent P-CSMA-CD is clearly superior to p-persistent P-CSMA-CD, since T_c is equivalent to a reservation-slot.

B. Notes on Reservation Overhead

1) Hierarchical Reservations: If the number of priority levels is large, then the overhead incurred in the reservation process may be high, especially if it is expected that the bulk of traffic will be in the lower levels of priority. This overhead can be decreased if a hierarchical reservation scheme (i.e., a tree priority resolution algorithm) is used. A burst in the first reservation-slot designates that messages in the highest group of priority levels are present. Following that, each level in the group is assigned its own reservation-slot for reservations, etc.

2) Message Delay Performance Versus Protocol Overhead: In the above described schemes, the higher the priority is, the smaller is the delay in gaining access right to the channel, and thus the better is the delay performance. Such a property is important if message delay for high priority classes is a critical performance measure. On the other hand, to guarantee a low delay performance for high priority classes, it is important to limit their load on the channel; as a result, it is expected that the bulk of traffic falls into the lower classes

incurring high overhead at each priority resolution period. This in turn limits the overall achievable channel capacity. An alternative to the above scheme consists of having all ready users start transmitting the unmodulated carrier in view of a reservation immediately following EOC, but such that the higher the priority is, the longer is the number of reservation-slots in which carrier is transmitted. As a result, the highest priority class present gains access by persisting the longest.

VI. CONCLUSION

We described in this paper a new version of CSMA which provides message-based priority functions. We also presented an analysis of the nonpreemptive P-CSMA which allowed us to derive analytically the throughput-delay characteristics for each priority class. Limitations in the analysis were overcome by also simulating the scheme. We evaluated the performance of prioritized CSMA, and discussed the effect of various system parameters and of preemption on the performance. Finally, we made a few comments on the overhead and on the use of the 1-persistent protocol.

P-CSMA satisfies the requirements set forth for prioritized access schemes, in that it is robust, efficient, fair to messages present within each priority class, and requires low overhead to implement.

APPENDIX

We give here the expressions for the expected duration of a cycle and the expected sum of backlogs over a cycle. For simplicity of expression, we let, by abuse of notation, the sum $I + Q + \cdots + Q^l$ be represented by the expression $(I - Q)^{-1}(I - Q^{l+1})$.

A. Expected Duration of a C_2-Cycle

$$
\begin{aligned}
&E[t_{e2}^{(r+1)} - t_{e2}^{(r)} \mid n_2(t_{e2}^{(r)}) = i \neq 0] \\
&\quad = \sum_{k=i}^{M_2} [Q_2{}^4]_{i,k} \left\{ 4 + \bar{I}_k^{(2)} + p_s^{(2)}(k) \left[T_2 + 1 \right.\right. \\
&\qquad \left. + \sum_{m=0}^{M_1} [Y_k{}^{(s)*}(Q_1)]_{0,m} \bar{L}_m \right] \\
&\qquad + [1 - p_s^{(2)}(k)] \left[T_c^{(2)} + 1 \right. \\
&\qquad \left.\left. + \sum_{m=0}^{M_1} [Y_k{}^{(f)*}(Q_1)]_{0,m} \bar{L}_m \right] \right\}
\end{aligned} \tag{A1}
$$

where $\bar{I}_k^{(2)}$ is the expectation of $I_k^{(2)}$ and is given by

$$
\bar{I}_k^{(2)} = \frac{1}{1 - \delta_k^{(2)}} \tag{A2}
$$

and where $\bar{L}_m$ denotes the expectation of L_m; and

$$
\begin{aligned}
&E[t_{e2}^{(r+1)} - t_{e2}^{(r)} \mid n_2(t_{e2}^{(r)}) = 0] \\
&\quad = 5 + \alpha(0,0)\bar{I}_0 \\
&\qquad + \sum_{k_1=0}^{M_1} \left\{ [a(1,0) + b(1,0)][Q_1{}^{T_1+1} J_1]_{1,k_1} (T_1 + \bar{L}_{k_1}) \right. \\
&\qquad + [a(0,1) + b(0,1)] [Q_1{}^{T_2+1}]_{0,k_1} (T_2 + \bar{L}_{k_1}) \\
&\qquad + \sum_{j=2}^{M_1} [a(j,0) + b(j,0)] [Q_1{}^{T_c^{(1)}+1}]_{j,k_1} (T_c^{(1)} + \bar{L}_{k_1}) \\
&\qquad + \sum_{j=2}^{M_2} [a(0,j) + b(0,j)] [Q_1{}^{T_c^{(2)}+1}]_{0,k_1} (T_c^{(2)} + \bar{L}_{k_1}) \\
&\qquad + \sum_{j_1=1}^{M_1} \sum_{j_2=1}^{M_2} [a(j_1,j_2) + b(j_1,j_2)] \\
&\qquad \cdot [Q_1{}^{\max(T_c^{(1)}, T_c^{(2)})+1}]_{j_1,k_1} \\
&\qquad \left. \cdot (\max(T_c^{(1)}, T_c^{(2)}) + \bar{L}_{k_1}) \right\}
\end{aligned} \tag{A3}
$$

where $\bar{I}_0$ is given by

$$
\bar{I}_0 = \frac{1}{1 - (1 - \sigma_1)^{M_1} (1 - \sigma_2)^{M_2}}. \tag{A4}
$$

B. Expected Sum of C_2-Backlogs Over a C_2-Cycle

$$
\begin{aligned}
&E\left[\sum_{t=t_{e2}^{(r)}}^{t_{e2}^{(r+1)}-1} n_2(t) \mid n_2(t_{e2}^{(r)}) = i \neq 0 \right] \\
&\quad = [(I_2 + Q_2 + Q_2{}^2 + Q_2{}^3) H_2]_i + \sum_{k=i}^{M_2} [Q_2{}^4]_{ik} k \bar{I}_k^{(2)} \\
&\qquad + \sum_{k=i}^{M_2} [Q_2{}^4]_{i,k} \left[S_2 (I_2 - Q_2)^{-1} (I_2 - Q_2{}^{T_2+1}) H_2 \right. \\
&\qquad + F_2 (I_2 - Q_2)^{-1} (I_2 - Q_2{}^{T_c^{(2)}+1}) H_2 \\
&\qquad + S_2 Q_2{}^{T_2+1} (I_2 - Q_2)^{-1} \\
&\qquad \cdot \left(I_2 - \sum_{m=1}^{M_1} [Y_k{}^{(s)*}(Q_1)]_{0,m} L_m{}^*(Q_2) \right) H_2 \\
&\qquad + F_2 Q_2{}^{T_c^{(2)}+1} (I_2 - Q_2)^{-1} \\
&\qquad \left. \cdot \left(I_2 - \sum_{m=1}^{M_1} [Y_k{}^{(f)*}(Q_1)]_{0,m} L_m{}^*(Q_2) \right) H_2 \right]_k
\end{aligned} \tag{A5}
$$

$$E\left[\sum_{t=t_{e2}^{(r)}}^{t_{e2}^{(r+1)}-1} n_2(t) \mid n_2(t_e^{(r)}) = 0\right]$$

$$= [(I_2 + Q_2 + {Q_2}^2 + {Q_2}^3)H_2]_0 + [a(1,0) + b(1,0)]$$

$$\cdot \left[(I_2 - Q_2)^{-1}\left(I_2 - \sum_{m=0}^{M_1} [{Q_1}^{T_1+1} J_1]_{1,m} {Q_2}^{T_1+1} \cdot {L_m}^*(Q_2)\right)H_2\right]_0 + [a(0,1) + b(0,1)]\left[(I_2 - Q_2)^{-1}\right.$$

$$\left.\cdot \left(I_2 - \sum_{m=0}^{M_1} [{Q_1}^{T_2+1}]_{0,m} {Q_2}^{T_2+1} {L_m}^*(Q_2)\right)H_2\right]_1$$

$$+ \sum_{j=2}^{M_1} [a(j,0) + b(j,0)]\left[(I_2 - Q_2)^{-1}\right.$$

$$\left.\cdot \left(I_2 - \sum_{m=j}^{M_1} [{Q_1}^{T_c^{(1)}+1}]_{j,m} {Q_2}^{T_c^{(1)}+1} \cdot {L_m}^*(Q_2)\right)H_2\right]_0 + \sum_{j=2}^{M_2} [a(0,j) + b(0,j)]$$

$$\cdot \left[(I_2 - Q_2)^{-1}\left(I_2 - \sum_{m=0}^{M_1} [{Q_1}^{T_c^{(2)}+1}]_{0,m} \cdot {Q_2}^{T_c^{(2)}+1} {L_m}^*(Q_2)\right)H_2\right]_j$$

$$+ \sum_{j_1=1}^{M_1} \sum_{j_2=1}^{M_2} [a(j_1,j_2) + b(j_1,j_2)] \cdot \left[(I_2 - Q_2)^{-1}\right.$$

$$\left.\cdot \left(I_2 - \sum_{m=j_1}^{M_1} [{Q_1}^{\max(T_c^{(1)}, T_c^{(2)})+1}]_{j_1,m} \cdot {Q_2}^{\max(T_c^{(1)}, T_c^{(2)})+1} {L_m}^*(Q_2)\right)H_2\right]_{j2}. \quad \text{(A6)}$$

C. Expected Duration of a C_1-Cycle

$$E[{t_{e1}}^{(r+1)} - {t_{e1}}^{(r)} \mid n_1(t_{e1}) = i \neq 0]$$

$$= \sum_{k=i}^{M_1} [{Q_1}^2]_{i,k}[1 + \bar{I}_k^{(1)} + {p_s}^{(1)}(k)(T_1 + 1) + (1 - {p_s}^{(1)}(k))(T_c^{(1)} + 1)] \quad \text{(A7)}$$

where

$$\bar{I}_k^{(1)} = \frac{1}{1 - \delta_k^{(1)}} \quad \text{(A8)}$$

$$E[{t_{e1}}^{(r+1)} - {t_{e1}}^{(r)} \mid n_1({t_{e1}}^{(r)}) = 0, n_2({t_{e1}}^{(r)}) = j \neq 0]$$

$$= \sum_{k=j}^{M_2} [{Q_2}^4]_{j,k}[4 + \bar{I}_k^{(2)} + {p_s}^{(2)}(k)(T_2 + 1) + [1 - {p_s}^{(2)}(k)](T_e^{(2)} + 1)] \quad \text{(A9)}$$

$$E[{t_{e1}}^{(r+1)} - {t_{e1}}^{(r)} \mid n_1({t_{e1}}^{(r)}) = n_2({t_{e1}}^{(r)}) = 0]$$

$$= 5 + a(0,0)\bar{I}_0 + [a(1,0) + b(1,0)]T_1 + [a(0,1) + b(0,1)]T_2 + \sum_{j=2}^{M_1} [a(j,0) + b(j,0)]T_c^{(1)} + \sum_{j=2}^{M_2} [a(0,j) + b(0,j)]T_c^{(2)} + \sum_{j_1=1}^{M_1} \sum_{j_2=1}^{M_2} [a(j_1,j_2) + b(j_1,j_2)] \max(T_c^{(1)}, T_c^{(2)}) \quad \text{(A10)}$$

$$E[{t_{e1}}^{(r+1)} - {t_{e1}}^{(r)} \mid n_1({t_{e1}}^{(r)}) = 0] = \sum_{j=0}^{M_2} {\pi_j}^{(2)} E[{t_{e1}}^{(r+1)} - {t_{e1}}^{(r)} \mid n_1({t_{e1}}^{(r)}) = 0, n_2({t_{e1}}^{(r)}) = j]. \quad \text{(A11)}$$

D. Expected Sum of C_1-Backlogs Over a C_1-Cycle

$$E\left[\sum_{t=t_{e1}^{(r)}}^{t_{e1}^{(r+1)}-1} n_1(t) \mid n_1(t_{e1}^{(r)}) = i \neq 0\right]$$

$$= [I_1 + Q_1]H_1 + \sum_{k=i}^{M_1} [{Q_1}^2]_{i,k} k \bar{I}_k^{(1)} + \sum_{k=i}^{M_1} [{Q_1}^2]_{i,k}[S_1(I_1 - Q_1)^{-1}(I_1 - {Q_1}^{T_1+1})H_1 + F_1(I_1 - Q_1)^{-1}(I_1 - {Q_1}^{T_c^{(1)}+1})H_1]_k \quad \text{(A12)}$$

$$E\left[\sum_{t=t_{e1}^{(r)}}^{t_{e1}^{(r+1)}-1} n_1(t) \mid n_1(t_{e1}^{(r)}) = 0, n_2(t_{e1}^{(r)}) = j \neq 0\right]$$

$$= \sum_{k=j}^{M_2} [{Q_2}^4]_{j,k}[{p_s}^{(2)}(k)(I_1 - Q_1)^{-1} \cdot (I_1 - {Y_k}^{(s)*}(Q_1))H_1 + (1 - {p_s}^{(2)}(k))(I_1 - Q_1)^{-1} \cdot (I_1 - {Y_k}^{(f)*}(Q_1))H_1]_0 \quad \text{(A13)}$$

$$E\left[\sum_{\tau=t_{e1}^{(r)}}^{t_{e1}^{(r+1)-1}} n_1(t) \mid n_1(t_{e1}^{(r)}) = n_2(t_{e1}^{(r)}) = 0\right]$$

$$= [(I_1 + Q_1 + Q_1^2 + Q_1^3)H_1]_0 + [a(1,0) + b(1,0)]$$
$$\cdot [(I_1 - Q_1)^{-1}(I_1 - Q_1^{T_1+1})H_1]_1$$
$$+ [a(0,1) + b(0,1)]$$
$$\cdot [(I_1 - Q_1)^{-1}(I_1 - Q_1^{T_2+1})H_1]_0$$
$$+ \sum_{j=2}^{M_1} [a(j,0) + b(j,0)]$$
$$\cdot [(I_1 - Q_1)^{-1}(I_1 - Q_1^{T_c^{(1)}+1})H_1]_j$$
$$+ \sum_{j=2}^{M_2} [a(0,j) + b(0,j)]$$
$$\cdot [(I_1 - Q_1)^{-1}(I_1 - Q_1^{T_c^{(2)}+1})H_1]_0$$
$$+ \sum_{j_1=1}^{M_1} \sum_{j_2=1}^{M_2} [a(j_1,j_2) + b(j_1,j_2)]$$
$$\cdot [(I_1 - Q_1)^{-1}(I_1 - Q_1^{\max(T_c^{(1)}, T_c^{(2)})+1})H_1]_{j1}. \quad \text{(A14)}$$

In this last case where $n_1(t_{e1}^{(r)}) = 0$, we again remove the condition $n_2(t_{e1}^{(r)}) = j$ by noting that the probability of this event in steady state is $\pi_j^{(2)}$.

ACKNOWLEDGMENT

The author would like to acknowledge the Telecommunications Sciences Center, SRI-International, for having provided the initial support for this research, and R. Rom of SRI-International for the fruitful discussions during the initial phase of this work. The author would also like to acknowledge M. Fine and G. Hahn for their assistance in programming, the Stanford Linear Accelerator Center, Stanford University, for providing the computer time used in the analysis, and N. Gonzalez-Cawley for performing the simulation work.

REFERENCES

[1] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Commun. Ass. Comput. Mach.*, vol. 19, pp. 395–403, July 1976.

[2] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, pp. 1468–1496, Nov. 1978.

[3] 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.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[4] 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.*, vol. COM-25, pp. 1103–1120, Sept. 1977.

[5] F. A. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," in *Proc. Local Area Commun. Network Symp.*, Boston, MA, May 1979; also in *Comput. Networks*, vol. 4, pp. 245–259, Oct.–Nov. 1980.

[6] F. A. Tobagi, "Multiaccess protocols in packet communication systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 468–488, Apr. 1980.

[7] N. Gonzalez-Cawley and F. A. Tobagi, "Simulation of carrier sense multiple access with message-based priority functions," Stanford Electron. Labs., Stanford Univ., Stanford, CA, Tech. Rep. 213, June 1981.

[8] W. R. Franta and M. B. Bilodeau, "Analysis of prioritized CSMA protocol based on staggered delays," Dep. Comput. Sci., Univ. of Minnesota, Minneapolis, Tech. Rep. TR-77-18, Sept. 1977.

[9] M. Onoe, Y. Yasuda, and M. Ishizuka, "A random access packet communication system with priority function—Priority Ethernet," in *Proc. Nat. Conv. Inform. Processing Soc.*, Japan, Aug. 1978, paper 3A-1.

[10] I. Iida, M. Ishizuka, Y. Yasuda, and M. Onoe, "Random access packet switched local computer network with priority function," in *Proc. Nat. Telecommun. Conf.*, Houston, TX, Dec. 1980, pp. 37.4.1–37.4.6.

★

Fouad A. Tobagi (M'77) was born in Beirut, Lebanon, on July 18, 1947. He received the Engineering degree from the Ecole Central des Arts et Manufactures, Paris, France, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1971 and 1974, respectively.

From 1971 to 1974, he was with the University of California, Los Angeles, where he participated in the ARPA Network Project as a Postgraduate Research Engineer and did research in packet radio communication. During the summer of 1972, he was with the Communications Systems Evaluation and Synthesis Group, IBM T. J. Watson Research Center, Yorktown Heights, NY. From December 1974 to June 1978, he was a Research Staff Project Manager with the ARPA project at the Computer Science Department, UCLA, and engaged in the modeling, analysis, and measurements of packet radio systems. In June 1978, he joined the faculty of the School of Engineering at Stanford University, Stanford, CA, where he is now Associate Professor of Electrical Engineering. His current research interests include computer communication networks, packet switching in ground radio and satellite networks, modeling and performance evaluation of computer communication systems, and VLSI implementation of network components.

Dr. Tobagi was the winner of the IEEE 1981 Leonard G. Abraham Prize Paper Award in the field of Communications Systems.

Distributions of Packet Delay and Interdeparture Time in Slotted ALOHA and Carrier Sense Multiple Access

FOUAD A. TOBAGI

Stanford University, Stanford, California

Abstract. Packet communication systems of the multiaccess/broadcast type, in which all communicating devices share a common channel that is multiaccessed in some random fashion, are considered. Among the various multiaccess schemes known, two prominent ones are considered: slotted ALOHA and Carrier Sense Multiple Access (CSMA). Existing analysis of these schemes has led to the determination of the average channel performance in terms of average throughput and average packet delay. This was achieved by formulating Markovian models for these channels with finite populations of users, each with a single packet buffer. Unfortunately, average performance is not adequate when designing communication systems intended for real-time applications, such as digitized speech, or when analyzing multihop packet radio networks, and the analysis has to be extended so as to provide delay distributions. Using the same Markovian models, the distributions of packet delay and interdeparture time for slotted ALOHA and CSMA channels are derived, and expressions for their moments are given.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*distributed networks*; C.2.5 [**Computer-Communication Networks**]: Local Networks—*access schemes*; C.4 [**Computer Systems Organization**]: Performance of Systems—*modeling techniques*

General Terms: Performance, Theory

Additional Key Words and Phrases: Random access schemes

1. *Introduction*

Slotted ALOHA and Carrier Sense Multiple Access (CSMA) are random access methods for multiplexing a population of users communicating over a shared packet-switched channel [9]. In slotted ALOHA the time axis is divided into slots of duration equal to the transmission time of a single packet (assuming constant-length packets). Users transmit any time they desire, as long as they start transmission of their packet at the beginning of a slot. If a conflict occurs (owing to time-overlapping transmissions), conflicting users reschedule transmission of their packets to some random time in the future [1, 2, 7, 9]. CSMA is a highly efficient random access scheme for environments where the propagation delay is short compared to the transmission time of a packet on the channel. Briefly, CSMA reduces the level of interference (caused by overlapping packets) in the random multiaccess environment by allowing terminals to sense the carrier due to other users' transmissions; on the basis of this channel state information (busy or idle) the terminal takes an action prescribed by

This work was supported by the Defense Advanced Research Projects Agency under Contract No. MDA 903-79-C-0201, Order No. A03717, monitored by the Office of Naval Research; and by the U.S. Army, CECOM, Fort Monmouth, N.J., under Army Research Office Contract No. DAAG 29-79-C-0138.

Author's address: Computer Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

the particular CSMA protocol in use. In particular, terminals never transmit when they sense that the channel is busy [4, 9].

The difficulty in analyzing multiaccess schemes such as slotted ALOHA and CSMA arises from the fact that the system's outcome is at all times dependent on the system's state and its evolution in time; for example, the time required to successfully transmit a packet is a function of the evolution of the number of contending users during the lifetime of the packet. To analyze the performance of slotted ALOHA and CSMA, Markov and semi-Markov models have been formulated for channels with *finite* populations of users, each user possessing a *single* packet buffer [3, 10, 11]. Average stationary performance has been derived in terms of average throughput and average packet delay. As the average performance may not be adequate when designing systems intended for real-time applications such as digitized speech, the analysis has to be extended so as to include delay distributions. Also, when analyzing multihop systems, it is important to be able to characterize the departure process from a collection of nodes, as this corresponds to the arrival process to other nodes. In this paper we show that using the same Markovian models, one can derive the actual distribution of packet delay, as well as the distribution of time separating consecutive successful transmissions (referred to as the interdeparture time). Moreover, it is shown that the analysis provides simple expressions for all moments of these distributions.

The body of the paper is divided into two sections, one devoted to slotted ALOHA and the other to CSMA. Although the basic technique of analysis is the same for both schemes, it is believed that readability is improved by treating the two schemes separately for two reasons: (i) Readers may be interested in only one scheme; (ii) treatment of the simpler case first, namely, slotted ALOHA, sets the stage for the more complex case of CSMA. For each scheme we begin by describing the model and the transmission protocol considered for analysis. We then review the derivation of the average performance as presented in references [3, 10, 11]. Following that, we address the issue of the interdeparture time distribution and its moments. Finally, we treat packet delay and derive its distribution and its moments.

2. *Slotted ALOHA*

2.1. THE MODEL. We consider a slotted ALOHA channel with a user population consisting of M users. Each user possesses a single packet buffer and therefore can be in one of two states, thinking or backlogged, depending on whether its packet buffer is empty or full. Backlogged users transmit their packet independently according to a Bernouilli process with parameter p; that is, in any slot t a backlogged user transmits its packet with probability p and delays action until the next slot with probability $1 - p$. A thinking user generates a packet (and thus joins the set of backlogged users) in a slot with probability λ. The generation of new packets is assumed to be instantaneous and to occur at the *end* of the slot. A packet transmission in a slot is successful if it is the only one in that slot. A user is assumed to learn about its success or failure instantaneously. Immediately following the successful transmission of its packet, a backlogged user switches to the thinking state. With respect to a user who has completed a successful transmission in some slot t, we distinguish two cases: (i) The user may generate a new packet (and thus rejoin the set of backlogged users) at the end of slot t with probability λ; (ii) the user may generate a new packet starting only at the end of slot $t + 1$. Thus in case (ii) we force the user to remain in the thinking state for at least one slot. The treatment of both cases is very much the same. In this paper we opt for case (i). Note that in the transmission protocol we

have just described, the first transmission of a newly generated packet is delayed by a geometrically distributed time following its generation, with mean $1/p$ slots. We therefore refer to this protocol as the *delayed-first-transmission* (*DFT*) protocol [8].

A slight variation of the above slotted ALOHA transmission protocol consists of transmitting, with probability 1, a newly generated packet, at its generation time. In this case it is assumed that the generation of a new packet by a thinking user occurs with probability λ instantaneously at the *beginning* of a slot; its first transmission takes place in that same slot. If the first transmission is unsuccessful, then the user joins the set of backlogged users and operates as in the DFT protocol described above, namely, transmits the packet in a slot with probability p and delays action to the next slot with probability $1 - p$. This protocol is referred to as the *immediate-first-transmission* (*IFT*) protocol [8].

Given the memoryless nature of the packet generation and transmission processes, the model we have just formulated for a slotted ALOHA channel is Markovian. We show that we can exploit this Markovian property to derive the distributions of packet delay, defined as the time elapsed from when the packet is generated until it is successfully transmitted, and of the interdeparture time, defined as the time separating two successive successful transmissions.

2.2. Average Channel Performance. Let $n(t)$ denote the number of backlogged users at the end of slot t. This number includes all new arrivals to the set of backlogged users that have occurred in slot t and excludes the user who may have just completed a successful transmission in that slot (unless, of course, it has rejoined the backlog, as devised by case (i) of the DFT protocol described above). It is clear that the process $\{n(t), t = 0, 1, 2, \dots\}$ is a Markov chain. Let $p_{ij} \triangleq \Pr\{n(t+1) = j \mid n(t) = i\}$. These transition probabilities, for $i = 0, 1, 2, \dots, M$, are expressed as

DFT protocol:

$$p_{ij} = \begin{cases} 0, & j < i - 1, \\ P_s(i)\binom{M-i+1}{j-i+1}\lambda^{j-i+1}(1-\lambda)^{M-j} & \\ \quad + [1 - P_s(i)]\binom{M-i}{j-i}\lambda^{j-i}(1-\lambda)^{M-j}, & i - 1 \le j \le M; \end{cases} \tag{1}$$

IFT protocol:

$$p_{ij} = \begin{cases} 0, & j < i - 1, \\ P_s(i)(1-\lambda)^{M-i}, & j = i - 1, \\ [1 - P_s(i)](1-\lambda)^{M-i} + (M-i)\lambda(1-\lambda)^{M-i-1}(1-p)^i, & j = i, \\ (M-i)\lambda(1-\lambda)^{M-i-1}[1-(1-p)^i], & j = i+1, \\ \binom{M-i}{j-i}\lambda^{j-i}(1-\lambda)^{M-j}, & j \ge i+2; \end{cases} \tag{2}$$

where we have adopted the convention that $\binom{m}{k} = 0$ for $k < 0$, and where $P_s(i)$ is the probability of a successful transmission in the DFT proposal given i users in the backlog and is expressed as

$$P_s(i) = ip(1-p)^{i-1}. \tag{3}$$

Let $\mathbf{P}$ denote the transition probability matrix. The stationary distribution $\Pi = \{\pi_0, \pi_1, \dots, \pi_M\}$, where $\pi_i \triangleq \lim_{t\to\infty}\Pr\{n(t) = i\}$, is simply obtained by solving the

system $\Pi = \Pi\mathbf{P}$. Given the special form of matrix $\mathbf{P}$, namely, that $p_{ij} = 0$ for $j < i - 1$, the numerical solution of $\Pi = \Pi\mathbf{P}$ is obtained recursively. Let $\bar{n}$ denote the average backlog. This is computed for both DFT and IFT as

$$\bar{n} = \sum_{i=0}^{M} i\pi_i. \tag{4}$$

We now derive the channel throughput and average packet delay for each protocol.

(1) *DFT protocol.* The average rate at which users join the backlog is, in steady state, equal to the rate at which they leave it; the latter, denoted by S, is also the average channel throughput (i.e., the average number of successful transmissions per slot) and is given by

$$S = \sum_{i=0}^{M} \pi_i P_s(i). \tag{5}$$

The average packet delay $\bar{D}$ is equal to the average time that a packet spends in the backlog state. Applying Little's result [5], this is simply given by

$$\bar{D} = \frac{\bar{n}}{S}. \tag{6}$$

(2) *IFT protocol.* The average channel throughput is given by

$$S = \sum_{i=0}^{M} \pi_i [P_s(i)(1-\lambda)^{M-i} + (M-i)\lambda(1-\lambda)^{M-i-1}(1-p)^i]. \tag{7}$$

The average rate at which users join and leave the backlog is given by

$$\mu = \sum_{i=0}^{M} \pi_i P_s(i)(1-\lambda)^{M-i}. \tag{8}$$

The difference $S - \mu$ is the average rate of packets successful at first transmission. The average time a user spends in the backlogged state, $\bar{B}$, is by Little's result expressed as

$$\bar{B} = \frac{\bar{n}}{\mu}. \tag{9}$$

A new packet is either successful at first transmission, in which case its average delay is just one slot, or joins the backlog, in which case its average delay is $1 + \bar{B}$. The average packet delay $\bar{D}$ is then given by

$$\bar{D} = 1 + \frac{\mu}{S}\bar{B} = 1 + \frac{\bar{n}}{S}. \tag{10}$$

2.3. Distribution of Interdeparture Times

THEOREM 2.1. *The z-transform of the interdeparture time distribution in slotted ALOHA is given by*

$$ID^*(z) = \sum_{l=1}^{\infty} \Delta\mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d\mathbf{H}z^l = z\Delta(\mathbf{I} - z\mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d\mathbf{H}, \tag{11}$$

where $\mathbf{P}_d$ and $\mathbf{P}_{\bar{d}}$ are matrices defined in eqs. (12)–(15) in the proof, Δ is a row vector solution of the system $\Delta = \Delta(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d$, $\mathbf{H}$ is the column vector with $M + 1$ elements all equal to one, and $\mathbf{I}$ is the $(M + 1) \times (M + 1)$ identity matrix.

PROOF. We augment the system state description to include an indicator $\delta(t)$ such that

$$\delta(t) = \begin{cases} 0 & \text{if no departure occurred in slot } t, \\ 1 & \text{if a departure occurred in slot } t. \end{cases}$$

The state of the system in slot t is now described by the pair $\{n(t), \delta(t)\}$. It is essential to note that the state in slot $t+1$, $\{n(t+1), \delta(t+1)\}$, depends on $n(t)$ but does not depend on $\delta(t)$. For $i = 0, 1, 2, \ldots, M$, define $p_{ij}^{(d)}$ and $p_{ij}^{(\bar{d})}$ as

$$p_{ij}^{(d)} \triangleq \Pr\{n(t+1) = j, \delta(t+1) = 1 \mid n(t) = i\},$$

$$p_{ij}^{(\bar{d})} \triangleq \Pr\{n(t+1) = j, \delta(t+1) = 0 \mid n(t) = i\}.$$

These transition probabilities are expressed as

DFT protocol:

$$p_{ij}^{(d)} = P_a(i) \binom{M-i+1}{j-i+1} \lambda^{j-i+1}(1-\lambda)^{M-j}, \qquad i-1 \le j \le M, \tag{12}$$

$$p_{ij}^{(\bar{d})} = [1 - P_a(i)] \binom{M-i}{j-i} \lambda^{j-i}(1-\lambda)^{M-j}, \qquad i \le j \le M. \tag{13}$$

IFT protocol:

$$p_{ij}^{(d)} = \begin{cases} P_a(i)(1-\lambda)^{M-i}, & j = i-1, \\ (M-i)\lambda(1-\lambda)^{M-i-1}(1-p)^i, & j = i, \\ 0, & \text{otherwise;} \end{cases} \tag{14}$$

$$p_{ij}^{(\bar{d})} = \begin{cases} 0, & j = i-1, \\ [1 - P_a(i)](1-\lambda)^{M-i}, & j = i, \\ p_{ij}, & \text{otherwise.} \end{cases} \tag{15}$$

Let $\mathbf{P}_d$ and $\mathbf{P}_{\bar{d}}$ denote the matrices with elements $P_{ij}^{(d)}$ and $P_{ij}^{(\bar{d})}$, respectively. For any integer $l \ge 1$ and any vector $(\delta_1, \delta_2, \ldots, \delta_l)$, where $\delta_k \in \{0, 1\}$ for all $k \in \{1, 2, \ldots, l\}$, we have

$$\begin{aligned} &\Pr\{n(t+l) = j, \delta(t+l) = \delta_l, \delta(t+l-1) = \delta_{l-1}, \ldots, \delta(t+1) = \delta_1 \mid n(t) = i\} \\ &\quad = [\mathbf{P}_{\delta_1}\mathbf{P}_{\delta_2} \cdots \mathbf{P}_{\delta_l}]_{ij}, \end{aligned} \tag{16}$$

where $\mathbf{P}_{\delta_k}$ is $\mathbf{P}_d$ if $\delta_k = 1$ and $\mathbf{P}_{\bar{d}}$ if $\delta_k = 0$, and where we adopt the notation $[\mathbf{B}]_{ij}$ to denote the (i, j)th element of an arbitrary matrix $\mathbf{B}$. We also have

$$\Pr\{\delta(t+l) = \delta_l, \ldots, \delta(t+1) = \delta_1 \mid n(t) = i\} = \sum_{j=0}^{M} [\mathbf{P}_{\delta_1}\mathbf{P}_{\delta_2} \cdots \mathbf{P}_{\delta_l}]_{ij}. \tag{17}$$

Let $\ldots, t_d^{(r)}, t_d^{(r+1)}, \ldots$ denote the slots at which a departure took place. $n(t_d^{(r)})$ represents the number of backlogged users left behind the departure in slot $t_d^{(r)}$. Let ID denote the interdeparture time. It is clear from eq. (17) that

$$V_i^{(l)} \triangleq \Pr\{\mathrm{ID} = l \mid n(t_d^{(r)}) = i\} = \sum_{j=i-1}^{M} [\mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d]_{ij}. \tag{18}$$

Let $\mathbf{V}^{(l)}$ be the column vector whose ith element, for $i = 0, 1, \ldots, M$, is precisely $V_i^{(l)}$. Let $\mathbf{H}$ be the column vector with all elements equal to 1. Equation (18) can be written in matrix form as

$$\mathbf{V}^{(l)} = \mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d\mathbf{H}. \tag{19}$$

To remove the condition on $n(t_d^{(r)})$, we now seek its stationary distribution. The process $\{n(t), t \in (\ldots, t_d^{(r)}, t_d^{(r+1)}, \ldots)\}$ is a Markov chain with transition probabilities given by

$$\Pr\{n(t_d^{(r+1)}) = j \mid n(t_d^{(r)}) = i\} = \sum_{l=1}^{\infty} [\mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d]_{ij} = [(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d]_{ij}. \tag{20}$$

Let $\Delta = (d_0, d_1, \ldots, d_M)$ denote the stationary distribution[1] of process $n(t_d^{(r)})$. Δ is obtained as a solution of the system $\Delta = \Delta(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d$.

Removing the condition on $n(t_d^{(r)})$ in eq. (18), we finally obtain

$$\Pr\{\mathrm{ID} = l\} = \Delta\mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d\mathbf{H}, \tag{21}$$

and hence the result in eq. (11). Q.E.D.

A SECOND PROOF. The same result can also be easily proved as follows. Let $\mathrm{ID}_i^*(z)$ be the z-transform defined as

$$\mathrm{ID}_i^*(z) \triangleq \sum_{l=1}^{\infty} z^l \Pr\{\mathrm{ID} = l \mid n(t_d^{(r)}) = i\}, \tag{22}$$

and let $\mathbf{ID}^*(z)$ be the column vector $(\mathrm{ID}_0^*(z), \mathrm{ID}_1^*(z), \ldots, \mathrm{ID}_M^*(z))^T$, where the superscript T denotes the transpose operation. Given $n(t_d^{(r)}) = i$, the interdeparture time ID is one slot if a departure takes place in $t_d^{(r)} + 1$; if there is no departure in $t_d^{(r)} + 1$ and $n(t_d^{(r)} + 1) = j$, then the distribution of ID has a z-transform given by $z\mathrm{ID}_j^*(z)$. Hence we have the relationship

$$\mathrm{ID}_i^*(z) = \sum_{j=i-1}^{M} z[p_{ij}^{(d)} + p_{ij}^{(\bar{d})}\mathrm{ID}_j^*(z)], \tag{23}$$

which in matrix notation can be written as

$$\mathbf{ID}^*(z) = z\mathbf{P}_d\mathbf{H} + z\mathbf{P}_{\bar{d}}\mathbf{ID}^*(z) \tag{24}$$

or

$$\mathbf{ID}^*(z) = z(\mathbf{I} - z\mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d\mathbf{H} = \sum_{l=1}^{\infty} \mathbf{P}_{(\bar{d})}^{l-1}\mathbf{P}_d\mathbf{H}z^l. \tag{25}$$

We finally note that $\mathrm{ID}^*(z) = \Delta\mathbf{ID}^*(z)$, hence eq. (11). Q.E.D.

Simple closed-form expressions exist for all moments of ID. Let $\mathrm{ID}^{(m)}$ denote the mth derivative of $\mathrm{ID}^*(z)$ evaluated at $z = 1$; that is, $\mathrm{ID}^{(m)} = d^m\mathrm{ID}^*(z)/dz^m|_{z=1}$. Clearly, $\mathrm{ID}^{(1)} = E[\mathrm{ID}]$, $\mathrm{ID}^{(2)} = E[\mathrm{ID}(\mathrm{ID} - 1)]$, $\mathrm{ID}^{(3)} = E[\mathrm{ID}(\mathrm{ID} - 1)(\mathrm{ID} - 2)]$, etc., ...

COROLLARY 2.1. *$ID^{(m)}$ is given by*

$$ID^{(m)} = m!\Delta[(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_{\bar{d}}]^{m-1}(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{H}. \tag{26}$$

PROOF. Let $\mathbf{ID}^{*(m)}(z)$ denote the column vector whose ith element is $d^m\mathrm{ID}_i^*(z)/dz^m$. Differentiating eq. (23), we get

$$\mathrm{ID}_i^{*(1)}(z) = \frac{d\mathrm{ID}_i^*(z)}{dz} = \sum_{j=i-1}^{M} [p_{ij}^{(d)} + p_{ij}^{(\bar{d})}\mathrm{ID}_j^*(z) + p_{ij}^{(\bar{d})}z\mathrm{ID}_j^{*(1)}(z)]. \tag{27}$$

[1] Note that for the IFT protocol, $d_M = 0$.

Letting $z = 1$ and noting that $\sum_{j=i-1}^{M} [p_{ij}^{(d)} + p_{ij}^{(\bar{d})} \mathrm{ID}_j^*(1)] = 1$, we get

$$\mathbf{ID}^{*(1)}(1) = \mathbf{H} + \mathbf{P}_{\bar{d}} \mathbf{ID}^{*(1)}(1) \tag{28}$$

or

$$\mathbf{ID}^{*(1)}(1) = (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{H}. \tag{29}$$

By successive differentiation of eq. (23) one can easily establish for $m > 1$ that

$$\mathrm{ID}_i^{*(m)}(z) = \sum_{j=i-1}^{M} p_{ij}^{(\bar{d})} [m \mathrm{ID}_j^{*(m-1)}(z) + z \mathrm{ID}_j^{*(m)}(z)], \tag{30}$$

which leads to

$$\mathbf{ID}^{*(m)}(1) = m[(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{P}_{\bar{d}}] \mathbf{ID}^{*(m-1)}(1). \tag{31}$$

Noting that $\mathrm{ID}^{(m)} = \Delta \mathbf{ID}^{*(m)}(1)$, we get the result in eq. (26). Q.E.D.

2.4. Distribution of Packet Delay

Theorem 2.2. *The generating function for the distribution of packet delay in slotted ALOHA is given by*

DFT protocol:

$$\begin{aligned} D^*(z) &= z\boldsymbol{\Gamma}(\mathbf{I} - z\mathbf{P}_{\bar{s}})^{-1} \mathbf{P}_s \mathbf{H} \\ &= \sum_{l=1}^{\infty} \boldsymbol{\Gamma} \mathbf{P}_{\bar{s}}^{l-1} \mathbf{P}_s \mathbf{H} z^l; \end{aligned} \tag{32}$$

IFT protocol:

$$\begin{aligned} D^*(z) &= z[\gamma_0 + \boldsymbol{\Gamma}(\mathbf{I} - z\mathbf{P}_{\bar{s}})^{-1} \mathbf{P}_s \mathbf{H}] \\ &= \gamma_0 z + \sum_{l=1}^{\infty} \boldsymbol{\Gamma} \mathbf{P}_{\bar{s}}^{l-1} \mathbf{P}_s \mathbf{H} z^{l+1}; \end{aligned} \tag{33}$$

where $\mathbf{P}_s$ and $\mathbf{P}_{\bar{s}}$ are matrices defined in eqs. (34) *and* (35) *in the proof,* $\mathbf{H}$ *is the column vector with $M + 1$ elements all equal to one,* $\mathbf{I}$ *is the $M \times M$ identity matrix,* $\boldsymbol{\Gamma} = (\gamma_1, \ldots, \gamma_m)$ *is a row vector determined by eq.* (39) *in the proof for the DFT protocol and eq.* (44) *for the IFT protocol, and γ_0 is given by eq.* (43).

Proof. The general approach used to derive the distribution of packet delay is similar to that used for the distribution of interdeparture times. It consists of first deriving the delay distribution for a tagged user conditioned on the number of backlogged users among which it finds itself and then removing the condition.

Let $n(t) = i \neq 0$, and let $D_i^*(z)$ denote the z-transform of the distribution of delay (counted as of the end of slot t) of a tagged user in the backlog of size i. Let $\mathbf{D}^*(z)$ be the column vector of dimension M such that the ith element, $1 \leq i \leq M$, is precisely $D_i^*(z)$. We now derive $\mathbf{D}^*(z)$. Consider a tagged user in the backlog $n(t) = i$. For $i = 1, 2, \ldots, M$ and $j = 0, 1, \ldots, M$, define $p_{ij}^{(s)}$ and $p_{ij}^{(\bar{s})}$ as

$$p_{ij}^{(s)} \triangleq \Pr\{n(t+1) = j, \text{ tagged user successful in } t+1 \mid n(t) = i\},$$
$$p_{ij}^{(\bar{s})} \triangleq \Pr\{n(t+1) = j, \text{ tagged user unsuccessful in } t+1 \mid n(t) = i\}.$$

They are given by

DFT protocol:

$$p_{ij}^{(s)} = \frac{1}{i} p_{ij}^{(d)}, \tag{34a}$$

$$p_{ij}^{(\bar{s})} = p_{ij} - p_{ij}^{(s)}; \tag{35a}$$

IFT protocol:

$$p_{ij}^{(s)} = \begin{cases} \frac{1}{i} p_{i,i-1}^{(d)} & \text{if } j = i-1, \\ 0 & \text{otherwise,} \end{cases} \tag{34b}$$

$$p_{ij}^{(\bar{s})} = p_{ij} - p_{ij}^{(s)}. \tag{35b}$$

If the tagged user is successful in slot $t+1$, then its delay is exactly one slot. If the tagged user is unsuccessful in slot $t+1$ and finds itself in a backlog of size j, then its delay distribution has a z-transform given by $zD_j^*(z)$. Thus we have

$$D_i^*(z) = \sum_{j=i-1}^{M} z p_{ij}^{(s)} + \sum_{\substack{j=i-1 \\ j\neq 0}}^{M} p_{ij}^{(\bar{s})} z D_j^*(z). \tag{36}$$

Let $\mathbf{P}_s$ denote the $M \times (M+1)$ matrix with elements $p_{ij}^{(s)}$, $1 \le i \le M$, $0 \le j \le M$, and let $\mathbf{P}_{\bar{s}}$ denote the $M \times M$ matrix with elements $p_{ij}^{(\bar{s})}$, $1 \le i \le M$, $1 \le j \le M$. Writing eq. (36) in matrix form, one can easily deduce

$$\mathbf{D}^*(z) = z(\mathbf{I} - z\mathbf{P}_{\bar{s}})^{-1}\mathbf{P}_s\mathbf{H} = \sum_{l=1}^{\infty} \mathbf{P}_{\bar{s}}^{l-1}\mathbf{P}_s\mathbf{H}z^l. \tag{37}$$

To complete the proof, we now need to derive the distribution of the backlog as seen by an arbitrary newly generated packet. Let $\gamma_j, j = 1, 2, \ldots, M$, denote the probability that an arbitrary packet finds itself in the system upon arrival in a group of j backlogged users. For the IFT protocol we also let γ_0 denote the probability that an arriving packet is successful in its first transmission, in which case its delay is just one slot.

Consider first the DFT protocol. Given that $n(t) = i$, the number of arrivals which find themselves in a backlog of size j, $j \ge i$, is $j - i + 1$ with probability $p_{ij}^{(d)}$, and $j - i$ with probability $p_{ij}^{(\bar{d})}$. Since γ_j is the fraction of arrivals which find themselves in a backlog of size j, we can write

$$\gamma_j = K \sum_{i=0}^{j} \pi_i[(j-i+1)p_{ij}^{(d)} + (j-i)p_{ij}^{(\bar{d})}], \tag{38}$$

where K is a constant, such that $\sum_{j=1}^{M} \gamma_j = 1$. Therefore we have

$$\gamma_j = \frac{\sum_{i=0}^{j} \pi_i[(j-i+1)p_{ij}^{(d)} + (j-i)p_{ij}^{(\bar{d})}]}{\sum_{j=1}^{M}\sum_{i=0}^{j} \pi_i[(j-i+1)p_{ij}^{(d)} + (j-i)p_{ij}^{(\bar{d})}]}, \qquad 1 \le j \le M. \tag{39}$$

Let $\Gamma = (\gamma_1, \gamma_2, \ldots, \gamma_M)$. The generating function $D^*(z)$ is then simply expressed as

$$D^*(z) = \Gamma\mathbf{D}^*(z), \tag{40}$$

where $\mathbf{D}^*(z)$ is as given in eq. (37), hence eq. (32).

Consider now the IFT protocol. Given that $n(t) = i$, then with probability $p_{ii}^{(d)}$ there is one packet which is successful upon its arrival, and with probability

$p_{ij}^{(\bar{d})}$ there are $j - i$ arrivals which find themselves in a backlog of size j, $j \geq i + 1$. By the same argument as above, we can write

$$\gamma_0 = K \sum_{i=0}^{M-1} \pi_i p_{ii}^{(d)}, \tag{41}$$

$$\gamma_j = K \sum_{i=0}^{j-1} \pi_i (j - i) p_{ij}^{(\bar{d})}, \qquad j \geq 1. \tag{42}$$

One may easily verify that for the IFT protocol, $\gamma_1 = 0$, as one expects. Calculating K by setting $\sum_{i=0}^{M} \gamma_j = 1$, we get

$$\gamma_0 = \frac{\sum_{i=0}^{M-1} \pi_i p_{ii}^{(d)}}{\sum_{i=0}^{M-1} \pi_i p_{ii}^{(d)} + \sum_{j=2}^{M} \sum_{i=0}^{j-1} \pi_i (j - i) p_{ij}^{(\bar{d})}}, \tag{43}$$

$$\gamma_j = \frac{\sum_{i=0}^{j-1} \pi_i (j - i) p_{ij}^{(\bar{d})}}{\sum_{i=0}^{M-1} \pi_i p_{ii}^{(d)} + \sum_{j=2}^{M} \sum_{i=0}^{j-1} \pi_i (j - i) p_{ij}^{(\bar{d})}}, \qquad 1 \leq j \leq M. \tag{44}$$

The z-transform $D^*(z)$ for the IFT protocol is finally expressed as

$$D^*(z) = \gamma_0 z + \Gamma z D^*(z), \tag{45}$$

hence eq. (33). Q.E.D.

As with the interdeparture time analysis, simple closed-form expressions can be derived for all moments of D. Let $D^{(m)} = d^m D^*(z)/dz|_{z=1}$; we have

COROLLARY 2.2. *For the DFT protocols, $D^{(m)}$ is given by*

$$D^{(m)} = m! \Gamma [(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{P}_{\bar{d}}]^{m-1} (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{H}, \qquad m \geq 1, \tag{46}$$

and for the IFT protocol, $D^{(m)}$ is given by

$$D^{(m)} = \begin{cases} 1 + \Gamma (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{H}, & m = 1, \\ m! \Gamma [(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{P}_{\bar{d}}]^{m-2} [\mathbf{I} + (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{P}_{\bar{d}}] (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1} \mathbf{H}, & m \geq 1. \end{cases} \tag{47}$$

The proof is identical to that for the interdeparture time.

3. *Carrier Sense Multiple Access*

3.1. THE MODEL. Although the operation of CSMA does not require the devices to be time synchronized, it is assumed here, for simplicity in analysis, that the channel time axis is slotted with the slot size equal to τs, the maximum propagation delay between all pairs of users,[2] and that all users are synchronized and begin transmission only at slot boundaries. The CSMA scheme under consideration here consists of the following. A user with a packet ready for transmission (i.e., with a packet which has just been generated or has been rescheduled for transmission at that instant) senses the channel and (i) if the channel is idle, starts transmitting the packet at the beginning of the next slot, and (ii) if the channel is busy, reschedules the transmission of the packet to some random time in the future.

We consider a finite population of M users, all in line of sight and within range of each other, such that each user can be in one or two states: backlogged or thinking. In the thinking state, a user generates a new packet (and starts transmitting the

[2] Note the difference between the definition of a slot in slotted ALOHA (which corresponds to the transmission time of a packet) and the definition of a slot in CSMA (which corresponds to the maximum propagation delay between users).

packet, if the channel is sensed idle) in a slot with probability σ. A user is said to be backlogged if it has a packet in transmission or awaiting transmission. It remains in that state until it completes successful transmission of the packet, at which time it switches to the thinking state. Thus a user in the backlogged state cannot generate a new packet for transmission. The rescheduling delay of a backlogged packet is assumed to be geometrically distributed, that is, each backlogged user is scheduled to resense the channel in the current slot with probability ν, as specified by the protocol, a retransmission would result only if the channel is sensed idle. In this model it is assumed that all packets are of a fixed size equal to T slots, with $T \geq 1$. In some implementation of CSMA, such as in local networks of the ETHERNET type [6], it is possible for users to detect collisions when they occur and abort the colliding transmissions. In such a case it is assumed that it takes T_c slots to perform the detection and abortion procedure, where $T_c \leq T$. If the collision detection feature is not in effect, then it is assumed that a user learns about its success or failure instantaneously at the end of its transmission.

3.2. Average Stationary Performance [10, 11]. Let $n(t)$ denote the number of backlogged users at the beginning of slot t. We observe on the time axis an alternate sequence of idle and busy periods as shown in Figure 1. We follow the approach used in [10, 11] and consider the imbedded Markov chain identified by the first slot of each idle period. Using properties of regenerative processes, we derive the average channel performance.

Let $t_e^{(r)}$ and $t_e^{(r+1)}$ be two consecutive imbedded slots; the period of time between $t_e^{(r)}$ and $t_e^{(r+1)}$ is called a cycle. Let $\mathbf{P}$ denote the transition probability matrix between $t_e^{(r)}$ and $t_e^{(r+1)}$; that is, the (i, j)th element of $\mathbf{P}$ is defined as

$$p_{ij} \triangleq \Pr\{n(t_e^{(r+1)}) = j \mid n(t_e^{(r)}) = i\}, \qquad 0 \leq i, j \leq M. \tag{48}$$

We let TP denote the length of the transmission period. If the transmission of the message is successful, then $\mathrm{TP} = T + 1$, where the additional slot accounts for the propagation delay, since it is only one slot after the end of transmission that the channel will be sensed idle by all users. If the transmission of the message is unsuccessful, then $\mathrm{TP} = T_c + 1$, where $T_c \leq T$ is the time to detect the collision and abort all transmissions if the collision detection feature is in effect, and $T_c = T$ otherwise. $n(t_e^{(r)})$ remains invariant over the entire idle period I (since according to the CSMA procedure, a new arrival sensing the channel idle would transmit with probability one). See Figure 1. Thus for $t \in [t_e^{(r)}, t_e^{(r)} + I - 1]$, $n(t) = n(t_e^{(r)})$. Let $\mathbf{R}$ denote the transition matrix between slot $t_e^{(r)} + I - 1$ and $t_e^{(r)} + I$. Since the success or failure of the transmission is a function of the number of users becoming ready in slot $t_e^{(r)}$, we write $\mathbf{R}$ as $\mathbf{R} = \mathbf{S} + \mathbf{F}$, where the (i, k)th elements of $\mathbf{S}$ and $\mathbf{F}$ are defined and expressed as

$$s_{ik} \triangleq \Pr\{n(t_e^{(r)} + I) = k \text{ and transmission is successful} \mid n(t_e^{(r)} + I - 1) = i\}$$

$$= \begin{cases} 0, & k < i, \\ \dfrac{(1-\sigma)^{M-i}[i\nu(1-\nu)^{i-1}]}{1-(1-\nu)^i(1-\sigma)^{M-i}}, & k = i, \\ \dfrac{(M-i)\sigma(1-\sigma)^{M-i-1}(1-\nu)^i}{1-(1-\nu)^i(1-\sigma)^{M-i}}, & k = i+1, \\ 0, & k > i+1; \end{cases} \tag{49}$$

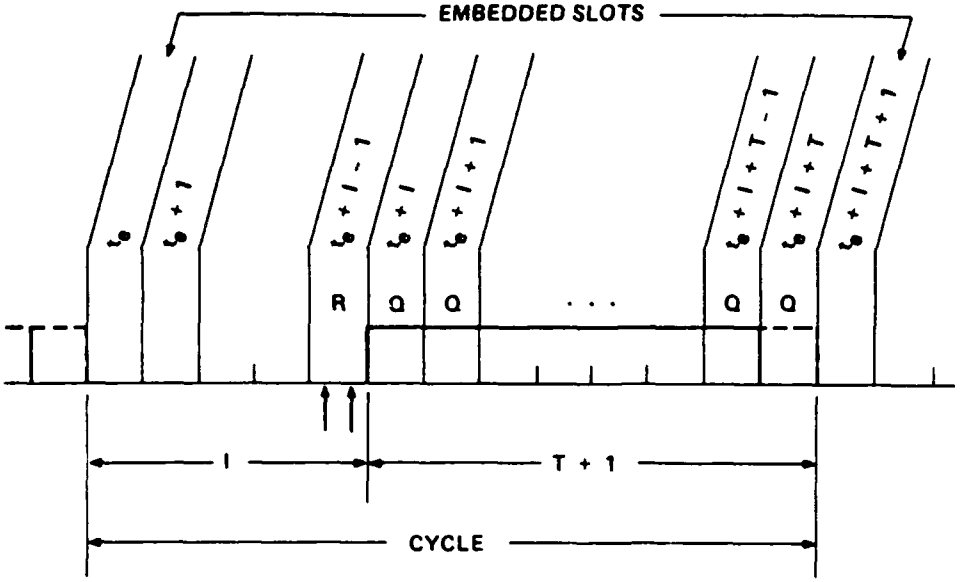


FIG. 1. The imbedded Markov chain in CSMA.

$$f_{ik} \triangleq \Pr\{n(t_e^{(r)} + I) = k \text{ and transmission is unsuccessful} \mid n(t_e^{(r)} + I - 1) = i\}$$

$$= \begin{cases} 0, & k < i, \\ \dfrac{(1-\sigma)^{M-i}[1-(1-\nu)^i - i\nu(1-\nu)^{i-1}]}{1-(1-\nu)^i(1-\sigma)^{M-i}} & k = i, \\ \dfrac{(M-i)\sigma(1-\sigma)^{M-i-1}[1-(1-\nu)^i]}{1-(1-\nu)^i(1-\sigma)^{M-i}}, & k = i+1, \\ \dfrac{\binom{M-i}{k-i}(1-\sigma)^{M-k}\sigma^{k-i}}{1-(1-\nu)^i(1-\sigma)^{M-i}}, & k > i+1. \end{cases} \tag{50}$$

During the transmission period, all new arrivals join the backlog. Thus for any $t \in [t_e^{(r)} + I + 1, t_e^{(r)} + I + \mathrm{TP}]$, we let Q denote the one-step transition matrix, for which the (i, k)th element is defined as $q_{ik} \triangleq \Pr\{n(t) = k \mid n(t-1) = i\}$ and expressed as

$$q_{ik} = \begin{cases} 0, & k < i, \\ \binom{M-i}{k-i}(1-\sigma)^{M-k}\sigma^{k-i}, & k \geq i. \end{cases} \tag{51}$$

Finally, to represent the fact that a successful transmission decreases the backlog by 1, we introduce matrix J such that its (i, k)th element is given by

$$j_{ik} = \begin{cases} 1, & k = i-1, \\ 0, & \text{otherwise.} \end{cases} \tag{52}$$

The transition matrix **P** is then expressed as

$$\mathbf{P} = \mathbf{S}\mathbf{Q}^{T+1}\mathbf{J} + \mathbf{F}\mathbf{Q}^{T_c+1}. \tag{53}$$

Let $\Pi = \{\pi_0, \pi_1, \ldots, \pi_M\}$ denote the stationary probability distribution of $n(t_e^{(r)})$. Π is obtained by the recursive solution of $\Pi = \Pi\mathbf{P}$.

Since $n(t_c^{(r)})$ is a regenerative process, the average stationary channel throughput is computed as the ratio of the time the channel is carrying successful transmission during a cycle averaged over all cycles to the average cycle length. Therefore we have

$$S = \frac{\sum_{i=0}^{M} \pi_i P_s(i) T}{\sum_{i=0}^{M} \pi_i [1/(1-\delta_i) + 1 + P_s(i)T + (1 - P_s(i))T_c]}, \tag{54}$$

where $P_s(i)$ is the probability of a successful transmission during a cycle with $n(t_c^{(r)}) = i$ and is given by

$$P_s(i) = \frac{(M-i)\sigma(1-\sigma)^{M-i-1}(1-\nu)^i + i\nu(1-\nu)^{i-1}(1-\sigma)^{M-i}}{1 - (1-\nu)^i(1-\sigma)^{M-i}}, \tag{55}$$

and where $(1-\delta_i)^{-1}$, with $\delta_i = (1-\nu)^i(1-\sigma)^{M-i}$, is the average idle period given $n(t_c^{(r)}) = i$.

Similarly, the average channel backlog is computed as the ratio of the expected sum of backlogs over all slots in a cycle (averaged over all cycles) to the average cycle length. Therefore we have

$$\bar{n} = \frac{\sum_{i=0}^{M} \pi_i (i/(1-\delta_i) + A(i))}{\sum_{i=0}^{M} \pi_i [1/(1-\delta_i) + 1 + P_s(i)T + (1 - P_s(i))T_c]}, \tag{56}$$

where $A(i)$ is the expected sum of backlogs over all slots in the busy period with $n(t_c^{(r)}) = i$ and is given by[3]

$$\begin{aligned} A(i) &= \sum_{l=0}^{T} \sum_{j=i}^{M} j[\mathbf{S}\mathbf{Q}^l]_{ij} + \sum_{l=0}^{T_c} \sum_{j=i}^{M} j[\mathbf{F}\mathbf{Q}^l]_{ij} \\ &= \sum_{j=i}^{M} j \left[\mathbf{S} \sum_{l=0}^{T} \mathbf{Q}^l + \mathbf{F} \sum_{l=0}^{T_c} \mathbf{Q}^l \right]_{ij}. \end{aligned} \tag{57}$$

By Little's result [5], the average packet delay (normalized to T) is simply expressed as

$$\bar{D} = \frac{\bar{n}}{S}. \tag{58}$$

3.3. Interdeparture Time Distribution

THEOREM 3.1. *The generating function for the interdeparture time distribution in CSMA is given by*

$$ID^*(z) = \boldsymbol{\Delta}[\mathbf{I} - \mathbf{P}_d^*(z)]^{-1}\mathbf{P}_d^*(z)\mathbf{H}, \tag{59}$$

where

(i) $\boldsymbol{\Delta}$ *is solution of* $\boldsymbol{\Delta} = \boldsymbol{\Delta}[\mathbf{I} - \mathbf{F}\mathbf{Q}^{T_c+1}]^{-1}\mathbf{S}\mathbf{Q}^{T+1}\mathbf{J}$;

(ii) $\mathbf{P}_d^*(z)$ *and* $\mathbf{P}_{\bar{d}}^*(z)$ *are matrices such that their* (i, j)*th elements are defined as*

$$[\mathbf{P}_d^*(z)]_{ij} = [\mathbf{S}\mathbf{Q}^{T+1}\mathbf{J}]_{ij} \frac{(1-\delta_i)z^{T+2}}{1-\delta_i z}, \tag{60}$$

$$[\mathbf{P}_{\bar{d}}^*(z)]_{ij} = [\mathbf{F}\mathbf{Q}^{T_c+1}]_{ij} \frac{(1-\delta_i)z^{T_c+2}}{1-\delta_i z}, \tag{61}$$

$$\delta_i = (1-\nu)^i(1-\sigma)^{M-i}; \tag{62}$$

(iii) $\mathbf{H}$ *is a column vector with all elements equal to one.*

[3] Recall that for an arbitrary matrix $\mathbf{B}$ we adopt the notation $[\mathbf{B}]_{ij}$ to represent the (i, j)th element of $\mathbf{B}$.

PROOF. The proof is similar to that given in Section 2 for slotted ALOHA. Consider an imbedded slot $t_e^{(r)}$ such that $n(t_e^{(r)}) = i$. Let $ID_i^*(z)$ denote the generating function of the distribution of time until completion of the first successful transmission following $t_e^{(r)}$. Let $I_i^*(z)$ denote the generating function of the distribution of the idle period. Since the latter is geometrically distributed with mean $1/(1 - \delta_i)$, where $\delta_i = (1 - \nu)^i(1 - \sigma)^{M-i}$, $I_i^*(z)$ is given by

$$I_i^*(z) = \frac{(1 - \delta_i)z}{1 - \delta_i z}. \tag{63}$$

$ID_i^*(z)$ is just $I_i^*(z)z^{T+1}$ if the first transmission is successful, and $I_i^*(z)z^{T_c+1}ID_j^*(z)$ if the first transmission is unsuccessful and $n(t_e^{(r+1)}) = j$. Thus, letting $p_{ij}^{(d)} \triangleq [\mathbf{S}\mathbf{Q}^{T+1}\mathbf{J}]_{ij} \triangleq [\mathbf{P}_d]_{ij}$ and $p_{ij}^{(\bar{d})} \triangleq [\mathbf{F}\mathbf{Q}^{T_c+1}]_{ij} \triangleq [\mathbf{P}_{\bar{d}}]_{ij}$, we have

$$ID_i^*(z) = \sum_{j=i-1}^{M} [p_{ij}^{(d)} I_i^*(z)z^{T+1} + p_{ij}^{(\bar{d})} I_i^*(z)z^{T_c+1} ID_j^*(z)]. \tag{64}$$

Let $\mathbf{ID}^*(z)$ denote the column vector $(ID_0^*(z), \ldots, ID_M^*(z))^T$ (the superscript T representing the transpose operation); let $\mathbf{P}_d^*(z)$ and $\mathbf{P}_{\bar{d}}^*(z)$ be as defined in eqs. (60) and (61); we can rewrite eq. (63) in matrix notation as

$$\mathbf{ID}^*(z) = \mathbf{P}_d^*(z)\mathbf{H} + \mathbf{P}_{\bar{d}}^*(z)\mathbf{ID}^*(z) \tag{65}$$

or

$$\mathbf{ID}^*(z) = [\mathbf{I} - \mathbf{P}_{\bar{d}}^*(z)]^{-1}\mathbf{P}_d^*(z)\mathbf{H}. \tag{66}$$

To obtain $\mathbf{ID}^*(z)$, we need to remove the condition on $n(t_e^{(r)})$. Let $\ldots t_d^{(r)}, t_d^{(r+1)}, \ldots$ denote the sequence of imbedded points immediately following a successful transmission. The process $\{n(t), t \in (\ldots, t_d^{(r)}, t_d^{(r+1)}, \ldots)\}$ is an imbedded Markov chain with transition probabilities given by

$$\begin{aligned} \Pr\{n(t_d^{(r+1)} = j | n(t_d^{(r)}) = i\} &= \sum_{l=1}^{\infty} [\mathbf{P}_{\bar{d}}^{l-1}\mathbf{P}_d]_{ij} \\ &= [(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d]_{ij}. \end{aligned} \tag{67}$$

The stationary distribution of $n(t_d^{(r)})$, $\Delta = (d_0, d_1, \ldots, d_M)$, is the solution of $\Delta = \Delta(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\mathbf{P}_d$. We finally have $\mathbf{ID}^*(z) = \Delta\mathbf{ID}^*(z)$, hence eq. (59). Q.E.D.

A simple recursive procedure exists for the computation of the mth moment of ID. Let $ID^{(m)} \triangleq d^m ID^*(z)/dz^m|^{z=1}$, $\mathbf{ID}^{*(m)}(z) \triangleq d^m\mathbf{ID}^*(z)/dz^m$, $\mathbf{P}_d^{*(m)}(z) \triangleq d^m\mathbf{P}_d^*(z)/dz^m$ and $\mathbf{P}_{\bar{d}}^{*(m)}(z) \triangleq d^m\mathbf{P}_{\bar{d}}^*(z)/dz^m$. (We use the convention that the derivative of a vector or matrix is the vector or matrix whose elements are derivatives of the corresponding elements in the original vector or matrix.)

COROLLARY 3.1. *$ID^{(m)}$ is given by*

$$ID^{(m)} = \Delta \cdot \mathbf{ID}^{*(m)}(1), \tag{68}$$

where $\mathbf{ID}^{*(m)}(1)$ is recursively determined by

$$\begin{aligned} \mathbf{ID}^{*(m)}(1) = (\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}\Bigg[&\mathbf{P}_d^{*(m)}(1)\cdot\mathbf{H} + \mathbf{P}_{\bar{d}}^{*(m)}(1)\cdot\mathbf{H} \\ &+ \sum_{k=1}^{m-1} \binom{m}{k} \mathbf{P}_{\bar{d}}^{*(k)}(1)\cdot\mathbf{ID}^{*(m-k)}(1)\Bigg]. \end{aligned} \tag{69}$$

PROOF. It can be easily proved by induction that differentiating eq. (65) m times leads to the relation,

$$\mathbf{ID}^{*(m)}(z) = \mathbf{P}_d^{*(m)}(z)\mathbf{H} + \sum_{k=0}^{m} \binom{m}{k} \mathbf{P}_{\bar{d}}^{*(k)}(z) \cdot \mathbf{ID}^{*(m-k)}(z). \tag{70}$$

Letting $z = 1$ and observing that $\mathbf{ID}^*(1) = \mathbf{H}$ and $\mathbf{P}_{\bar{d}}^*(1) = \mathbf{P}_{\bar{d}}$, we get eq. (69). Q.E.D.

The average interdeparture time, in particular, is given by

$$\mathbf{ID}^{(1)} = \mathbf{\Delta}(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}[\mathbf{P}_d^{*(1)}(1) + \mathbf{P}_{\bar{d}}^{*(1)}(1)]\mathbf{H}, \tag{71}$$

where $[\mathbf{P}_d^{*(1)}(1) + \mathbf{P}_{\bar{d}}^{*(1)}(1)]\mathbf{H}$ is a column vector whose ith element is simply

$$\frac{1}{1-\delta_i} + \sum_{j=i-1}^{M} [p_{ij}^{(d)}(T+1) + p_{ij}^{(\bar{d})}(T_c+1)].$$

The variance of ID is given by

$$\begin{aligned}\mathrm{Var}[\mathrm{ID}] = {} & \mathbf{\Delta}(\mathbf{I} - \mathbf{P}_{\bar{d}})^{-1}[[\mathbf{P}_d^{*(2)}(1) + \mathbf{P}_{\bar{d}}^{*(2)}(1)]\mathbf{H} + 2\mathbf{P}_d^{*(1)}(1)\mathbf{ID}^{*(1)}(1)] \\ & + \mathbf{ID}^{(1)} - [\mathbf{ID}^{(1)}]^2.\end{aligned} \tag{72}$$

3.4. DISTRIBUTION OF PACKET DELAY. Consider an imbedded slot $t_e^{(r)}$ such that $n(t_e^{(r)}) = i$. Consider a *tagged* user in the backlog of size i, and let $D_i^*(z)$ denote the z-transform of the distribution of delay (counted starting from $t_e^{(r)}$) until the tagged user is successful. Let $\mathbf{D}^*(z)$ be the column vector $(D_1^*(z), \ldots, D_M^*(z))^{\mathrm{T}}$.

THEOREM 3.2. *$\mathbf{D}^*(z)$ is given by*

$$\mathbf{D}^*(z) = [\mathbf{I} - \mathbf{P}_{\bar{s}}^*(z)]^{-1}\mathbf{P}_s^*(z)\mathbf{H}, \tag{73}$$

where $\mathbf{H}$ is the column vector with $M+1$ elements all equal to one, $\mathbf{I}$ is the $M \times M$ identity matrix, and $\mathbf{P}_s^*(z)$ and $\mathbf{P}_{\bar{s}}^*(z)$ are matrices of size $M \times (M+1)$ and $M \times M$, respectively, with their (i,j)th elements defined as

$$[\mathbf{P}_s^*(z)]_{ij} = [\mathbf{S}_s\mathbf{Q}^{T+1}\mathbf{J}]_{ij}\frac{(1-\delta_i)z^{T+2}}{1-\delta_i z}, \qquad 1 \le i \le M, \quad 0 \le j \le M, \tag{74}$$

$$\begin{aligned}[\mathbf{P}_{\bar{s}}^*(z)]_{ij} = {} & [\mathbf{S}_{\bar{s}}\mathbf{Q}^{T+1}\mathbf{J}]_{ij}\frac{(1-\delta_i)z^{T+2}}{1-\delta_i z} \\ & + [\mathbf{F}\mathbf{Q}^{T_c+1}]_{ij}\frac{(1-\delta_i)z^{T_c+2}}{1-\delta_i z}, \qquad 1 \le i \le M, \quad 1 \le j \le M,\end{aligned} \tag{75}$$

$$[\mathbf{S}_s]_{ik} = \begin{cases} \dfrac{1}{i}[\mathbf{S}]_{i,i}, & 1 \le i \le M, \quad k = i, \\ 0, & 1 \le i \le M, \quad 0 \le k \le M, \quad k \ne i, \end{cases} \tag{76}$$

$$[\mathbf{S}_{\bar{s}}]_{ik} = [\mathbf{S}]_{ik} - [\mathbf{S}_s]_{ik}, \qquad 1 \le i \le M, \quad 0 \le k \le M. \tag{77}$$

PROOF. The proof is similar to that given in Theorem 3.1 for $\mathbf{ID}^*(z)$. Noting that

$$[\mathbf{S}_s]_{ik} = \Pr\{n(t_e^{(r)} + I) = k \text{ and tagged user successful} \mid n(t_e^{(r)}) = i\},$$

we have

$$D_i^*(z) = \sum_{j=i-1}^{M} [\mathbf{S}_s\mathbf{Q}^{T+1}\mathbf{J}]_{ij} \frac{(1-\delta_i)z^{T+2}}{1-\delta_i z} + \sum_{\substack{j=i-1 \\ j\neq 0}}^{M} \left[[\mathbf{S}_F\mathbf{Q}^{T+1}\mathbf{J}]_{ij} \frac{(1-\delta_i)z^{T+2}}{1-\delta_i z} + [\mathbf{F}\mathbf{Q}^{T_c+1}]_{ij} \frac{(1-\delta_i)z^{T_c+2}}{1-\delta_i z} \right] D_j^*(z), \quad (78)$$

hence eq. (73). Q.E.D.

THEOREM 3.3. *The distribution of delay in CSMA is given by*

$$D^*(z) = \gamma_0^{(T+1)} z^{T+1} + \sum_{l=0}^{T} z^l \Gamma^{(l)} \mathbf{D}^*(z), \quad (79)$$

where $\gamma_0^{(T+1)}$ *and* $\Gamma^{(l)} \triangleq (\gamma_1^{(l)}, \gamma_2^{(l)}, \ldots, \gamma_M^{(l)})$, $0 \le l \le T$, *are defined in eq.* (87) *in the proof.*

PROOF. To complete the delay calculation, we need to compute $\gamma_j^{(l)}$, the probability that an arbitrary new packet arrives in a slot which is l slots away from the next imbedded slot and finds itself at the beginning of this imbedded slot in a backlog of size j; indeed, its delay is then $z^l D_j^*(z)$. We use the index $j = 0$ to represent the case where the arriving packet starts its successful transmission upon its arrival; clearly, in this case the arrival must have taken place in slot $t_e^{(r)} + I - 1$, l must equal $T + 1$, and its delay is given by z^{T+1}. We also note the following. Given that a user has generated a packet in a transmission period of size TP, then the generating function of the time since the generation of the packet until the next imbedded slot is given by

$$A_{\mathrm{TP}}^*(z) \triangleq \sum_{l=0}^{\mathrm{TP}} \alpha_l^{(\mathrm{TP})} z^l = \sum_{l=0}^{\mathrm{TP}} \frac{\sigma(1-\sigma)^{\mathrm{TP}-l-1}}{1-(1-\sigma)^{\mathrm{TP}}} z^l. \quad (80)$$

From the point of view of delay we distinguish four types of packet arrivals.

Type 1. packets which arrive in slot $t_e^{(r)} + I - 1$ and are successful in their first transmission; the distribution of delay is simply

$$d_0^{*(1)}(z) = z^{T+1}. \quad (81)$$

Type 2. packets which arrive during a successful transmission period and find themselves at the end of the transmission period in a backlog of size j; the distribution of their delay is

$$d_j^{*(2)}(z) = A_{T+1}^*(z) D_j^*(z). \quad (82)$$

Type 3. packets which arrive in slot $t_e^{(r)} + I - 1$, are not successful in their first transmission, and find themselves at the end of the transmission period in backlog of size j; the distribution of delay is given by

$$d_j^{*(3)}(z) = z^{T_c+1} D_j^*(z). \quad (83)$$

Type 4. packets which arrive during an unsuccessful transmission period and find themselves at the end of the transmission period in a backlog of size j; the distribution of delay is given by

$$d_j^{*(4)}(z) = A_{T_c+1}^*(z) D_j^*(z). \quad (84)$$

Given that $n(t_e^{(r)}) = i$, then with probability $[\mathbf{S}]_{i,i+1}[\mathbf{Q}^{T+1}\mathbf{J}]_{i+1,j}$, $i \le j \le M$, there are one arrival of type 1 and $j - i$ arrivals of type 2; with probability $[\mathbf{S}]_{i,i}[\mathbf{Q}^{T+1}\mathbf{J}]_{i,j}$,

$i-1 \le j \le M$, there are $j+1-i$ arrivals of type 2; and with probability $[\mathbf{F}]_{i,k}[\mathbf{Q}^{T_c+1}]_{k,j}$, $i \le j \le M$, there are $k-i$ arrivals of type 3 and $j-k$ arrivals of type 4. Given $0 \le j \le M$ and $1 \le m \le 4$, we let $\xi_j^{(m)}$ denote the probability that an arbitrary arrival is of type m and finds itself in a backlog of size j (clearly for $m = 1$ we have $j = 0$, and for $m > 1$, $j \neq 0$). $\xi_j^{(m)}$ is also the fraction of such packet arrivals, and therefore we have

$$\xi_j^{(m)} = \begin{cases} 0, & m = 1, \quad j \neq 0, \\ K \sum_{i=0}^{M-1} \pi_i [\mathbf{S}]_{i,i+1}, & m = 1, \quad j = 0, \\ K \sum_{i=0}^{j} \pi_i \{(j-i)[\mathbf{S}]_{i,i+1}[\mathbf{Q}^{T+1}\mathbf{J}]_{i+1,j} \\ \qquad + (j+1-i)[\mathbf{S}]_{i,i}[\mathbf{Q}^{T+1}\mathbf{J}]_{i,j}\}, & m = 2, \quad 1 \le j \le M, \\ K \sum_{i=0}^{j} \sum_{k=i}^{j} \pi_i (k-i)[\mathbf{F}]_{i,k}[\mathbf{Q}^{T_c+1}]_{k,j}, & m = 3, \quad 1 \le j \le M, \\ K \sum_{i=0}^{j} \sum_{k=i}^{j} \pi_i (j-k)[\mathbf{F}]_{i,k}[\mathbf{Q}^{T_c+1}]_{k,j} & m = 4, \quad 1 \le j \le M, \end{cases} \tag{85}$$

where K is a normalizing constant such that $\xi_0^{(1)} + \sum_{m=2}^{4} \sum_{j=1}^{M} \xi_j^{(m)} = 1$. As a result, we have

$$D^*(z) = \xi_0^{(1)} d_0^{*(1)}(z) + \sum_{m=2}^{4} \sum_{j=1}^{M} \xi_j^{(m)} d_j^{*(m)}(z). \tag{86}$$

From eqs. (80)–(86), we easily deduce eq. (79), where

$$\gamma_j^{(l)} = \begin{cases} 0, & j = 0, \quad l \neq T+1, \\ K \sum_{i=0}^{M-i} \pi_i [\mathbf{S}]_{i,i+1}, & j = 0, \quad l = T+1, \\ K \sum_{i=0}^{j} \pi_i \Big\{ \alpha_l^{(T+1)}(j-i)[\mathbf{S}]_{i,i+1}[\mathbf{Q}^{T+1}\mathbf{J}]_{i+1,j} \\ \qquad + \alpha_l^{(T+1)}(j+1-i)[\mathbf{S}]_{i,i}[\mathbf{Q}^{T+1}\mathbf{J}]_{i,j} \\ \qquad + \sum_{k=i}^{j} \alpha_l^{(T_c+1)}(j-k)[\mathbf{F}]_{i,k}[\mathbf{Q}^{T_c+1}]_{k,j} \Big\}, & 1 \le j \le M, \\ & 0 \le l \le T_c, \\ K \sum_{i=0}^{j} \pi_i \Big\{ \alpha_l^{(T+1)}(j-i)[\mathbf{S}]_{i,i+1}[\mathbf{Q}^{T+1}\mathbf{J}]_{i+1,j} \\ \qquad + \alpha_l^{(T+1)}(j+i-1)[\mathbf{S}]_{i,i}[\mathbf{Q}^{T+1}\mathbf{J}]_{i,j} \\ \qquad + \sum_{k=i}^{j} (k-i)[\mathbf{F}]_{i,k}[\mathbf{Q}^{T_c+1}]_{k,j} \Big\}, & 1 \le j \le M, \\ & l = T_c + 1, \\ K \sum_{i=0}^{j} \pi_i \Big\{ \alpha_l^{(T+1)}(j-i)[\mathbf{S}]_{i,i+1}[\mathbf{Q}^{T+1}\mathbf{J}]_{i+1,j} \\ \qquad + \alpha_l^{(T+1)}(j+1-i)[\mathbf{S}]_{i,i}[\mathbf{Q}^{T+1}\mathbf{J}]_{i,j} \Big\}, & 1 \le j \le M, \\ & T_c + 2 \le l \le T. \end{cases} \tag{87}$$

Q.E.D.

As with interdeparture times, a simple recursive procedure exists for the computation of all moments of the delay. Let $D^{(m)} \triangleq d^m D^*(z)/dz^m|_{z=1}$, $\mathbf{D}^{*(m)}(z) \triangleq d^m \mathbf{D}^*(z)/dz^m$, $\mathbf{P}_s^{*(m)}(z) \triangleq d^m \mathbf{P}_s^*(z)/dz^m$, and $\mathbf{P}_{\bar{s}}^{*(m)}(z) \triangleq d^m \mathbf{P}_{\bar{s}}^*(z)/dz^m$.

COROLLARY 3.2. *$D^{(m)}$ is given by*

$$D^{(m)} = \frac{(T+1)!}{(T+1-m)!}\gamma_0^{(T+1)} + \sum_{l=0}^{T}\left[\sum_{k=0}^{\min(l,m)}\binom{m}{k}\frac{l!}{(l-k)!}\Gamma^{(l)}\mathbf{D}^{*(m-k)}(1)\right], \quad (88)$$

where $\mathbf{D}^{*(m)}(1)$ is recursively determined by

$$\mathbf{D}^{*(m)}(1) = [\mathbf{I} - \mathbf{P}_T^*(1)]^{-1}\left[\mathbf{P}_s^{*(m)}(1)\mathbf{H} + \mathbf{P}_T^{*(m)}(1)\mathbf{H} + \sum_{k=1}^{m-1}\binom{m}{k}\mathbf{P}_T^{*(k)}(1)\mathbf{D}^{*(m-k)}(1)\right]. \quad (89)$$

PROOF. By successive differentiations of eqs. (78) and (79), and letting $z = 1$, we can easily establish eqs. (88) and (89). Q.E.D.

4. *Simple Numerical Examples*

We illustrate the analytic results obtained in this paper by considering some simple numerical examples. To keep this task simple, we restrict ourselves to the slotted ALOHA case.

With $M = 2$ it is possible to reach closed-form solutions. Let, for example, $p = \lambda = 0.5$. The analysis of average performance leads to $S = \frac{8}{17}$, $\bar{n} = \frac{26}{17}$, and $\bar{D} = \frac{13}{4}$ for DFT, and $S = 0.5$, $\bar{n} = 1$, and $\bar{D} = 3$ for IFT. The calculation of packet delay distribution leads to the following:

DFT protocol:

$$\Gamma = \left[\frac{1}{4}, \frac{3}{4}\right],$$

$$D_1^*(z) = \frac{0.5z(1-0.5z)}{1-0.875z+0.125z^2},$$

$$D_2^*(z) = \frac{0.25z}{1-0.875z+0.125z^2},$$

$$D^*(z) = \frac{(1.25/4)z(1-0.2z)}{1-0.875z+0.125z^2}$$

$$= 0.3125z + 0.2109375z^2 + 0.1455078z^3 + 0.1009521z^4 + 0.0701445z^5 + 0.0495573z^6 + \cdots.$$

IFT protocol:

$$\gamma_0 = \frac{1}{2}, \qquad \Gamma = \left(0, \frac{1}{2}\right),$$

$$D_1^*(z) = D_2^*(z) = \frac{0.25z}{1-0.75z},$$

$$D^*(z) = \frac{1}{2}z + \frac{1}{2}z\frac{0.25z}{1-0.75z},$$

$$= \frac{1}{2}z + \frac{1}{2}[0.25z^2 + \cdots + 0.25(0.75)^{l-2}z^l + \cdots].$$

The probability mass functions for these examples are shown in Figure 2. One can easily verify for both cases that $\bar{D} = D^{(1)}$ and $\mathrm{ID}^{(1)} = 1/S$. For larger values of M the calculation of interdeparture time and packet delay distributions requires the use of

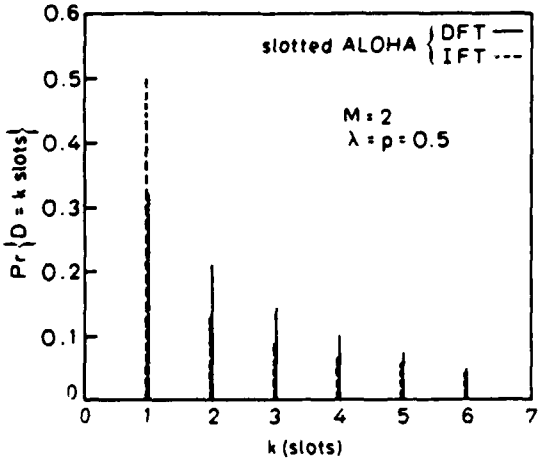


FIG. 2 Pr{$D = k$ slots} versus k for slotted ALOHA ($M = 2$, $\lambda = p = 0.5$).

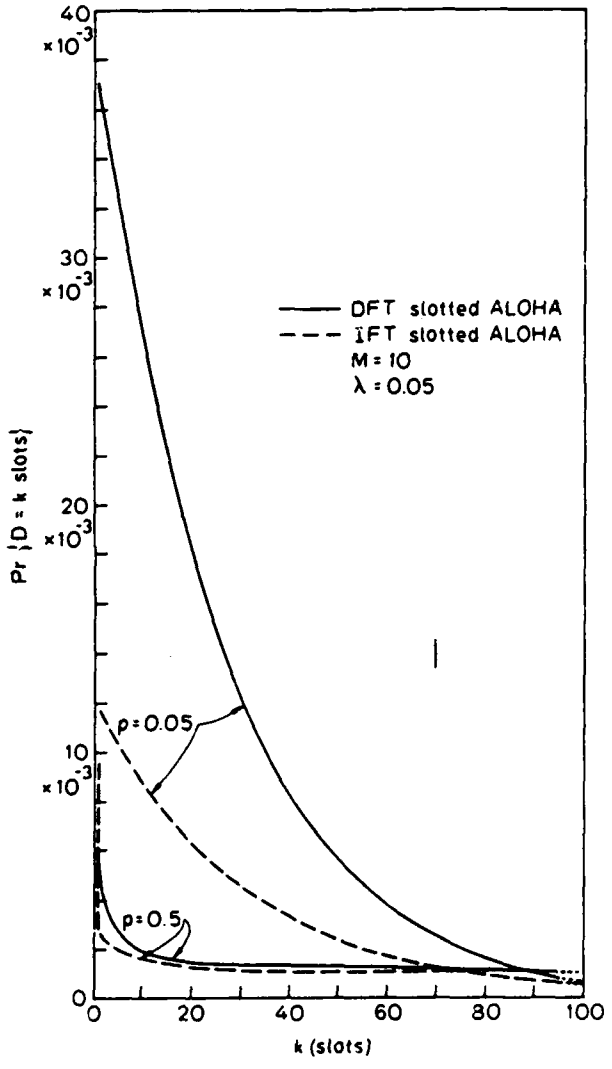


FIG. 3. Packet delay distribution for slotted ALOHA ($M = 10$, $\lambda = 0.05$, $p = 0.05$ and 0.5).

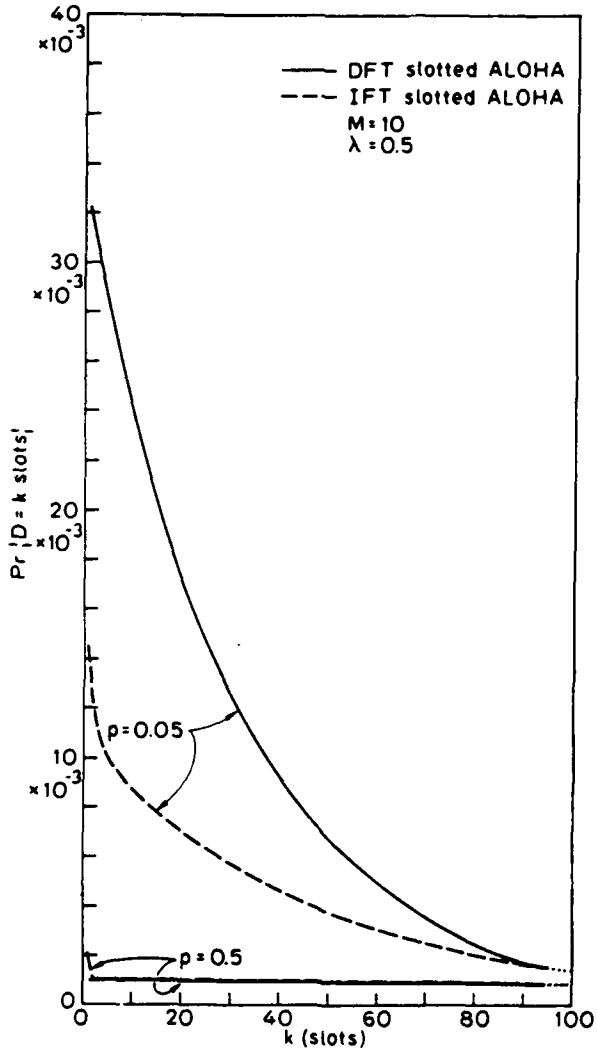


FIG. 4. Packet delay distribution for slotted ALOHA ($M = 10$, $\lambda = 0.5$, $p = 0.05$ and 0.5).

a computer. We show in Figures 3 and 4 some numerical results obtained for $M = 10$ and various values of λ and p.

5. *Conclusion*

We have derived in this paper the distribution of packet delay and interdeparture time for slotted ALOHA and CSMA channels with a finite population of interactive users. In slotted ALOHA it is interesting to note the "geometric" form that these distributions have, namely, $\mathbf{\Delta P}_d^{k-1}(\mathbf{P} - \mathbf{P}_d)\mathbf{H}$ and $\mathbf{\Gamma P}_r^{k-1}(\mathbf{P} - \mathbf{P}_r)\mathbf{H}$, and the special form that successive derivatives of their generating functions have, a form which is similar to that encountered in geometric distributions. Simple closed-form expressions for their moments have been obtained.

For CSMA we have derived simple recursive procedures to compute all moments of packet delay and interdeparture time and closed-form expressions for their

generating functions. Contrary to their counterpart in slotted ALOHA, the generating functions derived for CSMA may not prove very practical in the numerical computation of the distributions. Indeed, they require symbolic inversion of matrices whose elements are themselves z-transforms. However, a numerical procedure which allows us to compute approximations to the distributions can be devised as follows. Consider for example the case of interdeparture time. It is easily realized that

$$\mathbf{ID}^*(z) = \sum_{m=1}^{\infty} \mathbf{ID}^*_{(m)}(z), \tag{90}$$

where

$$\mathbf{ID}^*_{(m)}(z) \triangleq (\mathrm{ID}^*_{0,(m)}(z), \ldots, \mathrm{ID}^*_{M,(m)}(z)), \tag{91}$$

$$\mathbf{ID}^*_{(m)}(z) = \mathbf{P}^*_d(z)\mathbf{ID}^*_{(m-1)}(z), \qquad \text{for} \quad m \geq 2, \tag{92}$$

$$\mathbf{ID}^*_{(1)}(z) = \mathbf{P}^*_d(z)\mathbf{H}. \tag{93}$$

Equations (92) and (93) are equivalent to

$$\mathrm{ID}^*_{i,(1)}(z) = \sum_{j=i-1}^{M} [\mathbf{P}_d]_{ij}(1 - \delta_i)z^{T+2}[1 + \delta_i z + \delta_i^2 z^2 + \cdots], \tag{94}$$

and for $m \geq 2$,

$$\mathrm{ID}^*_{i,(m)}(z) = \sum_{j=i-1}^{M} [\mathbf{P}_d]_{ij}(1 - \delta_i)z^{T_c+2}[1 + \delta_i z + \delta_i^2 z^2 + \cdots]\mathrm{ID}^*_{j,(m-1)}(z). \tag{95}$$

Thus by successive polynomial multiplications and additions one can generate numerically an approximation of the distribution of ID, the accuracy of which is a function of the position at which the infinite series are truncated and of the maximum value given to m. A similar procedure can be devised for the distribution of delay.

It is interesting to note that the approach used in this paper is applicable to a more general class of models, namely, discrete-time queuing systems with finite queue size and state-dependent arrival and departure processes. The analysis presented here corresponds to the random-order service discipline, but can be simply extended to the first-come-first-served discipline as well. The finite-queue-size restriction can be relaxed, but in that case the conditions for ergodicity will have to be established.

REFERENCES

1. Abramson, N. The throughput of packet broadcasting channels. *IEEE Trans. Commun. COM-25* (Jan. 1977), 117–128.
2. Kleinrock, L., and Lam, S. Packet switching in a slotted satellite channel. Proc. AFIPS 1973 NCC, Vol. 42, AFIPS Press, Arlington, Va., pp. 703–710.
3. Kleinrock, L., and Lam, S.S. Packet switching in a multi-access broadcast channel: Performance evaluation. *IEEE Trans. Commun. COM-23* (Apr. 1975), 410–423.
4. Kleinrock, L., and Tobagi, F.A. Packet switching in radio channels, Part I: Carrier sense multiple-access modes and their throughput–delay characteristics. *IEEE Trans. Commun. COM-23* (Dec. 1975), 1400–1416.
5. Little, J. A proof of the queuing formula $L = \lambda W$. *Oper. Res. 9* (Mar.–Apr. 1961), 383–387.
6. Metcalfe, R.M., and Boggs, D.R. Ethernet: Distributed packet switching for local computer networks. *Commun. ACM 19*, 7 (July 1976), 395–404.
7. Roberts, L.G. ALOHA packet system with and without slots and capture. *Comput. Commun. Rev. 5* (Apr. 1975), 28–42.
8. Tobagi, F. Analysis of a two-hop centralized packet radio network, Part I: Slotted ALOHA. *IEEE Trans. Commun. COM-28* (Feb. 1980), 196–207.
9. Tobagi, F.A. Multiaccess protocols in packet communication systems. *IEEE Trans. Commun. COM-28*, 4 (Apr. 1980), 468–488.

10. TOBAGI, F., AND HUNT, V.B. Performance analysis of carrier sense multiple access with collision detection. *Comput. Networks 4*, 5 (Oct./Nov. 1980), 245–259.
11. TOBAGI, F., AND KLEINROCK, L. Packet switching in radio channels, Part IV: Stability considerations and dynamic control in carrier sense multiple access. *IEEE Trans. Commun. COM-25* (Oct. 1977), 1103–1120.

RECEIVED MARCH 1980; REVISED AUGUST 1981; ACCEPTED NOVEMBER 1981

ON CSMA-CD LOCAL NETWORKS AND VOICE COMMUNICATION*

Fouad A. Tobagi and Noel Gonzalez-Cawley

Computer Systems Laboratory
Stanford University
Stanford, CA 94305
(415) 497-1708

ABSTRACT

We consider in this paper local networks of the CSMA-CD broadcast bus type, exemplified by ETHERNET, and investigate their performance when supporting voice communication. For such real-time application, we define network performance as the maximum number of voice sources accommodated for a given maximum delay requirement and a tolerable packet loss rate. We study the effect on this performance of various system parameters such as channel bandwidth, vocoder rate, delay requirement and packet loss rate.

I. INTRODUCTION

A great deal of discussion can be seen in the recent literature regarding local networks and their applicability to many of today's local area communications needs. These needs have primarily consisted of data communication applications such as computer-to-computer data traffic, terminal-to-computer data traffic, and the like. More recently, a new line of thought has been apparent. It is the desire to integrate voice communication on local data networks. The reason for this is threefold: (i) voice is an office communication application just as computer data, facsimile, etc.; (ii) recent advances in vocoder technology have shown that digitized speech constitutes a digital communication application which is within the capabilities of local area data networks; and (iii) today's local network architectures, especially the broadcast bus type, offer very elegant solutions to the local communications problem, from both the point of view of simplicity in topology and device interconnection, and the point of view of flexibility in satisfying growth and variability in the environment.

While existing solutions are elegant, they are not without their limitations in performance. Some of these limitations arise as the characteristics of the environment and data traffic requirements being supported by these solutions deviate from those assumed in the original design. Examples of such characteristics are: packet length distribution, packet generation pattern, channel data rate, delay requirements, geographical area to be spanned, etc.

In this paper we consider local networks of the broadcast bus type, exemplified by Ethernet [1], and investigate the performance of such systems when supporting voice communication. In particular we study the effect on performance of various system parameters, such as channel bandwidth, vocoder rate, delay requirement, allowable packet loss rate, etc.... For comparison purposes, we also consider an ideal conflict-free TDMA case which is undoubtedly the most suitable for voice traffic exhibiting a deterministic generation process, and thus provides the ultimate performance one can achieve.

We begin by describing, in Section II, the network in question; namely, the broadcast bus system architecture, and the Carrier Sense Multiple Access scheme used. In Section III, we discuss the main characteristics of voice traffic and its requirements. Finally, in Section IV, we present and discuss numerical results obtained from a simulation of this system supporting voice communication.

II. BROADCAST BUS SYSTEMS AND THE CARRIER SENSE MULTIPLE ACCESS SCHEME

In a broadcast bus network, all devices share a single communication medium, typically a coaxial cable, to which they are connected via passive taps. When transmitted by a device, signals propagate in both directions, thus reaching all other devices. The device interface is such that it recognizes and accepts messages addressed to it.

The difficulty in controlling access to the channel by users who can only communicate via the channel itself has given rise to what is known as random access techniques. The best known such scheme appropriate to broadcast bus networks is Carrier Sense Multiple Access (CSMA) [2]. In CSMA, the risk of a collision (consisting of

* This research was supported by the U.S. Army, CECOM, Fort Monmouth, New Jersey, under Army Research Office Contract No. DAAG 29-79-C-0138.

overlapping packet transmissions) is decreased by having users sense the channel prior to transmission. If the channel is sensed busy, then transmission is inhibited. Conflicting users schedule retransmission of their packets to some later time, incurring a random rescheduling delay. There are several CSMA protocols. In the so-called nonpersistent CSMA, a user which finds the channel busy simply schedules the retransmission of the packet to some later time. In the p-persistent CSMA, a user which finds the channel busy monitors the channel, waits until the channel goes idle, and then performs the "p-process", which consists of transmitting the packet with probability p, and waiting the maximum end-to-end propagation delay with probability 1-p; at this new point in time, it senses the channel and again, if the channel is sensed idle, it repeats the p-process, otherwise it repeats the entire procedure.

Given the physical characteristics of data transmission on coaxial cables, in addition to sensing carrier it is possible for transceivers to detect interference among several transmissions (including their own) and abort the transmission of their colliding packets. This gives rise to a variation of CSMA which we refer to as Carrier Sense Multiple Access with Collision Detection (CSMA-CD) [1,3].

A new version of CSMA which includes message-based priority functions, referred to as P-CSMA, has been recently proposed and analyzed [4]. The scheme is based on the principle that access right to the channel is exclusively granted to ready messages of the current highest priority level. This is simply done by the means of reservation bursts and carrier sensing. For more details, the reader is referred to [4].

The performance of a CSMA broadcast bus system is normally characterized by two main measures: channel capacity and the throughput-delay tradeoff. Channel capacity is defined as the maximum throughput that the network is able to support. The throughput-delay measure is the relationship which exists between the average packet delay and the channel throughput. It should be clear that, due to collisions and retransmissions, channel capacity is always below the available channel bandwidth, and that throughput and delay have to be traded off: the larger the throughput is, the larger is the average packet delay.

Let W denote the channel bandwidth (in bits/seconds), d the length of the cable between the extreme users, and B the number of bits per packet (assuming fixed size packets). We let τ denote the end-to-end delay defined as the time from the starting of a transmission to the starting of reception between the extreme users, and T denote the transmission time of a packet; i.e., $T=B/W$. In all CSMA protocols, given that a transmission is initiated on an empty channel, it is clear that it takes <u>at most</u> τ sec. for the packet transmission to reach all devices; beyond this time the channel is guaranteed to be sensed busy for as long as data transmission is in progress. A collision can occur only if another transmission is initiated before the current one is sensed. Thus, the first τ sec. of a packet transmission represents its (maximum) vulnerable period and has a key effect on the performance of CSMA protocols. In particular we note that the performance of CSMA degrades as the ratio $\tau/T=\tau W/B$ increases; that is, as the propagation delay τ increases, the channel bandwidth W increases and/or the packet size B decreases. Among all protocols previously mentioned, the p-persistent CSMA-CD provides the best performance [2,3].

Both stochastic analysis and computer simulation have been used to evaluate the average stationary performance of CSMA and P-CSMA [2,3,4]. In that modeling effort it was assumed that for each user the packet intergeneration time is a random variable with a memoryless distribution (either exponential, or discrete-time geometric with the time unit equal to τ sec.). When dealing with voice applications, such an assumption is not adequate as the packet generation process is to a first approximation deterministic (see Section III below). Moreover, due to the real-time constraints encountered in voice communication, average performance is not sufficient, and one has to derive the distribution of delay or delay percentiles. This renders stochastic analysis rather difficult, and therefore we resort to simulation techniques for our study. The version of the simulator used in this investigation is that corresponding to P-CSMA. This was done with the intent that if voice and data were to be integrated on the same network, then, due to the strict end-to-end delay requirement in voice applications, one suspects that the prioritized scheme would be more appropriate. Indeed, by giving priority to voice packets over data packets, the scheme will help guarantee to a certain extent the delay constraint for voice packets even in the presence of data traffic. In fact, analysis and simulation of P-CSMA with two classes of traffic has already provided indication to that effect [4]. Note, however, that in the present study we consider that there exists only one class of traffic, namely voice, and that it is given the highest priority. The only difference between P-CSMA and CSMA in this case is that with the former there is an additional overhead incurred in the implementation of the priority function which degrades the performance slightly as compared to CSMA. This overhead is function of the ratio $\tau W/B$, and thus the degradation is more important as the ratio $\tau W/B$ becomes larger.

III. CHARACTERISTICS OF VOICE TRAFFIC AND VOICE SOURCES

It is assumed that vocoders digitize voice at some constant rate of V bits per second. Bits are grouped to form packets, which are then transmitted via the network to the destination vocoder. Let B denote the number of bits per packet. B is the sum of two components: B_h, which encompasses all overhead bits comprising the preamble, the packet header and the checksum, and B_v, the

information bits. The time to form a packet, T_f, is given by

$$T_f = \frac{B_v}{V}$$

T_f is also the packet intergeneration time for a vocoder. To achieve interactive speech and smooth playback operation, it is important to keep the end-to-end delay for most bits of voice information within tight bounds. End-to-end delay is defined as the time from when the bit is generated at the originating vocoder until it is received at the destination vocoder. Two components of delay are identified: the packet formation delay, T_f, and the packet network delay, D_n. The network delay is defined as the time since the packet is formed until it is successfully received at the destination. Denoting by D_m the maximum allowable delay for voice bits, a voice packet is acceptable only if $T_f+D_n \leq D_m$. Packets which do not satisfy this inequality are assumed to be lost. Usually speech can be effectively synthesized at the destination if the rate of lost packets does not exceed a maximum L. In voice applications, the performance measure is defined as the maximum number of vocoders that can be supported by the network under the delay constraint D_m and a tolerable loss rate L.

We assume that each voice source possesses a transmit buffer with room for exactly one packet. Whenever this buffer is nonempty, the station attempts transmission of the packet on the channel according to P-CSMA. We furthermore assume that, if the buffer is nonempty when a new packet is generated, then the former is lost and the latter occupies the buffer (i.e., the order of service is last-come-first-served). Although this model appears to be restrictive a priori, we shall show in the following section that this is not so. In fact, for a given delay requirement D_m, the optimum packet size which maximizes the number of voice sources is such that $D_n \leq T_f$ for all values of L. That is, at optimum we have $D_m \leq 2T_f$, and therefore there is no need for a transmit buffer of size larger than one, and LCFS is the appropriate queueing discipline. At optimum, packet loss contributing to L is only due to excess delay and not to lack of buffers.

Before we proceed with the discussion of the numerical results obtained from the simulation of P-CSMA, we undertake here an idealized analysis which provides an upper bound on the performance. In essence, it consists of assuming that network delay D_n is ideally deterministic and equal to only the transmission time on the channel of bandwidth W. With this assumption, $D_n=B/W$ and the condition $T_f+D_n \leq D_m$ leads to

$$\frac{B_v}{V} + \frac{B_h+B_v}{W} \leq D_m \qquad (1)$$

and thus,

$$B_v \leq \frac{(WD_m-B_h)V}{V+W} \qquad (2)$$

Given that M voice sources are active, the bandwidth constraint is then written as

$$W \geq MV \frac{(B_v+B_h)}{B_v} \qquad (3)$$

Eqs. (2) and (3) lead to a maximum value of M given by

$$M^* = \frac{WD_m-B_h}{VD_m+B_h} \qquad (4)$$

This ideal analysis in fact corresponds to TDMA in which perfect synchronization is achieved; i.e., the voice packet for a user is ready for transmission exactly at the beginning of the slot assigned to that user. This is simply achieved by having the vocoder synchronize the formation time of the packet with the boundary of its assigned slot. Therefore, with M users and TDMA frames of M slots, D_n is equal to one slot (i.e., $(B_h+B_v)/W$), while T_f is equal to M slots; hence, equations (1) and (3).

Equation (4) illustrates the effect of the overhead B_h on M*. If $B_h=0$, then $M^*=W/V$, independent of D_m. In that case, however, the optimum packet size is function of D_m, given by

$$B_v = D_m/(\frac{1}{V} + \frac{1}{W})$$

The smaller D_m is, the smaller the packet size is. With $B_h \neq 0$, M* is function of D_m, and decreases as D_m decreases. Indeed, with $B_h \neq 0$, the packet size cannot be arbitrarily decreased, as the effect of overhead becomes more and more severe. In Figures 3 and 4 below, the dashed curves represent M* as a function of V for W=1MBPS and W=10MBPS respectively, B_h=200 bits and various values of D_m. D_m=200 msec. corresponds roughly to the case $B_h=0$ or $D_m \to \infty$. For D_m=2 msec., the effect of B_h is so important that M* is limited to very small values and is rather insensitive to V.

IV. DISCUSSION OF NUMERICAL RESULTS

We consider a P-CSMA network, 1 Km long with an end-to-end propagation delay τ of approximately 10 μsec. We assume B_h=200 bits, which account for a 64-bit preamble, a 32-bit CRC and 104 bits for addressing and other control information. We consider various values for the vocoder bandwidth: V=16, 24, 32 and 64 KBPS; and two values for the channel bandwidth: W=1 and 10MBPS.

The delay constraint on voice bits depends on the type of voice communication being considered. The first type we identify is that of real-time voice communication within the local environment (i.e., all parties are on the local network); in this case, interactive communication can be effectively accomplished if D_m is on the order of 200 msec. (or even higher). The second type is that of real-time voice communication via the Public Switched Telephone Network (i.e., where all parties are not on the local network); in this case, D_m must be restricted to a value on the order of 20 msec. Finally, we distinguish a third type which arises when the goal is to have network delay which does not exceed that experienced in a PBX switch; in this case, the delay constraint must be set at 2 msec. As previously mentioned with voice communication, packet loss is tolerable as long as the loss rate L is limited to a small value. In this study, we shall assume that L=0.02 is adequate.

We first examine the effect of packet size B_h+B_v on the delay performance. Consider W, V and M to be fixed. Let k denote the number of packets sampled in the simulation, and let $D_1, D_2, \ldots, D_k$ be the delay incurred by the k packets respectively. Let $D_{i_1} < D_{i_2} < \ldots < D_{i_k}$ be the ordered sequence of delay samples. We let $\text{Max}\{D|L\} \triangleq D_{i_{k(1-L)}}$. $\text{Max}\{D|L\}$ is the value of packet delay which is exceeded by exactly a fraction L of all samples. We can similarly define $\text{Max}\{D_n|L\}$. Clearly, $\text{Max}\{D|L\} = T_f + \text{Max}\{D_n|L\}$. If B_v is arbitrarily large, then T_f is the predominant term. Indeed, the number of packets contending is small, and, with the packet size being large, the performance of P-CSMA (which is a function of $\tau W/B$) is relatively good. In this case, $\text{Max}\{D|L\}$ is also larger, on the same order of magnitude as $B_v/V+B/W$. As B_v decreases, T_f decreases, and so does $\text{Max}\{D|L\}$ until B_v is small enough as to cause a high degree of contention and an important increase of $\text{Max}\{D_n|L\}$. Clearly, further decrease in B_v causes the P-CSMA channel to saturate and $\text{Max}\{D_n|L\} \to \infty$ with probability one. Thus, there exists an optimum value for B_v which minimizes $\text{Max}\{D|L\}$. We illustrate these effects in Figure 1, in which we plot $\text{Max}\{D|L\}$ versus B_v for the case

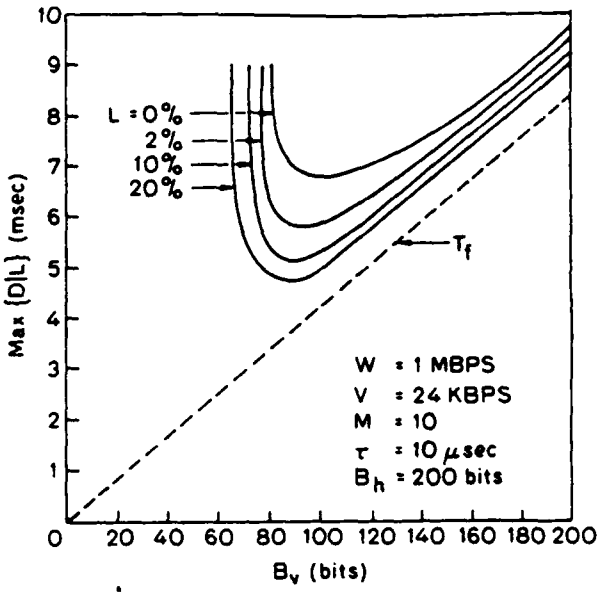


Fig. 1. $\text{Max}\{D|L\}$ versus B_v for fixed M and various values of L.

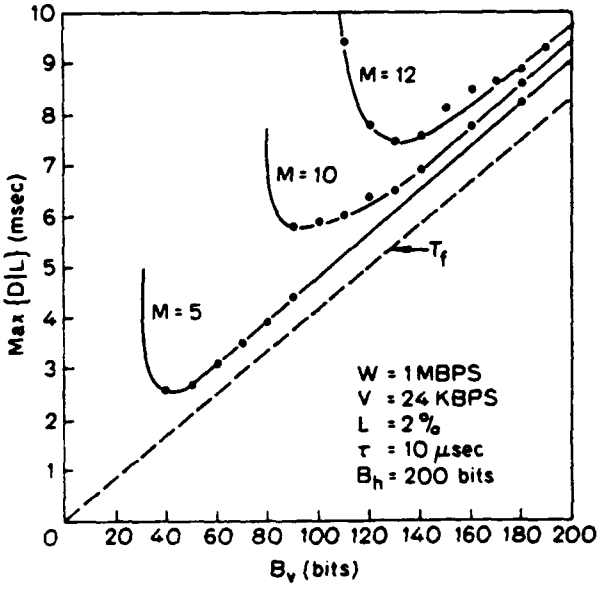


Fig. 2. $\text{Max}\{D|L\}$ versus B_v for fixed L and various values of M.

V=24KBPS, W=1MBPS, M=10 and various values of L. It is interesting to note that given M, the optimum packet size B_v is roughly the same for all values of L. Setting L=0.02, we show in Figure 2 $\text{Max}\{D|0.02\}$ versus B_v for various values of M. This figure shows that for a given value of D_m and a given value of L, there exists a unique

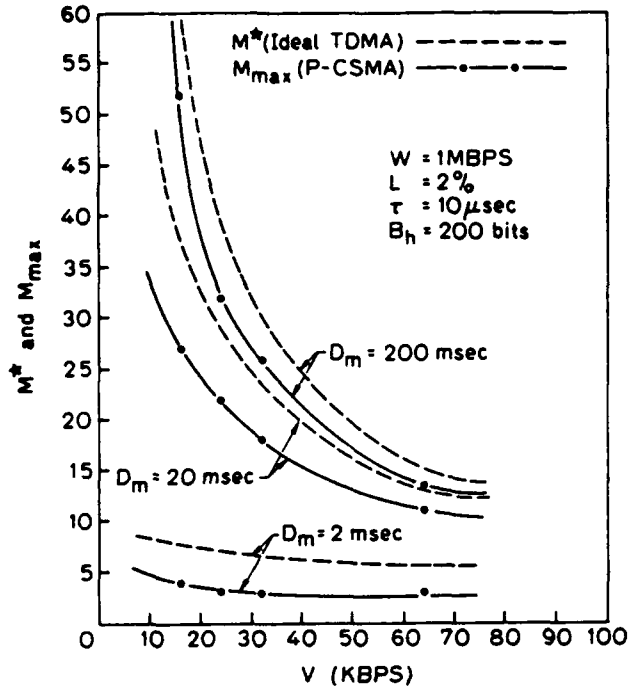


Fig. 3. M^* and M_{max} versus V for W=1MBPS

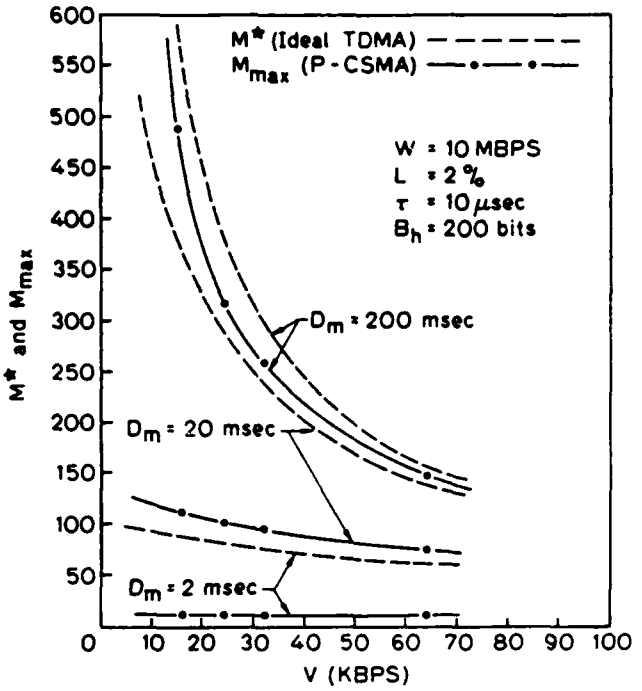


Fig. 4. M^* and M_{max} versus V for W=10MBPS

optimum value of B_v and a maximum value of M which satisfy the constraint $Max\{D|L\} \leq D_m$. This maximum value of M, M_{max}, represents the performance of P-CSMA when supporting voice communication. From Figures 1 and 2, as well as the results obtained for all other cases studied, we note that $Max\{D_n|L\}$ *at optimum* is always inferior to T_f, and therefore $Max\{D|L\} \leq 2T_f$, regardless of D_m and L. This clearly justifies that the model adopted for the vocoder's transmit buffer (single packet buffer and LCFS) is not restrictive. In Figure 3, we plot M_{max} as a function of the vocoder rate V for L=2% and W=1MBPS and the three values of D_m: 2 msec., 20 msec., and 200 msec. The dashed curves correspond to the ideal TDMA case. Figure 4 displays similar results for the case W=10MBPS. We note that when D_m=200 msec., both M^* and M_{max} decrease rapidly as V increases; while if D_m=2 msec., then M^* and M_{max} are rather insensitive to V. This is due to the existence of a nonzero overhead B_h, whose effect is more important as the delay requirement is more critical. To best compare the performance of P-CSMA to that of the ideal TDMA, we consider the ratio M_{max}/M^*. This ratio has the property of isolating the effect of contention as opposed to that of V and B_h, and therefore allows us to evaluate the relative performance of P-CSMA when supporting voice. We plot in Figures 5 and 6 M_{max}/M^* versus V for W=1MBPS and 10MBPS respectively. We note

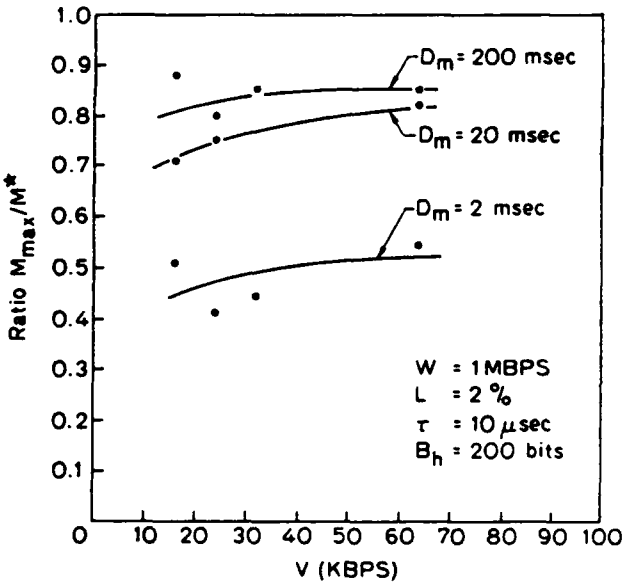


Fig. 5. Ratio M_{max}/M^* versus V for W=1MBPS

that the degradation in performance due to contention is more significant as the delay requirement is more severe and/or as the channel bandwidth is larger. Both these trends are due to the higher degree of contention caused by a larger ratio $\tau W/B$. Indeed, with smaller D_m, B_v is bound to be smaller, and therefore $\tau W/B$ is larger.

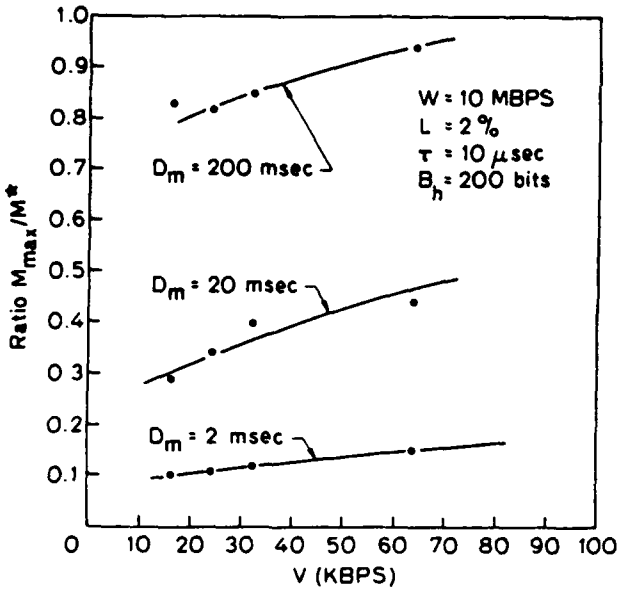


Fig. 6. Ratio M_{max}/M^* versus V for W=10MBPS

Finally, in Figure 7, we display the optimum packet size B_v as a function of V for L=0.02 and D_m=2, 20 and 200 msec.

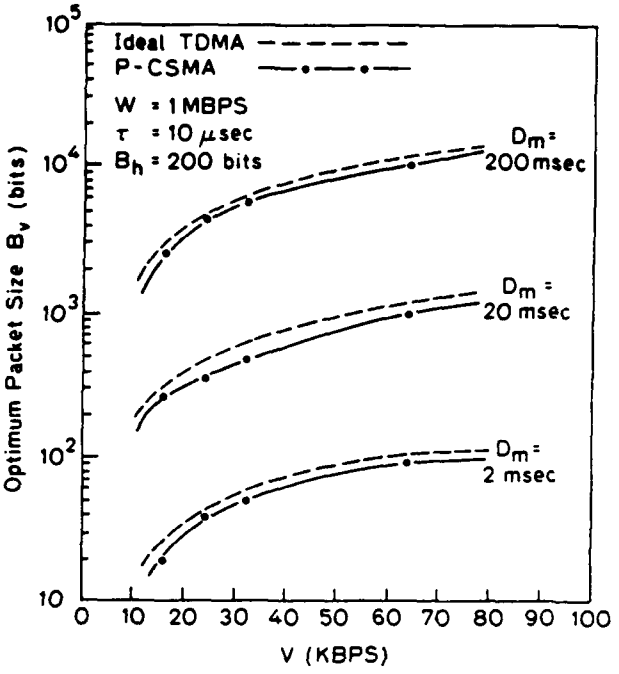


Fig. 7. Optimum B_v versus V for W=1MBPS

V. CONCLUSION

We examined in this paper the performance of CSMA-CD local networks when supporting voice communication, and compared it to an ideal TDMA system which provides the ultimate best achievable performance. The results show that, for a given delay constraint D_m and a given tolerable loss rate L, there is an optimum packet size B_v which provides a maximum number of voice sources. As long as the delay requirement D_m is not too severe (on the order of ≈200 msec.) and the channel bandwidth W is not too large (on the order of ≈1MBPS), then the performance of P-CSMA is comparable to that of the ideal TDMA. However, if either D_m is small (≤20 msec.), or W is large (≥10MBPS), or both, then P-CSMA becomes inferior to the ideal case, regardless of the vocoder rate. This is basically due to the relatively small transmission time of a packet for which P-CSMA is known to have a poor performance. As a result, we note that, when the delay requirement is low, an increase in channel bandwidth with the expectation to increase the maximum number of voice sources is rewarded by smaller than a proportional improvement. As an example, we see that, when D_m=20 msec. and V=32KBPS, M_{max} is about 20 for W=1MBPS and about 90 for W=10MBPS.

REFERENCES

[1] R.M. Metcalfe and D.R. Boggs, "ETHERNET: Distributed Packet Switching for Local Computer Networks", CACM, Vol. 19, n.7, July 1976.

[2] 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, Vol. COM-25, December 1975.

[3] F.A. Tobagi and V.B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection", Computer Networks, Vol. 4, n.5, October 1980.

[4] F.A. Tobagi, "Carrier Sense Multiple Access with Message-Based Priority Functions", IEEE Trans. on Comm., Vol. COM-30, n.1, January 1982.

APPENDIX II.

F. A. Tobagi and R. Rom, "Efficient Round-Robin and Priority Schemes for Unidirectional Broadcast Systems," *Proceedings of the IFIP-IBM Zurich Workshop on Local Area Networks*, Zurich, Switzerland, August 27-29, 1980.

F. A. Tobagi, F. Borgonovo, and L. Fratta, "EXPRESS-NET: A High-Performance Integrated-Services Local Area Network," *IEEE Journal on Selected Areas in Communications Special Issue on Local Area Networks*, November 1983.

F. A. Tobagi and M. Fine, "Performance of Unidirectional Broadcast Local Area Networks: Express-Net and Fasnet," *IEEE Journal on Selected Areas in Communications Special Issue on Local Area Networks*, November 1983.

M. Fine and F. A. Tobagi, "Scheduling-Delay Multiple Access Schemes for Broadcast Local Area Networks," in *Proceedings of the 1st African Conference on Computer Communications, AFRICOM'84*, Tunis, Tunisia, May 1984.

Zurich, Aug. 1980

EFFICIENT ROUND-ROBIN AND PRIORITY SCHEMES FOR UNIDIRECTIONAL BROADCAST SYSTEMS

Fouad A. Tobagi
Dept. of Electrical Engineering
Stanford University[1]
Stanford, California, 94305

Raphael Rom
Telecommunications Science Center
SRI International
Menlo Park, California, 94025

---

[1]Dr. Tobagi is a consultant to SRI International, Menlo Park, CA.

August 1980

ABSTRACT

Local area communication networks based on the packet broadcasting technology have received considerable attention in recent years, due to their simple architectures and efficient operation. One of the major concerns in designing such networks, however, has been the design of efficient multiaccess protocols.

We destinguish in this paper two main architectures for cable broadcast systems: the bidirectional broadcast system (BBS) architecture in which transmission is broadcast in both directions of the cable, and the unidirectional broadcast system (UBS) architecture in which transmission is forced into only one direction. Broadcast communication in the UBS architectures is achieved by folding the cable so that each device is connected to both the outbound and the inbound portions of the cable.

Following a brief discussion of multiaccess protocols and priority functions in distributed multiaccess environments we describe here a new and efficient round robin scheduling scheme suitable for UBS architectures, describe a simple and efficient mechanism for priority assessment in both BBS and UBS architectures, and then extend the applicability of the round robin scheme to a prioritized unidirectional environment.

I. INTRODUCTION

Significant advances in local area communication networks have been achieved in the past few years. Several architectures have been proposed and implemented [1, 2, 3, 4]. They have a particular element in common: they are all based on packet broadcasting technology.

Packet broadcasting is attractive in that it combines the advantages of both packet switching and broadcast communication. Packet switching offers the efficient sharing of communication resources by many contending users with random demands. Broadcast communication eliminates complex topological design and routing problems. Packet broadcasting is simply achieved by providing available communication bandwidth as a single high-speed channel to be shared by the many contending users.

A. Bidirectional and Unidirectional Broadcast Systems

Local communication networks of the Ethernet type use a cable to which all the communicating devices are connected. The device connection interface is a passive cable tap so that failure of an interface does not prevent the remaining devices from communicating. The use of a single coaxial cable with bidirectional transmission naturally achieves broadcast communication. We refer to such systems as bidirectional broadcast systems (BBS).

Broadcast communication with unidirectional transmission can also be achieved by folding the cable so that each device is connected to both the forward (or outbound) and reverse (or inbound) portions of the cable. Both taps are passive, and, in the simplest case, the tap on the forward portion of the cable is a write-only tap while the tap on the reverse portion of the cable is a read-only tap. This configuration results in an inherent physical ordering of the subscribers, a feature we take advantage of in the sequel. We refer to such systems as unidirectional broadcast systems (UBS). Figure 1 displays schematically the BBS and UBS architectures.

B. Multiaccess Protocols

One of the major concerns of designing such networks has been the design of efficient multiaccess protocols. The difficulty in controlling access to the channel by users who can only communicate via the channel itself has given rise to what is known as random-access techniques. In the ALOHA random access scheme, users transmit any time they desire [5]. When conflicts occur, the conflicting users schedule retransmission of their packets to some later time. In carrier-sense multiple access (CSMA), the risk of a collision is decreased by having users sense the channel prior to transmission [6]. If the channel is sensed busy, then transmission is inhibited.

Many CSMA protocols exist which differ in the action taken by a ready subscriber who finds the channel busy. In the nonpersistent CSMA, the terminal simply schedules the retransmission of the packet to some later time. In the 1-persistent CSMA, the terminal monitors the channel, waits until the channel goes idle (persisting on transmitting), and then transmits the packet with probability one. In the p-persistent CSMA, the terminal monitors the channel as in 1-persistent, but when the channel goes idle, it transmits the packet only with probability p, and with probability 1-p it waits the maximum propagation delay and then repeats the process provided that the channel is still sensed idle.

Given the physical characteristics of data transmission on coaxial cables, in addition to sensing carrier it is possible for transceivers to detect interference among several transmissions (including their own) and abort the transmission of their colliding packets. This produces a variation of CSMA which we refer to as carrier-sense multiple access with collision detection (CSMA-CD) [4, 7].

While random access techniques such as CSMA are suitable for both BBS and UBS architecutres a very efficient round-robin scheme is achievable in the UBS architecture as a result of the physical ordering of the subscribers on the cable. This scheme is described below in section II.

C. Priority Functions In Multiaccess Broadcast Systems

The need for priority functions in multiaccess environments is clear. Having multiplexed traffic from several users and different applications on the same bandwidth-limited channel in order to achieve a higher utilization of the latter, we require that a multiaccess scheme be responsive to the particular specifications of each. Priority functions in multiaccess environments are also addressed in this paper.

Little work has been done in the past to incorporate priority functions into multiaccess protocols, the distributed nature of the environment being the major obstacle. Priority functions here are viewed in their most general sense; that is, priority is defined as a function of the message to be transmitted and not just the device transmitting the message. The target requirements for a priority scheme to be acceptable are:

1. Hierarchical independence of performance: The performance of the scheme as seen by messages of a given priority class should be insensitive to the load exercised on the channel by lower priority classes. Increasing loads from lower classes should not degrade the performance of higher priority classes.

2. Fairness within each priority class: Several messages of the same priority class may be simultaneously present in the system. These should be able to contend on the communication bandwidth with equal right.

3. Robustness: A priority scheme must be robust in that its proper operation and performance should be insensitive to eventual errors in status information.

4. Low overhead: The volume of control information to be exchanged among the contending users, as required by the scheme, and the overhead required to implement the priority scheme must both be kept minimal.

To satisfy property (1) above, a priority scheme must be based on the principle that access right "at any instant" be exclusively given to ready messages of the highest current priority level. While this is easily achieved in nondistributed environements, in a distributed environment, there are three basic problems that need to be addressed:

(1) Identify the instants at which to assess the highest current priority class with ready messages; (2) Design a mechanism by which to assess the highest non-empty priority class; and (3) Design a mechanism which assigns the channel to the various ready users within a class. True hierarchical independence of performance can be achieved only with a preemptive priority scheme and even then not to a full extent for in a strict sense the overhead incurred by the preemption might be considered degradation of performance. Nonetheless, property (1) should be considered a prime target for all priority schemes.

An efficient mechanism for priority assessment, suitable for UBS and BBS architectures with a carrier sense capability has been proposed in [8] along with CSMA contention schemes. We review this mechanism in section III below, and describe in section IV an efficient prioritized round robin scheme for UBS architectures which makes use of it. The content of this paper consist of extracts from a more complete version [9].

II. AN EFFICIENT ROUND ROBIN ALGORITHM

The UBS considered here has two separate channels--the outbound channel which all subscribers access in order to transmit, and the inbound channel which subscribers access in order to read the transmitted information. In addition to transmitting capability on the outbound channel we assume that subscribers can also sense activity on that channel in a way similar to that required in other channel sensing systems such as CSMA. In a UBS this capability results in an interesting feature. Assume subscribers are numbered sequentially S1, S2, S3, etc., and subscriber S1 is defined as the "farthest", i.e., has the longest round trip delay (see figure 1). Because of the unidirectional signalling property, S2 is able to sense signals generated by S1 on both the inbound and outbound channels whereas the opposite does not hold; that is, S1 can sense signals generated by S2 only on the inbound channel. This asymmetry will be utilized in establishing the ordering in a round-robin scheme.

A subscriber is considered to be in one of three states. A subscriber is in the IDLE state if it does not have any message awaiting transmission. A non-IDLE subscriber, called a ready subscriber, can assume one of two states--ACTIVE if it has not transmitted its message in the "current round" or DORMANT if it did transmit and is waiting for the completion of the round. In order to achieve fair scheduling, DORMANT subscribers defer to all other ACTIVE subscribers. Consequently, we are assured that no subscriber will transmit its second message before other ready subscribers have a chance to transmit their first ones. Eventually all ready subscribers will have transmitted their messages (i.e. all are DORMANT); this constitutes the end of a round; at this time all reset their state and a new round starts.

While each subscriber distinguishes between its DORMANT and ACTIVE states (with a 1-bit flag), arbitration among active subscribers must be provided by additional means. To that end each ACTIVE subscriber transmits a short burst of unmodulated carrier after the end of the previous message to indicate its ACTIVEness and, at the same time,

senses the outbound channel. All but one ACTIVE subscriber will sense the outbound channel busy (due to transmission from lower indexed subscribers--see Fig. 1) thus singling out the next subscriber to transmit. Here we make use of the asymmetry of the outbound channel as explained earlier. If a given subscriber senses the outbound channel busy, there exists at least one ready subscriber "ahead" of it that generated that signal; a subscriber will always defer its transmission in favor of those "ahead" of it.

Initially all subscribers reset their state, meaning that all ready subscribers are ACTIVE. An ACTIVE subscriber will operate follows:

1. Wait until the next end-of-carrier (EOC) at the end of a message, detected on the inbound channel.

2. Transmit a short burst of unmodulated carrier and listen to the outbound channel for one round trip delay.

3. If the outbound channel is sensed idle, the subscriber transmits its message (free of conflict) and moves to the DORMANT state. Otherwise, the subscriber repeats the algorithm.

A DORMANT subscriber will become ACTIVE, if the inbound channel is sensed idle for one round trip delay or longer and then perform the above steps.

The algorithm is extremely efficient because a nonconflicting schedule is usually achieved within one half round-trip delay regardless of the position of the consecutive ready users on the bus, where a full round-trip delay is considered to be the propagation time through both the outbound and inbound channels. Since ordering is implied by information extracted from the outbound channel alone, one half round trip delay (through the outbound channel only) is sufficient to establish scheduling. An extra (idle) round-trip delay is required to signal the end of a round to all subscribers (which is nominal overhead especially in a loaded system).

The algorithm presented here differs slightly from a conventional round-robin algorithm. In a conventional round-robin discipline, each subscriber, in a prescribed order, is given a chance to transmit; it

transmits if it has a message ready and declines if it has not. This subscriber will not be given a second chance before all other subscribers have had their chance. In our algorithm, while no subscriber transmits more than once within each "round", the order of transmission within the round may vary depending on the instants at which messages arrive. For example, assume that subscriber S1 just completed transmission of its message. Assume also that at this time S2 does not have a message ready and therefore S3, assumed ready, transmits next. While S3 is transmitting, a message arrives at S2, and consequently when S3 is finished S2 transmits next. The order of transmission in this case was S1, S3, S2 while if all had a ready message at the beginning of the round the order would have been S1, S2, S3.

III. A PRIORITY ASSESSMENT MECHANISM FOR UBS AND BBS ARCHITECTURES

We review innthis section the priority assessent mecanism proposed in [8]. With the broadcast nature of transmission, users can monitor the activity on the channel at all times. The assessment of the highest priority class with ready messages is done, at least (as is the case in the nonpreemptive discipline), at the end of each transmission period, whether successful or not, i.e., every time the carrier on the channel goes idle. When detected at a subscriber, end of carrier (EOC) establishes a time reference for that subscriber. Following EOC, the channel time is considered to be slotted with the slot size equal to 2*TAU+GAMMA, where TAU denotes the maximum one-way propagation delay between pairs of subscribers, and GAMMA is a period of time sufficiently long to detect a short burst of unmodulated carrier. Within each subscriber, messages are ordered according to their priority. The priority of a subscriber at any time is the highest priority class with messages present in its queue.

Let us denote an arbitrary subscriber, and EOC(s) denote the time of end of carrier at subscriber s. Let p(s) denote the priority level of subscriber s at time EOC(s). The priority assessment algorithm consists of having subscriber s operate as follows:

1. If, following EOC(s), carrier is detected in slot i, with i<p(s), (thus meaning that some subscriber has priority i higher than p(s) and access right must be granted to class i), then subscriber s awaits the following end of carrier (at the end of the next transmission period) at which time it reevaluates its priority and repeats the algorithm.

2. If no carrier is detected prior to the j-th slot, where j=p(s), then subscriber s transmits a short burst of unmodulated carrier at the beginning of slot j (thus reserving channel access to priority class p(s) and, immediately following this slot, operates according to the contention resolution algorithm decided upon within class p(s) (such as p-persistent CSMA, for example). At the next end of carrier, subscriber s reevaluates its priority level and repeats the algorithm.

Thus, by the means of short burst reservations following EOC, the

highest nonempty priority class is granted exclusive access right, and messages within that class can access the channel according to any contention algorithm. If the contention algorithm is CSMA, then we refer to the scheme as prioritized CSMA (P-CSMA).

Note that the above algorithm corresponds to a nonpreemptive discipline, since a subscriber which has been denied access does not reevaluate its priority until the next end of carrier. However, note that, by assessing the highest priority level at the end of each transmission period whether successful or not, the scheme allows higher priority messages to regain the access right without incurring substantial delays.

The scheme is robust since no precise information regarding the demand placed on the channel is exchanged among the users. Information regarding the existing classes of priority is implied from the position of the burst of unmodulated carrier following EOC. Note also that there is no need to synchronize all users to a universal time reference. By choosing the slot size to be 2*TAU+GAMMA we guarantee that a burst emitted by some subscriber in its k-th slot is received within the k-th slot of all other subscribers.

We illustrate this procedure in Fig. 2 by displaying a snapshot of the activity on the channel for p-persistent P-CSMA. (For simplicity and without loss of generality, we consider that there are only two possible priority levels in the system.) We denote by n1 and n2 the number of active subscribers in class 1 (C1) and class 2 (C2), respectively. We adopt the convention that C1 has priority over C2. We also show a reservation burst as occupying the entire slot in which it is transmitted. Finally we represent by a vertical upward arrow the arrival of a new message to the system; the label C1 or C2 indicates the priority class to which the message belongs.) We assume in Fig. 2 that, at the first EOC, we have n1=0 and n2>0. Following EOC a reservation burst is transmitted in the second slot. The priority resolution period, also called priority assessment period (PAP), is, in this case, equal to two slots. Following the reservation, we observed a channel-access

period (CAP) which consists of the idle time until the channel is accessed by some user(s) in class 2. Clearly CAP is a function of the channel access procedure employed by class 2. Following CAP we observe the transmission period (TP) itself, the end of which establishes the new EOC time reference. (A crosshatched TP signifies a collision.) The time period between a reservation and the following EOC, called the contention period and equal to CAP+TP, is the time period during which exclusive access right is given to the class which succeeded in reserving the channel. In this nonpreemptive case, message arrival C1, although of higher priority, is not granted access right until the EOC following its arrival, at which time it reserves the channel.

Since the priority assessment period is of nonzero length, one may envision each subscriber continuously updating its priority during the priority resolution period. Clearly, unless we allow messages to change priority levels, the priority of a subscriber may only change at the generation times of new messages. As a result, given that we are in slot k of the reservation period, indicating that no priority class higher than k reserved access to the channel, a subscriber may still make reservation for its most current priority as long as this priority is lower than priority class k, and no reservation burst is detected before its corresponding slot. However, if following EOC no reservation burst is detected for K consecutive slots, where K is the total number of priority classes available in the system, then the channel becomes free to be accessed by any subscriber regardless of its priority, until a new EOC is detected.

A variant of this nonpreemptive P-CSMA algorithm is to require that each subscriber record, at the end of the priority assessment period, the priority level that is granted access (say i), so that i-level messages generated during the period of time when access right is granted to level i may also contend on the channel.

Note that the overhead incurred in a reservation period following EOC is a function of the currently highest priority level. The higher this class is, the smaller the overhead is and the smaller is the delay

to gain access.

Different contention algorithms may be used by different classes of priority. Considerations regarding the use of 1-persistent CSMA versus p-persistent CSMA have been made. The reader is referred to [8, 9] for details.

The prioritized CSMA can accommodate preemption fairly simply. Consider that, after the reservation process has taken place, the channel has been assigned to class j. Assume that, before a transmission takes place, a message of level i, i<j, is generated at some subscriber s. The nonpreemptive scheme dictates that subscriber s await the next time reference before it can ascertain its (higher) level i. The semipreemptive scheme allows subscriber s to preempt access right to class j, as long as no transmission from class j has yet taken place, by simply transmitting the message. If the generation of the message of level i takes place after a transmission period is initiated, then subscriber s waits until end of carrier is detected. Both nonpreemptive and semipreemptive schemes are applicable whether collision detection is in effect or not.

In a fully preemptive P-CSMA scheme, a subscriber with a newly generated packet may also preempt an on-going transmission of a lower priority level by intentionally causing a collision. Clearly this scheme is only appropriate if collision detection is in effect! It can offer some benefit if lower priority classes have long messages. One may also envision an adaptive preemption scheme whereby an on-going transmission is preemptive only if the already elapsed transmission time is short.

IV. A PRIORITIZED ROUND ROBIN SCHEME FOR UBS ARCHITECTURE

In this section we adapt the round-robin algorithm described in section II to a prioritized environment. Hence we must modify the algorithm to ensure that fairness is administered only within each priority class and that high priority messages are transmitted first. To achieve this, contention among subscribers is resolved in two stages. First, ready subscribers exchange information regarding the priority of their current message (i.e., undertake priority class assessment) and then the round-robin algorithm described previously is used to resolve contention among subscribers of the currently highest priority.

Here, again, we distinguish between ACTIVE and DORMANT subscribers depending on whether they did or did not transmit a message in the current round. However, to achieve fair scheduling within each priority class, subscribers maintain separate such states for each priority class; i.e., a subscriber can be ACTIVE with respect to one priority class and DORMANT with respect to another. Since only these two states (for each priority class) must be distinguished the total memory required is one bit per priority class.

All ready subscribers, ACTIVE or DORMANT, participate in the priority class assessment. A mechanism similar to the one described in Section III may be used. When the priority assessment period is over the currently highest priority class is established (independent of the internal state of the ready subscriber(s) holding these messages); the channel is then considered as operating at this priority level. Let p(channel) denote the latter. Those subscribers for which p(s)#p(channel) refrain from proceeding and those for which these priorities are equal operate according to the round-robin algorithm described previously.

A ready subscriber will therefore wait until the next end of message and operate according to the following:

1. Participate in the priority class assessment (at which p(channel) is established).

2. If p(s)≠p(channel) wait until the next EOC and then repeat the algorithm

3. If p(s)=p(channel) and the subscriber is ACTIVE with respect to this priority class then

    a. It transmits a short burst and listens to the outbound channel for one round trip delay.

    b. If the outbound channel is sensed idle the subscriber transmits its message and moves to the DORMANT state (with respect to this priority class). Otherwise, the subscriber repeats the algorithm.

4. A DORMANT subscriber for which p(s)=p(channel) senses the inbound channel for one round trip delay and, if sensed busy, it repeats the algorithm, otherwise it becomes ACTIVE with respect to p(channel) and performs step (3) above.

The scheme presented here is nonpreemptive. A semipreemptive scheme, in which a subscriber of high priority interferes between the end of priority class assessment period and the actual transmission is not meaningful since the relevant time window is only one half round-trip delay long which is too short for any preemptive activity. A full preemption scheme can be introduced by allowing a subscriber to jam the channel and force a new priority class-resolution-period.

V. CONCLUSION

We examined in this paper two local area communication system architectures: the bidirectional broadcast system architecture and the unidirectional broadcast system architecture. We briefly discussed the two important issues: (i) multiaccess protocols and (ii) priority functions in distributed multiaccess environments. The inherent ordering of subscribers due to unidirectionality in transmission in the UBS architecture leads to an efficient, robust, fair, and conflict free round robin scheme. In this round robin scheme, the time needed to switch access control from one active user to the other is an end-to-end propagation delay. Moreover, a simple priority assessment mechanism has been described which permits to implement in the UBS architecture a prioritized round robin scheme. Provided that each subscriber keeps one bit of information per priority-class representing its state in relation to that priority class, the conflict-free round robin scheduling scheme is achieved for each priority class independently.

REFERENCES

1. D. G. Willard, "Mitrix: A sophisticated Digital Cable Communications System,," in Proc. of the National Telecommunications Conference (IEEE, November 1977).
2. A. G. Fraser, "A Virtual Channel Network," Datamation, Vol. 21, No. 2, pp. 51-56 (February 1975).
3. D.J. Farber et al, "The Distributed Computing System," in Proceedings of the 7th Annual IEEE Computer Society International Conferene (IEEE, February 1973).
4. R.M. Metcalfe, D.R.Boggs, "ETHERNET: Distributed Packet Switching for Local Computer Networks," Communications ACM, Vol. 19, No. 7, pp. 395-403 (1976).
5. N. Abramson, "The ALOHA System - Another Alternative for Computer Communications," in Procceddings of the Fall Joint Computer Conference, pp. 281-285 (AFIPS, 1970).
6. 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. on Communications, Vol. COM-23, No. 12, pp. 1400-1416 (December 1975).
7. F.A. Tobagi and V.B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," in Proceedings of the Local Area Communications Networks Symposium, pp. 217-244 (The MITRE Corp., May 1979).
8. F.A. Tobagi, "Message Based Priority Functions in Multiaccess/Broadcast Communication Systems with Carrier Sense Capability," in Proc. of the Pacfic Telecommunicatons Conference, pp. 1A-33 - 1A-43 (IEEE, January 1980) Also available as Stanford University Electronic Laboratory Technical Report #187, October 1979.
9. R. Rom, and F.A. Tobagi, "Message Based Priority Functons in Local Multiaccess Communication Systems," SRINET Technical Report #5, SRI International (March 1980) [Submitted for publication in Computer Networks].

Expressnet: A High-Performance Integrated-Services Local Area Network

FOUAD A. TOBAGI, SENIOR MEMBER, IEEE, FLAMINIO BORGONOVO, AND LUIGI FRATTA, MEMBER, IEEE

Abstract — Expressnet is a local area communication network comprising an inbound channel and an outbound channel to which the stations are connected. Stations transmit on the outbound channel and receive on the inbound channel. The inbound channel is connected to the outbound channel so that all signals transmitted on the outbound channel are duplicated on the inbound channel, thus achieving broadcast communication among the stations. In order to transmit on the bus, the stations utilize a distributed access protocol which achieves a conflict-free round-robin scheduling. This protocol is more efficient than existing round-robin schemes as the time required to switch control from one active user to the next in a round is minimized (on the order of a carrier detection time), and is independent of the end-to-end network propagation delay. This improvement is particularly significant when the channel data rate is so high, or the end-to-end propagation delay is so large, or the packet size is so small as to render the end-to-end propagation delay a significant fraction of, or larger than, the transmission time of a packet. Moreover, some features of Expressnet make it particularly suitable for voice applications. In view of integrating voice and data, a simple access protocol is described which meets the bandwidth requirement and maximum packet delay constraint for voice communication at all times, while guaranteeing a minimum bandwidth requirement for data traffic. Finally, it is noted that the voice/data access protocol constitutes a highly adaptive allocation scheme of channel bandwidth, which allows data users to recover the bandwidth unused by the voice application. It can be easily extended to accommodate any number of applications, each with its specific requirements.

I. Introduction

A great deal of discussion can be seen in the recent literature regarding local networks and their applicability to many of today's local area communications needs. These needs have primarily consisted of data communication applications such as computer-to-computer data traffic, terminal-to-computer data traffic, and the like. More recently, a new line of thought has been apparent. It is the desire to integrate voice communication on local data networks. The reason for this is threefold: 1) voice is an office communication application just as computer data, facsimile, etc.; 2) recent advances in vocoder technology have shown that digitized speech constitutes a digital communication application which is within the capabilities of local area data networks; and 3) today's local network architectures offer very elegant solutions to the local communications problem, from both the viewpoint of *simplicity* in topology and device interconnection, and the viewpoint of *flexibility* in satisfying growth and variability in the environment. In addition to voice, one may envision a number of other applications in the office environment of the future which will require much higher bandwidth than what is offered in today's systems. These include high resolution graphics and video.

The local networks that are available today differ in many aspects: the topology, the transmission medium, the signaling scheme, the encoding scheme, and the multiaccess methods. The pros and cons of many alternatives have been debated at length at various occasions (such as, for example, at the IEEE Local Network Standardization Committee meetings). The throughput–delay performance of many of these systems has also been analyzed. A simple comparison of the performance of these systems has shown that rings with a token-passing access scheme outperform all other solutions (see, for example, [1]). No attempt is being made in this paper to address the issue of comparing all these solutions from their engineering aspects or performance. Instead we restrict ourselves to those schemes referred to as broadcast bus systems with a passive transmission medium (i.e., systems in which the medium has no active electronics).

While the broadcast bus networks available today constitute elegant solutions to the local networking problem, they are not without their limitations in performance. Some of these limitations arise as the characteristics of the environment and data traffic requirements being supported by these solutions deviate from those assumed in the original design. Examples of such characteristics are: packet length distribution, packet generation pattern, channel data

Manuscript received December 7, 1981; revised July 15, 1983. This work was supported in part by the U.S. Defense Advanced Research Projects Agency under Contract MDA 903-79-C-0201, Order A03717, monitored by the Office of Naval Research, and by the Italian Council for National Research under Contract C-NET 104520.97.8007745. This paper was presented at the International Conference on Performance of Data Communication Systems and Their Applications, Paris, France, September 14–16, 1981.

F. A. Tobagi is with the Computer Systems Laboratory, Department of Electrical Engineering and Computer Science, Stanford University, Stanford, CA 94305.

F. Borgonovo and L. Fratta are with the Dipartimento di Elettronica, Politecnico di Milano, 20133 Milano, Italy.

0733-8716/83/1100-0898$01.00

rate, delay requirements, geographical area to be spanned, etc.

In Section II, we briefly review existing broadcast bus systems and their underlying access protocols. We examine the source of limitations in each and characterize these limitations quantitatively. We then present in Section III a new proposal, the Expressnet, as an alternative. (See also [20].) It is a broadcast bus system with a passive transmission medium. Its access protocol is completely distributed and achieves conflict-free round-robin scheduling. In Section IV the performance of Expressnet is examined. It is shown to be more efficient than existing schemes, overcomes their limitations, and provides a performance similar to that seen with token-passing rings. The details for the analysis of delay are not given in this paper but can be found in a companion paper [23]. In Section V, we address the issue of voice communication. We show that Expressnet is particularly suitable for the integration of voice and data applications. We describe a simple voice/data access protocol which meets the bandwidth requirement and maximum delay constraint for voice communications at all times, while guaranteeing a minimum bandwidth requirement for data traffic. It is noted here that all engineering aspects of Expressnet are not addressed in detail in this paper. The focus is rather on the salient features of Expressnet, its topology and its access protocol.

Finally, we note the existence of two other systems which have recently been proposed, which address the same objectives as Expressnet, and which bear great resemblance to it, namely, Fasnet [2] and BID [3].

II. Broadcast Bus Systems (BBS)

In BBS all devices share a single communication medium, typically a coaxial cable, to which they are connected via passive taps. The devices transmit their packets on the bus according to some common distributed access protocol. It has been quite apparent that the throughput–delay performance of these networks is mostly determined by the access protocol used. These protocols fall basically into two categories, those which are of the contention type and those which achieve conflict-free scheduling [4].

The most prominent example of contention systems is Ethernet [5], [6]. It uses carrier sense multiple access with collision detection (CSMA-CD). The fundamentals of CSMA-CD are well known and its performance has been extensively analyzed [7]–[9]. The performance is a function of the ratio

$$a \triangleq \frac{\tau W}{B} \tag{1}$$

where τ denotes the end-to-end propagation delay between the two extreme users connected to the bus, W denotes the channel bandwidth in bits per second, and B is the number of bits per packet. Assuming for example an infinite population model in which users become ready to transmit according to a Poisson process, it can be shown that the channel capacity of CSMA-CD is given by[1] (see the Appendix for more details)

$$C(\infty, a) = \begin{cases} \dfrac{1}{1+Ha} & a \leqslant 0.5 \\ \dfrac{1}{(2+H)a} & a > 0.5 \end{cases} \tag{2}$$

where H is a constant (in the neighborhood of 3–6) which depends on the particular version of the protocol. The performance of CSMA-CD can also be evaluated by considering a finite population of M users, among which an assumed constant number N is always ready to transmit. In this case the channel utilization is independent of M and is given by (see the Appendix for more details)

$$C(M, N, a) = \begin{cases} \dfrac{1}{1+F(N)a} & a \leqslant 0.5 \\ \dfrac{1}{[2+F(N)]a} & a > 0.5 \end{cases} \tag{3}$$

where again the function $F(N)$ depends on the particular version of CSMA-CD. For the slotted p-persistent CSMA-CD considered here and in the Appendix, for example, we have $H = 5.145$ and

$$F(N) = \min_{0 \leqslant p \leqslant 1} \left\{ \frac{4 - 3(1-p)^N}{Np(1-p)^{N-1}} - 2 \right\}. \tag{4}$$

Note that for the same CSMA-CD version $\lim_{N \to \infty} F(N) = H$ and, as expected, (3) reduces to (2) as $N \to \infty$. In Fig. 1 the channel utilization for CSMA-CD is plotted versus a for various values of N. The utilization is rather insensitive to N and decreases with increasing values of a. The channel capacity attained under heavy load is obtained by setting $N = M$ in (3).

Conflict-free access to the bus can be obtained by means of carrier sensing. BRAM, MSRR, and MSAP are early examples [10], [11]. They provide round-robin scheduling based on the ability to sense the end-of-carrier due to a transmission and to acquire knowledge of the identity of the transmitting device. In BRAM, for example, given that n_1 is the identity of the node which just completed transmission, the next node to transmit is node n_2 such that $H(n_1, n_2)$ is the smallest, where

[1]The results are derived from a worst case analysis in which the propagation delay between any two users is always considered equal to τ, its maximum value. It is possible to predict better performance if one took into account the fact that the propagation delay is a function of the transmitting users; the result would become dependent on the geographical distribution of devices and the source-destination traffic pattern, and would be rather difficult to evaluate. In order to study the limitations of the scheme, and in an attempt to keep the results as general as possible, a worst case analysis is considered. (However, note that the capacity of CSMA-CD for $a > 0.5$ is upper bounded by $1/2a$, and this bound still shows severe degradation as a gets large.) On the other hand, detection time is considered negligible, and the preamble is considered part of the packet transmission time. If one is to compare these results to the performance of a synchronous system where preambles are not required, then these results must be discounted by the fraction of a packet occupied by the preamble.

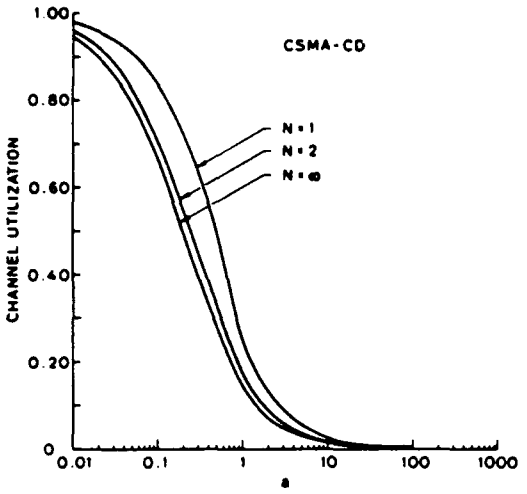


Fig. 1. Channel utilization versus a for CSMA-CD.

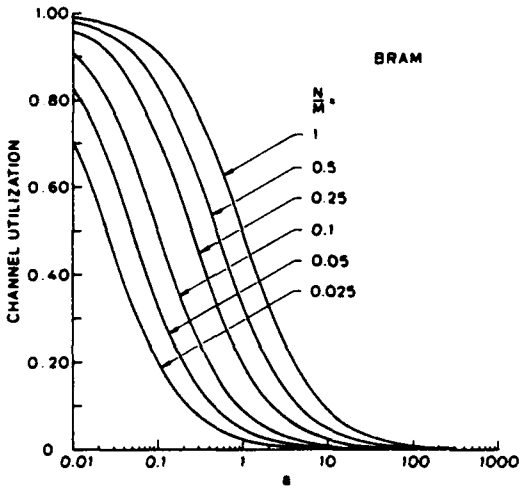


Fig. 2. Channel utilization versus a for BRAM.

$$H(n_1, n_2) = \begin{cases} (n_1 - n_2 + M) \bmod M & \text{if } n_1 \neq n_2 \\ M & \text{if } n_1 = n_2. \end{cases} \tag{5}$$

This transmission takes place $\min_{n_2} H(n_1, n_2)\tau$ seconds following the end of the previous transmission. At the next end of carrier the procedure is repeated. Given M users in total among which a constant number N is always ready to transmit, and neglecting detection time and processing delays, the channel utilization from BRAM is given by[2]

$$C(M, N, a) = \frac{1}{1 + \frac{M}{N}a}. \tag{6}$$

Note that, contrary to CSMA-CD, the channel utilization in BRAM is function of the total population size M. Accordingly, BRAM is rather inefficient if M is large and the number of busy users is small. In Fig. 2, the channel utilization for BRAM is plotted versus a for various values of the ratio N/M. The channel capacity attained under heavy load is $1/(1+a)$.

[2]Note that we considered here, as in [10], that a slot size equal to τ seconds is sufficient to accomplish the scheduling task; hence (6). In reality, however, we note that it takes anywhere from 0 to 2τ seconds in order to 1) detect end-of-carrier and 2) guarantee that all other users hear the presence or absence of carrier due to the next user in line, depending on the geographical locations of the users and their logical ordering. Thus, (6) is not the most pessimistic expression one may derive for the capacity of BRAM; a lower bound is obtained by replacing a in (6) by $2a$.

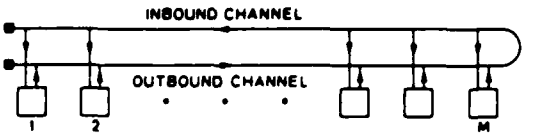


Fig. 3. Topology of UBS-RR.

A recent type of network which achieves conflict-free round-robin scheduling, yet overcomes the dependence on M is the so-called unidirectional broadcast system (UBS) type, which uses a unidirectional transmission medium [12], [13]. Broadcast communication is then achieved by various means, such as folding the cable or repeating all signals on a separate channel (or frequency) in the reverse direction, so that signals transmitted by any user reach all users on the reverse path. Thus, the system may be considered as consisting of two channels: the outbound channel which all users access in order to transmit, and the inbound channel which users access in order to read the transmitted information. The propagation delays between the two extreme users on the outbound and inbound channels are the same and are denoted by τ seconds. Since the access scheme in Expressnet is an improved version of UBS's round-robin scheme described in [12], we give here a brief description of the latter. The topology is shown in Fig. 3. The (round-robin) access scheme works as follows. A user who has a message to transmit is said to be backlogged. Otherwise it is said to be idle. In addition, with respect to a given round, any user may be either ACTIVE if it has not yet transmitted in the current round or DORMANT if it has. A user who is idle or DORMANT does not contend for the channel. An ACTIVE backlogged user contends for the channel as follows.

1) Wait for the next end-of-carrier on the inbound channel (EOC(IN)) due to a message transmission.

2) Transmit a short burst of unmodulated carrier and listen to the outbound channel for a period of time equal to the time that it takes for EOC(IN) to propagate to the end of the inbound channel and then for a possible reservation burst from the beginning of the outbound channel to propagate to this user.

3) If the outbound channel is sensed idle during this entire period then transmit the packet and go to the DORMANT state. Otherwise repeat the algorithm.

It is clear that according to this algorithm only one user transmits at a time conflict-free. Since DORMANT users do not contend for the channel, we are assured that no user will transmit more than one message in a round and, thus, fair scheduling is attained. Looking at the activity on the inbound channel, one will observe that the time separating two consecutive packets in the same round is one round-trip delay (2τ seconds), assuming the delay between the transmit and receive tap of the most downstream user is equal to zero. When the inbound channel is observed idle for longer than this time, then it means that all users are either idle or DORMANT. This indicates the end of a round, at which time all DORMANT users set their state to AC-

TIVE and contend for the channel starting at step 2 in the above algorithm. Since the channel remains idle for one round-trip delay before DORMANT users reset their state to ACTIVE, a total idle period of two round-trip delays between two consecutive rounds will result. Given that N users out of M are always ready to transmit, the channel utilization for the UBS-RR scheme is independent of M and is given by

$$C(M, N, a) = \frac{1}{1 + 2a + \frac{2a}{N}}. \tag{7}$$

The channel utilization for UBS-RR is plotted in Fig. 4 versus a for various values of N. As in CSMA-CD the capacity is not sensitive to N and decreases as a increases. While with CSMA-CD the capacity degrades as N increases, with UBS-RR the opposite is true. The round-robin algorithm just described can be achieved on a bidirectional broadcast bus by using an additional unidirectional control wire to schedule transmissions [14]. The performance of such a system is expected to be equivalent to that of UBS-RR.

Another conflict free access scheme for a bidirectional bus which uses a control wire is DSMA [15]. Users request bus allocation by transmitting their respective addresses serially (and synchronously as well as simultaneously) on the control wire. The latter acts as an OR circuit allowing the active user with the highest address to be identified and to be given access right. This user transmits on the bus as soon as the latter becomes free, i.e., following the end of carrier of the current transmission. With M users in total, the number of bits in a binary address is $\lceil \log_2(M+1) \rceil$ (the "0" address being used to indicate end of a cycle). Given the system description in [15] and ignoring the bandwidth required for the control wire, it can be easily shown that the channel utilization is given by

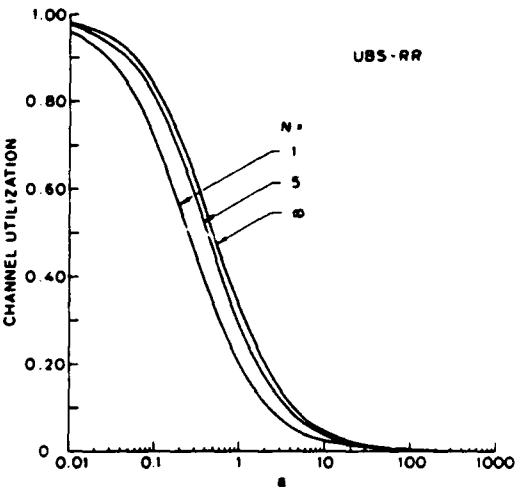


Fig. 4. Channel utilization versus a for UBS-RR.

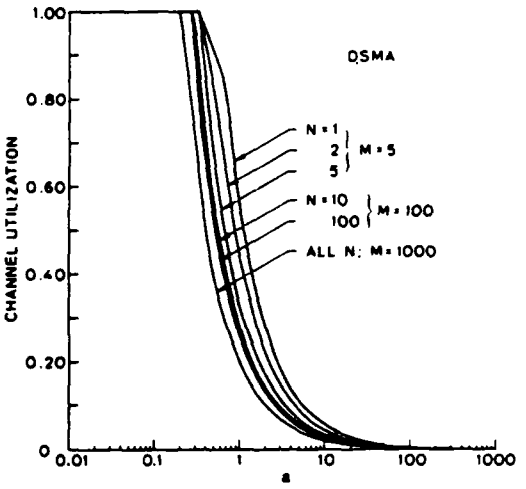


Fig. 5. Channel utilization versus a for DSMA.

$$C(M, N, a) = \begin{cases} 1 - & a \leq \dfrac{1}{\lceil \log_2(M+1) \rceil} \\ \dfrac{1}{1 + \dfrac{\lceil \log_2(M+1) \rceil a - 1}{N}} & \dfrac{1}{\lceil \log_2(M+1) \rceil} \leq a \leq \dfrac{2}{\lceil \log_2(M+1) \rceil} \\ \dfrac{2}{\left(1 + \dfrac{1}{N}\right) \lceil \log_2(M+1) \rceil a} & a \geq \dfrac{2}{\lceil \log_2(M+1) \rceil}. \end{cases} \tag{8}$$

The channel utilization for DSMA is plotted in Fig. 5 for various values of M and the ratio N/M. We note that the capacity degrades with increasing values of M However, contrary to BRAM, when M is not too small ($M = 100$ or 1000) the channel capacity is rather insensitive to the ratio N/M. When a is not too high ($a \leq 0.2$), DSMA is superior to all schemes discussed above (achieving a capacity equal to or close to 1), but as a increases, DSMA suffers similar degradation in capacity.

For the systems and applications contemplated at the present time, the parameter a does not exceed 1. Using for the various delays through interface components as given in [6], one finds that a good estimate for τ is 10 μs for each km of cable (assuming one repeater for each 500 m of cable). For 1 km cable, $W = 10$ Mbits/s and $B = 1000$, we get $a = 0.1$. For such a value, all schemes discussed above provide somewhat adequate channel capacity, 0.65 or higher. If one increases the bandwidth to 100 Mbits/s, then a increases to 1. If furthermore $B = 100$ bits, then a becomes 10, etc. Thus, given the future needs in local communications (voice, graphics, video, etc.), one sees that a number of combinations for the values of the three parameters W, B, and τ may lead to larger values of a such as 1, 10, or even 100, for which the existing schemes

comes these limitations and provide superior performance for small and large values of a.

III. The Expressnet

As with the UBS architecture, Expressnet is a broadcast communication system comprising an outbound channel and an inbound channel, a plurality of stations connected to both the outbound and inbound channels, transmitting on the outbound channel and receiving on the inbound channel. Transmitters on the outbound channel are considered to be of the unidirectional type rendering the Expressnet a UBS. (See Fig. 3.) The communication medium constituting the outbound and inbound channels may be a twisted pair, a coaxial cable, an optical fiber, or a waveguide. The channel access protocol for transmission on the outbound channel is based upon the round-robin scheme described for the UBS architecture with the advantage of utilizing the channel bandwidth more efficiently than the RR algorithm even when a is large.

The gist is basically the following. Contrary to the RR algorithm where the time reference used in determining the right of way is the end-of-carrier on the inbound channel (EOC(IN)), in the express access protocol the time reference used is the end-of-carrier on the outbound channel (EOC(OUT)). The mechanism used in determining the access right to users in a given round is thus made independent of the propagation delay τ, thus decreasing the gaps between consecutive transmissions to values on the same order as the time needed to detect carrier. Second, the idle time separating two consecutive rounds is kept as small as a round-trip propagation delay. The details of the scheme are as follows.

A. The Events EOC(IN) and EOC(OUT)

Let the Boolean function $c(t, \text{OUT})$ be defined as

$$c(t, \text{OUT}) = \begin{cases} 1 & \text{if carrier is detected present on the outbound channel at time } t \\ 0 & \text{otherwise.} \end{cases}$$

Note that $c(t, \text{OUT})$ signals the presence or absence of carrier with a delay of t_d seconds, where t_d is the time required for the detection operation. It is assumed here that the carrier detector is placed very close to the channel. (Other arrangements are also possible as set forth below.) The event EOC(OUT) is said to occur when $c(t, \text{OUT})$ undertakes a transition from 1 to 0. In a similar way $c(t, \text{IN})$ and EOC(IN) are defined.

B. Transmission Units (TU)

In an asynchronous mode of operation of Expressnet, a transmission unit consists of a preamble followed by the information packet itself. Transmission units and information packets may be of fixed or variable size. The preamble is for synchronization purposes at the receivers. It is sufficiently long for the receivers to detect presence of the unit, and then to synchronize with bit and packet boundaries.

C. Bus Transceivers

For every station in the system, the transmit tap is connected to the outbound channel while the receive tap is connected to the inbound channel. The transmit tap is unidirectional. If the transmission medium is a coaxial cable, then the transmit tap is identical to that seen in CATV installations.[3] Other media such as optical fibers are inherently unidirectional in nature. In addition to transmitting on the outbound channel, one of the basic features required for the UBS-RR algorithm and the express access protocol is the ability for each station to sense the carrier due to transmissions by stations on the upstream side of its transmitter. As in Ethernet, line amplifiers may have to be incorporated on the medium in order to regenerate the signal. The spacing of such amplifiers is determined by the attenuation characteristics of the medium as well as the taps.

D. Basic Mechanism to Transmit Transmission Units

A station S_i, which senses the outbound channel busy, waits for EOC(OUT). Immediately following the detection of EOC(OUT), it starts transmission of its unit. Simultaneously, it senses carrier on the outbound channel (on the upstream side of its transmit tap). If carrier is detected (which may happen in the first t_d seconds of the transmission, and which means that some station S_j with a lower index has also started transmission following its detection of EOC(OUT)), then station S_i immediately aborts its current transmission. Otherwise, it completes the transmission of its unit. Note that all ready stations which detect EOC(OUT) act as described above. The only station to complete transmission is the one with the lowest index, among those ready stations which were able to detect EOC(OUT). Clearly, during and following the transmission of its TU, a station will sense the outbound channel idle, and therefore will encounter no EOC(OUT), and will not be able to transmit another TU in the current round. Thus, in this mechanism, there is no need to distinguish between DORMANT and ACTIVE states, as required in the UBS-RR algorithm.

Note that the possible overlap among several transmission units is limited to the first t_d seconds of these transmissions. It is expected that the loss of the first t_d seconds of the preamble of the nonaborted transmission will not jeopardize the synchronization process at the receivers. According to the above basic mechanism, two consecutive transmission units are separated by a gap of duration t_d seconds, the time necessary to detect EOC(OUT).

The succession of transmission units transmitted in the same round is called a *train*. A train can be seen by a station on the outbound channel only as long as the TU's in it are being transmitted by stations with lower indexes. A train generated on the outbound channel is entirely seen by all stations on the inbound channel. Since there is a gap of duration t_d seconds between consecutive TU's within a

[3] Due to their widespread use, such unidirectional taps are commonly available, reliable, and relatively inexpensive [16].

train, the detection of presence of a train on the inbound channel can be best achieved by defining the new Boolean function TRAIN(t, IN) = $c(t - t_d$, IN) + $c(t$, IN). Clearly we have

$$\text{TRAIN}(t, \text{IN}) = \begin{cases} 1 & \text{as long as a train is in progress} \\ 0 & \text{otherwise.} \end{cases}$$

The transition TRAIN(t, IN): $1 \rightarrow 0$ defines the end of a train (EOT(IN)), and the transition TRAIN(t, IN): $0 \rightarrow 1$ defines the beginning of a train (BOT(IN)).

E. The Topology of Expressnet

After the last TU in a train has completed transmission, a mechanism is needed to start a new train of transmission units. Clearly, it is essential that this mechanism gives access right to the ready station with the lowest index. One may use a mechanism similar to the UBS-RR algorithm itself. (That is, as soon as EOT(IN) is detected, a ready station operates as in step 2 of the UBS-RR algorithm.) The drawback of this approach is twofold: the mechanism needed to initiate a new train is different from the basic transmission mechanism and the implementation of the scheme in each station is made dependent on the position of the station (as required by step 2 of the UBS-RR algorithm). A better solution can be obtained by guaranteeing that the event EOT(IN) visits the receivers in the same order as the stations' indexes, which is also the order in which they can transmit. This is achieved if the inbound channel is such that signals on it propagate in the same direction as on the outbound channel.

Thus, we consider the network topology to comprise an outbound channel and an inbound channel which are parallel and on which signals propagate in the same direction (i.e., visiting stations in the same order), and a connection between the outbound channel and the inbound channel to allow the broadcast of all outbound signals on the inbound channel. The propagation delay along the connection τ_c is anywhere between 0 and τ seconds (where τ is again the end-to-end propagation delay on the outbound or inbound channel), depending on the geographical distribution of the users and the way the inbound and outbound channels are connected. The minimum of zero seconds is obtained, for example, if the inbound and outbound channels have a loop shape (or, more generally speaking, the stations with the lower and highest indexes are colocated), as illustrated in Fig. 6(a). The maximum of τ seconds is observed if the *connection* cable is made parallel to the inbound and outbound channels, as shown in Fig. 6(b). In all cases, the propagation delay between the outbound tap and inbound tap for all stations is fixed and equal to $\tau + \tau_c$.

The major feature of this topology rests on the fact that, when the inbound channel is made *exactly parallel* to the outbound channel, the event EOT(IN), used by *all* stations as the synchronizing event to start a new train, will reach any station exactly at the same time as the carrier on the outbound channel due to a possible transmission by a station with a lower index. This helps resolve the overlap of several transmissions at the beginning of a train in just the same manner as the resolution obtained in the basic mechanism for transmitting TU's. This mechanism again allows the ready station with the lowest index to be the first to complete transmission of its TU, and following that the new train will take its normal course. The time gap between two consecutive trains, defined as the time between the end of the last TU in a train and the beginning of the first TU in the subsequent train, is now $\tau + \tau_c + 2t_d$ seconds. (See Fig. 7.)

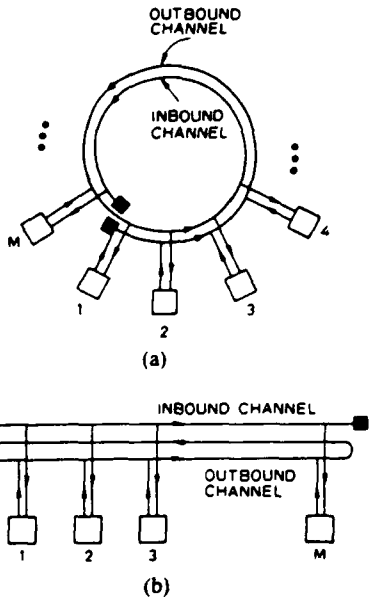


Fig. 6. Examples of Expressnet topologies.

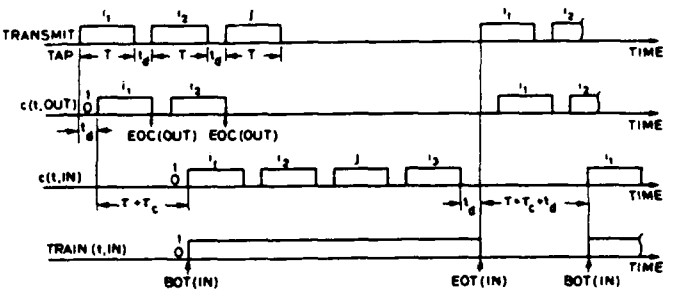


Fig. 7. Signals and events as observed by station j, assuming that stations with indexes $i_1 < i_2 < j < i_3$ are nonidle.

F. The Cold-Start Procedure, and Keeping the Expressnet "ALIVE"

The above algorithm and mechanisms are valid only if there are always events to which actions are synchronized, namely EOC(OUT) and EOT(IN). This assumes that at all times some station is ready, and therefore, trains contain at least one TU and are separated by gaps of fixed duration $\tau + \tau_c + 2t_d$. When this is not the case, the idle time on the inbound channel exceeds $\tau + \tau_c + 2t_d$ seconds. A station which becomes ready at time t_0 such that 1) TRAIN (t_0, OUT) = 0 (thus indicating that no EOC(OUT) will be encountered to synchronize action to) and 2) TRAIN (t, IN) = 0 during the entire period of time $[t_0, t_0 + \tau + \tau_c + t_d]$ (thus indicating that no EOT(IN) will be detected) has to undertake the so-called *cold-start* procedure. This procedure must be designed such that if executed by several

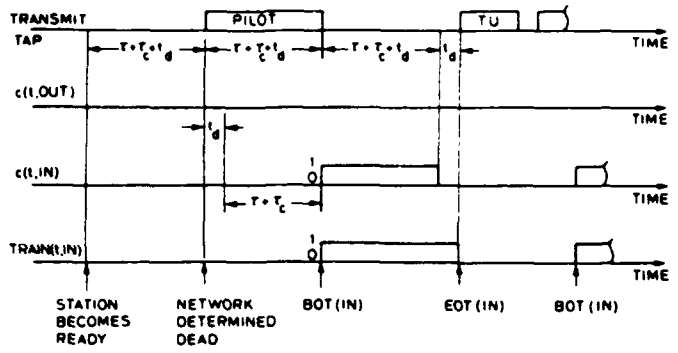


Fig. 8. Signals and events as observed at station j undertaking a cold-start.

stations becoming ready under these conditions, it leads to a single synchronizing event to be used as a time reference, followed then by an orderly conflict-free operation of the network.

The simple one proposed here consists of the following. Once a station has determined that a cold-start procedure is needed, it transmits continuously an unmodulated carrier (called PILOT) until BOT(IN) is detected. At this time the station aborts transmission of the pilot. It then waits for EOT(IN) (consisting of the end of the pilot) and uses that event as the synchronizing event. (Of course, an alternative could be to use the BOT(IN) due to the pilot as the synchronizing event.) It is possible that pilots transmitted by several users overlap in time. This, however, will cause no problem. Note that as long as pilots are aborted as soon as BOT(IN) is *detected*, it is guaranteed that the resulting PILOT as observed on the *inbound channel* is continuous and of length $\tau + \tau_c + t_d$ seconds. Following its end, there will be the normal gap of size $\tau + \tau_c + 2t_d$ before a TU follows. (Of course, this gap is absent if BOT(IN) is used as the synchronizing event.)

Assume that no station is ready when EOT(IN) is detected. The network is then said to go *empty*. The *first* station to become ready when the network is empty, spends $\tau + \tau_c + t_d$ seconds to determine the empty condition, by examining TRAIN(t, IN), after which it starts transmission of the pilot. Then it takes between $\tau_c + t_d$ and $\tau + \tau_c + t_d$ seconds before it detects BOT(IN). (The minimum $\tau_c + t_d$ is observed if the station in question is the lowest index station, and the highest index station happened to become ready exactly at the same time.) Regardless of which is the case, following BOT(IN) a pilot of length $\tau + \tau_c + t_d$ seconds is observed on the *inbound* channel, followed by the gap of $\tau + \tau_c + 2t_d$ seconds and then the transmission unit. Therefore, with the use of the PILOT, the time between the moment at which the *first* station becomes ready in an empty network and the moment at which the next EOT(IN) (due to the end of PILOT, and representing the synchronizing event) reaches the lowest index station is between $2\tau + 3\tau_c + 4t_d$ and $3(\tau + \tau_c) + 4t_d$ seconds. This is the time needed to start a new round and is to be compared to $\tau + \tau_c + 2t_d$ which is the time needed when the network does not become empty. Fig. 8 shows the timing of the cold-start procedure when only one station becomes ready.

To avoid the cold-start operation each time the network goes empty, one needs to guarantee that, as long as some stations may still become ready in the future, synchroniz ing events [namely, EOT(IN)] are created artificially by a such stations. More precisely, we consider a station to be i one of two states: DEAD or ALIVE. A station which is i the ALIVE state has responsibility to perpetuate the ex istence of the synchronizing event EOT(IN) for as long a it remains in that state. To accomplish this, each tim EOT(IN) is detected, the station transmits a short burst o unmodulated carrier, of duration sufficiently long to b very reliably detected (i.e., of a duration of at least t_d seconds). Such a burst is called LOCOMOTIVE. If th train were to be empty (i.e., no stations were to be ready when EOT(IN) is detected), now the LOCOMOTIVE constitutes the TRAIN, and EOT(IN) is guaranteed to take place. Clearly, if some station which is ALIVE is also ready, then immediately following the LOCOMOTIVE, i initiates transmission of its TU and follows the transmission mechanism giving access right to the lowest index. The network is said to be ALIVE if at least one station in the network is ALIVE; otherwise it is said to be DEAD. A station is said to be in the DEAD state if it is not engaging in keeping the network ALIVE and, therefore, is prohibited from transmitting any TU. To be able to transmit, a station has to become ALIVE. For a DEAD station to become ALIVE, it must first determine whether the network is ALIVE or not. Letting t denote the time at which a dead station wishes to become ALIVE, the network is determined ALIVE if a train is detected on the inbound channel anytime in the interval $[t, t + \tau + \tau_c + t_d]$. Otherwise, it is determined DEAD. If the network is determined ALIVE, then the station simply switches to the ALIVE state and acts accordingly. Otherwise, it executes the cold-start procedure following which it becomes ALIVE.

A station which is ALIVE can be either READY or NOT-READY at any moment. This is determined by the state of its transmit buffer, empty or nonempty. To that effect, we define for each station a function TB(t) as

$$\mathrm{TB}(t) = \begin{cases} 1 & \text{if its transmit buffer is nonempty} \\ 0 & \text{otherwise.} \end{cases}$$

An ALIVE station which becomes ready does not have to wait for EOT(IN) to undertake the attempts to transmit its packet. In fact, if an outbound train is observed, the station synchronizes transmissions with EOC(OUT). If, however, at the time it becomes ready, no train is observed on the outbound channel, then EOT(IN) is the synchronizing event.

G. The Express Access Protocol

We have defined above $c(t, \mathrm{OUT})$, $c(t, \mathrm{IN})$, TRAIN(t, IN), TB(t), BOT(IN), EOT(IN), EOC(IN), and EOC(OUT). We now define CTX as the event corresponding to the completion of transmission of the current TU, given that such a transmission has been initiated. We also define TIME-OUT (α) as the event corresponding to the completion of a period of time of duration α, starting the clock at the time when waiting for the event is initiated. From the above discussion, we may define PILOT as a continuous unmodulated carrier, and LOCOMOTIVE as an unmodulated carrier of duration t_d.

We consider that initially station X is in the DEAD state. Upon command (for bringing the station to the state ALIVE and eventually for transmission of data), the following basic algorithm is executed.

Step 1: [Check whether the Expressnet is ALIVE or not. If it is, then proceed with Step 2, otherwise undertake the "cold-start" procedure and then proceed with Step 2.] If TRAIN $(t, \text{IN}) = 1$ (i.e., the Expressnet is already ALIVE,) go to Step 2. Otherwise, wait for the first of the following two events: BOT(IN) or TIME-OUT $(\tau + \tau_c + t_d)$. If BOT(IN) occurs first (then again it means that the Expressnet is ALIVE), go to Step 2. If on the contrary TIME-OUT $(\tau + \tau_c + t_d)$ occurs first (indicating that the Expressnet is not ALIVE), transmit PILOT immediately at the occurrence of TIME-OUT $(\tau + \tau_c + t_d)$, and maintain transmitting it until BOT(IN) is detected, at which time abort transmission of PILOT and proceed with Step 2.

Step 2: Wait for the first of the following two events: EOC(OUT) and EOT(IN). If EOC(OUT) occurs first then go to Step 4. Otherwise, go to Step 3.

Step 3: [A new train has to be stated.] Transmit a LOCOMOTIVE, and go to Step 4.

Step 4: [Determine the current state of the station. If it is ready, then attempt transmission of the TU packet.] If $\text{TB}(t) = 0$ go to Step 2. Otherwise, initiate transmission of the TU. If t_d seconds later $c(t, \text{OUT}) = 1$ (meaning it is not station X's turn), then abort transmission and go to Step 2. Otherwise, complete transmission of the packet and go to Step 2.

In Fig. 9 we present the flowchart of this basic algorithm. In Fig. 10 we give the diagram for a finite state machine which performs the algorithm. It contains seven states. The states labeled D_1, D_2, and D_3 are assumed when the station is DEAD and is in the process of becoming ALIVE; the states labeled $A_1 - A_4$ are assumed when the station is ALIVE. Each possible transition is labeled by the combination of events which causes the transition, followed by the action taken.

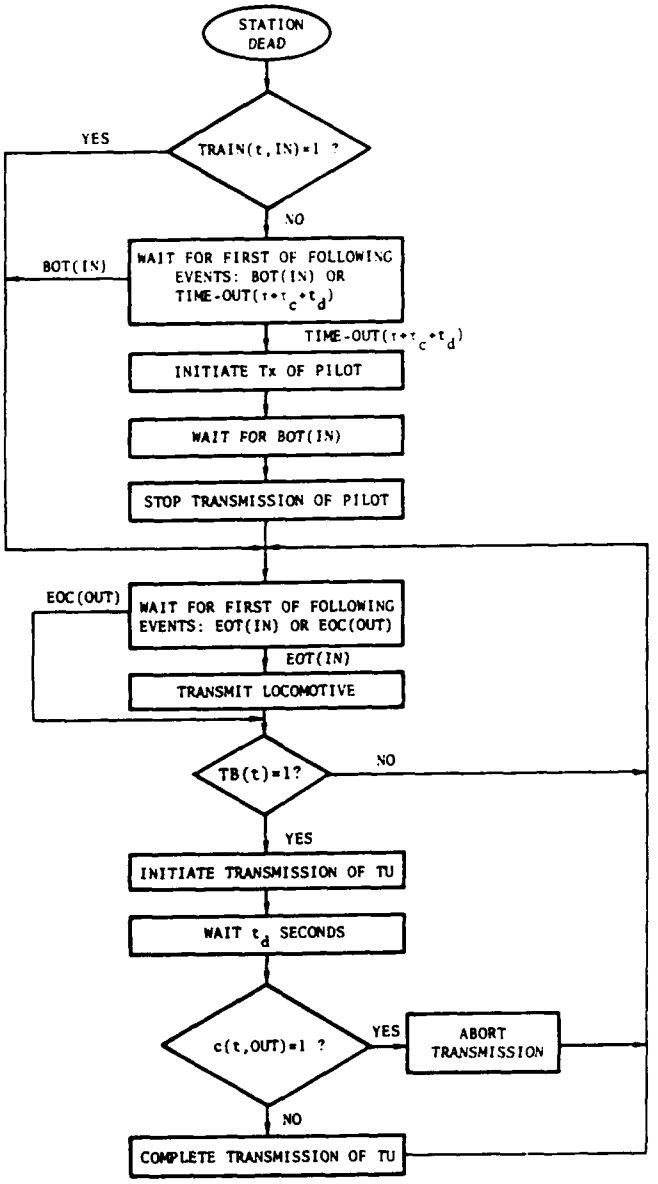


Fig. 9. Flowchart for the express access protocol.

IV. Performance Analysis

In this section we examine the performance of the express access protocol and compare it to that obtained with the schemes of Section II. The channel utilization is evaluated as the ratio of the average time in a train spent for data transmission to the average time separating the start of two consecutive trains. Given that N stations are always busy, the channel utilization is independent of the total number of stations M and is given by

$$C = \frac{NT}{N(T + t_d) + \tau + \tau_c + 2t_d} \tag{9}$$

where the additional t_d seconds in the intertrain gap is due to the existence of the LOCOMOTIVE. Neglecting t_d in comparison to T and τ, and taking $\tau_c = \tau$, (9) is rewritten in terms of $a = \tau/T$ as

$$C(M, N, a) = \frac{1}{1 + \frac{2a}{N}}. \tag{10}$$

Clearly, the channel capacity attained under heavy traffic is given by (10) where $N = M$.

It has been assumed above that the carrier detector and transmit logic of a station are collocated with its taps on the channel. Accordingly, the time it takes for the station's logic to respond to any of the four events EOC(OUT), EOT(IN), BOC(IN), and BOC(OUT) is just the detection time t_d. This is how implementation is expected to be. However, when this is not the case then adjustments in the protocol and its performance analysis have to be made to take into account the propagation delay between a station and its tap. Let τ_s seconds denote the maximum such delay over all stations. By a simple argument it can be shown that it takes a station $t_d + 2\tau_s$ seconds to respond to the occurrence of any of the above-mentioned events on the channel. Note also that the possible overlap among several transmission units is now equal to $2\tau_s + t_d$ instead of just t_d, meaning that the preamble length has to be increased by $2\tau_s W$ bits, an additional overhead which needs to be taken into account in the performance analysis. Denoting

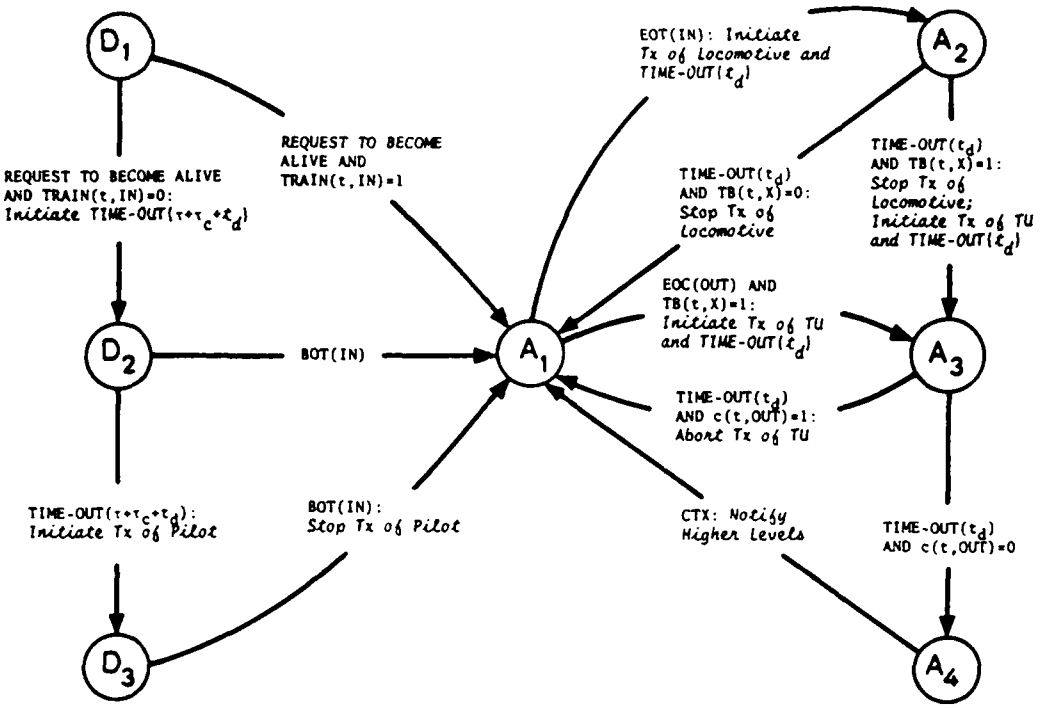


Fig. 10. Finite state machine implementing the express access protocol.

by t_g the gap between two consecutive transmission units in a train, and by t_{ov} the maximum duration of overlap among several transmission units, the throughput with N stations always busy is then given by

$$C = \frac{NT}{N(T + t_g + t_{ov} - t_d) + \tau + \tau_c + t_g + t_d} \qquad (11)$$

where $t_g = t_{ov} = 2\tau_s + t_d$. Neglecting t_d in comparison to T and τ, and letting $\tau_c = \tau$, (11) reduces to

$$C = \frac{1}{1 + \frac{2a}{N} + \frac{4\tau_s}{T} + \frac{2\tau_s}{NT}}. \qquad (12)$$

As the introduction of τ_s causes degradation in channel throughput, one may conceive placing the critical functions of carrier detection and transmission abortion in the transceiver (close to the channel); then t_{ov} can be kept as small as t_d, and (12) now becomes

$$C = \frac{1}{1 + \frac{2a}{N} + \frac{2\tau_s}{T} + \frac{2\tau_s}{NT}}. \qquad (13)$$

Note that the introduction of τ_s calls for slight modifications to the various parameters used in the above description of the algorithm.

In Fig. 11 we plot the channel utilization $C(M, N, a)$ versus a for various values of N. In this figure we neglect t_d and let $\tau_c = \tau$. Contrary to CSMA-CD or UBS-RR, the channel utilization is not insensitive to N, and improves with increasing values of N. For large a, a high utilization is achieved only if N is large. However, even in the worst case $N = 1$, Expressnet performs at least as good as, if not better than, all schemes considered in Section II. Indeed, the curve labeled $N = 1$ in Expressnet coincides with the

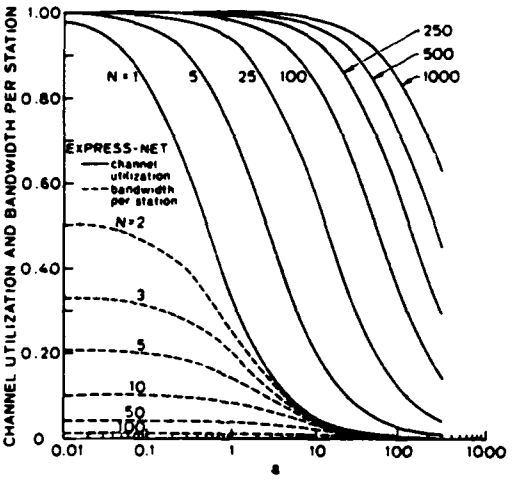


Fig. 11. Channel utilization and bandwidth acquired per station versus a for Expressnet.

curve with the same label in CSMA-CD when $a \leqslant 0.5$, but is superior to the latter when $a > 0.5$. It coincides with BRAM's curve labeled $N/M = 0.5$ (which, when $N = 1$, can be obtained with a total population size M of only 2). It also coincides with UBS-RR's curve labeled $N = \infty$ (which is superior to UBS-RR's curve labeled $N = 1$). As for the comparison of Expressnet with DSMA, we note that for $a \geqslant 1$ the channel utilization in Expressnet with $N = 1$ exceeds that in DSMA as long as $M \geqslant 100$. But with $N \geqslant 5$, Expressnet outperforms DSMA for $a \geqslant 1$ regardless of M. We also plot in Fig. 11 the fraction of the bandwidth acquired per station versus a. This amount decreases with increasing values of N but slower than $1/N$ since the throughput improves with N.

In order to attain a tractable analysis of packet delay and to compare Expressnet to CSMA-CD and BRAM, we consider a model consisting of M users each with a single

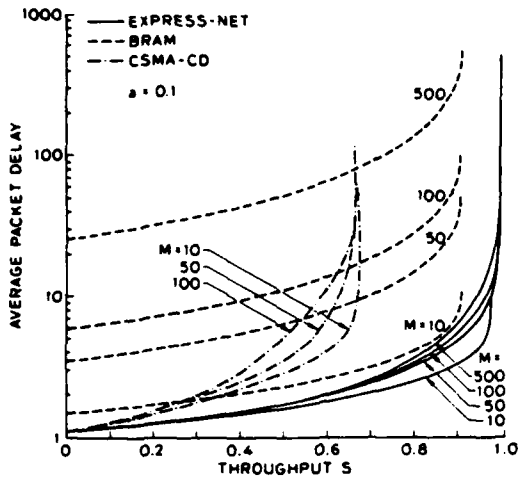


Fig. 12. Throughput-delay performance of CSMA-CD, BRAM, and Expressnet for $a = 0.1$.

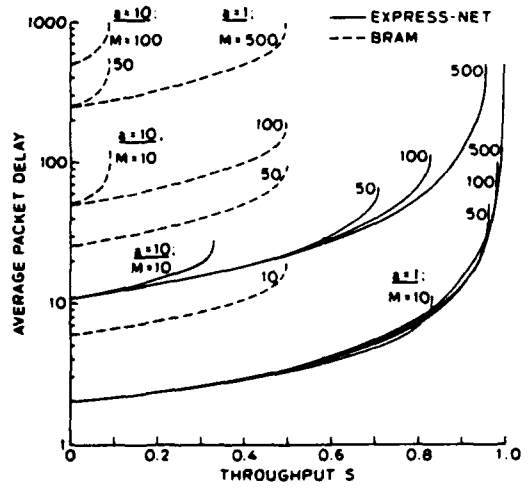


Fig. 13. Throughput-delay performance of BRAM and Expressnet for $a = 1$ and 10.

packet buffer. A user is either idle or backlogged. An idle user generates a packet in a random time which is exponentially distributed with mean $1/\lambda$ seconds. A backlogged user does not generate any packet and becomes idle upon successful transmission of its buffer. A packet transmission time is considered to be fixed equal to T seconds (including the preamble). This model which corresponds to the case of interactive users and which has been referred to in the literature as the "*linear feedback model*" has been used previously to analyze ALOHA and CSMA-CD. It has also been recently used in [17] and [23] in a study on the performance of unidirectional broadcast systems (namely, UBS-RR, Expressnet, and Fasnet) and several round-robin service disciplines achievable in these systems. The analysis of Expressnet (and BRAM) where users are serviced in a predescribed sequence is based on the work in [18], [19] as detailed in [23]. In this section we present the results which are most relevant to the understanding of the performance of Expressnet.

The throughput delay tradeoff is displayed in Figs. 12 and 13 where the normalized average delay is plotted versus the throughput S for various values of M and a. Fig. 12 corresponds to a small value of a, precisely 0.1, and compares Expressnet to CSMA-CD and BRAM. (The CSMA-CD scheme considered is again the slotted p-persistent version analyzed in [8].) Fig. 13 corresponds to $a = 1$ and 10, and compares Expressnet only to BRAM, as CSMA-CD achieves a very small network capacity. These figures clearly show the superiority of Expressnet to both CSMA-CD and BRAM for all values of a. In particular, note that for a given throughput S, while the delay in BRAM is highly sensitive to M, the delay in Expressnet is relatively insensitive to M. For each value of M, packet delay is bounded from above by the *finite* value attained at saturation, i.e., when $\lambda \to \infty$. This maximum delay is precisely $(M + 2a)T$ seconds. Finally, notes on the variance of delay can be found in [23].

V. Integrating Voice and Data on the Expressnet

A. Characteristics of Voice Traffic

It is assumed that vocoders digitize voice at some constant rate. Bits are grouped into packets which are then transmitted via the network to the destination vocoder. To achieve interactive speech and smooth playback operation, it is important to keep the end-to-end delay for each bit of information (from the time the bit is generated at the originating vocoder until it is received at the destination vocoder) within tight bounds. Two components of delay are identified: the packet formation delay and the network delay. The sum must not exceed the maximum allowed in order for all bits to satisfy the delay requirement of speech. An interesting property of round-robin schemes with finite number of stations is that the delay incurred in the transmission of a packet is always finite and bounded from above. This renders it particularly attractive for the packet voice application which we now examine in more detail.

Let W_v be the bandwidth required per voice user (i.e., the vocoder's rate in bits/s), and D_v the maximum delay allowed for any bit of digitized voice (not including the propagation delay). Let B_v denote the number of bits per voice packet. B_v is the sum of two components: $B_v^{(1)}$ which encompasses all overhead bits comprising the preamble, the packet header, and the checksum, and $B_v^{(2)}$, the information bits. Let T_f be the time required to form a packet; it is also the packet intergeneration time for a vocoder. Let T_v be the transmission time of a voice packet on a channel of bandwidth W. We clearly have

$$T_f = \frac{B_v^{(2)}}{W_v} \tag{14}$$

$$T_v = \frac{B_v}{W}. \tag{15}$$

Since packet generation is deterministic, occurring every T_f seconds, we can model each voice user by a $D/G/1$ queue. The packet service time D_n is the time from when the packet reaches the head of the queue until it is successfully received at its destination. Due to the bandwidth constraint we must have

$$T_f \geq D_n. \tag{16}$$

This is also the condition of stability for the $D/G/1$ queue.

Let N denote the number of *active* (off-the-hook) voice sources. (We consider again the case $\tau_s = 0$.) Assuming all queues are nonempty, a train is of length $N(T_v + t_d) + t_d$. The service time distribution of a packet can be bounded by a deterministic one, with the service time equal to a maximum cycle length (i.e., a train length plus the inter-train gap). That is, we now consider all queues to be (pessimistically) represented by $D/D/1$, where the inter-arrival time is T_f and the service time is $D_n = N(T_v + t_d) + \tau + \tau_c + 2t_d$. With these considerations, provided that the queue size is initially 0 and $T_f \geq D_n$, the waiting time of a packet is 0 and its total delay is D_n. The delay constraint for a voice bit is now written as

$$T_f + D_n \leq D_v. \tag{17}$$

The above two constraints lead to a maximum value for N when we choose $T_f = D_n = D_v/2$. Accordingly, we have the optimum packet size given by

$$B_v^{(2)} = \frac{D_v W_v}{2} \tag{18}$$

and the maximum number of voice users allowed at any one time given by

$$N_{\max} = \frac{D_v/2 - (\tau + \tau_c + 2t_d)}{B_v/W + t_d}. \tag{19}$$

We note that as long as $N \leq N_{\max}$, it is guaranteed that the length of a train never exceeds $N_{\max}$ transmission units, and the network packet service time never exceeds the maximum determined above: $N_{\max}(T_v + t_d) + \tau + \tau_c + 2t_d$; consequently, no queueing delay is incurred, the queue size at all users remains ≤ 1, and the total delay constraint for all voice bits is *always* satisfied.

B. Integrating Voice and Data

The principal constraints we have to satisfy here are 1) the delay constraint for voice packets and 2) a minimum bandwidth requirement for data. Although we do not impose a delay constraint on data packets, it is important to provide the bandwidth "reserved" for data on as continuous a basis as possible, and to fairly allocate that bandwidth to data users. Furthermore, we require that the protocols be dynamic in allocating the bandwidth to voice and data applications, allowing data users (or background traffic) to gracefully steal the bandwidth which is unused by voice. To accomplish these objectives on the Expressnet, we consider two types of trains, the voice train type and the data train type. Trains are always alternating between the two types, and stations transmit their packets on the train of the corresponding type. To satisfy the delay constraint for voice packets, it is important not only to limit the number of voice communications to a maximum, but also to limit the data trains to a certain maximum length.

Let W_d be the minimum data bandwidth required. Assuming that data trains are limited to a maximum length L, their effect on the calculation of the optimum value for $N_{\max}$ is to just increase the overhead between consecutive trains by the amount $L + \tau + \tau_c + 2t_d$. $N_{\max}$ is then given by

$$N_{\max} = \frac{D_v/2 - 2(\tau + \tau_c + 2t_d) - L}{B_v/W + t_d}. \tag{20}$$

Since the maximum period of time separating the beginning of two consecutive trains of the same type is $D_v/2$, L must satisfy

$$L = \frac{W_d D_v}{2W}. \tag{21}$$

It is important to limit data trains to the maximum length L at all times, even if the number of active voice users is smaller than $N_{\max}$. Otherwise, situations may arise where the packet delay for a voice packet will exceed D_v. This particularly will occur if, during a data train, a number of new voice users become ready, some of which might incur an initial delay longer than the maximum allowed.

Since a data train may not contain the TU's of all ready stations, it is important that the next data round resumes where the previous data train has ended. This is easily accomplished by the inclusion of the DORMANT/ACTIVE states for data users in the same way as in the UBS-RR algorithm. To switch from the DORMANT to the ACTIVE state, data users have to monitor the length of data trains on the inbound channel: a dormant user switches to the ACTIVE state whenever the length of a data train has not reached its maximum limit L.

In order to alternate between the two types of trains, a station maintains a flag ϕ which gets complemented at each occurrence of EOT(IN). We use the convention $\phi = 0$ for a data train and $\phi = 1$ for a voice train. Now we face the problem of having a station properly initialize ϕ when it becomes ALIVE. The simplest way is as follows. If the network is found DEAD, then following the pilot the station initializes ϕ such that the first train is of the voice type. If the network is found ALIVE, then the station monitors the inbound channel until either a valid packet is observed or the network has gone dead. In the former case, the type of train is derived from the type of packet observed, and ϕ is initialized accordingly. In the second case, the station undertakes a cold-start and the initialization of ϕ is independent of past history. Note that as long as the network is determined ALIVE, a station may not become ALIVE until it has observed a valid packet transmission; all empty trains are ignored. If it is highly likely that long successions of empty trains occur, the above mechanism may induce a high initial delay before the station becomes ALIVE. This can be overcome by including *explicit* information in the LOCOMOTIVE which indicates the type of train. That is, the LOCOMOTIVE now becomes a train-type indicator (TI) packet. The proper indicator packet must be transmitted following EOT(IN) (i.e., an attempt to do so is undertaken) by all ALIVE stations in the network, regardless of the type of packets they intend to transmit. Clearly, only one transmission of the train-type indicator packet is accomplished, by the station in the ALIVE state with the lowest index. With this mechanism, a station wishing to become ALIVE in a network determined ALIVE

waits for BOT(IN) following which it receives and decodes the train-type indicator packet, and initializes the flag ϕ accordingly. The use of the train-type indicator packet increases the overhead caused by the LOCOMOTIVE from t_d to the transmission time of a (relatively short) TI packet. This extra overhead has a small impact on the performance which is still approximated by the equations given above.

We have indicated that in order to satisfy the delay constraint for voice packets it is necessary to limit the number of phone calls in progress at the same time. This requires that a mechanism exists to check whether a new phone call can be accepted or not. The decision is based on the number of calls already set up, and this can be obtained by simply measuring the length of the previous voice train. However, we note that if voice packets representing silence are not transmitted, this measure may not be accurate and one may have to collect data regarding the length of several voice trains before deciding on the acceptance of a new call. The savings on voice bandwidth obtained by silence suppression may be utilized to increase the average data bandwidth. If used to increase the maximum number of phone calls on the network, then a service degradation will have to be allowed as some packets may be delayed beyond the maximum delay D_v. This topic is not carried any further in the present paper.

In the above discussion, it was assumed that the integration of different types of traffic is obtained by using different types of trains and by requiring that a packet be transmitted only on a train of the corresponding type. Another possible approach is to allow mixing of the different types of packets on the same train. In this case there is no need to provide train indicators. However, each station is then required to measure not only the length of the current train but also the period of time in the train already utilized by each type of traffic so that the bandwidth utilized by each traffic does not exceed the maximum value allowed. In order to fulfill the delay requirements for voice traffic, it is easy to see that the global amount of data transmitted in a train has to be limited to $L/2$. Note, however, that this limitation does not affect the overall efficiency of the system nor the bandwidth assigned to each type of traffic. Due to the difficulty foreseen in implementing this approach, we adopt in this paper the scheme consisting of different types of trains.

C. The Voice/Data Access Algorithm

Although the algorithm presented here is for only two types of trains, the concepts can be applied to any larger number of types regardless of the applications intended. Let STATE (ϕ) denote the DORMANT/ACTIVE state of a station with respect to train type ϕ, and TB(t, ϕ) denote the state of its buffer (empty or nonempty) with respect to packet type ϕ. We let $L(\phi)$ denote the maximum allowable length of a train of type ϕ, including the transmission time of the train indicator, and $R(\phi) = L(\phi) + \tau + \tau_c + t_d$. We let C denote a clock which is used to measure the length of a train in progress. In the following presentation, Steps 1–4 are performed by a station wishing to become ALIVE and to initialize its parameter ϕ, while the remaining steps are executed by a station which is ALIVE. Initially STATE(ϕ) = ACTIVE, $\phi = 0, 1$.

Step 1: If TRAIN (t, IN) = 0 then go to Step 2; otherwise, wait for EOT(IN), at which time proceed with Step 2.

Step 2: [Determine whether the network is ALIVE or not.] Wait for the first of the following two events: BOT(IN) and TIME-OUT($\tau + \tau_c + t_d$). If BOT(IN) occurs first then go to Step 3; otherwise, go to Step 4. (In case of a tie, BOT(IN) is considered to have precedence.)

Step 3: Set $C = \tau + \tau_c + t_d$. Receive and decode train-type indicator. If the decoding is unsuccessful then go to Step 1; otherwise, initialize ϕ accordingly, and go to Step 5. (It is assumed here that the operation of decoding a train-type indicator can be completed before EOT(IN) is detected, even when the current train contains only the train indicator. This is easily accomplished in practice by maintaining carrier on beyond the end of transmission of train indicator for a period of time sufficiently long so as to guarantee completion of the decoding operation before the occurrence of EOT(IN).)

Step 4: [The network is DEAD: execute the cold-start procedure.] Initiate transmission of PILOT. Wait for BOT(IN). At the occurrence of BOT(IN), stop transmission of PILOT. Set $\phi = 0$, $C = 0$, and proceed with Step 5.

Step 5: Wait for the first of the following two events: EOT(IN) and EOC(OUT). If EOC(OUT) occurs first, then go to Step 9; otherwise, proceed with Step 6.

Step 6: [At occurrence of EOT(IN), C contains a measure of the length of the previous train. This information is passed on to higher levels to determine acceptance or rejection of new voice calls.] Notify higher levels of the current values of ϕ and C.

Step 7: [Reset STATE (ϕ) if appropriate.] If $C < R(\phi)$ then set STATE (ϕ) = ACTIVE.

Step 8: [A new train is to be started.] Complement ϕ; set $C = 0$; and transmit train indicator. (In the case where the train indicator is an actual packet, its transmission is in accordance with the basic transmission mechanism as follows. Initiate transmission of TI and wait t_d seconds. If at this new point in time $c(t, \text{OUT}) = 1$ then abort transmission of TI and go to Step 5; otherwise, wait for CTX and then proceed with Step 9.)

Step 9: If STATE (ϕ) = ACTIVE and TB(t, ϕ) = 1 and $C < L(\phi)$, then go to Step 10; otherwise, go to Step 5.

Step 10: Initiate transmission of TU. Wait t_d seconds. If at this new point of time $c(t, \text{OUT}) = 0$ and $C < L(\phi)$ then proceed with Step 11; otherwise, abort transmission of TU and go to Step 5.

Step 11: Wait for the first of the following two events: CTX and $C = L(\phi)$. If $C = L(\phi)$ occurs first, then abort transmission of TU and go to Step 5; otherwise, set STATE (ϕ) = DORMANT and go to Step 5.

Note that in the above algorithm, in order to keep STATE(1) = ALIVE for all stations at all times, it is sufficient to assign to $L(1)$ an arbitrarily large number. The flow chart for the voice/data express algorithm is shown in Figs. 14 and 15. Fig. 14 represents Steps 1–4 performed by a DEAD station in becoming ALIVE, while Fig. 15 represents the steps executed by a station which is ALIVE. In case the train indicator is an actual packet, the box labeled

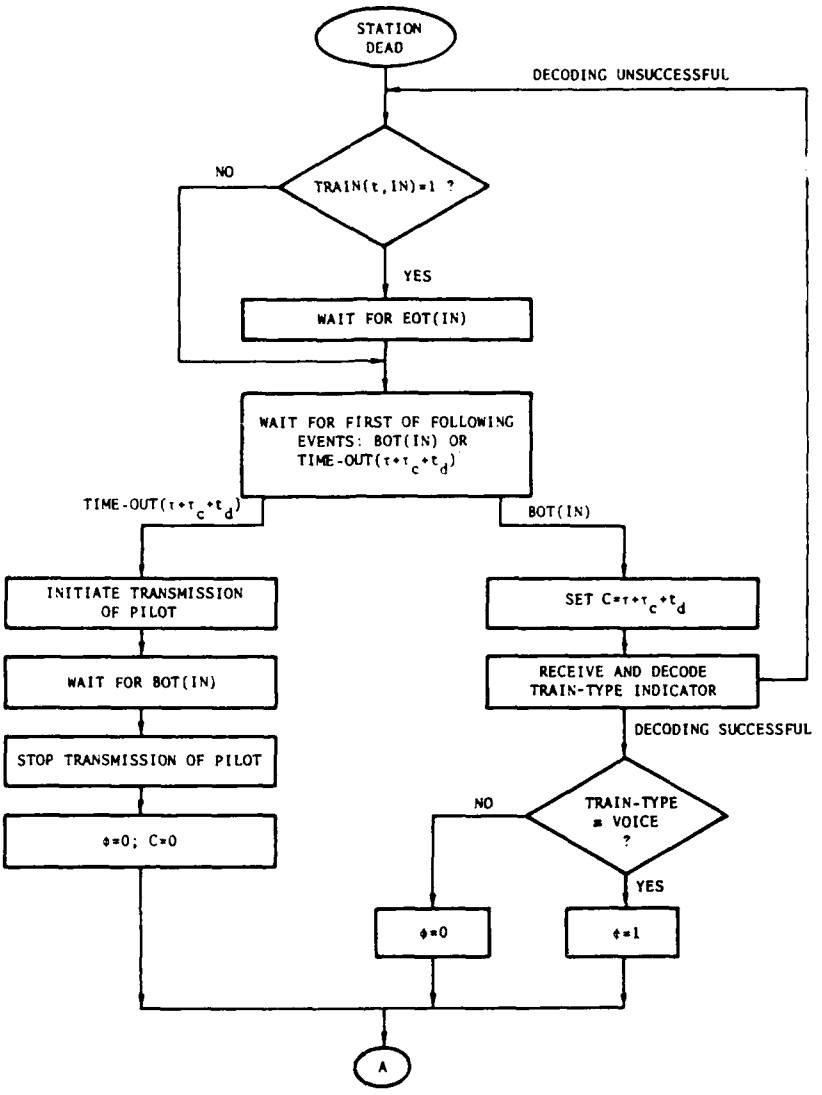


Fig. 14. Flowchart for the initialization portion of the voice/data express algorithm.

TRANSMIT TRAIN INDICATOR must contain all instructions outlined in Step 8 above. The state diagram for a finite state machine which performs the above algorithm is shown in Figs. 16 and 17. Fig. 16 contains the portion of the state diagram corresponding to the flowchart in Fig. 14, while Fig. 17 contains the portion of the state diagram corresponding to the flowchart in Fig. 15.

VII. Conclusions

We have described in this paper the Expressnet, a proposed local area communication network. The express access protocol used by all stations connected to the bus is a distributed algorithm which provides conflict-free transmission of messages. It is essentially a round-robin scheme in which the time to switch from one active user to the next in a round is kept very small, on the order of carrier detection time, thus achieving a performance which is relatively independent of the end-to-end network propagation delay. This feature represents the major improvement obtained with this protocol in comparison to other existing ones, such as CSMA-CD and UBS-RR; it makes it very suitable for local area networks in which, because of high channel speed, long end-to-end delay, and/or small packet size, the propagation delay constitutes a large fraction of or is even larger than the packet transmission time. Furthermore, we have shown that this protocol is particularly suitable for the transmission of packetized voice as it is able to guarantee an upper bound on the transmission delay for each packet. A possible way to integrate voice and data on the same network has been described in detail. In conclusion, we note that Expressnet seems to be most suitable for office automation including real-time applications.

Appendix

As an illustrative example of CSMA-CD, we consider the slotted p-persistent version described and analyzed in

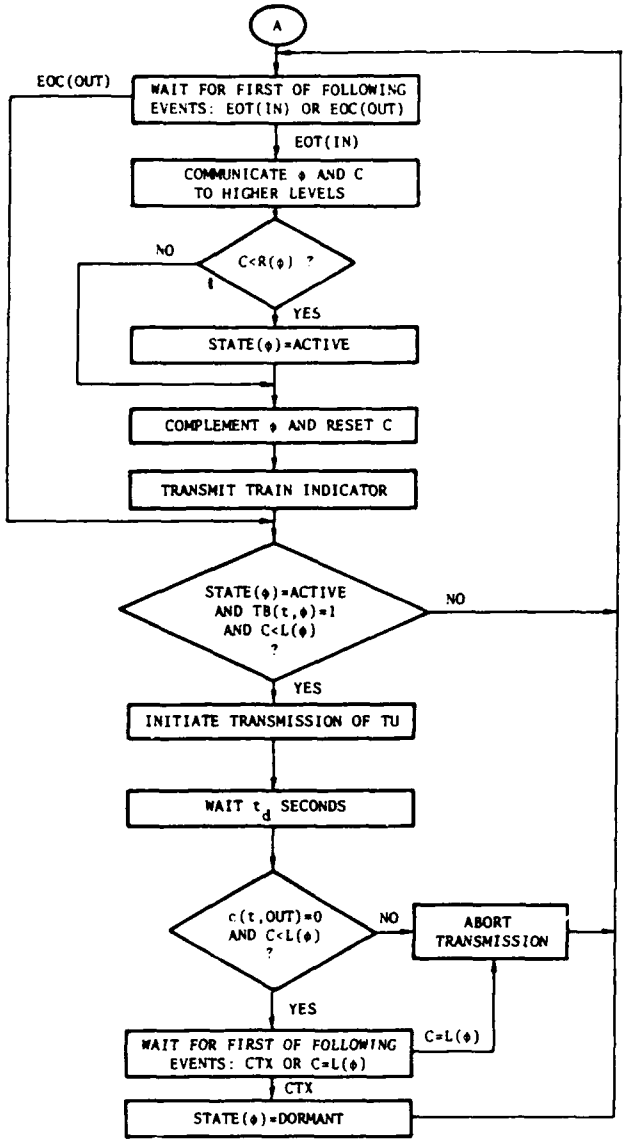


Fig. 15. Flow chart for the portion of the voice/data express algorithm executed by a station in the ALIVE state.

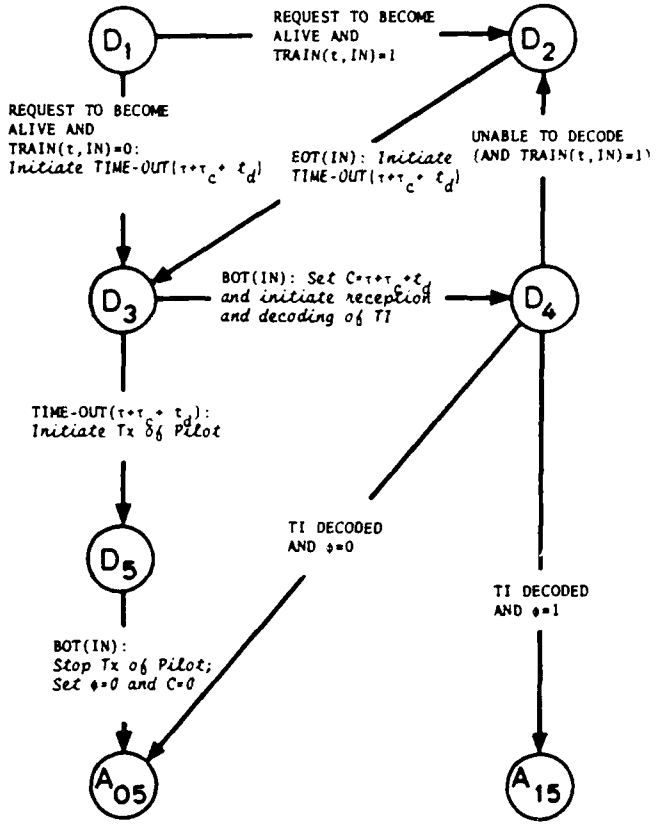


Fig. 16. Finite state machine implementing the initialization portion of the voice/data express algorithm.

[8]. The channel time axis is slotted with the slot size equal to τ seconds, and beginning and end of carrier coincide with slot boundaries. The carrier sensing function which is performed to determine the state of the channel is assumed to be done in the middle of the slot. (Accordingly, following a transmission period, there is at least one idle slot.) Assuming an infinite population model in which users become ready to transmit according to a Poisson process with rate g users per slot, the channel utilization can be derived as in [8] and is given by

$$S(\infty, g, a) = \frac{Tge^{-g}}{Tge^{-g} + (1 - e^{-g} - ge^{-g})T_c + (2 - e^{-g})\tau}. \tag{A.1}$$

T_c is the time needed to detect a collision and abort transmission, and takes the form

$$T_c = \gamma\tau + \xi + \zeta$$

where $\gamma\tau$ represents twice the propagation delay between the two transmitting devices, ξ represents the time it takes a device to detect the interference once the latter has reached it, and ζ is the duration of the jamming period used for collision consensus reinforcement. Ignoring ξ and ζ, we see that $T_c = \gamma\tau$ where γ must equal 2 to account for the worst case of the two extreme users colliding. With these considerations, (A.1) can be rewritten as

$$S(\infty, g, a) = \frac{1}{1 + H(g)a} \tag{A.2}$$

where

$$H(g) = \frac{4e^{g} - 3}{g} - 2. \tag{A.3}$$

Note that the result in (A.2) is valid only as long as $T \geq 2\tau$; in order to always be in a position to perform the collision detection function even when $T < 2\tau$, Ethernet specifies a minimum packet size equal to $2\tau W$ bits, whether the entire packet carries useful information or not. Accordingly for $a > 0.5$, the channel utilization is calculated as the fraction of time useful information is transmitted, and is given by $S(\infty, g, 0.5)/2a$. The channel capacity (or maximum channel utilization) denoted by $C(\infty, a)$ is obtained by maximizing $S(\infty, g, a)$ with respect to g; hence (2).

Assuming a constant number N is always busy, then a similar analysis as for the infinite population leads to

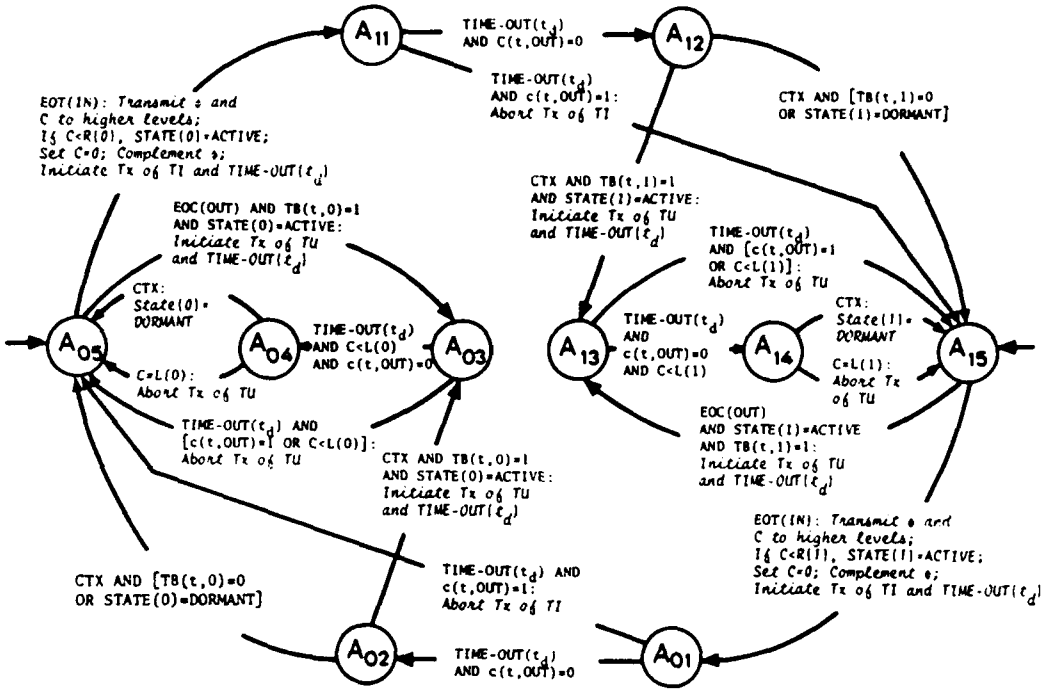


Fig. 17. Finite state machine implementing the portion of the voice/data express algorithm executed by a station in the ALIVE state.

$$S(M,N,p,a)=\begin{cases}\dfrac{1}{1+F(N,p)a} & a\leq 0.5\\[2ex] \dfrac{1}{[2+F(N,p)]a} & a>0.5\end{cases} \tag{A.4}$$

where

$$F(N,p)=\frac{4-3(1-p)^N}{Np(1-p)^{N-1}}. \tag{A.5}$$

To minimize $F(N,p)$, p must satisfy

$$4(Np-1)+3(1-p)^N=0. \tag{A.6}$$

Acknowledgment

The authors would like to acknowledge M. Fine and H. Kanakia for their assistance in the preparation of all performance figures shown in this paper.

References

[1] W. Bux, "Local-area subnetworks: A performance comparison," *IEEE Trans. Commun.*, vol. COM-29, pp. 1465–1473, Oct. 1981.
[2] J. O. Limb and C. Flores, "Description of FASNET, a unidirectional local area communications network," *Bell Syst. Tech. J.*, vol. 71, Sept. 1982.
[3] M. E. Ulug, G. M. White, and W. J. Adams, "Bidirectional token flow system," in *Proc. 7th Data Commun. Symp.*, Mexico City, Mexico, Oct. 1981.
[4] F. Tobagi, "Multiaccess protocols in packet communication systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 466–488, Apr. 1980.
[5] R. M. Metcalfe and D. R. Boggs, "ETHERNET: Distributed packet switching for local computer networks," *Commun. Ass. Comput. Mach.*, vol. 19, July 1976.
[6] "The ETHERNET: A local area network. Data link layer and physical layer specifications. Verson 1.0," Xerox Corp., Sept. 1980.
[7] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple access models and their throughput delay characteristics," *IEEE Trans. Commun.*, vol. COM-25, pp. 1400–1416, Dec. 1975.
[8] F. A. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," *Comput. Networks*, vol. 4, pp. 245–259, Oct. 1980.
[9] S. Lam, "A carrier sense multiple access protocol for local networks," *Comput. Networks*, vol. 4, pp. 21–32, 1980.
[10] I. Chlamtac, W. Franta, and K. D. Levin, "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.*, vol. COM-27, pp. 1183–1190, Aug. 1979.
[11] L. Kleinrock and M. Scholl, "Packet switching in radio channels: New conflict-free multiple access schemes," *IEEE Trans. Commun.*, vol. COM-28, pp. 1015–1029, July 1980.
[12] F. A. Tobagi and R. Rom, "Efficient round-robin and priority schemes for unidirectional broadcast systems," in *Proc. IFIP WG6.4 Zurich Workshop Local Area Networks*, Zurich, Switzerland, Aug. 27–29, 1980.
[13] R. Rom and F. A. Tobagi, "Message-based priority functions in local multiaccess communication systems," *Comput. Networks*, vol. 5, pp. 273–286, July 1981.
[14] K. P. Eswaran, V. C. Hamacher, and G. S. Shedler, "Asynchronous collision-free distributed control for local bus networks," *IEEE Trans. Software Eng.*, vol. SE-7, Nov. 1981.
[15] J. W. Mark, "Distributed scheduling conflict-free multiple access for local area communication networks," *IEEE Trans. Commun.*, vol. COM-28, pp. 1968–1976, Dec. 1980.
[16] "Product profile: Directional taps," *Commun. Eng. Dig.*, pp. 49–51, Nov. 1981.
[17] M. Fine and F. A. Tobagi, "Performance of round robin schemes in unidirectional broadcast local networks," in *Proc. Int. Conf. Commun.*, Philadelphia, PA, June 13–17, 1982, pp. 1C.5.1–1C.5.6.
[18] C. Mack, T. Murphy, and N. L. Webb, "The efficiency of N machines unidirectionally patrolled by one operative when walking time and repair times are constants," *J. Royal Statist. Soc., Ser. B*, vol. 19, pp. 166–172, 1957.
[19] A. R. Kaye, "Analysis of a distributed control loop from data transmission," in *Proc. Symp. Comput. Commun. Networks Teletraffic*, Polytech. Inst. Brooklyn, Brooklyn, NY, Apr. 1972.
[20] L. Fratta, F. Borgonovo, and F. A. Tobagi, "The Expressnet: A local area communication network integrating voice and data," in *Performance of Data Communication Systems*, G. Pujolle, Ed. Amsterdam, The Netherlands: North Holland, 1981, pp. 77–88.
[21] J. O. Limb, "High speed operation of broadcast local networks," in *Proc. Int. Conf. Commun.*, Philadelphia, PA, June 13–17, 1982, pp. 6C.1.1–6C.1.5.
[22] F. Borgonovo and L. Fratta, "B-Expressnet: A communication protocol for bidirectional bus networks," in *Proc. Conf. Commun., Distrib. Syst.*, Berlin, Germany, Jan. 1983.
[23] F. A. Tobagi and M. Fine, "Performance of unidirectional broadcast local area networks: Expressnet and Fasnet," this issue, pp. 913–926.

Fouad A. Tobagi (M'77–SM'83), for a photograph and biography, see this issue, p. 701.

Flaminio Borgonovo received the Doctorate in electrical engineering from the Politecnico di Milano, Milano, Italy, in 1971.

From 1971 to 1973 he worked as a Research Assistant at the Laboratory of Electrical Communication, Politecnico di Milano. In 1973 he reached the Centro di Telecomunicazioni Spaziali of C.N.R. as a Research Associate, and since 1979 he has been an Associate Professor at the Dipartimento di Elettronica, Politecnico di Milano. His main research interests are in the field of communication networks.

Luigi Fratta (M'74) received the Doctorate in electrical engineering from the Politecnico di Milano, Milano, Italy, in 1966.

From 1967 to 1970 he worked at the Laboratory of Electrical Communications, Politecnico di Milano. As a Research Assistant at the Department of Computer Science, University of California, Los Angeles, he participated in data network design under the ARPA project from 1970 to 1971. From November 1975 to September 1976 he was at the Computer Science Department of the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, working on modeling analysis and optimization techniques for teleprocessing systems. In 1979 he was a Visiting Associate Professor in the Department of Computer Science at the University of Hawaii. In the summer of 1981 he was at the Computer Science Department, IBM Research Center, San Jose, CA, working on local area networks. At present he is a full Professor at the Dipartimento di Elettronica of the Politecnico di Milano. His research interests concern communication networks.

Dr. Fratta is a member of the Association for Computing Machinery and the Italian Electrotechnical and Electronic Association.

0733-8716/83/1100-0913$01.00

Performance of Unidirectional Broadcast Local Area Networks: Expressnet and Fasnet

FOUAD A. TOBAGI, SENIOR MEMBER, IEEE, AND MICHAEL FINE, STUDENT MEMBER, IEEE

Abstract—Local area communication networks based on packet broadcasting techniques provide simple architectures and flexible and efficient operation. Unidirectional broadcast systems use a unidirectional transmission medium which, due to their physical ordering on the medium, users can access according to some efficient distributed conflict-free round-robin algorithm. Two systems of this type have been presented in the literature: Expressnet and Fasnet. In this paper we briefly describe these two. We identify three different service disciplines achievable by these systems and discuss and compare the performance of each. These systems overcome some of the performance limitations of existing random-access schemes, making them particularly well suited to the high bandwidth requirements of an integrated services digital local network.

Manuscript received February 1, 1983; revised July 15, 1983. This work was supported by the Defense Advanced Research Projects Agency under Contract MDA 903-79-C-0201, Order A03717, monitored by the Office of Naval Research. This paper was presented at the International Conference on Communications, Philadelphia, PA, June 13-17, 1982.

The authors are with the Computer Systems Laboratory, Department of Electrical Engineering and Computer Science, Stanford University, Stanford, CA 94305.

I. Introduction

LOCAL area communication networks have registered significant advances in recent years. Currently, networks operating in the 1-10 Mbit/s range and spanning a couple of kilometers are commercially available. Although they are adequately satisfying current needs for computer communications, it appears that, in the future, there will be an increasing demand for communication resources as new system architectures (such as distributed processing) evolve and as other services such as voice, graphics, and video are integrated onto the same networks.

Multiaccess broadcast bus systems have been popular since, by combining the advantages of packet switching with broadcast communication, they offer efficient solutions to the communication needs both in simplicity of topology and flexibility in satisfying growth and variabil-

0733-8716/83/1100-0913$01.00

ity. These systems have largely used random-access contention schemes such as carrier sense multiple access (CSMA). A prominent example is Ethernet [1]. Although they have proven to perform well in the environments for which they were designed, these schemes do exhibit performance limitations particularly when the channel bandwidth is high or the geographical area to be spanned is large. For example, in [2], [6] it has been shown that the performance of CSMA/CD degrades significantly as the ratio $\tau W/B$ increases, where τ is the end-to-end propagation delay, W is the channel bandwidth, and B is the number of bits per packet[1] [2], [3], [6].

In an attempt to overcome these limitations a new approach, also based on packet broadcasting, has emerged. This type of network, called the unidirectional broadcast system (UBS) type, uses a unidirectional transmission medium on which the users contend according to some distributed conflict-free round-robin algorithm. We examine two recent proposals, Expressnet [5], [6] and Fasnet [7], [8]. In these systems the access overhead between consecutive packets in a round is independent of both the end-to-end propagation delay and the number of users connected to the network. Due to this feature these systems overcome some of the performance limitations of the random-access schemes as well as earlier round-robin schemes such as the Distributed Computing System [9], the UBS proposed in [10], and BRAM [12]. In this study we present quantitative results showing the performance of Expressnet and Fasnet. In Section II we briefly describe the operation of these two networks with emphasis on the basic access protocol rather than on detailed functionality. As will become clear from the descriptions below, one may identify several different conflict-free round-robin service disciplines that can be achieved in these systems by simple modifications to the access protocols. These disciplines differ in certain aspects of the performance and it is our objective to highlight these differences. In Section III we describe a mathematical model for the systems followed by the analysis in Section IV. Finally, numerical results for the performance of these systems are discussed in Section V.

II. The Unidirectional Broadcast Systems Expressnet and Fasnet

The transmission medium in unidirectional broadcast systems comprises two channels which users access in order to transmit and to read the transmitted data. In Expressnet one channel, designated the *outbound* channel, is used exclusively for transmitting data and the other, designated the *inbound* channel, is used exclusively for reading the transmitted data. All signals transmitted on the outbound channel are duplicated on the inbound channel thus achieving broadcast communication among the stations. In Fasnet, the transmissions on the two unidirectional channels propagate in opposite directions. Users are able to write onto and read from both. Together the two channels provide a connection between any pair of stations on the network. In both systems the asymmetry created by the unidirectional signal propagation establishes a natural ordering among the users required for the round-robin access protocols described below.

[1] Including the preamble needed for synchronization.

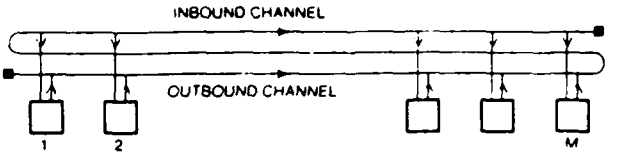


Fig. 1. Topology of Expressnet.

A. Expressnet [5], [6]

The topology of Expressnet is shown in Fig. 1. In addition to writing on the outbound channel each user has the capability to sense activity on that channel due to users on the upstream side of its transmit tap. A user who has a message to transmit is said to be backlogged. Otherwise it is said to be idle. An idle user does not contend for the channel. A backlogged user operates as follows.

1) Wait for the next end of carrier on the outbound channel. [We denote this event by EOC(OUT).]

2) Immediately begin transmitting the packet and at the same time sense the outbound channel for activity from the upstream side.

3) If activity is detected from upstream, then abort the transmission, otherwise complete the transmission. If still backlogged, go back to step 1, otherwise wait for the next packet.

Note that there is a single user which does not have to abort its transmission and hence it transmits successfully. Moreover, a user who has completed the transmission of a packet in a given round will not encounter the event EOC(OUT) again in that round, thus guaranteeing that no user will transmit more than once in a given round. Letting t_d denote the time that it takes to detect presence or absence of carrier, the gap between two consecutive packets in the same round is t_d [the time required to detect EOC(OUT)], and the possible overlap at the beginning of a packet is t_d (the time to detect activity due to upstream users). Thus, the overhead associated with each transmission is on the order of $2t_d$ (Fig. 2).

We now describe the mechanism for initiating a new round. Define a train to be a succession of transmissions in a given round. A train is generated on the outbound channel and entirely seen on the inbound channel by all users. The end of a train on the inbound channel [EOT(IN)] is detected whenever the idle time exceeds t_d. Using a topology for Expressnet as shown in Fig. 1, EOT(IN) will visit each user in the same order as they are permitted to transmit. Thus to start a new round, EOT(IN) is used as the synchronizing event, just as EOC(OUT) was used in the above description. Step 1 of the algorithm should be as follows.

1a) Wait for the first of the two events EOC(OUT) or EOT(IN). (Note that only one such event can occur at a given point in time.)

To avoid losing the synchronizing event EOT(IN) which happens if no packets are ready when it sweeps the inbound channel, all users (whether idle or backlogged)

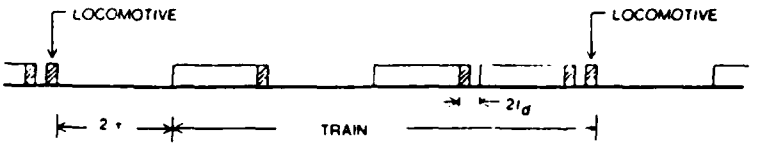


Fig. 2. Typical activity on Expressnet over one cycle as seen on the inbound channel.

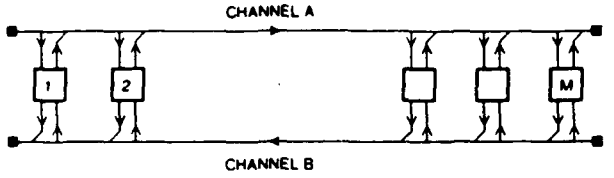


Fig. 3. Topology of Fasnet.

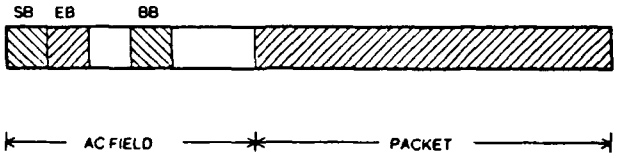


Fig. 4. Format of a slot in Fasnet.

transmit a short burst of unmodulated carrier of duration t_d whenever EOT(IN) is detected. (If the user is in the backlogged state it does so before attempting to transmit a packet.) This burst is referred to as a locomotive. If the train were to be empty, then the end of the locomotive constitutes EOT(IN). It is clear that the time separating two consecutive trains is the propagation delay between the transmit tap and the receive tap of a user, which is the same for all users. (For the topology shown in Fig. 1, this gap amounts to a round-trip delay.)

B. Fasnet [7], [8]

The topology of Fasnet comprises two unidirectional channels (*A* and *B*) where the signals propagate in opposite directions (see Fig. 3). All users can read from and write to both channels. A user wishing to send a packet will transmit on one of the channels such that the recipient is downstream from the sender. As the two channels are identical we consider events on channel *A*. The most upstream user (or head user) and the most downstream user (or end user) on each channel perform special functions. For channel *A* the head user is user 1 and the end user is user *M*. The head user transmits a clock signal which keeps the system bit synchronous.[2] From this clocking information users listening to the channel are able to identify fixed length slots traveling downstream. Each slot begins with an access control field (AC) which determines how and when each station may access the channel. The structure of the AC field, as shown in Fig. 4, consists of three bits. The *start* bit (SB), when set, indicates the start of a new round or cycle (SOC). The *busy* bit (BB), when set, indicates that a packet has been written into the slot. After each of these bits is a dead time which allows the user to read and process them as the slot in traveling by. The third bit, called the *end* bit (EB), is located in the dead time between the start and busy bits. This bit is used by the end user to instruct the head user *via channel B* to initiate a new cycle on channel *A*. We describe two different access protocols for Fasnet. The first which we call gated Fasnet is used in the most recent version of the system [7]. The second which we call nongated Fasnet is used in an earlier version [8].

In gated Fasnet a user with no packets to transmit on channel *A* is said to be IDLE. Upon arrival of a packet to be transmitted on channel *A* (i.e., destined for a user to the right of this one) the user goes to the WAIT state. The user reads SB of every slot. When SB = 1 the user goes to the DEFER state. In this state it simultaneously reads and sets BB of each slot; setting an already set bit is assumed to have no effect. When an empty slot is detected the user writes its packet into it. It then goes to the IDLE state or WAIT state depending on whether it has more packets to transmit or not.

In the nongated Fasnet an IDLE user is said to be either ACTIVE if it has not yet transmitted in the current round or DORMANT if it has. A DORMANT user does not attempt to access the channel. Upon arrival of a packet to an ACTIVE IDLE user, this user moves immediately to the DEFER state. It does not wait for the beginning of the next round as in the gated version. Having transmitted its packet the user becomes DORMANT and does not attempt another transmission in this round. A DORMANT user becomes ACTIVE at the beginning of the next round, i.e., when SB = 1 is detected.

In both versions of Fasnet SB is set by the head user in cooperation with the end user. The end user examines all slots on channel *A*, decoding the status of SB and BB. Upon detecting SB = 1, the end user looks for the first slot in which BB = 0 (indicating that all users are IDLE or DORMANT), at which time it sets EB = 1 in the next slot *on channel B*. The head user, detecting EB = 1, then sets SB = 1 in the next slot on channel *A*. Thus, in the worst case the overhead in initiating a new round will be twice the end-to-end propagation delay plus twice the slot size. The additional two slots are incurred as the end user, having detected BB = 0 on channel *A*, waits for the AC field of the next slot on channel *B* to set EB = 1 and the head user, having detected EB = 1 on channel *B*, waits for the next slot on channel *A* in order to set SB = 1. It is also possible to allow the end user to set EB = 1 every time it encounters BB = 0. This will result in higher throughputs since SOC's will occur at a higher frequency. However, this leads to an irregular pattern of cycle lengths and unfairness among users, giving preference to upstream users. In this paper the former scheme is adopted and analyzed and it is the scheme corresponding to Fasnet.

[2] This is to be contrasted with Expressnet as described above, which is assumed to operate in asynchronous mode. In this mode, a preamble is needed for each packet for synchronization purposes at the receiver. In Fasnet a synchronization pattern is also needed but rather infrequently.

C. The Various Service Disciplines

Clearly from the above descriptions, Expressnet achieves a "conventional" round-robin discipline where users are serviced in a predescribed order determined by their physical location on the network. If a user has no message when its turn comes up, it declines to transmit and then must wait for the next round before getting another turn. We refer to this type of discipline as the nongated sequential service discipline (NGSS). The gated Fasnet system also achieves sequential service in the same physically predescribed order. In this system however, only those users who are ready at the beginning of a given round are serviced in that round. We refer to this discipline as the gated sequential service discipline (GSS). In nongated Fasnet users are also ordered according to their position on the bus; however, following a transmission, the next user to transmit is always the most upstream user who has a packet and has not yet transmitted in the current round. This discipline is referred to as the most upstream first service discipline (MUFS).

Note that Expressnet can be made to operate in GSS mode merely by having each user, upon generating a packet, wait for the event EOT(IN) before attempting to transmit that packet. Similarly, one could operate Fasnet in NGSS mode by allowing each user, upon generating a packet in a given round, to transmit that packet in the next empty slot as long as this user has not yet seen an empty slot go by in the current round and has not yet transmitted in the current round. Otherwise it waits for the SOC. The SOC in Fasnet and the EOT(IN) in Expressnet are analogous events. In MUFS on the other hand, the transmission of each packet is synchronized to an event which sweeps the entire population of users from the most upstream to the most downstream. Thus, only Fasnet can support this discipline.

In this paper we consider only fixed length packets. In Expressnet however, the access protocol allows for packets of any length. In the most recent description of Fasnet [7], slots are required to be of a fixed length in order to simplify the hardware implementation. Nevertheless, in gated Fasnet, variable length packets can be accommodated simply by allowing a user to access a number of consecutive slots. This is feasible because downstream users may only transmit after the current user, and will have full knowledge of the slot usage. In nongated Fasnet only fixed length packets equal to the size of a slot can be achieved since the order of transmissions does not correspond to the physical order; therefore the user does not know how many consecutive empty slots (if any) follow the one in which it begins to transmit.

III. The Model

We consider now a model which is used to analyze the performance of the three service disciplines. Consider a system of M users. Each user has a single packet buffer and is either idle or backlogged. An idle user will generate a packet in a random time which is exponentially distributed with mean $1/\lambda$. A backlogged user does not generate any packets and becomes idle upon successful transmission of its buffer. This model corresponds to the case of interactive users, widely used in the past to analyze slotted ALOHA, CSMA, and other access schemes. The end-to-end propagation delay of the signal traveling across the network is denoted by τ. This corresponds to the propagation delay between the extreme users on one of the channels (e.g., the outbound channel on Expressnet or channel A on Fasnet). The time required to transmit a packet is $T = B/W$ where B is the number of bits in a packet (assumed fixed) excluding the preamble if any in Expressnet and the AC field in Fasnet, and where W is the bandwidth of the channel. The overhead before each transmission to determine which user gets access to the channel is denoted by t_o. In Fasnet t_o is given by the length of the AC field. In Expressnet t_o is given by $2t_d$. The time required to transmit the preamble is denoted by t_p. In Fasnet, since the system is synchronous, $t_p = 0$. In Expressnet, t_p is nonzero if the system is operated asynchronously. Thus, to transmit a packet of length T requires a transmission period of $X = T + t_o + t_p$. The time that the channel becomes idle between rounds is called the interround overhead and is denoted by Y. In Expressnet $Y = 2\tau$.[3] In Fasnet Y must be an integral number of slots and is taken to be $Y = \lceil 2\tau/X \rceil X + X$.

In the next section we present the analysis of this model for these service disciplines. The performance measures derived from these analyses are the channel throughput, the expected delay incurred by a packet, and the variance of this delay.

IV. Analysis

An analysis of a loop system where users are serviced in a predescribed sequence and which fits the NGSS discipline of Expressnet is given by Kaye [13] based on the work in [14]. A summary of this analysis as adapted to the NGSS discipline is presented below. For the GSS discipline we present some additional definitions and both a mean value analysis and a distribution of delay analysis. An analysis of the MUFS discipline is outlined in the appendix and consists of a generalization of the analysis of GSS. This analysis for MUFS is exact for the case $Y \leq X$ but becomes inexact, and in fact leads to pessimistic results, for the case $Y > X$. (For details see the Appendix.) In the discussion of numerical results in the following section, simulation is also used for MUFS when $Y > X$.

A. Analysis of the Nongated Sequential Service Discipline

The probability that there are n packet transmissions in a round, denoted by p_n, is given by [14]

$$p_n = p_0 \binom{M}{n} \prod_{j=0}^{n-1} \left[e^{\lambda(jX+Y)} - 1 \right] \quad 0 < n \leq M \qquad (1)$$

[3] If the topology of Expressnet is such that the two extreme users are collocated, then the interround gap Y is equal to τ. See [6].

where p_0 is determined by $\sum_{n=0}^{M} p_n = 1$. The probabilities p_n satisfy the following recursive formula

$$\frac{p_n}{p_{n-1}} = \frac{M-n+1}{n}\left[e^{\lambda[(n-1)X+Y]}-1\right]. \tag{2}$$

Based on this distribution, Kaye derived the distribution of waiting time $\tilde{w}$, defined as the period between the moment when a packet is generated by a station and the moment when its transmission commences [13]. The expected waiting time and second moment of $\tilde{w}$ are then derived to be given by

$$E[\tilde{w}] = \frac{1}{\bar{n}}\sum_{n=1}^{M} np_n \frac{[(n-1)X+Y]e^{\lambda[(n-1)X+Y]}}{e^{\lambda[(n-1)X+Y]}-1} - \frac{1}{\lambda} \tag{3}$$

$$E[\tilde{w}^2] = \frac{1}{\bar{n}}\sum_{n=1}^{M} np_n \frac{[(n-1)X+Y]e^{\lambda[(n-1)X+Y]}[(n-1)X+Y-2/\lambda]}{e^{\lambda[(n-1)X+Y]}-1} + \frac{2}{\lambda^2} \tag{4}$$

where

$$\bar{n} = \sum_{n=0}^{M} np_n. \tag{5}$$

The mean and variance of packet delay are obtained by adding X to $E[\tilde{w}]$ and X^2 to $E[\tilde{w}^2]-(E[\tilde{w}])^2$, respectively.

From the distribution $\{p_n\}$, one can also easily derive the average network throughput S for a given value of λ. It is simply given by

$$S = \frac{\bar{n}T}{\bar{n}X+Y}. \tag{6}$$

Note that as $\lambda \to \infty$, $\bar{n} \to M$ and the throughput reaches a maximum given by $MT/(MX+Y)$.

B. Analysis of the Gated Sequential Service Discipline

Let $n(t)$ denote the number of backlogged users at time t and let $t_c^{(r)}$ denote the start of the rth round. Since only those users who are backlogged at $t_c^{(r)}$ can transmit in round r, the number of transmissions in the rth round is given by $n(t_c^{(r)})$. The number of backlogged users at the start of round $r+1$ depends on the length of round r and the arrival of packets during this round. Hence, the number of backlogged users at $t_c^{(r+1)}$, denoted by $n(t_c^{(r+1)})$, depends only on $n(t_c^{(r)})$ and the events that occur during the rth round. Thus, $\{n(t_c^{(r)}), r \in (-\infty, \infty)\}$ constitutes an embedded Markov process. So we can use the properties of Markov processes to derive the analytic solution for the performance of the system.

1) Mean Value Analysis: For the mean value analysis the state of the system at an embedded point is described sufficiently by the number of users who are backlogged at this instant. Consider two consecutive embedded points $t_c^{(r)}$ and $t_c^{(r+1)}$. The time interval $[t_c^{(r)}, t_c^{(r+1)}]$ is called a cycle. Each cycle is considered to consist of two subcycles. The first is that part of the cycle where packets are being transmitted. The second is that period which is the interround overhead. Let P be the transition matrix for the embedded Markov process $n(t_c^{(r)})$. The elements of P are denoted by p_{ik} and are given by $p_{ik} \triangleq \Pr\{n(t_c^{(r+1)}) = k | n(t_c^{(r)}) = i\}$. Since those users who transmit during the round can only generate a new packet after they have transmitted the one backlogged from the previous round, the probability of generating a new packet is not the same for all users. Therefore, in computing the transition probabilities p_{ik}, we must account for all possible ways that k out of M users can become ready. To do this we use a recursive approach by considering the instants of time that correspond to the end of a transmission period.

Define the function $G(n, m, s)$ as the probability that, in a round of length n, m users have generated new packets by the end of the sth transmission period of that round. We compute $G(n, m, s)$ in terms of $G(n, m', s-1)$ and we do this by computing the probability that $m-m'$ new packets are generated in the sth transmission period out of a possible $M-(n-s+1)-m'$. (Since $n-s+1$ users are still waiting to transmit in this round, they cannot generate new packets.) Summing over all possible values of m' gives $G(n, m, s)$ as follows.

$$G(n,m,s) = \sum_{m'=0}^{m} \binom{M-(n-s+1)-m'}{m-m'} \cdot [p(X)]^{m-m'}[1-p(X)]^{M-(n-s+1)-m} \cdot G(n,m',s-1) \qquad s \neq 0 \tag{7}$$

where $p(t)$ is the probability of a single user generating a packet during an interval t. Since interarrival times are exponentially distributed with rate λ this is given by

$$p(t) = 1 - e^{-\lambda t}. \tag{8}$$

At the beginning of the round ($s = 0$), there must be with probability 1 no new packets generated. Therefore, $G(n, m, 0) = 1$ for $m = 0$ and $G(n, m, 0) = 0$ for $m \neq 0$. Starting with these initial conditions, the recursion in (7) ends at $G(n, m, n)$, the probability that m users have generated new packets by the end of the first subcycle.

Using (7) and considering additional arrivals during the interround overhead period allows us to compute the elements of the transition matrix P.

$$p_{ik} = \sum_{j=0}^{k} G(i,j,i)\binom{M-j}{k-j}[p(Y)]^{k-j}[1-p(Y)]^{M-k}. \tag{9}$$

Given P we can calculate the stationary distribution of the backlog at the embedded points, the average throughput, and average delay using results from the theory of regenerative processes. The stationary distribution is denoted by $\Pi = (\pi_0, \cdots, \pi_M)$.

Average Throughput: Since $n(t_c^{(r)})$ is a regenerative

process the channel throughput can be computed as the ratio of the expected time that the channel is busy in a cycle to the expected length of a cycle [4], [15]. Hence, the expected throughput, denoted by S, is simply

$$S = \frac{\sum_{i=1}^{M} \pi_i iT}{\sum_{i=0}^{M} \pi_i (iX + Y)}. \tag{10}$$

Average Packet Delay: Consider each user to be a single buffer queueing system with loss and exponential interarrival times. The expected delay of a packet in such a system can be computed as the difference between the expected interdeparture time and the expected interarrival time.[4] Letting s_i denote the expected throughput of packets from user i, the expected interdeparture time of packets from user i is simply $1/s_i$. Hence, the expected delay of a packet from user i is given by

$$d_i = \frac{1}{s_i} - \frac{1}{\lambda}. \tag{11}$$

Averaging over all the users gives the expected delay of a packet D as

$$\begin{aligned} D &= \sum_{i=1}^{M} \frac{s_i}{S} d_i \\ &= \frac{M}{S} - \frac{1}{\lambda} \end{aligned} \tag{12}$$

where we have used the fact that $S = \sum_{i=1}^{M} s_i$.

2) Distribution of Delay Analysis: We now derive the distribution of packet delay in order to compute the higher order moments of delay. In the distribution of delay analysis we select a single user and consider packets only from this user. We refer to this user as the tagged user. This approach will not only yield an expression for the distribution of delay but, by tagging different users on the network, will enable us to compare the performance achieved by the different users. From this we can see how a user's physical location on the network can affect the quality of the service it gets from the network.

Let N, $1 \leq N \leq M$, denote the tagged user. As in the mean value analysis we consider the beginning of a cycle to constitute an embedded point defining an embedded Markov chain. However, in order to completely describe the state of the system at the embedded point, the state descriptor must contain information about the state of the tagged user, the number of active users upstream from the tagged user, and the number of active users downstream from the tagged user. Thus, the state of the system at the beginning of the current round must be described by a vector with three elements $(\delta(t_e^{(r)}), n_u(t_e^{(r)}), n_d(t_e^{(r)}))$ where $n_u(t)$ and $n_d(t)$ are the number of active users upstream and downstream from the tagged user at time t, respectively, and $\delta(t)$ indicates the state of the tagged user at time t. $\delta(t)$ can take on the values 0 and 1 denoting the tagged user to be idle or busy, respectively. $n_u(t)$ and $n_d(t)$ are in the range $[0, N-1]$ and $[0, M-N]$, respectively. To simplify the notation for the state descriptor we define $\mathcal{S}(t) \triangleq (\delta(t), n_u(t), n_d(t))$.

We first compute the transition matrix P for the embedded Markov process $\mathcal{S}(t_e^{(r)})$. We partition the users into three groups. The first consists of those users on the upstream side of the tagged user; the second consists of those users on the downstream side of the tagged user; the third consists of the tagged user. We compute the state transition probabilities by considering new arrivals to the system from each group separately. We now present a generalized form of the recursive function that was used in the mean value analysis. We use this generalized version to compute the state transition probabilities for the upstream and downstream groups of users.

Consider a sequence of x consecutive transmissions by users from a single group. Define the function $G_Z(x, m, s|y)$ to be the probability that, in a transmission sequence of length x, m users have generated new packets by the end of the sth transmission period in this sequence, given that y users had already generated new packets at the beginning of the sequence. The subscript Z denotes the size of the population of users of this group. We can write this function as

$$\begin{aligned} &G_Z(x, m, s|y) \\ &\quad \triangleq \Pr\left\{ n_g(t_b + sX) = x - s + m | n_g(t_b) = x + y \right\} \end{aligned}$$

where $n_g(t)$ denotes the number of users from group g who are in the backlogged state and t_b is the time corresponding to the beginning of the first transmission in the sequence.

For $s > 0$ we can compute G_Z recursively by considering the number of new arrivals during the sth transmission period.

$$\begin{aligned} G_Z(x, m, s|y) = \sum_{m'=y}^{m} &\binom{Z-(x-s+1)-m'}{m-m'} \\ &\cdot p(X)^{m-m'} [1-p(X)]^{Z-(x-s+1)-m} \\ &\cdot G_Z(x, m', s-1|y) \end{aligned}$$
$$s \neq 0,\ s \leq x. \tag{13}$$

At the beginning of the sequence there must be exactly $x + y$ users backlogged and so the boundary conditions on (13) are given by $G_Z(x, m, 0|y) = 1$ for $m = y$ and 0 for $m \neq y$.

Since, in a given round, new arrivals to the system do not affect the order of transmissions in this round and since each user's arrival process is independent, the transition probabilities over one cycle can be computed as the product of the transition probabilities of each group of users over the cycle. Using (13) and conditioning on the size of the backlog of the upstream and downstream users at the beginning and end of their respective transmission sequences allows us to compute the elements of the transmission matrix P. For $\delta(t_e^{(r)}) = 0$ we get

[4] An alternative approach is to use Little's result to compute the average packet delay as the ratio of the average backlog of packets to the average channel throughput. The reader is referred to [16] for the details of this approach.

$$p(0,i,j)(\beta,k,l)=\begin{cases}[p((i+j)X+Y)]^{\beta}[1-p((i+j)X+Y)]^{1-\beta}\\ \cdot\left[\sum_{h=0}^{k}G_{N-1}(i,h,i|0)\binom{N-1-h}{k-h}[p(jX+Y)]^{k-h}[1-p(jX+Y)]^{N-1-k}\right]\\ \cdot\left[\sum_{h=0}^{l}\sum_{y=0}^{h}\binom{M-N-j}{y}[p(iX)]^{y}[1-p(iX)]^{M-N-j-y}G_{M-N}(j,h,j|y)\right.\\ \left.\cdot\binom{M-N-h}{l-h}[p(Y)]^{l-h}[1-p(Y)]^{M-N-l}\right].\end{cases}\tag{14}$$

For $\delta(t_c^{(r)})=1$ we get

$$p(1,i,j)(\beta,k,l)=\begin{cases}[p(jX+Y)]^{\beta}[1-p(jX+Y)]^{1-\beta}\\ \cdot\left[\sum_{h=0}^{k}G_{N-1}(i,h,i|0)\binom{N-1-h}{k-h}[p((j+1)X+Y)]^{k-h}[1-p((j+1)X+Y)]^{N-1-k}\right]\\ \cdot\left[\sum_{h=0}^{l}\sum_{y=0}^{h}\binom{M-N-j}{y}[p((i+1)X)]^{y}[1-p((i+1)X)]^{M-N-j-y}G_{M-N}(j,h,j|y)\right.\\ \left.\cdot\binom{M-N-h}{l-h}[p(Y)]^{l-h}[1-p(Y)]^{M-N-l}\right].\end{cases}\tag{15}$$

From P we can compute the stationary distribution at the embedded points $\Pi=(\pi_0\cdots\pi_M)$.

Consider now an arbitrary arrival to the system in cycle r from the tagged user. The delay incurred by this packet consists of two components; the delay incurred from the instant of arrival until the end of cycle r and the delay incurred from the beginning of round $r+1$ until the end of the transmission of this packet. The distribution of delay is given by the convolution of these two components of delay.

Since the arrival process is memoryless we recognize that the distribution of delay of the first component of delay is given by the distribution of delay of a packet over an interval $[0,t]$ given that the arrival occurs in this interval and that the packet remains backlogged for the remainder of the interval. Let t_a, $0\leq t_a\leq t$, denote the arrival time of the packet. Then, the delay incurred by the packet over the interval $[0,t]$, denoted by D, is $D=t-t_a$. Since interarrival times are exponentially distributed with mean $1/\lambda$, the cumulative distribution function of delay is given by

$$\begin{aligned}\Pr\{D<d|t_a\leq t\}&=\Pr\{t_a>t-d|t_a\leq t\}\\&=\frac{e^{-\lambda t}}{1-e^{-\lambda t}}[e^{\lambda d}-1].\end{aligned}$$

Differentiating with respect to d gives the probability density function. From this distribution function we can compute the Laplace transform of the distribution of delay of a packet over the interval $[0,t]$ given that the packet arrives in this interval. This distribution function denoted by $\mathcal{S}^*(t,s)$ is

$$\mathcal{S}^*(t,s)=\frac{\lambda}{\lambda-s}\frac{e^{-st}-e^{-\lambda t}}{1-e^{-\lambda t}}.\tag{16}$$

Given that this arbitrary arrival is in a round with $\mathcal{S}(t_c^{(r)})=(\alpha,i,j)$ and $n_u(t_c^{(r+1)})=k$ then the Laplace transform of the distribution of delay of the first component of packet delay is given by $\mathcal{S}^*(jX+Y,s)$ if $\alpha=1$ or $\mathcal{S}^*((i+j)X+Y,s)$ if $\alpha=0$. The second component of delay is simply $(k+1)X$. The Laplace transform of the distribution of the total delay incurred by a packet arriving in such a round, denoted by $d^*_{(\alpha,i,j)|(1,k,*)}(s)$, is given by the product of the transforms of the two distributions.

$$d^*_{(\alpha,i,j)|(1,k,*)}(s)=\begin{cases}\mathcal{S}^*((i+j)X+Y,s)e^{-s(k+1)X} & \alpha=0\\ \mathcal{S}^*(jX+Y,s)e^{-s(k+1)X} & \alpha=1.\end{cases}\tag{17}$$

The probability that this arbitrary packet arrives in a cycle with $\mathcal{S}(t_c^{(r)})=(\alpha,i,j)$ and $n_u(t_c^{(r+1)})=k$ is given by

$$\begin{aligned}&\zeta_{(\alpha,i,j)|(1,k,*)}\\&\triangleq\Pr\left\{\mathcal{S}(t_c^{(r)})=(\alpha,i,j),n_u(t_c^{(r+1)})=k|\delta(t_c^{(r+1)})=1\right\}\end{aligned}\tag{18}$$

where by conditioning on $\delta(t_c^{(r+1)})=1$ we have conditioned on the event of an arrival from the tagged user in cycle r. Using conditional probability, ζ can be expressed as

$$\zeta_{(\alpha,i,j)|(1,k,*)}=K\sum_{l=0}^{M}\pi_{(\alpha,i,j)}p_{(\alpha,i,j)|(1,k,l)}\tag{19}$$

where the constant K can be determined from

$$\sum_{\alpha=0}^{1}\sum_{i=0}^{N-1}\sum_{j=0}^{M-N}\sum_{k=0}^{N-1}\zeta_{(\alpha,i,j)|(1,k,*)}=1.$$

Using (19) to remove the conditions on α, i, j, and k in (17) we can express the Laplace transform of the distribution of delay of a packet from the tagged user as

$$D^*(s) = \sum_{a=0}^{1} \sum_{i=0}^{V-1} \sum_{j=0}^{M} \sum_{k=0}^{V} \zeta_{(a,i,j)(1,k,*)} d^*_{(a,i,j)(1,k,*)}(s). \tag{20}$$

By successive differentiation of (20) and letting $s = 0$ one can compute the moments of delay to any order.

V. Numerical Results

We discuss in this section numerical results showing the performance of Expressnet operating under the NGSS discipline and Fasnet operating under the GSS and MUFS disciplines. Let $a \triangleq \tau/T$. The unit of time is taken to be the transmission time of a packet (i.e., $T = 1$). In both Expressnet and Fasnet we neglect the interpacket overhead t_o since this is assumed to be small compared to the length of the packet. The interround overhead Y is then taken to be $2a$ for Expressnet (and hence for NGSS), and $\lceil 2a \rceil + 1$ for Fasnet (and hence for GSS and MUFS). The performance of these networks for various values of a and M is presented in terms of the throughput as a function of the generation rate of packets, the maximum channel utilization referred to as the network capacity, and the throughput-delay tradeoff. These results show that all three service disciplines exhibit similar performance characteristics. This is to be expected since they are merely variations of a basic round-robin algorithm. However, there are interesting differences which we will highlight in the discussion. All numerical results are obtained from analysis with the exception of MUFS when $Y > 1$ in which case simulation is used. The reason is that, as pointed out in Section IV and the Appendix, the analysis outlined in the Appendix for MUFS gives pessimistic results when $Y > 1$. In most of the results shown below, the preamble in Expressnet has been assumed to be negligible except for certain figures where its effect is explicitly shown.

In Fig. 5 we show, for each of the three service disciplines, the behavior of the throughput S as a function of the aggregate generation rate $M\lambda$ for $a = 1.0$ and 10, and $M = 20$ and 50. $M\lambda$ is the rate at which packets would be presented to the system if all users were in the idle state. The curves show that S increases steadily as $M\lambda$ increases from zero until some finite value of $M\lambda$ (in the vicinity of one), and remains practically constant as $M\lambda$ increases further. This shows that the system remains stable as the load increases to infinity. (Contrast this to CSMA/CD where stability can only be achieved by using some form of dynamic control or a long rescheduling delay leading to a high packet delay [17].) Note how, as a result of gating (i.e., the delaying of packets until the round following the one in which they were generated), the throughput achieved by GSS is always less than or equal to that achieved by NGSS and MUFS. For MUFS with $a = 10$, the curves (which are obtained by simulation), exhibit a slight hump before S levels off to its constant value. This occurs since, at the generation rate corresponding to the hump in S, all users are on the average transmitting in every round, but some users happen to generate and transmit their packets during the interround overhead; this results in a *lower* effective overhead and, hence, higher throughput than expected.

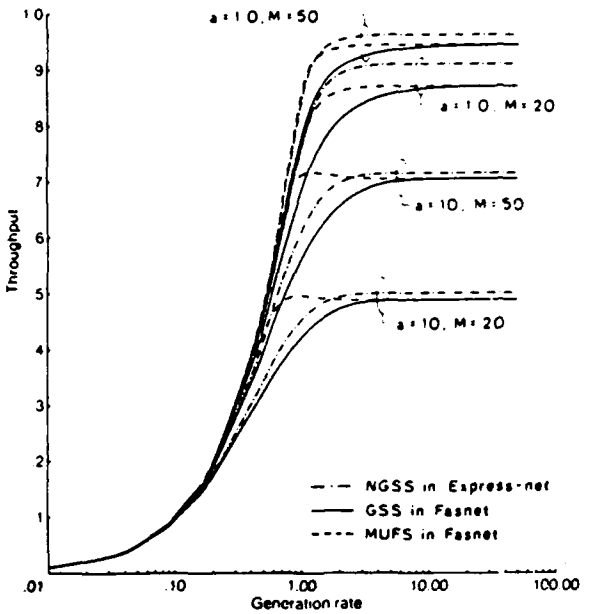


Fig. 5. Channel throughput as a function of the generation rate $M\lambda T$ for NGSS, GSS, and MUFS.

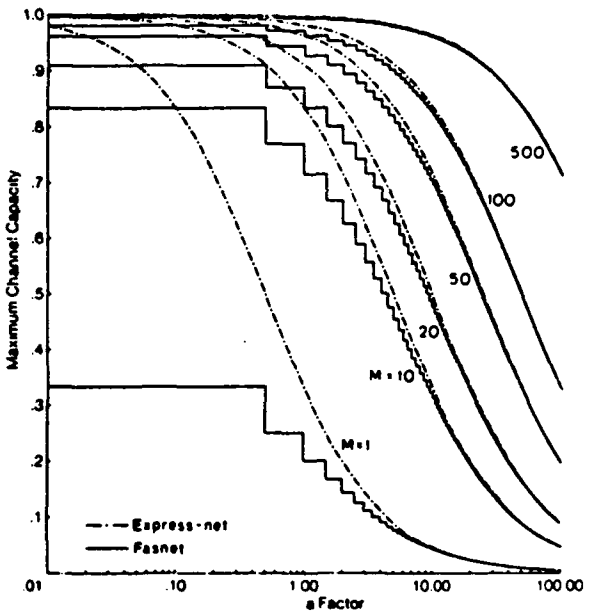


Fig. 6. Network capacity versus a for Expressnet and Fasnet.

As the network reaches saturation, S approaches a finite value, given by $M/(M(1+t_p)+Y)$ independent of the service discipline, which we call the network capacity. For NGSS and GSS the network capacity represents the maximum channel utilization. For MUFS, the maximum channel utilization is slightly higher than the network capacity for the reasons discussed above. The difference between the network capacity for NGSS and that for GSS and MUFS seen in Fig. 5 is a result of the different values of Y in Expressnet and Fasnet for the same value of a ($2a$ and $\lceil 2a \rceil + 1$, respectively); recall that the preamble t_p is assumed here to be zero. In Fig. 6 we plot for Expressnet and Fasnet the network capacity versus a for various values of M. Unlike CSMA/CD (see [6]), a high utilization can still be achieved for large a when M is large. If M is not sufficiently large, then one can alter the access protocol to

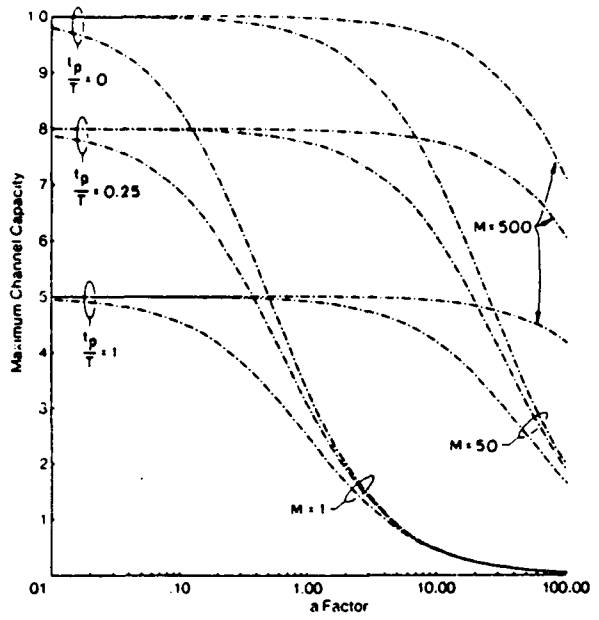


Fig. 7. Network capacity versus a for Expressnet with $M = 1$, 50, and 500, and with three values of the preamble corresponding to $t_p = 0$, $0.25T$, and T.

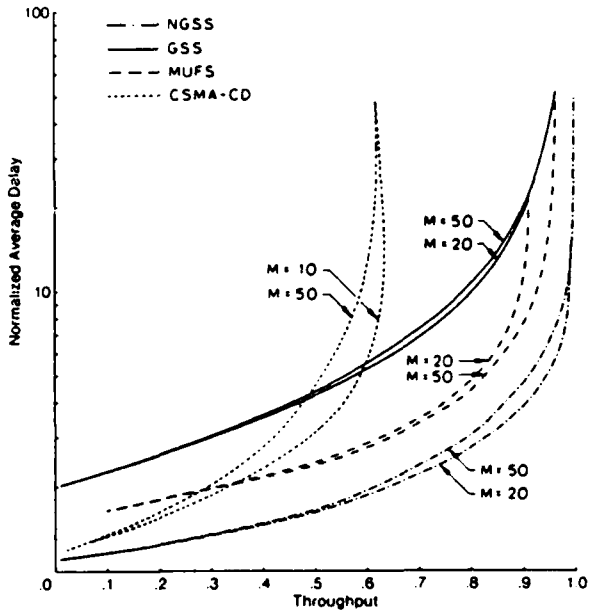


Fig. 8. Average delay normalized by the packet transmission time T versus the channel throughput for NGSS, GSS, MUFS, and CSMA/CD with $a = 0.1$.

allow each user to transmit more than one packet in a round thereby achieving a high utilization as for large M. Due to the fixed slot size in Fasnet, the interround overhead does not decrease below two slots even as a becomes very small; hence the poor channel utilization in the case of Fasnet for small a when $M = 1$. In Expressnet there is no slotting of the time axis and so as $a \to 0$ the overhead becomes zero and the maximum channel utilization goes to 1. The effect of a nonzero preamble on the network capacity for Expressnet can be seen in Fig. 7 where some representative curves are plotted. A preamble which is on the same order of magnitude as the packet transmission time will cause a significant degradation in the capacity.

The relationship between S and average delay D normalized to T for each of the three disciplines is shown in Fig. 8 for $a = 0.1$ and in Fig. 9 for $a = 1.0$ and 10. Also in Fig. 8 is plotted the relationship between S and D for CSMA/CD.[5] As with NGSS, we assume that the preamble for CSMA/CD is negligible. This figure shows how favorably the delay performance of the round-robin schemes compares to that of CSMA/CD. No throughput-delay curves are plotted for CSMA in Fig. 9 since for $a = 1.0$ and 10 this access scheme achieves a very small network capacity. For the three round-robin disciplines, we see that, for a given S, D is fairly insensitive to M as long as M is large enough so that this value of S can be achieved. We also see that the *normalized* average delay increases as a gets larger. However, if a ($= \tau W/B$) has become larger because the channel bandwidth W has increased or the packet size B has decreased, meaning that T ($= B/W$) has decreased, then the *actual* delay is slightly smaller than that obtained with small a; the packet transmission time has decreased thus reducing the size of a slot and the length of a round. On the other hand, if a has become larger because the size of

[5]The CSMA/CD scheme considered here is the slotted nonpersistent version analyzed in [2].

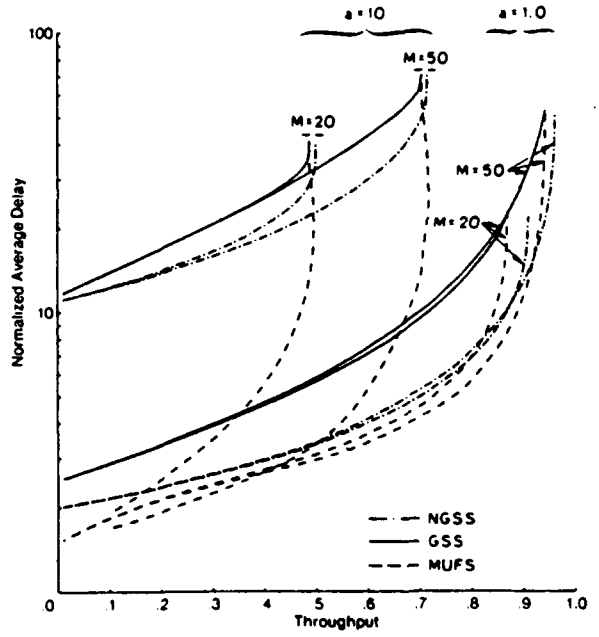


Fig. 9. Average delay normalized by the packet transmission time T versus the channel throughput for NGSS, GSS, and MUFS with $a = 1.0$ and 10. The curves for MUFS with $a = 10$ were obtained by simulation. All the other curves were obtained from the analysis.

the network has increased, meaning that T has not changed, then the actual delay will have increased as represented by the normalized delay. Although the performance trends for all three disciplines are similar, the results do show some differences. In particular one should note that, for large a ($a = 10$), MUFS achieves substantially lower delay than the other two schemes as long as the throughput is not close to saturation. This is due to the fact that in MUFS, having generated a packet, a nondormant user transmits this packet in the next available slot regardless of when the start of cycle appears. In particular, at $S = 0$, D will be equal to $1.5T$ since a user can transmit its packet in the slot immediately following the one in which it was generated.

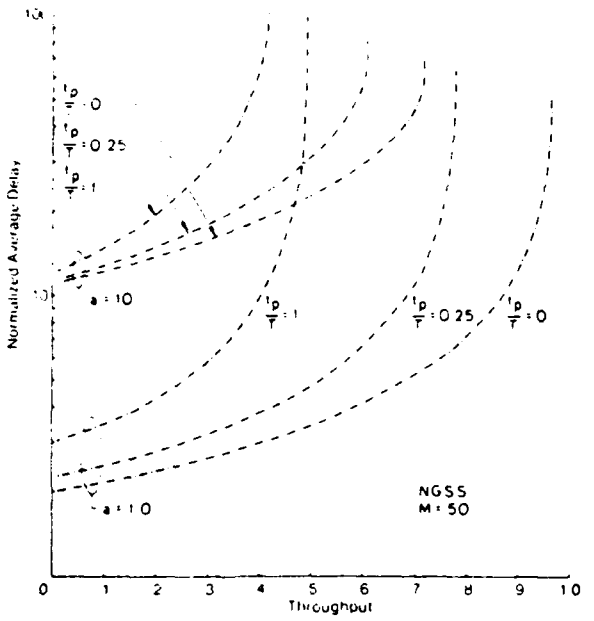


Fig. 10. Average delay normalized by the transmission time T versus the channel throughput showing the effect of the preamble on the throughput-delay performance of NGSS

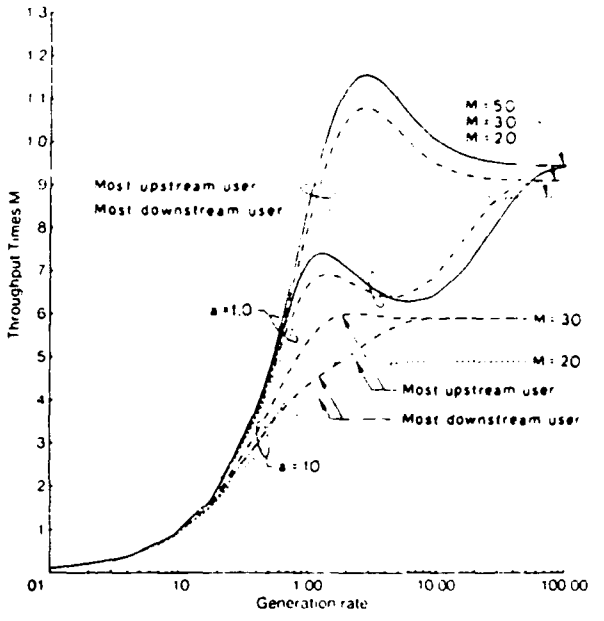


Fig. 11. Throughput multiplied by M versus the generation rate $M\lambda T$ for GSS as achieved by the most upstream user and the most downstream user

instead of incurring on the average a delay of aT while waiting for the SOC as in GSS, or the locomotive as in NGSS.

The average packet delay (normalized to T) versus S for NGSS, in the case where the preamble is not negligible, is plotted in Fig. 10 for $a = 1$ and 10 and $M = 50$. As expected, the delay for a given S increases as the length of the preamble increases.

The results presented above were obtained from the mean value analysis and represent the average performance over all users. For Expressnet with the NGSS discipline, service is offered to each user when it sees the EOT (either on the outbound or inbound channel). Therefore, the EOT can be viewed as an implicit token passed from each user to the next in sequence. Due to the symmetry of this organization, the system is fair and all users achieve the same performance. In GSS and MUFS on the other hand, the synchronizing event is the beginning of a slot which always sweeps the channel from the most upstream user to the most downstream user. As will be seen in the results discussed below, this mode of operation favors the upstream users by giving channel access to the most upstream of all the users contending for a given slot. In the distribution of delay analysis of GSS and MUFS we derived the performance achieved by each user. This enables us to determine the extent to which this performance is affected by the user's location on the network.

First we consider GSS. In Fig. 11 we show M times the throughput achieved by the most upstream user and the most downstream user as a function of $M\lambda$, for various values of M and a. We refer to these two users as user 1 and user M, respectively. The curves show that initially, S increases as $M\lambda$ increases from zero. At low loads there are long idle periods between packet generations, rounds are short, and the throughput achieved by each user is not sensitive to its position on the network. As the network capacity is approached, we see that user 1 achieves a significantly higher throughput than user M. This occurs since in any given round, user 1, having transmitted its packet at the beginning of the round, has the remainder of that round in which to generate a new packet before the next SOC. User M on the other hand, having transmitted at the end of the round, has only the interround overhead period before the next SOC in which to generate its new packet and thus is less likely to be ready at the beginning of the next cycle. As M increases the difference in S between these two users increases. For large values of a this difference is not as pronounced as for small a due to the fact that the interround overhead becomes the dominant factor affecting the performance results and its effect is the same on all users. Finally, as $M\lambda \to \infty$, user M will generate a new packet during the interround gap with probability 1 assuming that $a > 0$; hence, user 1 and user M will achieve the same throughput which is given by the network capacity divided by M. In the limiting case where $a = 0$, user M, having transmitted at the end of a given round, will be ready at the beginning of the next round with probability 0; in particular, at $\lambda = \infty$, user M will transmit once in every two rounds and achieve a throughput of only half that achieved by the other users. The throughput achieved by any of the other users lies within the bounds of user 1 and user M. In fact, any given user will achieve a throughput which is greater than any user downstream from it and less than any user upstream from it. Recall that each user has only a single buffer. If, however, a multiple packet buffer is provided, then a user could generate additional packets for transmission before transmitting the one at the front of the queue. This would reduce the extent of the unfairness suffered by those users on the downstream side of the network. In the limiting case where each user had an infinite buffer, all of the users would achieve the same throughput assuming that they were all generating packets at the same rate.

For the MUFS discipline, the throughput is plotted as a

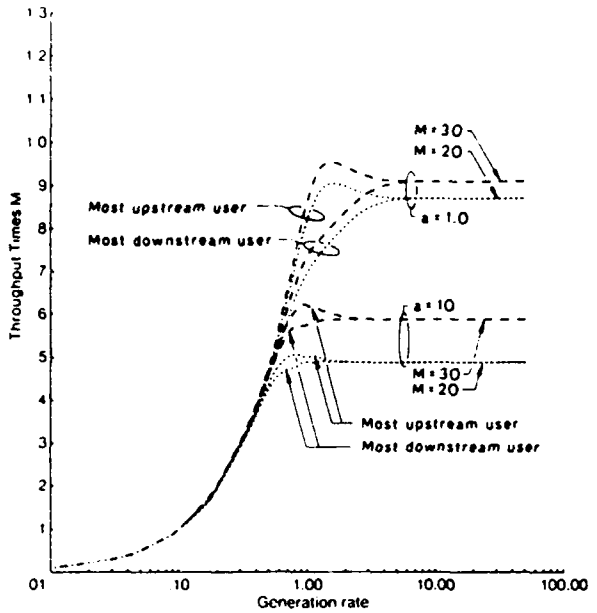


Fig. 12. Throughput multiplied by M versus the generation rate $M\lambda T$ for MUFS as achieved by the most upstream user and the most downstream user. The curves shown for $a = 10$ were obtained by simulation.

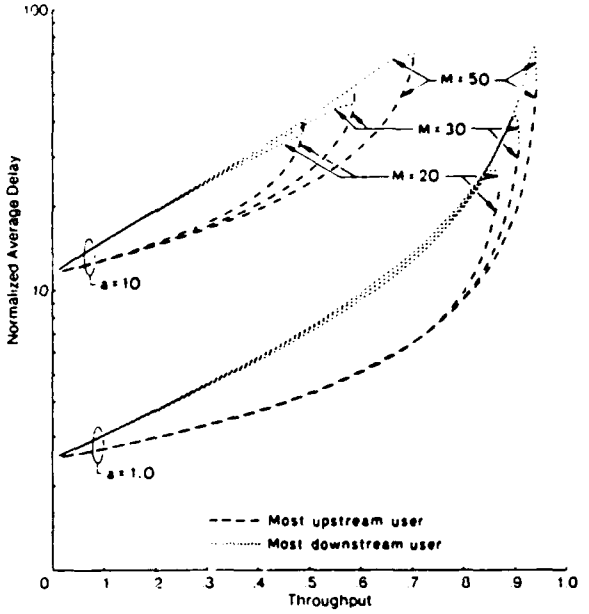


Fig. 13. Normalized average delay versus throughput for GSS as achieved by the most upstream user and the most downstream user.

function of $M\lambda$ for $a = 1$ and 10 and $M = 20$ and 30 in Fig. 12 and exhibits the same characteristics as for GSS. Note that for MUFS there is not as much of a discrepancy in the service achieved by the individual users.

It is important to point out that for Fasnet, there are two separate channels on which users can transmit data. In the analysis we consider only one of these and, in addition, we assume that all users are generating packets for this channel at the same rate. In actual fact the downstream users on a given channel will most probably require a lower throughput on this channel than the upstream ones since they will be transmitting mostly on the other channel. In fact, the most downstream user on a given channel will not transmit any packets on that channel since there is nobody further downstream to receive it.

The difference in average delay between the most upstream and most downstream users is shown for GSS in Fig. 13 and for MUFS in Fig. 14. Since in a given round user 1 is serviced before user M, it achieves a lower delay for a given S. It is interesting to note that in GSS the delay of user 1 is bounded from above by the maximum length of a cycle which is $MT + Y$. For user M the delay is bounded by twice the maximum length of a round plus an inter-round overhead period, that is $2MT + Y$, even though at saturation ($\lambda \to \infty$) the delay will be $MT + Y$. In MUFS and also NGSS the delay of a packet from any user is always bounded by $MT + Y$.

Finally, we examine the variance of delay. The relationship between the variance and the throughput for each of the three service disciplines is shown in Fig. 15 for $a = 1.0$ and 10 and for various values of M. For GSS and MUFS we show the variance of delay versus S as achieved by user 1 and user M. Since for NGSS all users achieve the same performance, we show in Fig. 15 the variance versus S as achieved by any user on the network. For $S = 0$ the variance is nonzero due to the randomness between the time of

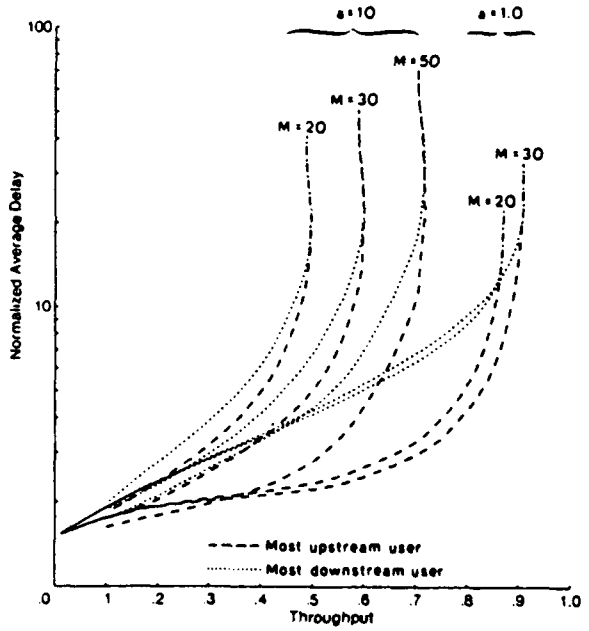


Fig. 14. Normalized average delay versus throughput for MUFS as achieved by the most upstream user and the most downstream user. The curves shown for $a = 10$ were obtained by simulation.

arrival of a packet and the time at which the user may transmit this packet. Depending on the service discipline, the time at which a user may transmit may be after the next locomotive or after the next SOC in the case of NGSS or GSS, respectively, or at the beginning of the next slot in the case of MUFS. This implies that, for large a, the variance for MUFS at low S is lower than for GSS and for NGSS since the randomness in the packet delay in this case is associated with the time of arrival taking place within a slot which is shorter than the period separating two consecutive locomotives or SOC's. As $\lambda \to \infty$, the variance drops to zero since at each user, a new packet is generated as soon as the previous one is transmitted, all rounds are of full length and the packet delay is determin-

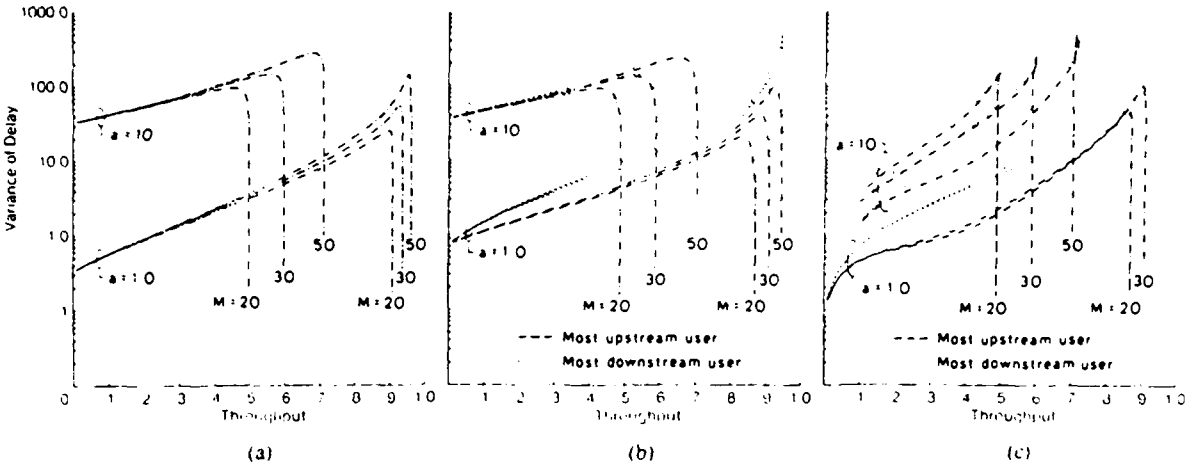


Fig. 15. Variance of delay versus throughput for (a) NGSS, (b) GSS, and (c) MUFS. For GSS and MUFS the variance as achieved by the most upstream user and the most downstream user is shown for each of the values of a and M. For NGSS the curves shown are for the variance as achieved by any user on the network. The curves shown for $a = 10$ in (c) were obtained by simulation.

istic and equal to $MX + Y$. It is interesting to note that the variance incurred is highest for S close to the network capacity while the variance is zero at the network capacity.

VI. Conclusion

In this paper we investigated the performance of two unidirectional broadcast systems that have been presented in the literature: Expressnet and Fasnet. Two versions of the access protocol have been presented for Fasnet. From these two protocols and the one for Expressnet, three service disciplines were identified which we called non-gated sequential service (Expressnet), gated sequential service (Fasnet), and most upstream first service (Fasnet). In addition we noted that, with a simple change in their respective access protocols, Expressnet could be operated in GSS mode and Fasnet could be operated in NGSS mode. However, only Fasnet could support MUFS.

From the analyses of these service disciplines numerical results were computed. We showed that these systems, unlike random-access techniques, can achieve a channel utilization close to 100 percent even when the channel bandwidth is high or the propagation delay of the signal over the network is large. In addition, the network remains stable as the load increases to infinity without the need for any dynamic control of the access protocol. The throughput delay characteristics are excellent and the maximum delay is bounded from above by a finite value which is easily computed. As the throughput approaches the network capacity the variance of delay reaches a peak and then drops to zero. At network capacity the system becomes deterministic with all users transmitting in every round.

Finally, we noted that all three service disciplines exhibit similar performance characteristics. However, in GSS and MUFS there is an element of unfairness which favors some users over others depending on their location on the network, while for NGSS the access protocol is completely fair with all users achieving the same performance.

Appendix
Analysis of the Most Upstream First Service Discipline

The approach for this analysis is similar to that of the analysis for GSS. A summary of the mean value analysis is given. For the complete analysis, including the distribution of packet delay, the reader is referred to [16]. Again we consider two consecutive embedded points $t_e^{(r)}$ and $t_e^{(r+1)}$, and define the state of the system at the embedded points by the number of backlogged users at that instant.

Let $\boldsymbol{P}$ be the transition matrix for the embedded Markov process $n(t_e^{(r)})$. For the MUFS discipline, the number of transmissions in cycle r may be greater than $n(t_e^{(r)})$. In order to compute the elements of $\boldsymbol{P}$ we condition on the number of transmissions in the first subcycle.

$$p_{ik} = \sum_{l=1}^{M} \Pr\left\{ n\left(t_e^{(r+1)}\right) = k | L = l, n\left(t_e^{(r)}\right) = i \right\} \cdot \Pr\left\{ L = l | n\left(t_e^{(r)}\right) = i \right\} \quad \text{(A.1)}$$

where L is the random variable denoting the number of transmissions in the round. Note that by conditioning on L we have removed the dependency of $n(t_e^{(r+1)})$ on $n(t_e^{(r)})$, that is, $\Pr\{ n(t_e^{(r+1)}) = k | L = l, n(t_e^{(r)}) = i \} = \Pr\{ n(t_e^{(r+1)}) = k | L = l \}$.

Let $\theta_i(l) \triangleq \Pr\{ L = l | n(t_e^{(r)}) = i \}$ and let $\phi_l(k) \triangleq \Pr\{ n(t_e^{(r+1)}) = k | L = l \}$. Instead of enumerating all possible events over the cycle, we use recursive functions in order to compute $\theta_i(l)$ and $\phi_l(k)$. Consider a round of L transmissions and the transmission period which is s transmission periods from the end of the round. Define the function $F(m, s)$ as the probability of m given users each generating a packet in the next s transmissions. $F(m, s)$

can be computed recursively by assuming that we know $F(m-j, s-1)$ and that there are j arrivals during transmission period s. This gives

$$F(m,s) = \sum_{j=0}^{m} \binom{m}{j} p(X)^j [1-p(X)]^{m-j} F(m-j, s-1). \tag{A.2}$$

To satisfy the constraints of the system we require that $F(m,s)=0$ for $m \geq s$ and $s \neq 0$, and $F(0,0)=1$. These are the boundary conditions that end the recursion in (A.2). Now $\theta_i(l)$ can be computed as $\binom{M-i}{l-i} F(l-i, l)$ times the probability that $M-l$ users do nothing in the first subcycle. For the case where $n(t_c^{(r)})=0$ there will be an idle period until some users generate packets. We assume that the round then begins with the first transmission from one of these users.

$$\theta_i(l) = \begin{cases} [1-p(l)]^{M-l} \binom{M-i}{l-i} F(l-i, l) & 0 < i \leq M \\ \sum_{j=1}^{M} \dfrac{\binom{M}{j} p(1)^j [1-p(1)]^{M-j}}{1-[1-p(1)]^M} \theta_j(l) & i = 0. \end{cases} \tag{A.3}$$

Having conditioned on the length of the round, we can compute $\phi_l(k)$ using a similar (and simpler) recursive function to the one used in the GSS analysis. Consider the sth transmission period in a given round. Define the function $G(m,s)$ as the probability that, out of the s users who transmitted in the previous s transmission periods, m of them have generated new packets. Since for MUFS only those users who have transmitted in the previous $s-1$ transmission periods could have generated packets to transmit in the next round, we do not need to consider any arrivals from a user who has not yet transmitted in the current round. $G(m,s)$ can be computed recursively by assuming $G(j, s-1)$ and $m-j$ new arrivals in the sth transmission period. We express $G(m,s)$ as

$$G(m,s) = \sum_{j=0}^{m} \binom{s-1-j}{m-j} p(1)^{m-j} [1-p(1)]^{s-1-m} G(j, s-1) \tag{A.4}$$

where the boundary conditions are $G(m,s)=0$ for $m \geq s$ and $G(0,1)=1$. $\phi_l(k)$ is given by

$$\phi_l(k) = \sum_{j=0}^{l} G(j,l) \binom{M-j}{k-j} p(Y)^{k-j} [1-p(Y)]^{M-k} \tag{A.5}$$

where the term $p(Y)^{k-j}[1-p(Y)]^{M-k}$ accounts for the probability that $k-j$ users generate packets during the second subcycle. Note that this assumes that any packets generated during the second subcycle remain backlogged until the next cycle. Although this assumption is exact when $Y \leq X$, it becomes inexact when $Y > X$ since, in the latter case, it may be possible for a nondormant user to both generate and transmit a packet during the second subcycle. This analysis leads to pessimistic results when $Y > X$.

The elements of $\boldsymbol{P}$ are now computed as $p_{ik} = \sum_{l=\max(1,i)}^{M} \theta_i(l) \phi_l(k)$. Given $\boldsymbol{P}$ we can calculate the stationary distribution Π.

Average Throughput: As in the analysis of GSS, the average channel throughput S is computed as the ratio of the expected time in a cycle that the channel is carrying packets to the expected length of a cycle. This is simply given by

$$S = \frac{\bar{L} T}{\pi_0 X/(1-e^{-M\lambda X}) + \bar{L} X + Y} \tag{A.6}$$

where $\bar{L}$ is the expected number of transmission periods in a round. Since we have assumed that no packets are transmitted during the second subcycle, $\bar{L}$ can be computed from the distribution in (A.3).

Average Packet Delay: As in the analysis of GSS, the average delay of a packet is given by $D = M/S - 1/\lambda$ where S is given by (A.6).

Acknowledgment

The authors would like to thank the reviewer for suggesting the method of calculating the average packet delay which was used in the mean value analysis. This approach is simpler than the one originally used.

References

[1] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Commun. ACM*, vol. 19, no. 7, pp. 395–403, 1976.

[2] F. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," in *Proc. Local Area Commun. Network Symp.*, Boston, MA, May 1979.

[3] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple access modes and their throughput delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[4] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-25, pp. 1103–1120, Oct. 1977.

[5] L. Fratta, F. Borgonovo, and F. A. Tobagi, "The Expressnet: A local area communication network integrating voice and data," in *Proc. Int. Conf. Performance Data Commun. Syst., Their Appl.*, Paris, France, Sept. 14–16, 1981.

[6] F. Tobagi, F. Borgonovo, and L. Fratta, "Expressnet: A high-performance integrated-services local area network," this issue, pp. 898–913.

[7] J. O. Limb and C. Flores, "Description of Fasnet, a unidirectional local area communications network," *Bell Syst. Tech. J.*, Sept. 1982.

[8] J. O. Limb, "Fasnet: A proposal for a high speed local network," in *Proc. Office Inform. Syst. Workshop*, St. Maximin, France, Oct. 1981.

[9] D. J. Farber *et al.*, "The distributed computing system," in *Proc. 7th Annu. IEEE Comput. Soc. Int. Conf.*, Feb. 1973.

[10] F. A. Tobagi and R. Rom, "Efficient round-robin and priority schemes for unidirectional broadcast systems," in *Proc. IFIP 6.4*

Zurich Workshop Local Area Networks, Zurich, Switzerland, Aug. 27–29, 1980.

[11] R. Rom and F. A. Tobagi, "Message-based priority functions in local multiaccess communications systems," *Comput. Networks*, vol. 5, pp. 273–286, July 1981.

[12] I. Chlamtac, W. Franta, and K. D. Levin, "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.*, vol. COM-27, pp. 1183–1190, Aug. 1979.

[13] A. R. Kaye, "Analysis of a distributed control loop for data transmission," in *Proc. Symp. Comput. Commun. Networks Teletraffic*, Polytech. Inst., Brooklyn, NY, Apr. 1972.

[14] C. Mack, T. Murphy, and N. L. Webb, "The efficiency of *N* machines unidirectionally patrolled by one operative when walking time and repair times are constants," *J. Royal Stat. Soc.*, ser. B, no. 19, pp. 166–172, 1957.

[15] F. Tobagi *et al.*, "Modeling and measurement techniques in packet communication networks," *Proc. IEEE*, pp. 1423–1447, Nov. 1978.

[16] M. Fine and F. A. Tobagi, "Performance of uni-directional broadcast local area networks: Expressnet and Fasnet," Stanford Univ. Comput. Syst. Lab. Tech. Rep., to be published.

[17] M. Fine and F. A. Tobagi, "Performance of round robin schemes in unidirectional broadcast local networks," in *Proc. Int. Conf. Commun.*, Philadelphia, PA, June 13–17, 1982, pp. 1C.5.1–1C.5.6.

Fouad A. Tobagi (M'77–SM'83), for a photograph and biography, see this issue, p. 701.

Michael Fine (S'78–M'78–S'79) was born in Pretoria, South Africa, in 1957. He received the B.Sc. degree in electrical engineering from the University of the Witwatersrand, Johannesburg, South Africa, in 1978 and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1981. He is currently working towards the Ph.D. degree at Stanford University.

He is currently a Research Assistant in the Computer Systems Lab, Stanford University. His doctoral research is focused on local area communication networks and packet switching.

0733-8716/83/1100-0926$01.00

SCHEDULING-DELAY MULTIPLE ACCESS SCHEMES FOR BROADCAST LOCAL AREA NETWORKS

Michael Fine and Fouad A. Tobagi

Computer Systems Laboratory
Department of Electrical Engineering
Stanford University, Stanford, CA 94305

Local area communications networks based on packet broadcasting techniques provide simple architectures and efficient and flexible operation. Various ring systems and CSMA contention bus systems have been in operation for several years. More recently, a number of distributed *demand assignment multiple access* (DAMA) schemes suitable for broadcast bus networks have emerged which provide conflict-free broadcast communications by means of various scheduling techniques. Among these schemes, the Token-Passing Bus Access method uses *explicit tokens*, i.e., control messages, to provide the required scheduling. Others use *implicit tokens*, whereby stations in the network rely on information deduced from the activity on the bus to schedule their transmissions. In this paper we present many implicit-token DAMA schemes in a unified manner, identify their basic access mechanisms, provide a classification thereof, and compare them in terms of performance and other important attributes.

1. INTRODUCTION

Local area communications networks can be broadly categorized into two basic types. These are broadcast busses and ring systems [1], [2], [3]. In ring systems the data flow is unidirectional, propagating around the ring from station to station. The interface between the station and the network is an active device which receives the signal from the incoming line and retransmits it on the outgoing line. Various techniques for accessing the channel exist which give rise to various types of ring networks such as token rings, slotted rings, and register insertion rings. Ring networks provide high channel utilization and bounded packet delay. However, reliable operation of the network relies on the integrity of explicit information such as a unique token, slot boundaries, and slot status, and on the proper operation of the active taps in relaying the packets and removing them at either the receiver or the sender.

In broadcast bus networks, random access methods such as CSMA have been effectively employed. The Ethernet [4] is a common example. These schemes are simple to implement, robust, and are considered more reliable than ring networks since the taps and medium used are generally passive. However, due to random conflicts, a fraction of the bandwidth is wasted and packet delay is unbounded. Moreover, it has been shown that the performance of CSMA/CD degrades significantly as the ratio $a \triangleq \tau W/B$ increases, where τ is the end-to-end propagation delay of the signal across the network, W is the channel bandwidth and B is the number of bits per packet [5], [20].

More recently, a number of new demand assignment multiple access (DAMA) schemes for broadcast busses have been proposed which provide conflict-free transmission using distributed access protocols with round robin scheduling functions. Using token passing techniques leading to bounded delay, these schemes are also suitable for bus systems using passive components. The stations that are "alive" from what is called a *logical ring*. In some of these schemes, such as the Token-Passing Bus access method [6], the token is an *explicit message* which gets sent

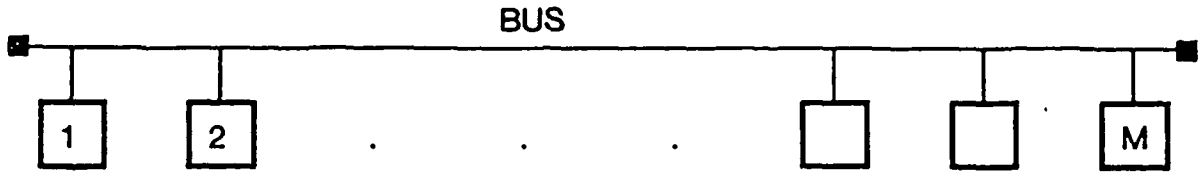


Fig. 1 Topology of the bidirectional bus system

around the logical ring to provide the required scheduling; the station holding the token at any instant is the one that has access to the channel at that instant. It relinquishes its right to access the channel by transmitting the token to the next one in turn. However, as in rings, the robustness of these networks depends on the integrity of the token and on the proper operation of the involved stations. As in random access networks, the performance degrades significantly with a.

In contrast to those schemes where a station transmits an explicit token to the next in turn, in others the stations rely on various events due to activity on the channel to determine when to transmit. Since the token passing operation is *implicit*, the overall robustness of the network is improved over token bus networks. Here too, packet delay is bounded; but in addition both throughput and delay are much less sensitive to a, thus rendering these schemes particularly suitable to networks with high bandwidth, small size packets (as those arising from real time applications), and long distances.

Most of these implicit-token DAMA schemes have been independently proposed and, from reading their descriptions, they appear to be completely different. However, with careful examination it is apparent that basic comonalities can be identified which become explicit by presenting the schemes in a unified manner. In addition, with such a presentation, the unique features of each scheme can be easily identified. This has been the objective of this work; namely, to provide a clear and consistent presentation of many implicit-token DAMA schemes, show their fundamental similarities, and identify their differences. It is possible to identify three basic access mechanisms according to which these schemes can be classified. These are the *scheduling-delay access mechanism*, the *reservation access mechanism* and the *attempt-and-defer access mechanism*. In section II we describe in general terms these three access mechanisms and their underlying network topologies. In this paper however, we focus only on those schemes that fall into the first class; that is, those schemes using the scheduling-delay access mechanism. This in depth presentation is given in section III. For clarity we avoid a chronological presentation but rather the schemes are described in an order which allows a logical development and clear understanding of their features. In a forthcoming paper to appear in the IEEE Transactions on Computers we present the remaining schemes.

II. CLASSIFICATION OF IMPLICIT TOKEN DAMA SCHEMES

The objective of each of the DAMA schemes under consideration is to provide a distributed conflict-free round-robin scheduling function without the use of explicit tokens. Although large in number, these schemes can be grouped into three subsets according to the basic mechanisms used in accomplishing the objective. These three basic access mechanisms are: the *scheduling delay access mechanism*, the *reservation access mechanism*, and the *attempt-and-defer access mechanism*.

In presenting these basic mechanisms, three distinct broadcast bus network configurations can be identified. The first is the *bidirectional bus system* (BBS) in which, as with Ethernet, the signal transmitted by a station propagates in both directions to reach all other stations on the bus. (See Fig. 1.) The second is the *unidirectional bus system* (UBS) in which the transmitted signal propagates in only one direction. Broadcast communications is then achieved in different ways. One way is to provide two unidirectional busses with signals propagating in opposite

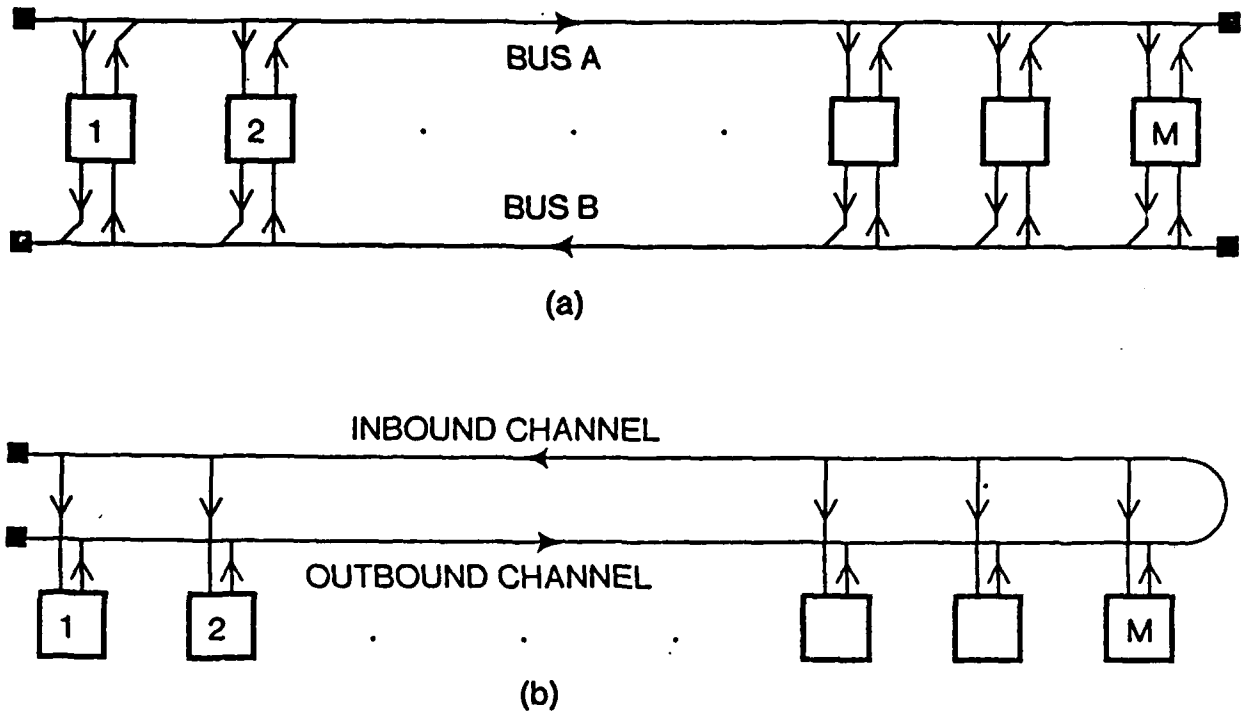


Fig. 2 Two configurations of the unidirectional bus system

directions as shown in Fig. 2(a). Another way is to fold the cable onto itself (or to use a separate frequency channel in the case of broadband signaling) so as to create two channels, an outbound channel onto which the users transmit packets and an inbound channel from which users receive packets, and such that all signals transmitted on the outbound channel are repeated on the inbound channel. (See Fig. 2(b).) The third configuration is the *bidirectional bus with control* (BBC) which consists of a bidirectional bus along with an auxiliary control wire used to control the allocation of the bus.

We now describe the three different basic access mechanisms. We consider that there are M stations in the network. We assume the stations to be numbered 1 through M. As it will be apparent in the sequel, for some schemes this numbering is a requirement and is explicitely made use of in the algorithm, while for other schemes it merely serves the purpose of clarity in presentation. We shall denote by S_i the station with index i. Furthermore, a station which has a message to transmit is said to be *backlogged*. Otherwise, it is said to be *idle*.

a) *The scheduling delay access mechanism.* This class is suitable for the BBS configuration where the only means for coordinating the access of the various users following the end of a transmission is by staggering the potential starting times of these users. More specifically, each station is assigned a unique index number. These indices form a logical ring which determines the order in which stations are allowed to transmit. Included with each transmission is a field for the index number of the sending station. Let S_i be the station currently transmitting. Let EOC(i) denote its *end-of-carrier*. Following the detection of EOC(i), station S_j assigns itself a *scheduling delay* $H_j(i)$, function of both i and j, according to which it schedules its potential transmission. $H_j(i)$ is sufficiently long such that, if at least one of the stations with indices between S_i and S_j is backlogged, then that backlogged station which is the next in sequence following S_i would have begun to transmit its packet and would have been detected by S_j before the scheduled transmission time of S_j, thus resulting in a round-robin scheduling. The network schemes considered in this paper that use this access mechanism are BRAM [7], MSAP [8], SOSAM [9], BID [10], Silentnet [11], and L-Expressnet [12].

b) *The reservation access mechanism.* This class is mainly suitable for the BBC configura-

tion in which the stations use the control wire to place reservations and to reach a consensus on the next station to transmit prior to the transmission on the bus, according to some measure such as the relative positions of the stations on the network, or their addresses. Examples of such schemes are DSMA [13], and the control wire systems of [14] and [16]. The reservation access mechanism can also be implemented on a UBS configuration. For robustness purposes, reservations consist of unmodulated bursts of carrier. These are transmitted on the same bus interleaved with packet transmissions. Consensus here can be reached due to the ordering of the stations, implied by the unidirectionality in transmission and the stations' positions on the bus. An example of this is UBS-RR [17].

c) *The attempt-and-defer access mechanism.* This mechanism can only be implemented on UBS configurations where there is an implicit ordering of the stations. Using this access mechanism, a station wishing to transmit waits until the channel is idle. It then begins to transmit thus establishing its desire to acquire the channel. However, if another transmission from upstream is detected then this station aborts its transmission and defers to the one from upstream. The upstream transmission is therefore allowed to continue conflict-free. Examples of network schemes that use this access mechanism are Expressnet [20], D-Net [21], Fasnet [22], U-Net [24], Token-Less Protocols [25], MAP [26], CSMA/CD-DCR [27], and Buzznet [28].

In the following section we present those schemes that use the scheduling-delay access mechanism. We discuss their similarities and differences, and examine their performance.

For all schemes, we consider that the bandwidth W is the same, and that all packets contain a fixed number of bits, B, giving a constant packet transmission time equal to $T = B/W$. All the schemes considered are asynchronous and hence the transmission of each packet is preceded by the transmission of a preamble needed for receiver synchronization. The transmission time of such a preamble is denoted by Ω. We consider that it takes a nonzero amount of time Δ for a station to detect the presence or absence of carrier on the bus. We also consider that it takes a nonzero amount of time Φ for a station to decode the index of the last station to have transmitted and to compute the scheduling delay. Due to the different amount of computation involved, it is possible that Φ takes on different values for different schemes. While τ denotes the maximum bus end-to-end propagation time, we let $\tau_{i,j}$ denote the signal propagation time between S_i and S_j. Normalizing time to T, we let $a \triangleq \tau/T$, $\delta \triangleq \Delta/T$, $\omega \triangleq \Omega/T$, and $\phi = \Phi/T$. In all schemes, there is an overhead incurred in the transfer of access right from one user to the next backlogged station in sequence. The amount of overhead associated with each scheme has a primary effect on the performance of that scheme. To keep the performance evaluation simple and yet be in a position to adequately compare the various schemes, we consider the situation in which a subset of stations of size N, $N \leq M$, is continuously backlogged, whereas the remaining $M - N$ users are idle. The case $N = M$ is referred to as the *heavy traffic* condition. The performance of a scheme is given in terms of the *channel utilization* $C(M, N)$ representing the fraction of time spent in packet transmission (as opposed to synchronization and protocol overhead), and in terms of the *packet delay* $D(M, N)$ for the head of the queue at each station, (that is, the time separating two consecutive packet transmissions from the same station). From these results one could also derive the network capacity as well as a bound on delay, by considering heavy traffic conditions.

III. SCHEMES USING THE SCHEDULING-DELAY ACCESS MECHANISM

In this section we describe those schemes that use the scheduling delay technique as their basic access mechanism. They differ according to (i) the way the delay function $H_j(i)$ is computed, (ii) the extent to which the scheme is distributed, that is, the extent to which it makes use of particular stations to perform specific functions, (iii) the need for (or lack thereof) a correspondence between the indexing of the stations and their relative positioning on the bus, and (iv) the performance achieved.

A) BRAM (Chlamtac, Franta, Levin, 1979) [7]: Stations are indexed arbitrarily, independent

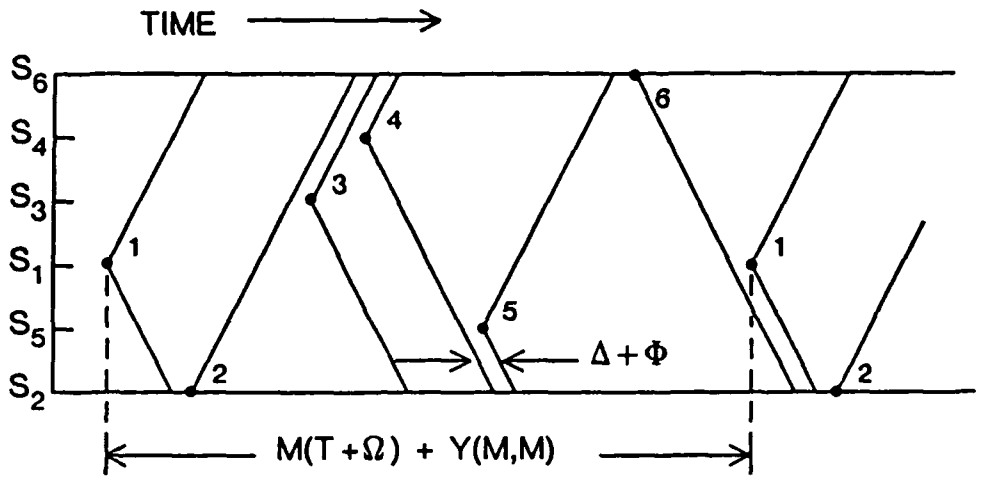


Fig. 3 Time-space diagram showing the activity on the channel for a typical six station BRAM network under heavy traffic conditions.

of their positions on the cable. Furthermore stations are assumed to have no knowledge of their respective positions, nor of the distances that separate them. As indicated in section II, the delay $H_j(i)$ should be sufficiently long such that if any station with index k, $i < k < j$, happened to transmit, then station j would be in a position to detect that transmission prior to its scheduled time. The detection time Δ must be accounted for in the computation of $H_j(i)$. The processing time Φ, however, which is incurred by all stations following the detection of EOC, does not affect $H_j(i)$ and thus need not be accounted for in its computation. Without the knowledge of exact propagation times between consecutively indexed stations, $H_j(i)$ is computed by using the maximum value possible, that is the bus end-to-end propagation delay τ. Under this condition, the scheme will accommodate all possible layouts. The scheduling delay function for BRAM is given by

$$H_j(i) = \begin{cases} (2\tau + \Delta)[(j - i + M - 1) \bmod M] & j \neq i \\ (2\tau + \Delta)M & j = i. \end{cases} \tag{1}$$

As stations are given their turn according to the sequence determined by their indices, a round can be defined as the time since the start of transmission of some station in the backlogged subset until the next start of transmission by that same station. Given N backlogged stations the round length is equal to the cumulative packet transmission times of all stations in the round, $N(T + \Omega)$, plus the cumulative channel overhead incured in the round. We denote the latter by $Y(M, N)$. The channel utilization is then given by

$$C(M, N) = \frac{N(T + \Omega)}{N(T + \Omega) + Y(M, N)}. \tag{2}$$

The packet delay as defined in section II is simply the total length of the round,

$$D(M, N) = N(T + \Omega) + Y(M, N). \tag{3}$$

While $H_j(i)$ is by design independent of the relative physical locations of the stations, the latter does affect the exact timing of the transmissions on the channel and the associated overheads; this is the case because the time until the next transmission following an EOC is based on the time at which that EOC is detected by the next backlogged station in sequence. To illustrate how events occur on the network and to compute the overhead associated with this scheme, which in turn allows us to evaluate the performance, we consider time-space diagrams in which

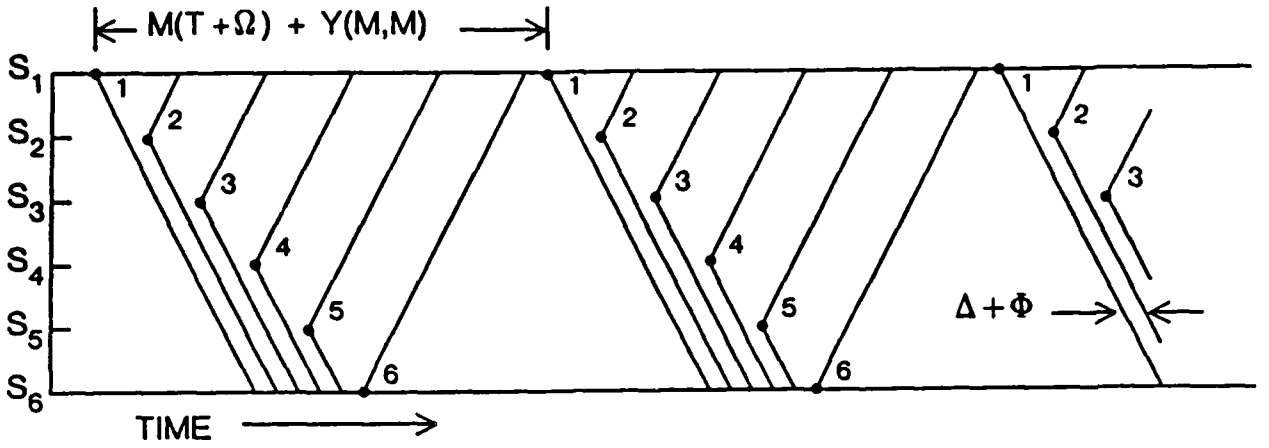


Fig. 4 Time-space diagram showing activity on the channel for BRAM when the stations' logical order is the same as their physical order on the channel.

the vertical axis represents distance along the network and the horizontal axis represents time increasing from left to right. Fig. 3 presents such a diagram for a network with six stations under the assumption of *heavy traffic*. The transmission time of the packet is represented in this and subsequent time-space diagrams by the thickness of a line. The dots represent the origin of an event (in this case a packet transmission with its beginning-of-carrier (BOC) and EOC collapsed into a single event) and the diagonal lines emanating from each dot represent the *time-space locus* of that event. Thus, we see in Fig. 3 a round beginning with the transmission of S_1 and ending with the next transmission of S_1. The overhead between two consecutive transmissions is the time taken for the EOC from one transmitter to propagate on the channel to the tap of the following transmitter plus $\Delta + \Phi$.* The total overhead in a round is the sum of the propagation delays between consecutively indexed stations plus $M(\Delta + \Phi)$. Thus

$$Y(M,M) = \sum_{i=1}^{M-1} \tau_{i,i+1} + \tau_{M,1} + M(\Delta + \Phi) \tag{4}$$

The round overhead is maximized, and hence the network capacity is minimized, when $\tau_{i,i+1} = \tau \ \forall i$. This situation arises in the case where all even numbered stations are collocated on one side of the network and all odd numbered stations on the other. Under these conditions $Y = M(\tau + \Delta + \Phi)$ and $C(M,M) = 1/(1 + \omega + \delta + \phi + a)$.

Clearly, the minimum overhead is incurred for a layout in which all stations are collocated, since then in the limit $\tau_{i,i+1} = 0 \ \ \forall i$. In this case $Y(M,M) = M(\Delta + \Phi)$ and $C(M,M) = 1/(1 + \omega + \delta + \phi)$. If, on the other hand, we insist that the layout be such that the farthest two stations are τ sec. apart, then $Y(M,M)$ is minimized when the stations are ordered in such a way so that their logical order is the same as their physical order on the bus. (See Fig. 4.) In this case $Y(M,M) = 2\tau + M(\Delta + \Phi)$ and $C(M,M) = 1/[1 + \omega + \delta + \phi + 2a/M]$, resulting in a throughput which is almost independent of a if M is sufficiently large.

The question now is how the overhead is affected when some stations do not transmit. Consider three stations numbered consecutively i, $i+1$ and $i+2$. If, in a given round, all three of these stations transmit when their turns come up, then the overhead between these transmissions is $\tau_{i,i+1} + \tau_{i+1,i+2} + 2(\Delta + \Phi)$. Suppose now S_{i+1} does not transmit. S_{i+2} will transmit $2\tau + 2\Delta + \Phi$ sec after EOC(i) reaches it. In this case the overhead is $2\tau + \tau_{i,i+2} + 2\Delta + \Phi$.

---

*The value of Φ may be null if it is possible for a station to decode the index of an on-going transmission and compute the resulting scheduling delay during the time of that transmission.

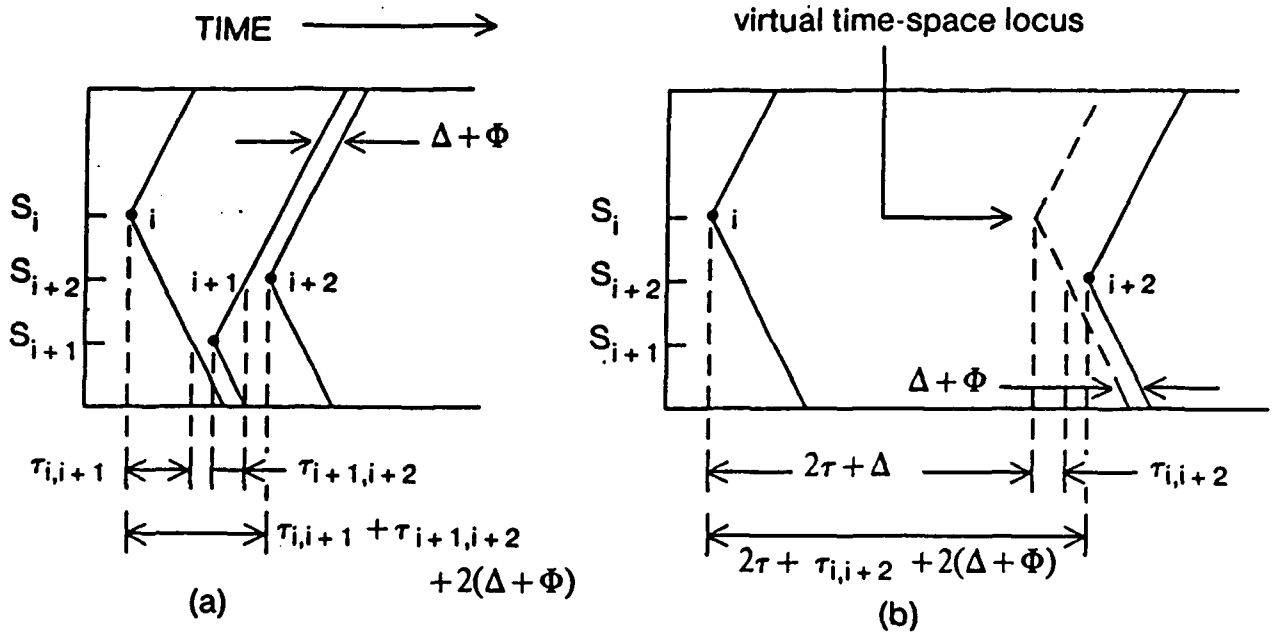


Fig. 5 The effect on the overhead of a station missing its turn to transmit in BRAM. In (a) the three station S_i, S_{i+1} and S_{i+2} all transmit. In (b) S_i and S_{i+2} transmit while S_{i+1} misses its turn.

These two cases are shown in Fig. 5. The effect of the missing transmission is to cause a *virtual time-space locus* for EOC(i) which is delayed in time by $2\tau + \Delta$ from the actual one. The interesting point is that, by missing a turn, S_{i+1} has not only reduced the total number of transmissions in a round but in addition has caused an increase in the overhead in this round. More generally, consider the case where N out of M stations are continuosly backlogged with packets to be transmitted. We note that $Y(M,N)$ depends on the stations' layout and the particular choice of the subset of backlogged stations. The maximum possible value is $Y(M,N) = N\tau + (M-N)(2\tau+\Delta) + N(\Delta+\Phi)$ giving $C(M,N) = 1/[1+\omega+\delta+\phi+a+\frac{M-N}{N}(2a+\delta)]$. The minimum value is $Y(M,N) = (M-N)2\tau + \Delta + N(\Delta+\Phi)$ giving $C(M,N) = 1/[1+\omega+\delta+\phi+\frac{M-N}{N}(2a+\delta)]$.

Comments: BRAM accommodates all layouts without requiring excessive knowledge by each station of the layout, paying a price in performance. The algorithm is entirely distributed. However, the original description of it in [7] fails to address important issues pertaining to the loss of the synchronizing event EOC in the event that all stations become temporarily idle, nor does it describe how the algorithm gets started. The robustness of the scheme is furthermore dependent on the ability to properly and accurately decode the index of each transmitting station by all stations in the network. Other schemes discussed in this section address these issues, (and their solutions certainly can be applied to BRAM) and provide improved performance. Nevertheless, BRAM and its cousins MSAP and MSRR (Kleinrock, Scholl, 1977) [8] which bear great resemblance to BRAM, are among the first conflict-free algorithms for distributed broadcast networks. In the original description of BRAM the detection time Δ and processing time Φ are ignored which in effect assumes that they are zero. It can be shown that under some conditions this leads to erroneous operation of the access scheme. In addition, in that description the stations scheduling delays are staggered by τ instead of $2\tau+\Delta$. Such a scheduling delay would work only in a network where $\tau_{i,j} = \tau \; \forall i,j \; i \neq j$. Obviously such a restriction is not desirable. In our opinion, the scheduling delay function given in eq. (1) is correct and complete.

B) SOSAM (Gold, Franta, 1982) [9]: This scheme, called the source synchronized access method

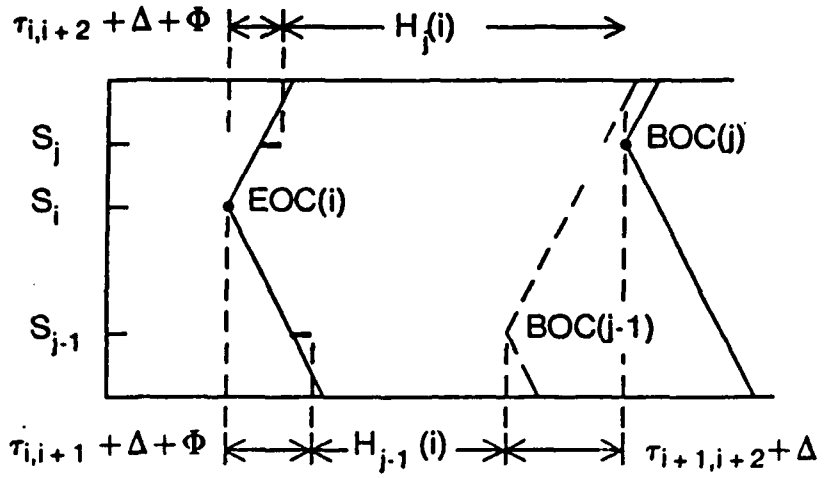


Fig. 6 Time-space diagram showing the recursive nature of the computation of $H_j(i)$ in SOSAM

(SOSAM), is similar to BRAM in that it requires no correspondence between the stations' indices and their locations, and achieves the same performance when all stations are backlogged. It provides however improved performance when stations miss their turns. To accomplish that, all stations must have explicit knowledge of the propagation delay between every pair of stations. Given this knowledge, S_j can determine $H_j(i)$ the minimum time required after detecting S_i's EOC required to detect a potential transmission from S_{j-1} and set $H_j(i)$ to this time. In particular, $H_{i+1}(i) = 0$. In the general case $H_j(i)$ can be computed recursively. That is, given $H_{j-1}(i)$, S_j can determine $H_j(i)$ in terms of $H_{j-1}(i)$ and the topological information consisting of the propagation delays between stations. Let BOC(i) denote the event corresponding to the beginning of carrier from S_i. Consider a transmission for S_i. With reference to Fig. 6, S_{j-1} detects EOC(i) at time $t_0 + \tau_{i,j} + \Delta$ and evaluates $H_{j-1}(i)$ by time $t_0 + \tau_{i,j-1} + \Delta + \Phi$. Were S_{j-1} to transmit it would do so after a scheduling delay of $H_{j-1}(i)$. In this case, BOC($j-1$) would be detected by S_j at time $(t_0 + \tau_{i,j-1} + \Delta + \Phi) + H_{j-1}(i) + (\tau_{j-1,j} + \Delta)$ which is the time at which S_j should schedule its transmission. Since S_j detects the synchronizing event EOC(i) at time $t_0 + \tau_{i,j} + \Delta$ and takes Φ sec. to compute $H_j(i)$, the latter can be defined recursively by

$$H_j(i) = H_{j-1}(i) + \tau_{i,j-1} + \tau_{j-1,j} - \tau_{i,j} + \Delta. \tag{5}$$

As with BRAM, the overhead in a round, $Y(M,N)$ for SOSAM, varies depending on the relative locations of the stations on the bus and their logical ordering, and this can take on a range of values depending on the topology. However, for a given configuration, this overhead is constant regardless of how many or which stations transmit in the round, and is computed as in eq. (4). Given $Y(M,N)$, the network capacity $C(M,N)$ and delay $D(M,N)$ can easily be derived. (See eq. (2) and (3).)

Comments: (i) To gain this improvement in performance over BRAM, in SOSAM S_k must store in its network interface either all the inter-station propagation delays $\tau_{i,j}$ or else precomputed values of the scheduling delay $H_j(i)$ $\forall i$. Either option requires substantial memory if the network is large (say 1000 stations). Also this memory would have to be updated at every station every time one is moved, added to, or removed from the network. This makes the task of maintaining such a network a difficult one. (ii) It was indicated in BRAM that the synchronizing event, EOC, is lost and the network stalls if all stations are idle at the time that their respective scheduling delays expire. SOSAM implements a mechanism to prevent this. If a station is idle when its scheduling delay expires, that station resets the latter to some predetermined constant which is larger than any scheduling delay, thereby maintaining the staggering of the potential transmission times. Clearly, the smallest constant that can be

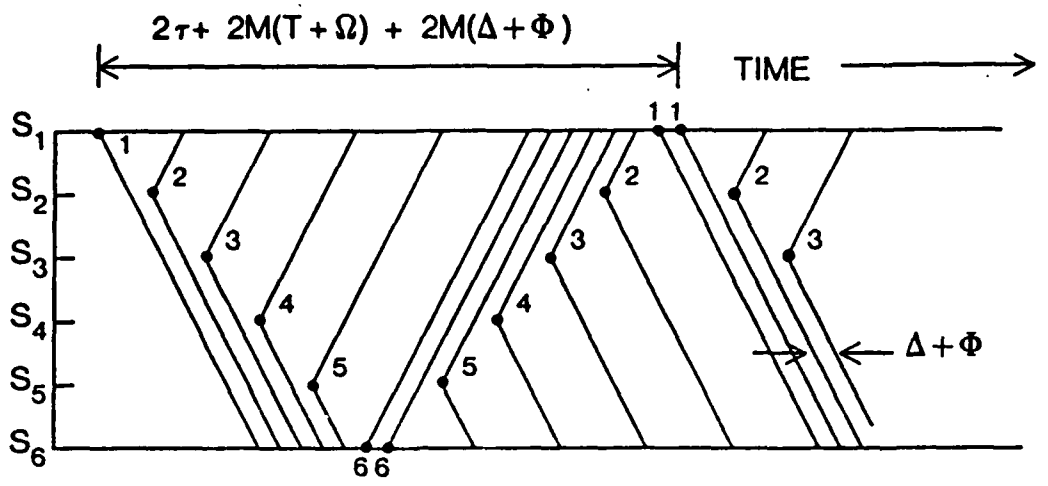


Fig. 7 Time-space diagram for BID showing the activity on the channel under heavy traffic conditions.

used is $\max_j\{H_j(i)\}+\Delta = H_i(i)+\Delta$. Furthermore, it can be shown that $H_i(i) = Y(M,M)$ and hence is independent of i. While this ensures that the network will operate under zero load, the robustness of the scheme nevertheless depends on the ability to correctly decode the index of each transmitting stations.

C) **BID (Ulug, White, Adams, 1981) [10]:** We indicated in SOSAM that, just as in BRAM, the overhead is minimized and the network capacity maximized by numbering the stations such that their logical order is the same as their physical order on the bus. In this case $\tau_{i,j-1}+\tau_{j-1,j} = \tau_{ij}$, and the scheduling delay function given for SOSAM becomes $H_j(i) = H_{j-1}(i)+\Delta = (j-i-1)\Delta$ which is independent of the propagation delays between stations. In fact, this is in essence what the scheduling delay in BID, a predecessor of SOSAM, is. We now describe those features specific to BID. The stations are numbered 1 through M as shown in Fig. 1. The end stations (S_1 and S_M) perform a special function called the *start-of-round*. A round or cycle is the time between two consecutive start-of-rounds. Each station on the network is allowed to transmit only once during a given round. The order in which stations are allowed to transmit, however, varies from round to round. In a *left-to-right round*, backlogged stations transmit in turn starting with S_1 and ending with S_M. In a *right-to-left round*, backlogged stations transmit in turn starting with S_M and ending with S_1. Each station can determine the "direction" of the current service order by an indicator which is transmitted along with the index number in each packet. Suppose that round r is a left-to-right round. Round r ends with the end of S_M's transmission, if S_M were backlogged in this round, or at the time S_M would have transmitted if it were idle. At this time S_M initiates round $r+1$ by transmitting a packet which has the direction indicator set to "right-to-left". If S_M is backlogged this packet would be a data packet, if S_M is idle this packet would be a start-of-round or token packet. By symmetry, round $r+2$ is initiated by S_1 at the end of round $r+1$. From the direction indicator and the index number of the transmitting station, each station can compute the scheduling delay as

$$H_j(i) = \begin{cases} (j-i-1)\Delta & \text{left-to-right round and } j > i \\ (i-j-1)\Delta & \text{right-to-left round and } j < i \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

In Fig. 7 we show a time-space diagram of the activity on the channel in BID. Due to the nature of the order of transmissions within a round, and by reversing this order from round to round, the overhead is clearly minimized. Ignoring the overhead due to a potential start-of-round packet from either S_1 or S_M, the overhead over *two* rounds in BID is $2\tau + 2M\Delta + 2N\Phi$ giving a network capacity of $C(M,N) = 1/(1+\omega+\phi+\frac{M}{N}\delta+a/N)$. We have assumed here

that there is a gap of $\Phi + \Delta$ between the two consecutive transmission of an end station. The Φ accounts for the time taken for the end station to determine that it should transmit again and the Δ is the delay required so that other stations can distinguish the two transmissions. It is possible that the processing be completed during the transmission time of the first of the two transmissions. In this case the overhead over the two rounds will be reduced by 2Φ. The delay $D(M,N)$ as defined in section II, is the delay incurred by a packet while at the head of the queue at each station plus the transmission time of that packet. Since in BID the order of service is reversed from round to round, $D(M,N)$ varies with each station and with the direction of the sweep. Bounds on $D(M,N)$ are given by the packet delay at the end stations where, normalized to T, $D(M,N)$ has the values $D(M,N) = 1 + \omega + \delta + \phi$ and $D(M,N) = 2N(1 + \omega + \phi) + 2M\delta + 2a$. For any other station $D(M,N)$ lies between these two values.

Comments: (i) In BID, no knowledge of τ or $\tau_{i,j}$ is required. The logical ordering of stations is the same as their physical order, BID is able to achieve a performance which is almost independent of τ if N is sufficiently large. However, this restriction on the ordering of stations makes it difficult to add stations to the network or move existing ones. (ii) BID is partially centralized in that end stations are required to initiate new rounds. As a result the network is robust in the sense that synchronizing events are periodically generated even when all stations are idle, and one end station can initiate a new round if the index number or direction indicator of a transmission cannot be decoded. In the event of an end station failure the adjacent station can assume the functions of the end station. In the case that stations $M, M-1, \ldots, i+1$ all fail then S_i will perform the functions of the end station on the right. This is accomplished in the following way. Consider a round in which the service is from left-to-right. S_i will determined that stations $M, M-1, \ldots, i+1$ have failed if it does not detect any activity on the network either due to a packet or due to a round-start token for a sufficiently long time after it has had its turn in this round. This time-out period is determined as follows. Consider the time reference at S_k to be either the end of its transmission (or the time that it would have transmitted if it were idle) S_k will expect to have detected a round start token from S_M $2\tau_{i,M} + (M - i + 1)\Delta$ sec. later. Allowing another $(M - i - 1)\Delta$ for each of the stations $M-1, M-2, \ldots, i+1$ to possibly start the new round, S_k must detect no activity for $2\tau_{i,M} + 2(M - i)\Delta$ to determine that stations $M, \ldots, i+1$ have failed. By symmetry one can determine the appropriate time out to determine that the stations on the left have all failed. To simplify the installations of the network the quantity $2(k-1)\Theta$ can be substituted for $2\tau_{1,k}$ where Θ is a constant greater than the maximum propagation delay between adjacent stations. In the description of BID the latter approach has been adopted. However, in that description, Δ has been ignored and so, in order that the preceding algorithm be correct, Θ must include Δ. Except for this recovery algorithm, no knowledge of τ or $\tau_{i,j}$ is required in BID. (iii) By having the end stations alternately initiate the rounds, the order in which stations are served within a round is reversed from round to round. While this improves the network capacity, it means that the upper bound on $D(M,N)$ is given by the length of two rounds as opposed to the length of one as in BRAM and SOSAM where the service order is fixed. However the average value is the length of one round and is less than that of BRAM and SOSAM due to the reduction in the overhead.

D) **Silentnet (Jensen, Tokoro, Sha, 1980) [11]:** Silentnet is similar to BID in that stations' logical ordering is the same as their physical order on the bus, and thus can apply the same efficient scheduling delay function. Like BID, the order in which stations are serviced reverses from round to round. The distinction in Silentnet is the distributed, as opposed to centralized, mechanism used to initiate a round. While BID makes use of the end stations for this purpose, in Silentnet, this functionality is part of the scheduling delay function. While in this system there are no explicit start-of-round events, we nevertheless define a round to be the sequence of transmissions which are in either left-to-right order or right-to-left order. Consider a round in which the service order is from left-to-right. For $j > i$, $H_j(i)$ is computed as is done in BID. For $j \leq i$, S_j has already had its turn in the current round and schedules its next transmission for

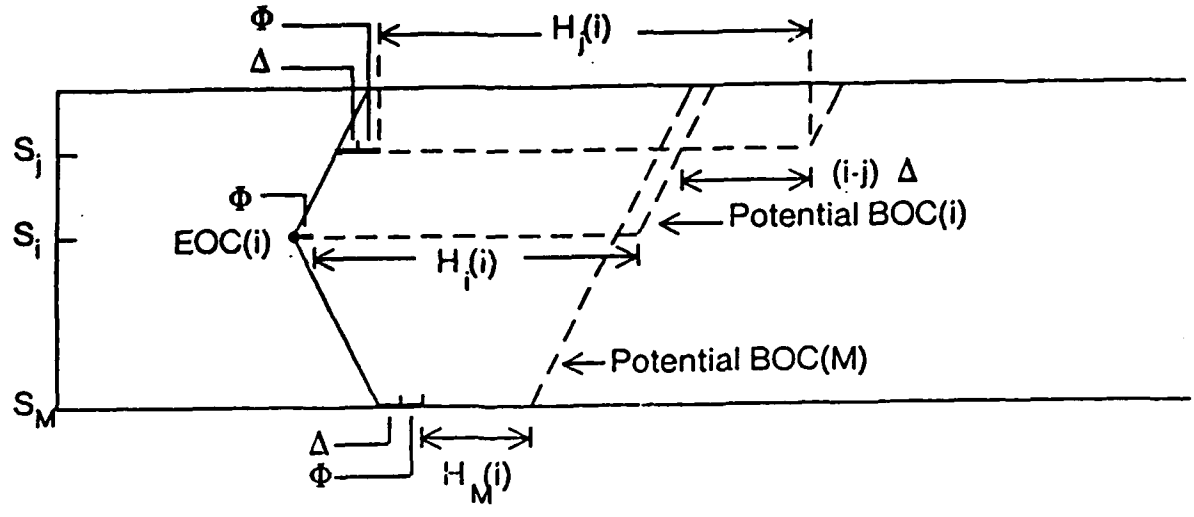


Fig. 8 Time-space diagram for Silentnet showing the relationship between the scheduling delays $H_i(i)$, $H_M(i)$ and $H_j(i)$, $j < i$.

a time when it exclusively can begin the next round. We now show how this time is evaluated. The reader is referred to Fig. 8. Suppose that S_i were the last station to transmit in this round. Were some other station S_k, $i < k \leq M$, to transmit after S_i, the latter would detect BOC(k) $\Delta + \Phi + (k-i)\Delta + 2\tau_{i,k}$ sec. after completing its transmission. Thus, if S_i detects no activity for $\Delta + \Phi + (M-i)\Delta + 2\tau_{i,M}$ sec. after it completes its transmission then none of the stations S_k, $i < k \leq M$ are backlogged; hence S_i can transmit at this time (beginning the new round in which the service order will be from S_i to S_1). If we assume that it takes S_i Φ sec. to re-evaluate its scheduling delay at the end of its own transmission, $H_i(i)$ for a left-to-right round is $H_i(i) = 2\tau_{i,M} + (M-i+1)\Delta$. Given this potential BOC(i), stations S_j, $j < i$, must stagger their potential transmission times appropriately. Thus, synchronizing to the actual event EOC(i), the scheduling delay for S_j, $j < i$, is $H_j(i) = (H_i(i) - \Delta) + (i-j)\Delta$. The scheduling delay for right-to-left round can be deduced by a symmetrical argument. Thus $H_j(i)$ can be computed as

$$H_j(i) = \begin{cases} (j-i-1)\Delta & j > i \\ 2\tau_{i,M} + (M-i+1)\Delta & j = i \\ 2\tau_{i,M} + (M-j)\Delta & j < i \end{cases} \quad ; \quad H_j(i) = \begin{cases} (i-j-1)\Delta & j < i \\ 2\tau_{1,i} + i\Delta & j = i \\ 2\tau_{1,i} + (j-1)\Delta & j > i. \end{cases} \tag{7}$$

left-to-right round — right-to-left round

Using this scheduling delay the performance of Silentnet is identical to BID. If, however, at the cost of some efficiency, one desires that the scheduling delay be independent of the stations location on the network, one could replace $\tau_{i,M}$ in eq. (7) by τ. This is in fact what is done in the original description of Silentnet (however the more efficient version described above is presented as a variation). In this case an additional overhead of 2τ is incured between consecutive rounds leading to a network capacity of $C(M,N) = 1/(1+\omega+\phi+\frac{M}{N}\delta+3a/N)$.

Comments: (i) Three variants of Silentnet are presented in the original description [11]. The first, called the "basic algorithm" is the one described above but with $\tau_{i,M}$ replaced by τ. The second, called the "distance algorithm" (since each station must have knowledge of its distance from each end of the network) is the one described above and its performance is superior to that of the basic algorithm. In the third, called "the see-saw algorithm" the start-of-round function is assigned to the end stations. This variant of Silentnet is identical to BID. (ii) As in SOSAM, in Silentnet a mechanism is provided to maintain the synchronizing event EOC(i) when the network load is close to zero. This is achieved by having the last station to have transmitted, if idle, set its scheduling delay to a constant sufficiently large such that it will have detected a

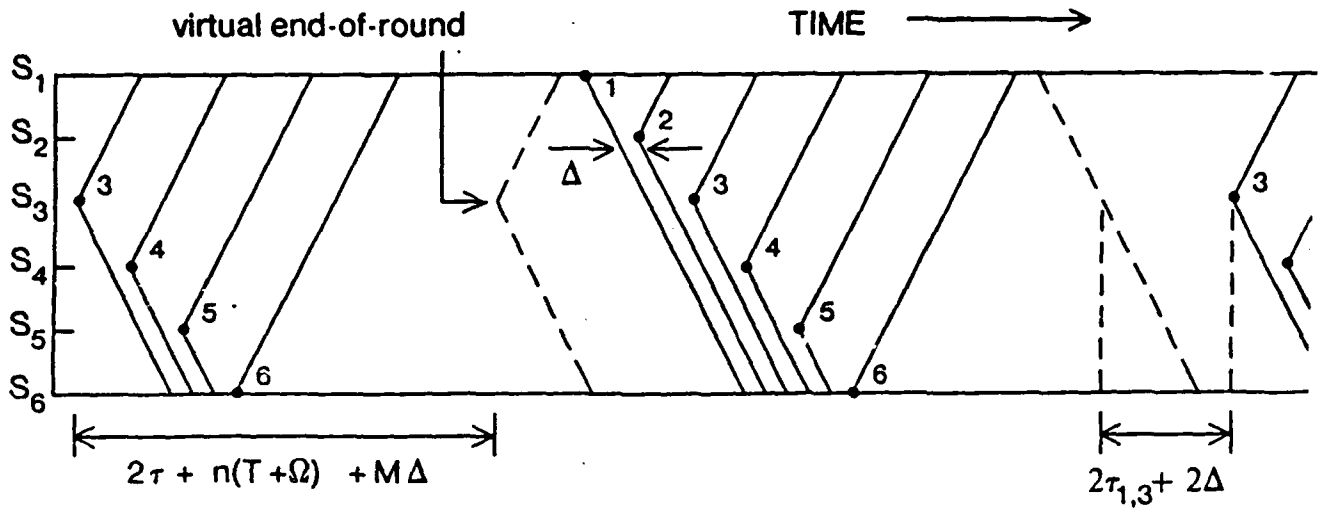


Fig. 9 Time-space diagram for L-Expressnet showing typical activity on the channel over three rounds. The first is begun by S_3, the second by S_1 and the third by S_3.

transmission by any backlogged station before this scheduling delay expires. If the scheduling delay does expire a dummy packet is transmitted, thereby regenerating the event EOC(i). If some other station transmits then that station becomes the last to have transmitted. The minimum value of the constant is given by $\max_{i,j}\{H_j(i)\} + 2\tau$ which is $4\tau + M\Delta$. In the original scheme, $H_i(i)$ is given by this amount regardless of whether S_i is backlogged or idle. As a result the last station to transmit in a given round misses its turn in the next leading to an unfair scheduling function. The scheduling delay function in eq. (7) overcomes this limitation in the original scheme and provides fair service to all stations.

E) **L-Expressnet (Borgonovo, Fratta, Tarini, Zini, 1983) [12]:** Like Silentnet, L-Expressnet also implements a distributed version of BID but, in this scheme, the order in which stations transmit is always from left to right. Define a round to be a sequence of transmissions in left-to-right order. The distributed mechanism for starting a round in L-Expressnet identifies the leftmost of the participating stations which then begins the round by transmitting a start-of-round token. The EOC of the start-of-round token (EOC(token)) serves as the time reference to which stations synchronize their transmissions. To determine when to transmit, S_j counts the time that the channel is idle starting from the last EOC(token). When this cumulative time reaches $H_j(i) = (j-1)\Delta$, S_j may transmit. Note that, for each station, $H_j(i)$ is a constant and is not re-evaluated after each transmission. As a result, no processing overhead is incurred. This EOC(token) also serves as time reference to determine when to start the next round as follows. All stations count for a cumulative idle time of $2\tau + M\Delta$ as measured from the event EOC(token). As can be seen in Fig. 9, this creates a virtual time reference which has the property that it occurs after the last transmission in the round. (A tight time reference, i.e., one which corresonds exactly to the potential event EOC(M), could be achieved but this would require knowledge of the $\tau_{i,j}$'s and the index number of the station that transmitted the start-of-round token.) Using this virtual time reference, S_k schedules a time at which to transmit the next start-of-round token using the same approach as the procedure to recover from an end station failure in BID. That is, if S_k fails to detect activity within $2\tau_{1,k} + 2(k-1)\Delta$ sec. as measured from this virtual time reference, it transmits the start-of-round token. In the description of L-Expressnet in [12], this scheduling delay is computed as $2(k-1)\Theta$ where, as in BID, Θ is a constant greater than the maximum propagation delay between adjacent stations and, (although not mentioned in this description,) must include an amount Δ for the detection time. Obviously, the overhead for this scheme, neglecting the token, is $Y(M,N) = 2\tau + M\Delta + 2(k-1)\Theta$ where S_k is the leftmost of the participating stations. Thus $C(M,N)$ and $D(M,N)$ can be easily computed from eqs. (2) and (3).

Comments: (i) In L-Expressnet, each station uses a fixed scheduling delay which is a function only of that stations index number. Hence no need exists to read the index of each transmission. In addition, $H_j(i)$ is not re-evaluated at each EOC and therefore no overhead due to processing is incurred. (ii) In L-Expressnet all stations participate in the start-of-round procedure. As a result, the scheme is robust in the sense that the synchronizing event EOC(token) is continuously regenerated. (iii) While, in general, stations take Δ sec. to detect EOC(token) plus Φ sec. to recognize it as such, the station that transmits the start-of-round token does not incur this overhead. Thus there is a discrepancy in the time references between this station and the rest which has been overlooked in the description of L-Expressnet. This can be compensated for by the former having transmitted the start-of-round token, delaying any further activity for $\Delta + \Phi$ sec.

IV. CONCLUSION

From the numerous implicit-token DAMA schemes that have recently been proposed, we have identified three basic access mechanisms according to which it has been possible to classify them. These are the scheduling delay access mechanism, the reservation access mechanism, and the attempt-and-defer access mechanism. In this paper we described these access mechanisms and discussed those schemes that use the scheduling-delay access mechanism. With this classification, it was possible to present the network schemes belonging to this class in a unified manner, showing their similarities, and identifying their differences. We have shown that some schemes can be seen as variations or special cases of others, and that a feature proposed for one could, in some cases, be applied to another. For each scheme, we identified the overhead due to the distributed access protocol and presented some performance results in terms of the network capacity and packet delay. We showed that, using an appropriate scheduling delay function, some of these schemes are able to achieve a network capacity which is almost independent of a if M is sufficiently large.

REFERENCES

[1] W. Stallings, "Local Network Performance," *IEEE Communications magazine*, vol. 22 no. 2, February 1984.

[2] D. D. Clark, K. T. Pogran, D. P. Reed, "An introduction to local area networks," *Proceedings of the IEEE*, vol. 66, no. 11, Nov. 1978.

[3] K. Kummerle and M. Reiser, "Local area communication networks–an overview," *Journal of Telecommunication Networks*, vol. 1, no. 4, Winter 1982.

[4] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, no. 7, pp. 395–403 (1976).

[5] F. Tobagi and V. Bruce Hunt, "Performance analysis of carrier sense multiple access with collision detection," *Computer Networks*, vol. 4, no. 5, Oct/Nov 1980.

[6] *IEEE Project 802 Local Area Network Standards*, Draft D 802.4, Token-Passing Bus Access Method and Physical Layer Specification, IEEE Computer Soc., Silver Spring, MD, 1983.

[7] I. Chlamtac, W. Franta and K. D. Levin, "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.*, Vol. COM-27, No. 8, August 1979, pp. 1183-1190.

[8] L. Kleinrock and M. Scholl, "Packet switching in radio channels: New conflict-free multiple access schemes for a small number of data users," in *Proc. of the International Conf. on Communications*, Chicago, IL, June 1977.

[9] Y. I. Gold and W. R. Franta, "An efficient scheduling function for distributed multiplexing of a communication bus shared by a large number of users," in *Proc. of the International Conf. on Communications*, Philadelphia, June 13-17, 1982.

[10] M. E. Ulug, G. M. White, W. J. Adams, "Bidirectional token flow system," in *Proceedings of the 7th Data Communications Symposium*, Mexico City, October 1981.

[11] E. D. Jensen, M. Tokoro, L. Sha, "Bus allocation scheme for distributed real time systems," *Carnegie-Mellon University Report*, December, 1980.

[12] F. Borgonovo, L. Fratta, F. Tarini, P. Zini, "L–Express-net: A communication protocol for local area networks," in *Proceedings INFOCOM '83* San Diego, April 1983.

[13] J. W. Mark, "Distributed scheduling conflict-free multiple access for local area communications networks," *IEEE Trans. Commun.*, vol. COM-28, pp.1968-1976, Dec. 1980.

[14] K. P. Eswaran, V. C. Hamacher, G. S. Shedler, "Collision-free access control for computer communication bus networks," *IEEE Transactions on Software Engineering*, vol. SE-7, no. 6, Nov. 1981.

[15] K. P. Eswaran, V. C. Hamacher, G. S. Shedler, "Asynchronous collision-free distributed control for local bus networks," IBM Research Report RJ2482, San Jose, Ca, 1979.

[16] L. Fratta, "An improved protocol for data communication bus networks with control wire," in *Proceedings SigComm*, 1983.

[17] F.A. Tobagi and R. Rom, "Efficient round-robin and priority schemes for unidirectional broadcast systems," *Proceedings of the IFIP 6.4 Zurich Workshop on Local Area Networks*, Zurich, Switzerland, August 27-29 1980.

[18] R. Rom and F.A. Tobagi, "Message-based priority functions in local multiaccess communications systems," *Computer Networks*, vol. 5, no. 4, July 1981, pp. 273-286.

[19] L. Fratta, F. Borgonovo and F. A. Tobagi, "The Express-net: A local area communication network integrating voice and data", in *Proceedings of the International Conference on the Performance of Data Communication Systems and Their Applications*, Paris France, 14–16 September 1981.

[20] F. Tobagi, F. Borgonovo, L. Fratta, "Express-net: A high-performance integrated-services local area network," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 5, Nov. 1983.

[21] C. Tseng and B. Chen, "D-Net, a new scheme for high data rate optical local area networks," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 3, April 1983.

[22] J. O. Limb and C. Flores, "Description of Fasnet, a unidirectional local area communications network," *Bell Systems Technical Journal*, September 1982.

[23] John O. Limb, "Fasnet: A proposal for a high speed local network," in *Proc. of Office Inform. Sys. Workshop*, St. Maximin, France, Oct. 1981.

[24] M. Gerla, C. Yeh, P. Rodrigues, "A token protocol for high speed fiber optics local networks," in *Proceedings Optical Fiber Communication Conference*, New Orleans, Louisiana, February 1983.

[25] P. Rodrigues, L. Fratta, M. Gerla, "Token less protocols for fiber optics local area networks," submitted to *ICC '84*.

[26] M. A. Marsan and G. Albertengo, "Integrated voice and data network," *Computer Communications*, vol. 5, no. 3, June 1982.

[27] A. Takagi, S. Yamada, S. Sugawara, "CSMA/CD with deterministic contention resolution," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 5, Nov. 1983.

[28] M. Gerla, P. Rodrigues, C. Yeh, "BUZZ-NET: A hybrid random access/virtual token local network," in *Proceedings Globecom '83*, San Diego, CA, December 1983.

APPENDIX III.

F. A. Tobagi, "Analysis of a Two-Hop Centralised Packet Radio Network—Part I: Slotted ALOHA," *IEEE Transactions on Communications*, Vol. COM-28, pp. 196–207, February 1980.

F. A. Tobagi, "Analysis of a Two-Hop Centralised Packet Radio Network—Part II: Carrier Sense Multiple Access," *IEEE Transactions on Communications*, Vol. COM-28, pp. 208–216, February 1980.

F. A. Tobagi and J. Brásio, "Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes," *Proceedings of INFOCOM '83*, San Diego, April 1983.

J. M. Brásio and F. A. Tobagi, "Theoretical Results in Throughput Analysis of Multihop Packet Radio Networks," in *Proceedings of the International Conference on Communications, ICC'84*, Amsterdam, The Netherlands, May 1984.

Analysis of a Two-Hop Centralized Packet Radio Network—Part I: Slotted ALOHA

FOUAD A. TOBAGI, MEMBER, IEEE

Abstract—**The design of packet radio systems involves a large number of design variables that interact in a very complex fashion. As this design problem in its general form is quite complex, a viable approach is to analyze some simple but typical configurations in an attempt to understand the behavior of these systems. In this paper, a two-hop centralized configuration is considered in which traffic originates at terminals, is destined to a central station, and requires for its transport the relaying of packets by store-and-forward repeaters. The throughput-delay performance is derived, and its dependence on such key system variables as the network topology, the transmission protocol, and the repeaters' storage capacities, is given. In this part, devices are assumed to be utilizing the slotted ALOHA access mode. Carrier sense multiple access is treated in Part II of this series [1].**

INTRODUCTION

THE economic sharing of computer resources has been made possible by the development of the packet-switching technique whereby packet switches are interconnected by *point-to-point data circuits* according to some topological structure [2]-[4]. Economic studies have subsequently shown that, for geographically distributed networks, a significant part of the overall system cost is incurred by the local collection of data from, or dissemination of data to, a large population of users [5]. Today, with the proliferation of computer applications, computer resources have to be brought increasingly close to the individual; this makes it extremely desirable to create more flexible and more economic communication techniques. The *packet-broadcasting* technique offers an attractive solution in that it brings together the advantages of both packet switching and broadcast communication. Packet switching offers the fair and efficient sharing of the communication resources by many contending users with unpredictable demands; the (radio) broadcast medium is a readily available resource, is easily accessible and particularly suitable for communication with mobile users. The ALOHA system at the University of Hawaii, a packet-switched computer communication system utilizing radio, is perhaps the first example illustrating the feasibility of this technique [6]. Originally, the ALOHA system was a one-hop system whereby all terminals are in line-of-sight and within range of the central computer (the station). Later on, packet repeaters were added to provide expansion of geographical coverage beyond the range of the station [7]. Another prominent example is typified by the Packet Radio system of the Defense Advanced Research Projects Agency [8], [9]. The target requirements of the system are more ambitious than with the ALOHA system and include many added features such as direct communication by a ground radio network between users over *wide* geographical areas, coexistence with possibly different systems in the same frequency band, antijam protection, etc. The key requirement of direct communication over wide geographical areas renders the repeaters integral components of the system.

The design of packet radio systems involves a large number of design variables which interact in a very complex fashion [8]-[13]. In summary these are: the *network topology*, which consists of the number of devices and their geographical setting; the *bandwidth management*, that is the allocation of the available bandwidth as dedicated channels, or high-speed channels to be shared by many users, or a mixture of these two modes; the *channel-access policy*, which is particularly crucial when we are in presence of shared channels and can consist of either a centrally controlled scheme or some random-access mode; the *modulation scheme*, which can be of the spread spectrum type or one of the more conventional narrow-band modulation schemes; the *operational protocols* which consist of the routing algorithms, the error control procedures, the flow control protocols and the monitoring functions required for the operation of the network; and finally the *nodal design*, that is the storage capacity required at each node, the buffer management strategy, the power requirement, and the nodal processing speed.

In its general form, the optimum solution is extremely hard to come by. However, it is often the case that the selection of some system parameters is dictated by physical constraints. For example, for rapid deployment in military applications, and for easy communication among mobile terminals, it is advantageous that all devices employ omnidirectional antennas and share a single high-speed channel. In fact a great advantage is gained by providing the available communication bandwidth as a single high-speed channel to be dynamically multiaccessed by the many devices; this advantage is due to the statistical load averaging. With these arguments we have somewhat decreased the space of design variables, and need to focus only on packet radio systems with the above characteristics. This task, however, is still of a very high caliber. One of two alternatives are present; either we create a simple but crude and approximate model suitable for general network configurations, or we analyze more accurately simple but typical configurations as a first attempt to understand the behavior

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication after presentation at the National Telecommunications Conference, Los Angeles, CA, December 1977. Manuscript received May 31, 1978; revised September 4, 1979. This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract MDA-903-77-C-0272 and Contract MDA-79-C-0201.

The author is with the Computer Systems Laboratory, Stanford University, Stanford, CA 94305.

0090-6778/80/0200-0196$00.75

of these systems, and to derive their performance. In this paper we opt for the latter approach.

II. NETWORK CONFIGURATIONS UNDER CONSIDERATION

A number of papers have already appeared in the literature that study various simple network topologies. Single-hop networks where terminals communicate directly with each other or with a central station have been investigated extensively [14]-[19]. A two-hop configuration involving a ring of repeaters around a station has been analyzed by Gitman [20]; network capacity was studied, but packet delay was not considered.

Here we consider again two-hop centralized configurations in which traffic originates at terminals, is destined to a central station, and requires for its transport that packets be relayed by store-and-forward repeaters. The basic performance measure sought is the throughput-delay tradeoff and its dependence on such key system parameters as the network topology (i.e., the number of repeaters and their connectivity pattern), the repeaters transmission policy, and their storage capacity. Two random-access schemes are considered: the slotted ALOHA scheme [7], [21], [22] studied in this part, and the nonpersistent carrier sense multiple-access scheme (CSMA) [14] analyzed in Part II [1].

All devices are provided with omnidirectional antennas and employ a random-access scheme over a single shared channel. With each repeater is associated a population of terminals, in line-of-sight and within range of only that repeater. Traffic originates at terminals and is destined to the station; thus, we consider *inbound traffic* only. Each repeater is provided with a *finite storage* capacity which can accommodate a *maximum of M packets*. The station has an infinite storage capacity. Packets are all of a fixed size. When the transport of a packet over a hop is successful (i.e., the transmission is free of interference and storage is available at the receiving device), the packet is deleted from the sender's queue; otherwise, the packet incurs a retransmission delay. It is assumed here that a device learns about its success or failure at the end of transmission; that is, acknowledgments are assumed to be instantaneous and for free. At any one time, a device can be either transmitting or receiving, but not both simultaneously. The station always has its receiver on. The packet processing time at any device is considered to be negligible. As for the connectivity among repeaters, we consider here two types. The first, depicted in Fig. 1, is called the *star configuration*; in this, each repeater is in line-of-sight and within range of the station only. The second, depicted in Fig. 2, is called the *fully connected* (FC) configuration and consists of having all repeaters within range and in line-of-sight of each other and of the station.

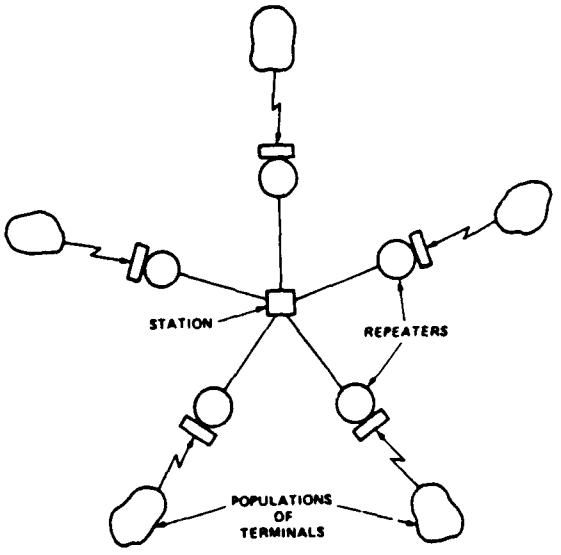


Fig. 1. A two-hop star configuration.

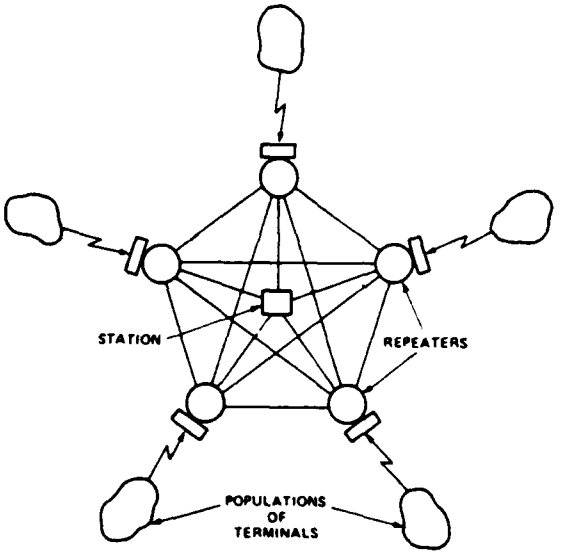


Fig. 2. A two-hop fully connected configuration.

III. ANALYSIS OF SLOTTED ALOHA SYSTEMS

We consider a universal time axis which is slotted into segments of duration equal to the transmission time of a packet. Each population of terminals is assumed to be infinite and to collectively generate new packets according to a Poisson distribution at a rate of s packets/slot. Terminals transmit their packets according to the slotted ALOHA scheme [22] (as described in Section III-A-3). Repeaters transmit their packets on a first-come-first-served basis; when its buffer is nonempty, a repeater transmits the head of its queue, in a slot, with probability p. With this protocol the first transmission of a newly received packet (at the repeater) incurs a geometrically distributed delay following its arrival at the head of the queue with mean $1/p$. We shall refer to this transmission protocol as the *delayed-first-transmission* (DFT) protocol. A slight variation of this transmission protocol, considered later in this section, consists of transmitting (with probability one) a newly received packet immediately following its arrival at the head of the queue. In case of an unsuccessful transmission the packet remains in the repeater's buffer and, as above, incurs the geometrically distributed delay. This protocol will be referred to as the *immediate-first-transmission* (IFT) protocol.

A packet successfully transmitted by a given population of terminals can be "blocked" at the immediate destination repeater; blocking is due to two factors: 1) the repeater (or any other repeater in the FC configuration) is in a transmit mode, or 2) the repeater's receiver is on (and in the FC configuration, all other repeaters are quiet), but the repeater's buffer is full. Due to the blocking of traffic at the receiving repeater and the need for retrials, the rate of successful transmissions of packets to a repeater from its corresponding population of terminals is actually greater than s and is denoted by λ. Furthermore, this process of packet arrivals to a repeater is assumed to be a Bernoulli one.[1]

Let N denote the number of repeaters present in the configuration. Given the transmission protocol adopted and given that the input process to each repeater is a Bernoulli process, the state of the system in slot t is entirely defined by the vector

$$\underline{n}^t = (n_1{}^t, n_2{}^t, \cdots, n_N{}^t)$$

where $n_i{}^t$ is the number of packets present in slot t at the ith repeater. Note that the arrival and departure of a packet are completed at the end of a slot. $\underline{n}^t$ includes packets in transmission, but does not include arrivals in process. It is clear that $\underline{n}^t$ is a Markov chain.

A. The Single Buffer Case, DFT Protocol

In this case, the state of the system can be equivalently described by the number of repeaters with nonempty buffers, referred to as the number of "active" repeaters. Let n^t, $0 \leq n^t \leq N$, denote that number in slot t.

1) *Star Configuration*: The Markov chain n^t for the star configuration has a transition matrix P whose (i, j)th element is given by

$$p_{ij} = \begin{cases} 0 & j < i-1 \\ P_s(i)(1-\lambda)^{N-i} & j = i-1 \\ [1-P_s(i)]\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j} & \\ \quad + P_s(i)\binom{N-i}{j-i+1}\lambda^{j-i+1}(1-\lambda)^{N-j-1} & \\ & j \geq i \end{cases} \quad (1)$$

[1] The validity of this assumption is demonstrated by simulation results which show that the process of packets successfully transmitted from a slotted ALOHA population of terminals approaches a Bernoulli one, especially when the system load is not too high. It is also substantiated by results obtained from a separate analytic study of the packet transport process from N repeaters (or terminals) to a station, $2 \leq N \leq 10$, contending on the same channel in a slotted ALOHA mode; the results show that this process can be approximated by a Bernoulli process for a large range of the system parameters N, λ, and p. The chi^2 value of a sample of 1000 interarrival times (at the station) is below 67, which corresponds to a level of confidence of over 99.5 percent, (degree of freedom = 100). This Bernoulli assumption is essential in the creation of the Markov chain model used in this analysis because of the underlying memoryless property.

where $P_s(i)$ denotes the probability of a successful transmission given i active repeaters and is expressed as

$$P_s(i) = ip(1-p)^{i-1}. \quad (2)$$

Let

$$\pi_i \triangleq \lim_{t\to\infty} \Pr\{n^t = i\}.$$

We compute the stationary distribution $\Pi = \{\pi_0, \pi_1, \cdots, \pi_N\}$ by solving *recursively* the system $\Pi = \Pi P$. Let $\bar{n}$ denote the average number of active repeaters. We have

$$\bar{n} = \sum_{k=0}^{N} k\pi_k. \quad (3)$$

Consider a packet successfully transmitted by a population of terminals. We denote by β the probability of blocking due to the repeater being in transmit mode, and by α the probability of blocking due to the buffer being full (and the repeater's receiver on). We have

$$\alpha = (1-p)^{\bar{n}}/N \quad (4)$$

$$\beta = p\bar{n}/N, \quad (5)$$

and the total probability of blocking is given by

$$B = \alpha + \beta = \bar{n}/N \quad (6)$$

The total network throughput, denoted by S, is defined as the rate of successful packets received at the station; it is given by

$$S = (N - \bar{n})\lambda. \quad (7)$$

The packet delay D is defined to be the time since the packet is originated at the terminal until it is successfully received at the station. We distinguish two components: 1) the *access delay* D_a, defined to be the time required for the packet to be correctly received at the repeater, and 2) the *network delay* D_n which consists of the time elapsed since the packet is accepted at the repeater until it is successfully received at the station. By Little's result, the average network delay is given by

$$D_n = \bar{n}/S. \quad (8)$$

2) *FC Configuration*: In the fully connected configuration, an arrival to a repeater in a slot will not be successfully received if any of the repeaters is actively transmitting in that slot. The transition matrix P is given by

$$p_{ij} = \begin{cases} 0 & j < i-1 \\ P_s(i) & j = i-1 \\ (1-p)^i(1-\lambda)^{N-i} + [1-(1-p)^i - P_s(i)] & \\ & j = i \\ (1-p)^i\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j} & j > i \end{cases} \quad (9)$$

Let β denote the probability that a terminal's transmission is blocked due to a transmission by *one or more* repeaters. Given that k repeaters are active, this probability is simply $1-(1-p)^k$. Removing the condition on k we get

$$\beta = 1 - \sum_{k=0}^{N} \pi_k (1-p)^k. \tag{10}$$

Let α denote the probability that a terminal's transmission is blocked due to the repeater's buffer being full, and that *no* repeater is transmitting. Given k active repeaters, this probability is simply $(k/N)(1-p)^k$, where

$$\binom{N-1}{k-1} \Big/ \binom{N}{k} = k/N$$

is the probability that a particular repeater R_i is active. Removing the condition we get

$$\alpha = \sum_{k=0}^{N} \pi_k \frac{k}{N} (1-p)^k. \tag{11}$$

The network throughout S is expressed as

$$S = \sum_{k=0}^{N} \pi_k \frac{k}{N} kp(1-p)^{k-1} = N\lambda(1-B) \tag{12}$$

and the network delay is simply given by (8).

3) *Access Delay*: To complete the delay analysis, we need to evaluate the access delay D_a for a given throughput S. Fig. 3 represents the state diagram for the population of terminals associated with a repeater. First, a terminal is in the thinking state. After a random period of time, the terminal generates *and transmits* a new packet. If the transmission is unsuccessful due to a collision with other contending terminals, the terminal joins the set of *colliding* terminals and reschedules transmission of its packet following a random retransmission delay, which we denote by X. The terminal retransmits its packet and repeats this process until its transmission is free of collision by other terminals. In the latter case, the packet will be successfully received at the repeater if and only if the repeater is not transmitting (as well as any other repeater in the FC configuration) *and* its buffer is not full; otherwise, the terminal joins the set of *blocked* terminals and reschedules transmission of its packet following the random retransmission delay. The process is repeated until the collision-free transmission of the packet is successfully received at the repeater, in which case the terminal rejoins the set of thinking terminals. It is clear from the diagram in Fig. 3 that the average access delay D_a is equal to the average time spent by a terminal in transiting from point A_1 to point A_5.

In the absence of blocking at the receiving device, slotted ALOHA in an infinite population environment has been analyzed by Kleinrock and Lam [22], [23]. The generation of new packets by the infinite population is modeled as a Poisson process with rate s packets/slot. The retransmission delay is considered to be uniformly distributed over K slots. Assuming that the blocking probability B is uniform over time, we modify Kleinrock and Lam's infinite population model to get the following equation:[2]

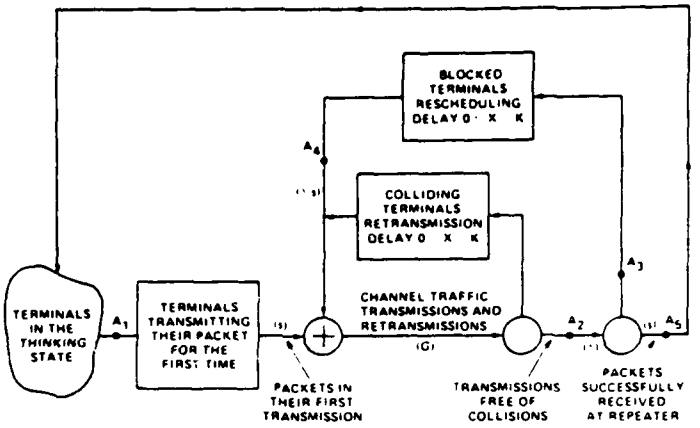


Fig. 3. State diagram for a population of terminals.

$$D_{S-\text{ALOHA}}(s, K, B) = 1 + \frac{E(K+1)}{2} + \tfrac{1}{2} \tag{13}$$

where

$$E = \frac{1-q_n}{q_t}$$

$$q_n = \left[e^{-G/K} + \frac{G}{K} e^{-G}(1-B) \right]^K e^{-s}(1-B)$$

$$q_t = \frac{e^{-G/K} - e^{-G}(1-B)}{1 - e^{-G}(1-B)} \left[e^{-G/K} + \frac{G}{K} \cdot e^{-G}(1-B) \right]^{K-1} e^{-s}(1-B)$$

$$s = G \frac{q_t}{q_t + 1 - q_n}.$$

Given S, the access delay is given by

$$D_a = \min_K D_{S-\text{ALOHA}} \left(\frac{S}{N}, K, B \right) \tag{14}$$

[2] Packet arrivals are not considered synchronized with slot boundaries, so that one-half of a slot is added to the access-delay equation. The CSMA scheme treated in [1] does not incur this additional synchronization delay.

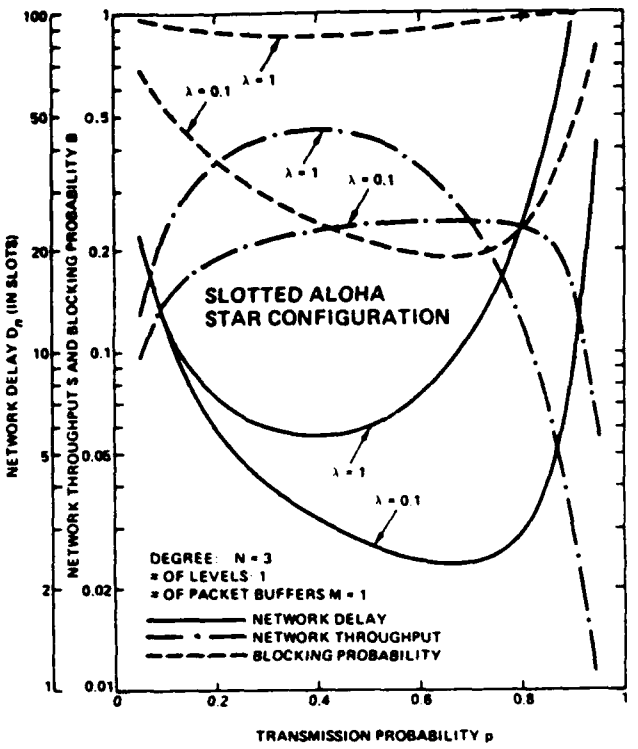


Fig. 4. Slotted ALOHA star configuration: Network delay, throughput and probability of blocking versus p.

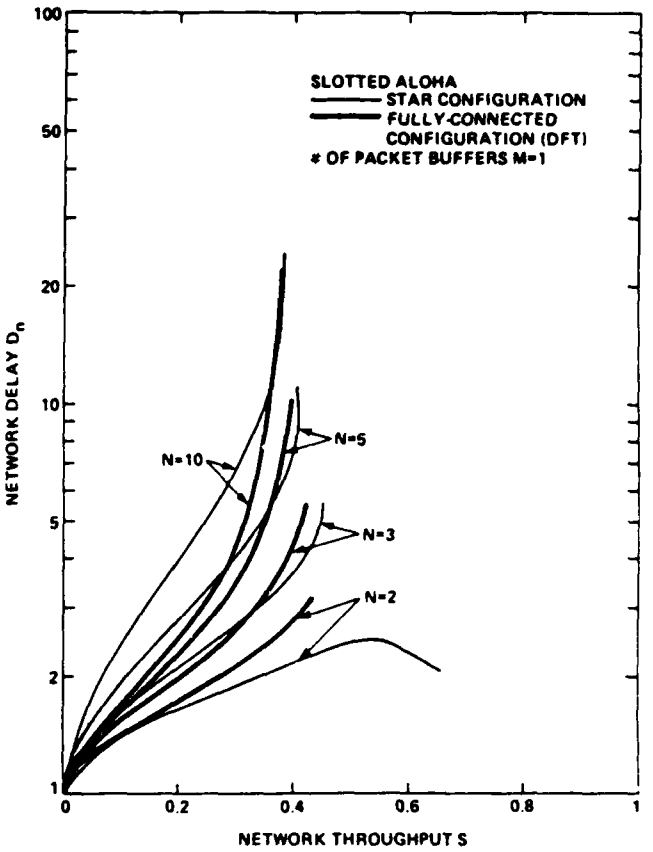


Fig. 5. Slotted ALOHA star and fully connected configurations: Optimum (network) throughput-delay curves.

The access delay can also be estimated by the following approximate formulas[3]

$$D_a \cong \frac{1}{1-B} D_{S-\text{ALOHA}}\left(\frac{S}{N(1-B)}, K_{\text{opt}}, 0\right) + \frac{B}{1-B}\frac{K_{\text{opt}}^{-1}}{2} + \tfrac{1}{2} \tag{14a}$$

where K_{opt} denotes the optimum retransmission delay minimizing $D_{S\text{-ALOHA}}(\lambda, K, 0)$.

4) *Numerical Results:* Consider first the *star* configuration. Fixing N and λ, we observe that $\bar{n}$ is a *convex* function of p. Thus there exists a value of p which minimizes $\bar{n}$. From (6), (7), and (8) we note that D_n and B are also convex functions of p while S is concave. Moreover, it is clear that the value of p which maximizes S, minimizes D_n and B. As an example we show, in Fig. 4, D_n, S, and B versus p for $N = 3$ and two values of λ. We observe that the throughput S is not as sensitive to p as are D_n and B. That is, if p is improperly tuned, while the system can maintain the throughput desired, the network delay D_n and the probability of blocking B (and thus the access delay) may suffer large increases! In Fig. 5, we plot the optimum delay versus the achieved throughput for various values of N. We note that the network delay increases with increasing values of N. The reverse behavior is observed for the probability of blocking B over a large range of $S(0 < S < 0.35)$ as shown in Fig. 6.

S, optimized with respect to p, is a monotonic function of λ;[4] the system capacity is achieved for $\lambda = 1/e$ and is expressed as $\max_p \{(N/e)(1 - B)\}$. In Fig. 7 we plot S (maximized over p with λ kept constant) versus λ for various values of N. The system capacity is precisely the network throughput at $\lambda = 1/e$. We note that for the larger values of $N(N \geq 3)$, S, which increases with increasing values of λ, levels off rather rapidly and approaches its maximum value for λ well below $1/e$. This is not so with the smaller values of N. Thus it is clear the the limiting hop is the terminal-to-repeater hop for $N = 2$ and 3, and the repeater-to-station hop for larger N. Moreover, we note that, for $N \geq 3$, the system capacity is a decreasing function of N. In Fig. 8 we plot the system capacity versus N for the two-hop configuration.

[3] We have compared numerical results for the access delay using both (14) and (14a). It was observed that (14a) was a good approximation for low throughput, but as the latter increased (and thus B increased), (14a) provided pessimistic results. For the sake of comparison with CSMA, Fig. 9 was plotted using (14a) since this approximation is the only available model for the access delay in CSMA.

[4] The maximum value of λ allowable in this model is a function of the access mode in use by the terminals. If a slotted ALOHA mode is used, it is well known that the maximum rate of successful packets that can be transmitted by an infinite population of terminals is $\lambda = 1/e = 0.368$. On the other hand, given the memoryless property of the Bernoulli input process, the above analysis corresponds also to the "linear-feedback" model whereby, following the successful transmission of its buffered packet, a repeater is assumed to generate a new packet after a geometrically distributed time with mean $1/\lambda$. In the linear-feedback model, the rate λ can take any value between 0 and 1. $B = \bar{n}/N$ represents the fraction of time a repeater is active; and D_n represents the total packet delay.

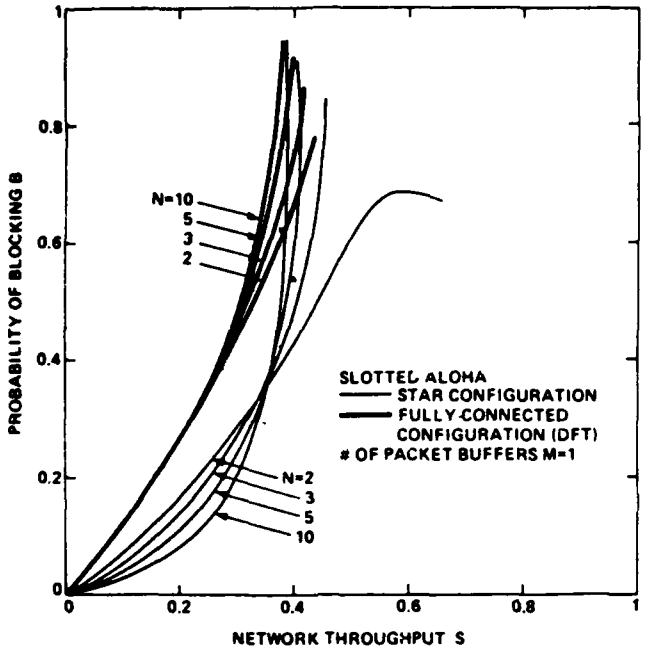


Fig. 6. Slotted ALOHA star and fully connected configurations: Minimum blocking versus throughput.

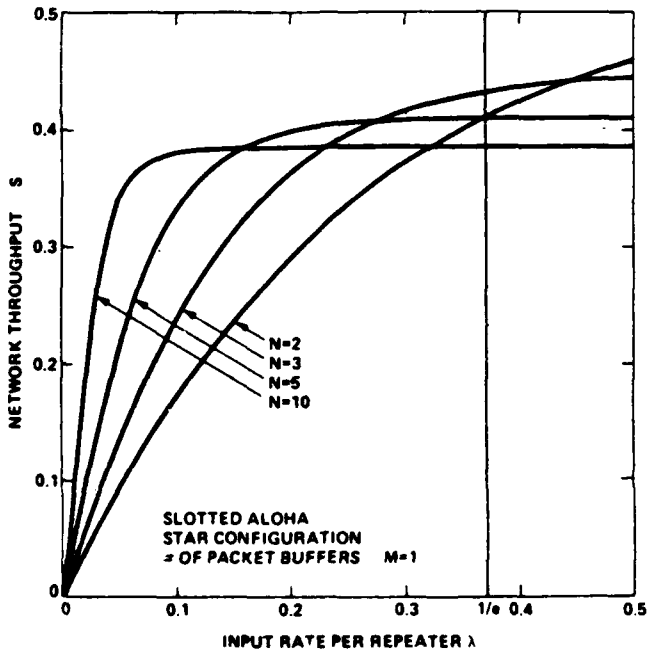


Fig. 7. Slotted ALOHA star configuration: Throughput versus λ.

We examine now the FC case. Contrary to the star configuration, the value of p which yields minimun D_n for a given p does not correspond to that which yields minimum blocking B (and thus maximum throughput). We get the optimum D_n for a given throughput S by plotting in the (S, D_n) plane the constant λ contours (varying p), and then by taking the lower envelope. Fortunately, the difference between the minimum blocking and the blocking achieved at optimum delay is rather insignificant! Optimum D_n and optimum B will therefore yield nearly the optimum total delay D for a given throughput S.

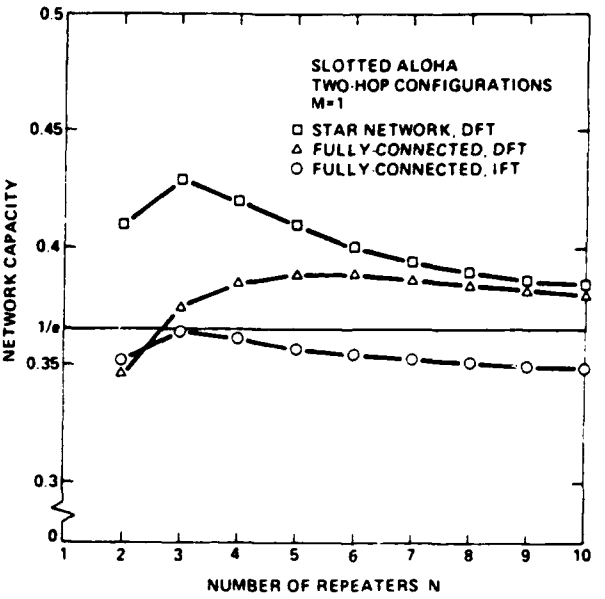


Fig. 8. Slotted ALOHA two-hop star and fully connected networks: Network capacity versus N.

Fig. 5 shows the optimum D_n versus S for various values of N along with the corresponding curves obtained in the star configuration. Fig. 6 shows the optimum blocking versus S. We note that, as expected, the probability of blocking is consistently higher for the fully connected configuration; this is simply due to the fact that transmissions by all repeaters contribute to the blocking of an incoming packet. Moreover, little discrepancy is observed as N varies between 2 and 10. The delay D_n, however, is smaller for lower throughput (with the exception of $N = 2$), and the difference becomes more significant as N gets larger. As for the system capacity, the FC configuration provides a smaller network capacity than the star configuration, especially for the smaller values of N ($N \leq 6$), as shown in Fig. 8; but as N gets larger ($7 \leq N \leq 10$), the capacity of the fully connected system approaches the one achieved in the star configuration.

The total packet delay $D_a + D_n$ for the above two cases is plotted in Fig. 9 (along with the results for other cases obtained and discussed in a later part of the paper). We note that for both $N = 2$ and $N = 5$ the delay obtained in the FC configuration is larger than or equal to the delay obtained with the star network; for $N = 10$, however, not only does the system capacity approach the one obtained with the star configuration, but the delay is also smaller for a wide range of S; this is simply explained by the fact that, as N gets larger, the value of λ that achieves a given throughput is smaller, and thus D_n becomes the predominant component of D; the improvement in D_n observed for $N = 10$ (see Fig. 5) overcomes the degrading effect of the larger blocking probability experienced (see Fig. 6).

B. The Multibuffer Case, DFT Protocol

1) *Analysis*: We consider here the star configuration; with $M > 1$, the state of the system is described by $\underline{n}^t = (n_1{}^t, n_2{}^t, \cdots, n_N{}^t)$. Let S denote the state space; that is, $S = \{(n_1,$

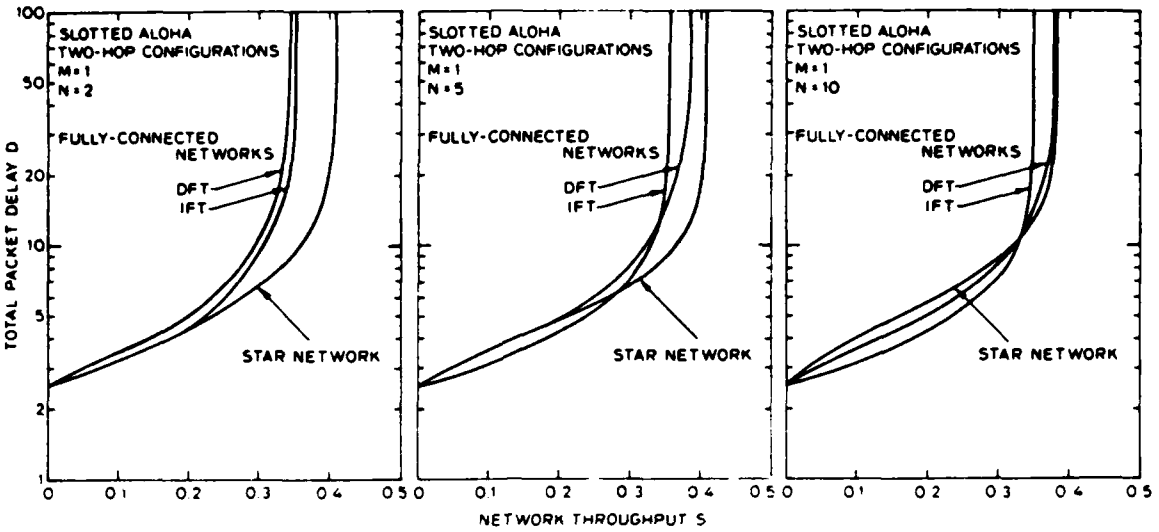


Fig. 9. Throughput-delay tradeoffs in two-hop slotted ALOHA star and fully connected networks.

$n_2, \cdots, n_N) \mid 0 \leq n_i \leq M, \forall i = 1, 2, \cdots, N\}$. The derivation of the transition matrix P for this case is given in Appendix A. Let $\Pi = \{\pi_{\underline{n}}\}_{\underline{n} \in S}$ be the stationary distribution of $\underline{n}^t$. Π is evaluated by *iteratively* solving the system: $\Pi = \Pi P$. The marginal distribution of n_i is given by

$$\Pr\{n_i = k\} = \sum_{\{\underline{n} \in S \mid n_i = k\}} \pi_{\underline{n}}. \tag{15}$$

The average queue length at a repeater, denoted by $\bar{q}$, is then given by

$$\bar{q} = \sum_{k=0}^{M} k \Pr\{n_i = k\}. \tag{16}$$

The blocking probabilities α and β defined in Section III-A-1) above are expressed as

$$\alpha = \Pr\{n_i = M\}(1 - p) \tag{17}$$

$$\beta = [1 - \Pr\{n_i = 0\}]\, p. \tag{18}$$

The network throughput is simply given by

$$S = N\lambda(1 - \alpha - \beta) \tag{19}$$

and by Little's result, the network delay is computed by

$$D_n = \frac{\bar{q}}{\lambda(1 - \alpha - \beta)}. \tag{20}$$

2) *Numerical Results*: In Fig. 10 we plot on the (S, D_n) plane the constant λ contours (varying p) for the example $N = 3, M = 2$. The optimum delay is obtained by taking the lower envelope. It is noted that given λ, the value of p yielding optimum delay again does not exactly correspond to the value of p which yields minimum blocking (and therefore maximum throughput). However, the probability of blocking at optimum delay is not significantly different from the minimum blocking achievable! The effect M has on network delay is shown in Fig. 11 where we plot, for $N = 2$, and 3, the optimum delay curves corresponding to various values of M. The increase with larger M is due to the additional queueing time incurred and to a larger fraction of time that the repeaters are active. The effect M has on the probability of blocking is shown in Fig. 12 where we plot the minimum blocking as a function of S. Note the (slight) decrease achieved by going from $M = 1$ to $M = 2$. Increasing M to 3, however, offers no further significant improvement. Thus, for a given network throughput S, an increase in M results in an increase in D_n and a decrease in D_a (due to a decrease in B). What is then the effect on the total delay D? As an example, in Fig. 13, we plot D versus S for $N = 3$ and various values of M. Again we only note a slight improve-

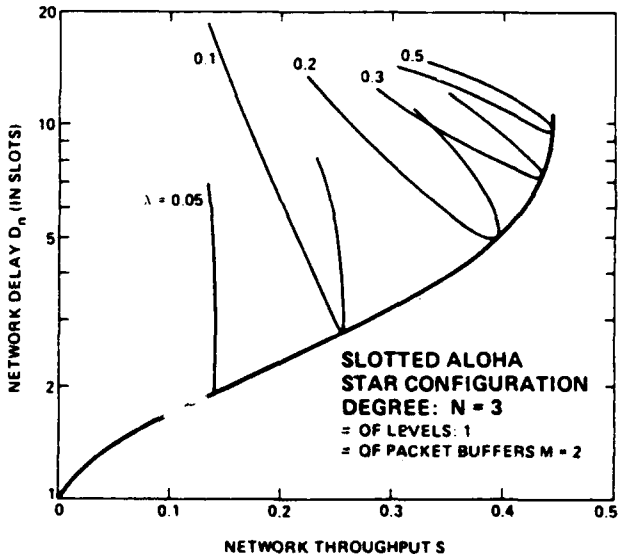


Fig. 10. Slotted ALOHA star configuration: Network delay versus throughput with $M > 1$.

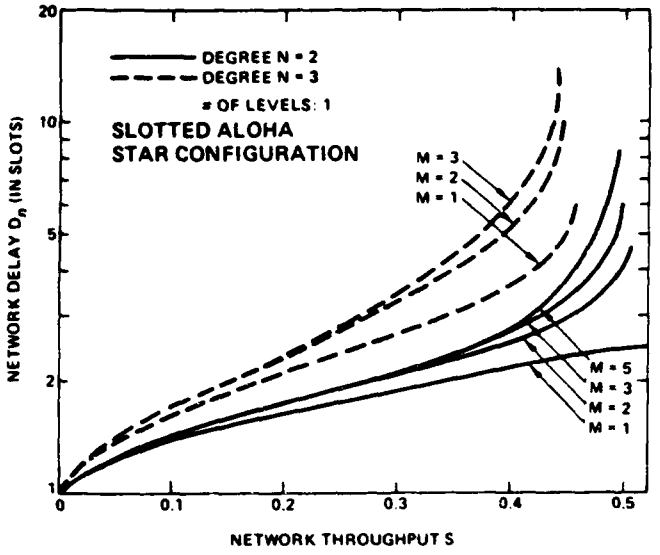


Fig. 11. Slotted ALOHA star configuration: Minimum network delay versus S for various values of M.

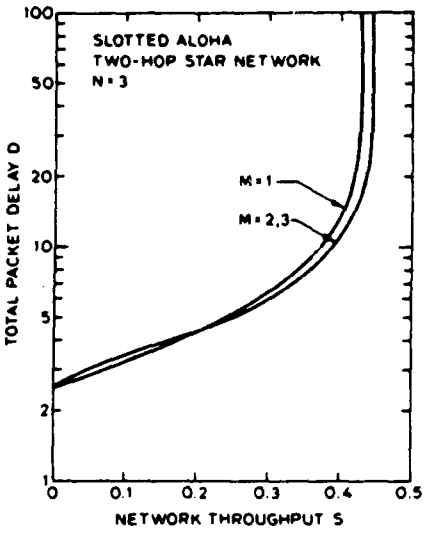


Fig. 13. Two-hop slotted ALOHA star networks: Total packet delay versus S for $N = 3$ and $M > 1$.

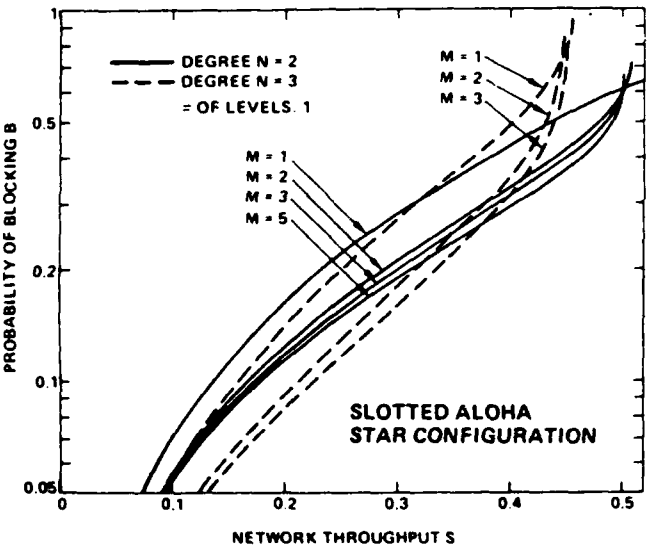


Fig. 12. Slotted ALOHA star configuration: Minimum blocking versus S for various values of M.

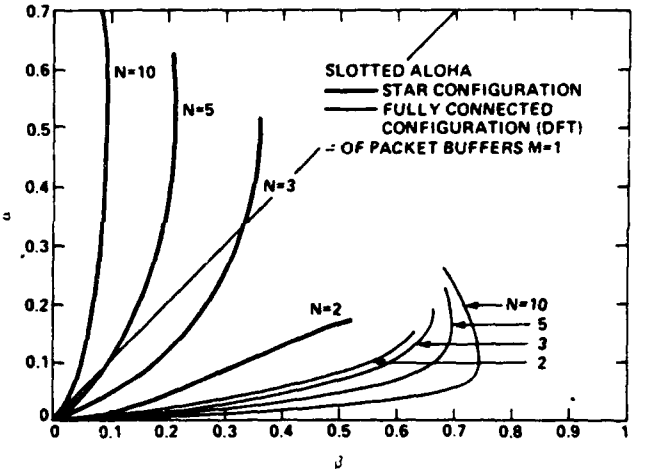


Fig. 14. Slotted ALOHA star and fully connected configurations: α versus β at minimum blocking.

ment in performance by going from $M = 1$ to $M = 2$. No further significant improvement is gained beyond $M = 2$.

The lack of important improvement experienced by increasing M is mainly explained by the fact that the system, at optimum, is mostly "*channel bound*" as opposed to "storage bound." To show that, we consider in Fig. 14 the (α, β) plane on which we plot the locus of optima, for both the star and FC configurations, for $M = 1$ and various values of N. For the star configuration, the curves corresponding to $N = 2$ and $N = 3$ lie almost entirely in the $\beta > \alpha$ half of the quadrant, showing that blocking is mostly due to the receiver being shut off. However, as N increases, the optimum drifts to the $\alpha > \beta$ region. Is the system then storage bound when N is large, say 10, for example? It is easy to show that there is still no significant improvement by increasing M. First, with large N, D_n is the predominant delay factor; indeed for a given S, D_n increases with N (see Fig. 5) while D_a decreases with S/N (for $N = 10$, for example, $S/N < 0.04$). Secondly, as S remains lower than 0.35 (value close to the capacity of these networks with large N), B is smaller for larger N rendering it ineffectual to further decrease it in an attempt to decrease D_a. For example, consider $N = 10$ and $S = 0.35$; we have $D_n \cong 10$, $B \cong 0.38$ and $D_a \cong 2.5$, yielding $D \cong 12.5$. By taking $B = 0$, we can decrease D_a to 1.15 providing thus a lower bound on D of 11.15, a rather small improvement. Moreover, due to the queueing effect, D_n increases with larger M.

As for the fully connected configuration, it is all too evident that β is the predominant factor and thus the FC configuration is even more channel bound than the star configuration. Moreover, the predominance of β relative to α is accentuated as N increases; this is due to the larger number of contending devices. Thus, in the sequel, we shall only consider $M = 1$.

C. The IFT Protocol

The motivation in considering the IFT protocol is simply an expected decrease in packet network delay due to the avoidance of an initial delay at the first transmission of the packet. In view of comparing this to the DFT protocol, we shall restrict ourselves to the FC configuration as it is simpler to analyze.

1) *Analysis (FC Configuration)*: Let n^t still denote the number of active repeaters in slot t. In this protocol, n^t is not a Markov chain since its transitions depend not only on n^{t-1}, but also on whether new arrivals had occurred in slot $t-1$ or not. Instead of formulating a Markov chain model for the system by increasing the state description to include an indicator for such events, we choose to utilize the imbedded Markov chain technique, and derive the steady-state performance measures using arguments from the theory of regenerative processes [24].

Denote by *empty slot* a slot in which no repeater undertook a transmission. Denote by d^k the number of active repeaters in the system at the end of the kth nonempty slot (see Fig. 15); d^k is a Markov chain. We derive its transition matrix P in Appendix B. Let

$$\pi_i{}^d = \lim_{k\to\infty} \Pr\{d^k = i\}.$$

The stationary distribution $\Pi^d = \{\pi_0{}^d, \pi_1{}^d, \cdots, \pi_N{}^d\}$ is obtained by solving recursively the system $\Pi^d = \Pi^d P$. We now derive the stationary performance measures. We define a *cycle* to be the interval of time separating two consecutive imbedded points. A cycle is entirely determined by the state of the system at the imbedded point which initiates it and can be labeled by that state. Given that the latter is i, the cycle length is equal to $I_i + 1$, where I_i denotes the number of empty slots in the cycle. The distribution of I_i and its average, $\bar{I}_i$, are also derived in Appendix B. Let S_i be the probability of a successful transmission in cycle i; we have

$$S_i = \Pr\{I_i = 0\}\frac{ip(1-p)^{i-1}}{1-(1-p)^i} + \Pr\{I_i > 0\} \cdot \frac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i + ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-p)^i(1-\lambda)^{N-i}}. \tag{21}$$

The average of the sum of active repeaters over the cycle is denoted by σ_i and is given by

$$\sigma_i = i\bar{I}_i + i + \Pr\{I_i > 0\}\frac{(N-i)\lambda}{1-(1-p)^i(1-\lambda)^{N-i}}. \tag{22}$$

By renewal theory arguments, the stationary system throughput is expressed as

$$S = \frac{\sum_{i=0}^{N}\pi_i S_i}{\sum_{i=0}^{N}\pi_i(\bar{I}_i + 1)} \tag{23}$$

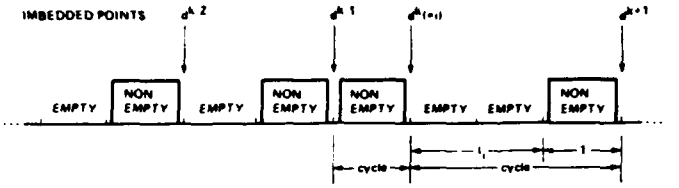


Fig. 15. The imbedded Markov chain in the slotted ALOHA IFT protocol.

and the stationary average number of active repeaters is given by

$$\bar{n} = \frac{\sum_{i=0}^{N}\pi_i\sigma_i}{\sum_{i=0}^{N}\pi_i(\bar{I}_i + 1)}. \tag{24}$$

By Little's result, $D_n = \bar{n}/S$. The probability of blocking is simply $B = 1 - (S/N\lambda)$. The access delay is given by (14) or (14a).

2) *Numerical Results*: The main focus here is to compare the performance obtained with this case to the one obtained with the DFT protocol. This we do by first plotting D_n versus S in Fig. 16 and B versus S in Fig. 17 (at optimum) along with the delay and blocking corresponding to the DFT protocol. We note that for the most interesting range of S, D_n is indeed smaller with IFT. The IFT-system capacity for a two-hop environment, however, is dominated by the DFT-system capacity (with the exception of $N = 2$) as shown in Fig. 8 above; this capacity is not too sensitive to variations in the size of the network, N. The throughput-delay curves are shown in Fig. 9 above. For $N = 2$, the IFT delay curve is consistently lower than the DFT curve. For $N = 5$ and 10, the IFT delay is lower over a significant range of the throughput, but as S increases, the relationship reverses as the IFT system reaches its capacity sooner. Thus we experience with the IFT protocol a slightly improved packet delay but a slightly degraded system capacity.

IV. CONCLUSION

The difficulty encountered in analyzing multihop packet radio systems led us to consider simple but typical configurations in an attempt to understand the behavior of these systems and derive their performance. We analyzed in this paper the performance of centralized two-hop packet radio networks employing slotted ALOHA, in terms of system capacity and throughput-delay tradeoffs. We have also shown the effect on system performance of various system parameters, namely the transmission probability p, the number of repeaters N, and the repeater's buffer size M.

The results show that, under the assumption that the processing time at the devices is negligible, packet radio systems are channel bound; a *slight* improvement may be gained by increasing the buffer size from $M = 1$ to $M = 2$, but no significant improvement is obtained beyond that.

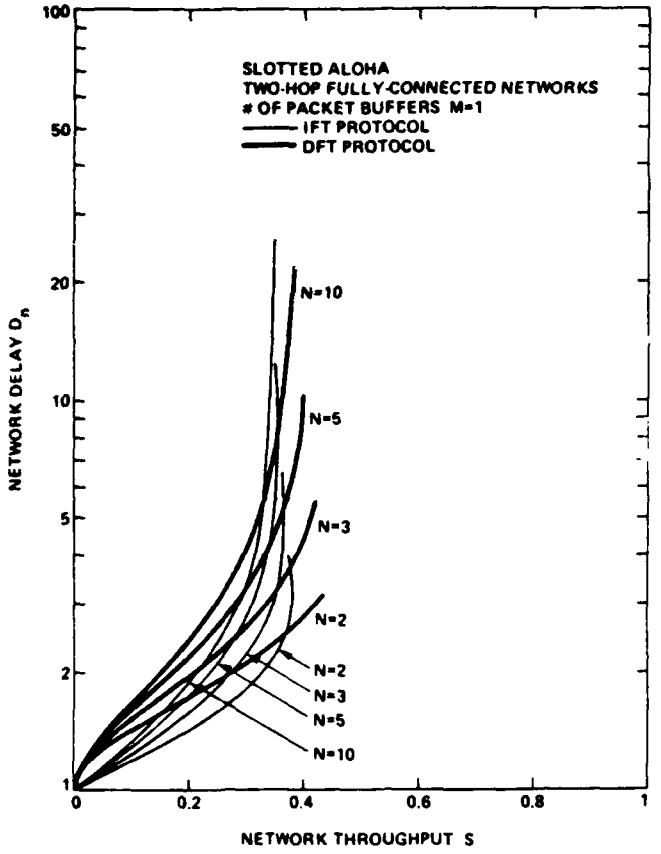


Fig. 16. Slotted ALOHA fully connected configuration: Optimum (network) throughput-delay curves.

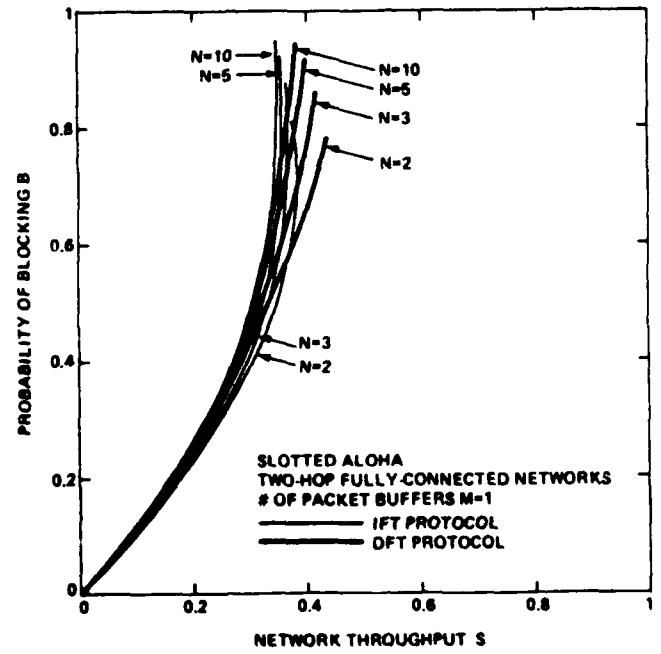


Fig. 17. Slotted ALOHA fully connected configuration: Minimum blocking versus S.

In the slotted ALOHA context, we studied the star configuration and the fully connected configuration, as well as two transmission protocols, the delayed first-transmission protocol and the immediate first-transmission protocol. For small N, the star configuration offers a higher system capacity than the FC configuration; this is due to the fact that the terminal access hop in the star case is more efficient resulting from smaller probabilities of blocking at the repeaters. But as N increases, the inner hop becomes the critical hop and both configurations become equivalent in capacity. For the larger values of N, the FC configuration provides smaller packet delays for low and moderate values of the throughput; this is due to smaller network delays in the FC configuration, and this is more noticeable for the larger values of N where network delay becomes the important component of the total packet delay.

System performance varies with the particular transmission protocol utilized at the repeater hop. Still in the context of slotted ALOHA, the sensitivity of the system performance to variations in the transmission protocol was observed by comparing the IFT-FC-configuration to the DFT-FC-configuration. We basically noted a lower system capacity with the IFT protocol which is not too sensitive to changes in N. The packet delay, however, has *slightly* improved over a significant range of the throughput, as anticipated.

In Part II [1] we examine the same problem with the nonpersistent carrier sense multiple-access mode used throughout the system.

APPENDIX A

TRANSITION MATRIX FOR THE SLOTTED ALOHA PROTOCOL WITH $M > 1$

We denote by $\Pr\{n \mid m\}$ the probability of the one-step transition from state $m = (m_1, m_2, \cdots, m_N)$ to state $n = (n_1, n_2, \cdots, n_N)$. In any transition, the amplitude of change in n_i cannot exceed 1. We distinguish the following cases:

1) If $\exists\, i$ such that $|m_i - n_i| > 1$, or if $\exists\, i, j, i \neq j$, such that $n_i = m_i - 1$ and $n_j = m_j - 1$, then $\Pr\{n \mid m\} = 0$.

2) Otherwise (either a successful transmission took place or no packet was successfully transmitted), if $\exists\, i_0$ such that $m_{i_0} = n_{i_0} + 1$ (indicating a successful transmission by repeater i_0) then

$$\Pr\{n \mid m\} = p \prod_{j \neq i_0} (1-p)^{\chi_j} \prod_{l \neq i_0} [\lambda \xi_l^{-} + (1 - \lambda + \lambda \zeta_l)\xi_l] \tag{A.1}$$

where

$$\chi_j = \begin{cases} 1 \text{ if } m_j > 0 \\ 0 \text{ if } m_j = 0 \end{cases} \qquad \xi_j = \begin{cases} 1 \text{ if } m_j = n_j \\ 0 \text{ if } m_j \neq n_j \end{cases}$$

$$\xi_j^{-} = \begin{cases} 1 \text{ if } m_j = n_j - 1 \\ 0 \text{ if } m_j \geq n_j \end{cases} \qquad \zeta_j = \begin{cases} 1 \text{ if } n_j = M \\ 0 \text{ if } n_j < M \end{cases} \tag{A.2}$$

The term $p\Pi_{j \neq i_0}(1-p)^{\chi_j}$ represents the probability that i_0 is the only transmitting repeater among all active ones. The sec-

ond product term represents the probability of all changes, that is, the presence or absence of arrivals, which occurred at the remaining queues in the current slot.

3) Otherwise (no successful transmission took place), letting $I_s = \{j \mid m_j = n_j\}$ we have

$$\Pr\{n \mid m\} = \left[\prod_{j \in I_s} [p^{\chi_j} + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)] - \sum_{j \in I_s} p^{\chi_j} \prod_{\substack{k \in I_s \\ k \neq j}} (1-p)^{\chi_k}(1-\lambda+\lambda\zeta_k)\right] \cdot \prod_{j \notin I_s} (1-p)^{\chi_j}\lambda$$

where χ_j and ζ_j are defined in (A.2) above. According to the model under consideration, an arrival to a repeater in a slot t is rejected (blocked) if that repeater is in transmit mode during the slot. Thus, the number of packets queued at repeater $j \in I_s$ remains unchanged with probability $p^{\chi_j} + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)$, *provided* that any transmission (represented by the term p^{χ_j}) is unsuccessful. Since the repeaters are all independent, the probability of the event $\{m_j = n_j\}$ for all $j \in I_s$ is then given by the expression in the first bracket, in which the summation

$$\left[\sum_{j \in I_s} p^{\chi_j} \prod_{\substack{k \in I_s \\ k \neq j}} (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_k)\right]$$

represents the probability of all possible *successful* transmissions. Now for all $j \notin I_s$, the number of packets increased by one; the probability of this event is simply given by the last product term.

APPENDIX B

TRANSITION MATRIX FOR THE SLOTTED ALOHA IFT PROTOCOL WITH $M = 1$

Let $p_{ij} \triangleq \Pr\{d^{k+1} = j \mid d^k = i\}$. For $i = 0$, we have

$$p_{0j} = \begin{cases} \dfrac{N\lambda(1-\lambda)^{N-1}}{1-(1-\lambda)^N} & j = 0 \\ 0 & j = 1. \\ \dbinom{N}{j}\dfrac{\lambda^j(1-\lambda)^{N-j}}{1-(1-\lambda)^N} & j = 2, 3, \cdots, N \end{cases} \tag{B.1}$$

Given that $d^k = i$, let I_i denote the number of empty slots separating two consecutive nonempty slots. Note that, in a fully connected configuration, it is only in an empty slot that an arrival from a terminal can be successfully received at the repeater. Also note that with the IFT protocol, an arrival in an empty slot ends the sequence of empty slots separating two consecutive nonempty slots. Thus, for $i \neq 0, N$, we have

$$\Pr\{I = 0\} = 1-(1-p)^i; \Pr\{I_i > 0\} = (1-p)^i \tag{B.2}$$

and the transition probabilities are given by ($i \neq 0, N$)

$$p_{ij} = \begin{cases} 0 & j < i-1 \\ \Pr\{I_i = 0\}\dfrac{ip(1-p)^{i-1}}{1-(1-p)^i} + \Pr\{I_i > 0\}\dfrac{ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-\lambda)^{N-i}(1-p)^i} & j = i-1 \\ \Pr\{I_i = 0\}\dfrac{1-(1-p)^i-ip(1-p)^{i-1}}{1-(1-p)^i} + \Pr\{I_i > 0\} \cdot \dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i + (1-\lambda)^{N-i}[1-ip(1-p)^{i-1}-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j = i \\ \Pr\{I_i > 0\}\dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}[1-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j = i+1 \\ \Pr\{I_i > 0\}\dfrac{\dbinom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j}}{1-(1-\lambda)^{N-i}(1-p)^i} & j \geq i+2. \end{cases} \tag{B.3}$$

Finally, for $i = N$ we simply have

$$p_{N,j} = \begin{cases} 0 & j < N-1 \\ \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j = N-1 \\ 1 - \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j = N. \end{cases} \quad \text{(B.4)}$$

The probability density function of I_i is given by

$$Pr\{I_i = l\} = \begin{cases} (1-\lambda)^{N(l-1)}[1-(1-\lambda)^N] & i = 0; l \geq 1 \\ 1-(1-p)^i & i \neq 0, N; l = 0 \\ (1-p)^i[(1-\lambda)^{N-i}(1-p)^i]^{l-1} \cdot [1-(1-\lambda)^{N-i}(1-p)^i] & i \neq 0, N; l > 0 \\ (1-p)^{Nl}[1-(1-p)^N] & i = N; l \geq 0. \end{cases} \quad \text{(B.5)}$$

Thus $\bar{I}_i$ is expressed as

$$\bar{I}_i = \begin{cases} \dfrac{1}{1-(1-\lambda)^N} & i = 0 \\ \dfrac{(1-p)^i}{1-(1-p)^i(1-\lambda)^{N-i}} & i \neq 0, N. \\ \dfrac{(1-p)^N}{1-(1-p)^N} & i = N. \end{cases} \quad \text{(B.6)}$$

REFERENCES

[1] F. Tobagi, "Analysis of a two-hop centralized packet radio network—Part II: Carrier sense multiple access," this issue, pp. 208–216.

[2] L. G. Roberts and B. D. Wessler, "Computer network development to achieve resource sharing," in *1970 Spring Joint Comput. Conf., AFIPS Conf. Proc.*, pp. 543–549.

[3] R. E. Kahn, "Resource sharing computer communication networks," *Proc. IEEE*, vol. 60, pp. 1397–1407, Nov. 1972.

[4] N. Abramson and F. Kuo, Eds., *Computer Communication Networks*. New York: Prentice-Hall, 1973.

[5] L. Roberts, "Data by the packet," *IEEE Spectrum*, Feb. 1974.

[6] N. Abramson, "The ALOHA-System—Another alternative for computer communications," in *1970 Fall Joint Comput. Conf., AFIPS Conf. Proc.*, pp. 695–702.

[7] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Trans. Commun.*, vol. COM-25, pp. 117–128, Jan. 1977.

[8] R. E. Kahn, "The organization of computer resources into a packet radio network," in *1975 Nat. Conf. Comput., AFIPS Conf. Proc.*, pp. 177–186.

[9] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, pp. 1468–1996, Nov. 1978.

[10] L. Kleinrock and F. Tobagi, "Random access techniques for data transmission over packet-switched radio channels," in *Nat. Conf. Comput., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1975, pp. 187–201.

[11] H. Frank, I. Gitman, and R. Van Slyke, "Packet radio system—Network considerations," in *Nat. Conf. Comput., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1975, pp. 217–231.

[12] S. Fralick and J. Garrett, "Technological considerations for packet radio networks, in *Nat. Conf. Comput., AFIPS Conf. Proc.*, vol 44 Montvale, NJ: AFIPS Press, 1975, pp. 233–243.

[13] J. Burchfiel, R. Tomlinson, and M. Beeler, "Functions and structure of a packet radio station," in *Nat. Conf. Comput., AFIPS Conf. Proc.*, vol. 44. Montvale, NJ: AFIPS Press, 1975, pp. 245–251.

[14] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple access modes and their throughput delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[15] F. 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.*, vol. COM-23, pp. 1417–1433, Dec. 1975.

[16] ——, "Packet switching in radio channels: Part III—Polling and (dynamic) split channel reservation multiple access," *IEEE Trans. Commun.*, vol. COM-24, pp. 832–845, Aug. 1976.

[17] L. Kleinrock and S. Lam, "Packet switching in a multiaccess broadcast channel: Performance evaluation," *IEEE Trans. Commun.*, vol. COM-23, pp. 410–423, Apr. 1970.

[18] S. Lam, "Packet switching in a multiaccess broadcast channel with applications to satellite communication in a computer network," School of Eng. Appl. Sci., Univ. California, Los Angeles, CA, UCLA-ENG-7429, Apr. 1974 (also published as Ph.D. dissertation).

[19] G. Fayolle, E. Gelembe, and J. Labetoulle, "Stability and optimal control of the packet-switching broadcast channels," *J. Ass. Comput Mach.*, vol. 24, pp. 375–386, July 1977.

[20] I. Gitman, "On the capacity of slotted ALOHA networks and some design problems," *IEEE Trans. Commun.*, vol. COM-23, pp. 305–317, Mar. 1975.

[21] L. G. Roberts, "ALOHA packet system with and without slots and capture," *Comput. Commun. Rev.*, vol. 5, pp. 28–42, Apr. 1975.

[22] L. Kleinrock and S. Lam, "Packet switching in a slotted satellite channel," in *Nat. Conf. Comput., AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, 1973, pp. 703–710.

[23] L. Kleinrock, *Queueing Systems. Vol. II: Computer Applications*. New York: Wiley-Interscience, 1976.

[24] F. Tobagi *et al.*, "On modeling and measurement techniques in packet communication networks," *Proc. IEEE*, vol. 66, pp. 1423–1447, Nov. 1978.

★

Fouad A. Tobagi (M'77) was born in Beirut, Lebanon, on July 18, 1947. He received the Engineering degree from Ecole Centrale des Arts et Manufactures, Paris, France, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1971 and 1974, respectively.

From 1971 to 1974, he was with the University of California, Los Angeles, where he participated in the ARPA Network Project as a Postgraduate Research Engineer and did research in packet radio communication. During the summer of 1972, he was with the Communications Systems Evaluation and Synthesis Group, IBM T. J. Watson Research Center, Yorktown Heights, NY. From December 1974 to June 1978, he was a Research Staff Project Manager with the ARPA project at the Computer Science Department, UCLA, and engaged in the modeling, analysis, and measurements of packet radio systems. In June 1978, he joined the faculty of the School of Engineering at Stanford University, where he is now Assistant Professor of Electrical Engineering. His current research interests include computer communication networks, packet switching in ground radio and satellite networks, modeling and performance evaluation of computer communications systems.

Analysis of a Two-Hop Centralized Packet Radio Network—Part II: Carrier Sense Multiple Access

FOUAD A. TOBAGI, MEMBER, IEEE

Abstract—We continue in this paper our study of two-hop centralized packet radio networks in view of understanding the behavior of these systems. Traffic originates at terminals, is destined to a central station, and requires for its transport the relaying of packets by store-and-forward repeaters. We consider here that all devices employ the nonpersistent carrier sense multiple-access mode. System capacity and throughput-delay tradeoffs are derived and compared to those obtained for slotted ALOHA in Part I [1].

I. INTRODUCTION

THE difficulty encountered in analyzing multihop packet radio systems led us to consider simple but typical configurations in an attempt to understand the behavior of these systems and derive their performance. In Part I we analyzed the performance of centralized two-hop packet radio networks employing slotted ALOHA in terms of system capacity and throughput-delay tradeoffs [1]. In the present paper, we continue the study by considering that all devices employ the nonpersistent carrier sense multiple access.

Carrier sense multiple access reduces the level of interference caused by overlapping packets in the random multiaccess environment by allowing the devices to sense the carrier due to transmissions by users within range [2]. In the simple nonpersistent CSMA protocol, a device with a packet ready for transmission senses the channel and operates as follows: 1) if the channel is sensed idle, the device transmits the packet; 2) if the channel is sensed busy, then the device reschedules the transmission of the packet to some later time incurring a random rescheduling delay; at this new point in time, the device senses the channel and repeats the algorithm. For simplicity in analysis, a slotted version of the above protocol is considered in which the time axis is slotted and the slot size is τ s, where τ is the *propagation delay* among pairs of devices.[1] Note that this definition of a slot is different from that used in Part I for slotted ALOHA; here a packet's transmission time is equivalent to several slots. All devices are synchronized and are forced to start transmission only at the beginning of a slot. When a packet's arrival occurs during a slot, the device senses the channel and operates according to the protocol described above.

As in Part I, we consider two-hop centralized configurations in which traffic originates at terminals, is destined to a central station, and requires for its transport that packets be relayed by store-and-forward repeaters. The basic performance measures sought here are, again, system capacity and throughput-delay tradeoffs. All devices are provided with onmidirectional antennas. With each repeater is associated a population of terminals, in line-of-sight and within range of only that repeater. Traffic originates at terminals and is destined to the station; thus, we consider *inbound traffic* only. Each repeater is provided with a *finite storage* capacity which can accommodate a single packet. The station has an infinite storage capacity. Packets are all of a fixed size. When the transport of a packet over a hop is successful (i.e., the transmission is free of interference and storage is available at the receiving device), the packet is deleted from the sender's queue; otherwise, the packet incurs a retransmission delay. It is assumed again that a device learns about its success or failure at the end of transmission; that is, acknowledgments are assumed to be instantaneous and for free. At any one time, a device can be either transmitting or receiving, but not both simultaneously. The station always has its receiver on. The packet processing time at any device is considered to be negligible.

In order to gain much of the advantage of CSMA, we consider the FC configuration depicted in Fig. 1, where all repeaters are within range and in line-of-sight of each other and of the station. Terminals follow the nonpersistent CSMA protocol described above. A repeater, which has completed the successful reception of a packet from its associated population of terminals, transmits the packet without delay. The repeater is guaranteed that the channel will be sensed idle at the end of a correct reception since, given the system connectivity, all repeaters must have been quiet during the entire reception time of the packet. This first transmission (and subsequent transmissions) of the received packet may however still be unsuccessful due to collisions with transmissions from other active repeaters. The rescheduling of the packet is considered to be geometrically distributed; that is, the repeater resenses the channel in the current slot with a fixed probability ν; clearly, a retransmission will result only if the channel is sensed idle. (Note that this transmission protocol is analog to the IFT protocol considered in Part I for the slotted ALOHA mode [1].)

II. ANALYSIS

A. Characterization of Repeaters's Traffic

Consider for each population of terminals T_i a time line which exhibits packet transmissions from T_i only. Consider

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received May 31, 1978; revised September 4, 1979. This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-77-C-0272 and Contract MDA-79-C-0201.

The author is with the Computer Systems Laboratory, Stanford University, Stanford, CA 94305.

[1] As in [2], we assume the propagation delay to be the same for all pairs.

0090-6778/80/0200-0208$00.75

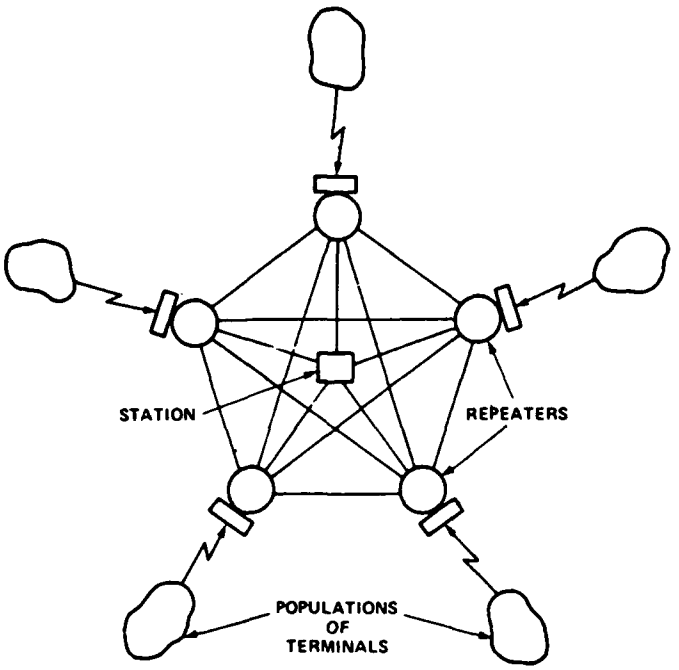


Fig. 1. A two-hop fully connected configuration.

also a time line R which exhibits packet transmissions from repeaters only. On each such time line we observe an alternate sequence of transmission and idle periods (see Fig. 2). The processes defining these time lines are evidently dependent on each other in a rather complex way; the dependence is determined by the particular system connectivity.

Since repeaters possess a single-packet buffer, it is clear that packet transmissions from a population of terminals to their associated repeater are useless if the latter has a non-empty buffer. Although such transmissions do not affect the system's operation, they do affect the network performance in that they may cause the repeater to delay its transmission due to sensing terminals' carrier. Accordingly, we consider here that the repeaters use a signaling scheme which allow them to distinguish between the presence of carrier due to other repeaters and carrier due to transmissions by their associated terminals. One such scheme consists of having repeaters transmit a busy-tone signal on a narrow-band busy-tone channel whenever they are undertaking packet transmissions.[2] From the analysis point of view, an important simplification is also gained, in that the decision made by a repeater regarding the transmission of its packet is solely dependent on the state of the repeaters.

B. Characterization of Terminals' Traffic

In the environment in question, a terminal is out-of-range of all but its associated repeater and thus can incur collisions with other repeaters' transmissions. However, by assuming that a terminal does not inhibit transmission even when its

[2] The busy-tone channel is assumed to be separate from the available bandwidth in question. Problems in detecting the busy tone that may arise with the use of narrow-band channels are ignored in this paper. [3]

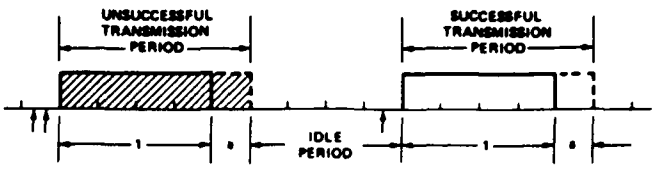


Fig. 2. Slotted nonpersistent CSMA: Transmission and idle periods (vertical arrows represent terminals becoming ready to transmit).

associated repeater is transmitting, we here too simplify the analysis considerably in that the processes defining each time line T_i become independent of the repeaters' time line R; the *successful transport* of a packet from a terminal to its associated repeater, on the other hand, will be considered dependent on the state of R (as seen in the analysis below). The effect of this assumption on the evaluation of the system performance is to provide rather slightly pessimistic results; indeed transmissions from T_i which start during a transmission period of repeater R_i are useless and contribute to a higher traffic rate on time line T_i. It is however important to note that this effect gets smaller as N, the number of repeaters, gets larger. For $N = 10$, for example, T_i can *normally* hear only 10 percent of the repeaters' traffic.

A transmission from T_i is said to be T_i–successful it if is free of collision from other terminals in T_i. Let λ denote the rate of T_i–successful transmissions from T_i (normalized to the packet transmission time). Due to blocking at repeater R_i, only a fraction $s \leq \lambda$ is correctly received at the repeater. Let G be the rate of sense points on time line T_i. By the above assumption, λ and G for this slotted nonpersistent CSMA are related by [2]

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

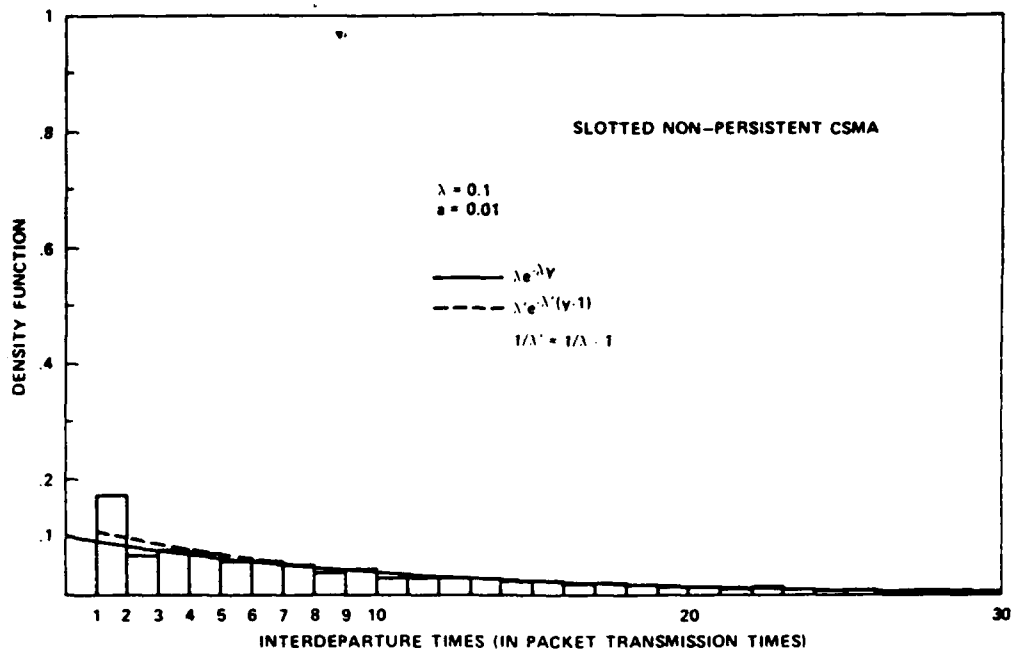
where $a = \tau/T$ and T is the transmission time of a packet. Moreover, the average idle period of time line T_i is $ae^{-aG}/(1 - e^{-aG})$, and the transmission period is $T + \tau$. We let τ be the unit time. T denotes then the number of slots per transmission time, and a equals $1/T$. We characterize now the process defining the T_i-successful transmission. Let Y denote the time (in units of T) between the end of two consecutive T_i-successful transmissions. Simulation results have shown that we can approximate Y by $1 + Z$ where Z is exponentially distributed with mean $1/\lambda' = 1/\lambda - 1$. That is

$$\Pr\{Y \leq y\} = 1 - e^{-(\frac{1}{\lambda} - 1)} \qquad y \geq 1. \tag{2}$$

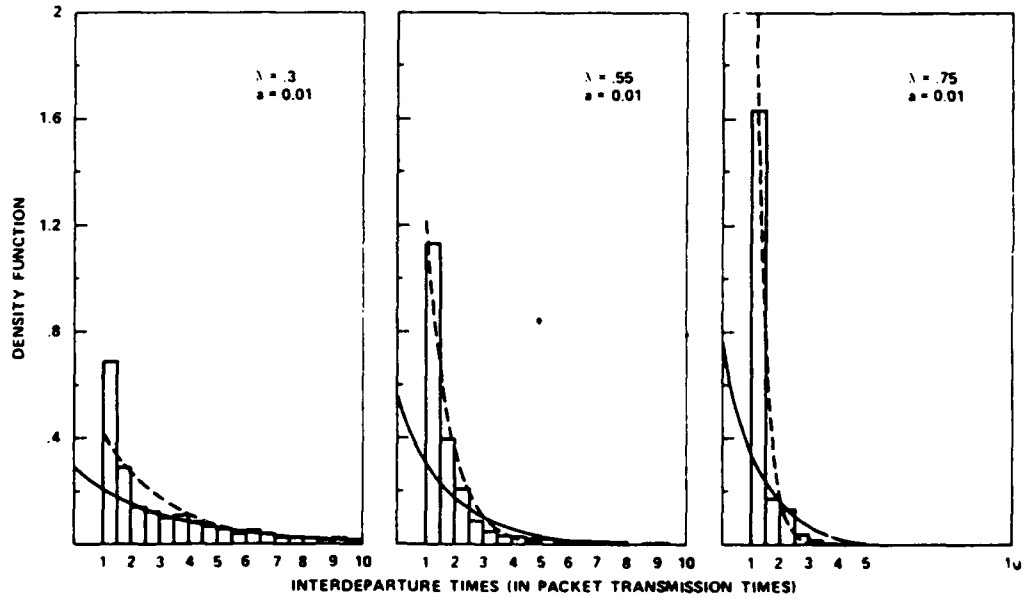
The goodness of the approximation is verified by comparing this density function with histograms of interdeparture times obtained from the simulation of an infinite population of terminals employing CSMA. Examples are shown in Fig. 3.

C. Analysis

Consider time line R on which we observe an alternate sequence of transmission periods and idle periods. As in [4], we consider the *imbedded slots* defined to be the first slot



(a)



(b)

Fig. 3. (a) Histograms of interdeparture times in slotted nonpersistent CSMA ($\lambda = 0.1$). (b) Histograms of interdeparture times in slotted nonpersistent CSMA ($\lambda = 0.3$, 0.55 and 0.75).

of each idle period (see Fig. 4). The intervals between two consecutive imbedded slots are defined as *cycles*. Let $n^t{}_e$ denote the number of active repeaters in slot t_e. We show that $n^t{}_e$ is a Markov chain and determine its transition probabilities.

Given $n^t{}_e = n$, let I_n denote the length of the idle period (in slots). An idle period ends in a slot if either an active repeater decides to start transmission in that slot, or a successful transmission to a passive repeater from its associated popula-

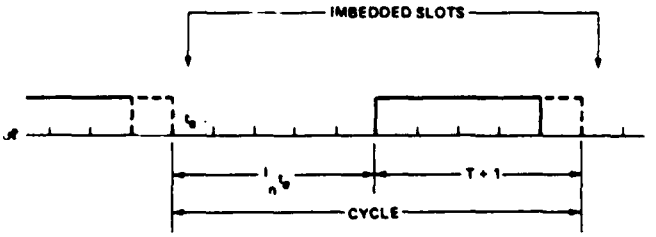


Fig. 4. The imbedded slots in time line R.

tion of terminals is completed in that slot (since the repeater immediately relays it), or both. It is clear that for a T_i-successful transmission to be successfully received at repeater R_i (considered inactive), this transmission should *entirely* take place during an *idle* period of time line R. Consider the imbedded slot t_e and assume $n^{t_e} = n$. Let then J_n denote the time until some active repeater decides to sense the channel (and hence to transmit if the channel is sensed idle). J_n is geometrically distributed; its density function is given by

$$\Pr\{J_n = k \text{ slots}\} = (1-\nu)^{n(k-1)}[1-(1-\nu)^n]. \tag{3}$$

With $n^{t_e} = n$, there are $N-n$ inactive repeaters. Let R_i again denote such a repeater. By the independence assumption between time line R and the terminal time lines, the end of a cycle on time line R represents, relative to time line T_i, a random look in time; accordingly, the probability that this point falls in a transmission period of T_i is precisely the ratio of the transmission period to the average cycle time (yielding $1-\lambda/G$); the probability that it falls in an idle period is clearly λ/G. We let Y_i denote the time since t_e (the end of a cycle on time line R) until time line T_i is idle; its distribution is then given by

$$\Pr\{Y_i \leq y\} = \frac{\lambda}{G} + \left(1-\frac{\lambda}{G}\right)\frac{y}{\tau} \qquad 0 \leq y \leq T. \tag{4}$$

Given that a transmission from T_i requires T slots, no successful reception at repeater R_i can take place *before* slot $t_e + Y_i + T$. Let $Y_i' = Y_i + T$. From the characterization of successful traffic introduced above, we note that, following $t_e + Y_i'$, the arrival process from T_i to R_i can be represented by a Bernoulli process, whereby the probability of completion of a correct reception in a slot is $a\lambda'$, with $\lambda' = 1/(1/\lambda - 1)$. Without loss of generality, we let $R_1, R_2, \cdots, R_{N-n}$ be the inactive repeaters and we let $Y_1' \leq \cdots \leq Y_{N-n}' < \infty$ and let $Y_0' = 0$. For any slot t, $t_e + Y_j' < t \leq t_e + Y_{j+1}'$, (and under the condition that no arrival took place to any inactive repeater prior to t), the arrival process in slot t is binomial such that

$$\Pr\{k \text{ packet receptions completed in } t, 0 \leq k \leq j\} = \binom{j}{k}(a\lambda')^k(1-a\lambda')^{j-k}. \tag{5}$$

With these considerations, it is clear that n^{t_e} is a Markov chain. To avoid the great complexity involved in keeping track of the position of time slot t in relation to the sequence $\{t_e + Y_j'\}$, we choose here to derive an upper and lower bound on performance by considering the following much simpler arrival processes. We let $Y_{\min}' = Y_1'$ and $Y_{\max}' = Y_{N-n}'$. The upper bound is obtained by considering the arrival process to be

$$\Pr\{k \text{ packet receptions completed in } t, 0 \leq k \leq N-n\} = \begin{cases} 0 & t < t_e + Y_{\min} \\ \binom{N-n}{k}(a\lambda')^k(1-a\lambda')^{N-n-k} & t_e + Y_{\min}' \leq t < \infty \end{cases} \tag{6}$$

The lower bound is obtained by substituting $Y_{\max}'$ for $Y_{\min}'$ in the above equation. Let Y_m' denote interchangeably $Y_{\min}'$ and $Y_{\max}'$, where the subscript m is replaced by min or max as needed. If $J_n < Y_m'$ then the idle period ends because of the start of a transmission from an active repeater; if $J_n \geq Y_m'$ then arrivals to passive repeaters are possible, and for each slot thereon it is the contention of both active repeaters and passive repeaters just completing reception that determine the end of the idle period in that slot. Clearly the system state (number of active repeaters) does not vary over a transmission period of time line R. With these considerations, it is straightforward to derive the transition matrix P for each case. This is given in the Appendix. Let $\Pi = \{\pi_0, \pi_1, \cdots, \pi_N\}$ denote the stationary distribution, where

$$\pi_i = \lim_{t_e \to \infty} \Pr\{n^{t_e} = i\}.$$

Π is obtained by solving recursively the system $\Pi = \Pi P$. Now, we proceed with the derivation of the performance measures, namely the network throughput S and the network delay D_n. We have defined a cycle to be the interval of time separating two successive imbedded slots; a cycle consists of an idle period followed by a transmission period. Given that $n^{t_e} = n$, let I_n denote the length of the idle period; the transmission period is of length $T+1$; the cycle length is $I_n + T + 1$. Let $\bar{I}_n$ denote the expected value of I_n; $\bar{I}_n$ is derived in the Appendix. The probability of a successful transmission by the repeaters over the cycle, which we denote by S_n is expressed as

$$S_n = \Pr\{J_n < Y_m'\}\frac{n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n} + \Pr\{J_n \geq Y_m'\} \cdot \frac{n\nu(1-\nu)^{n-1}(1-a\lambda')^{N-n} + (N-n)a\lambda'(1-a\lambda')^{N-n-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda')^{N-n}}. \tag{7}$$

In this expression, we distinguished the case $J_n < Y_m'$ where only active repeaters contend on the channel, and the case $J_n \geq Y_m'$ where newly received packets contend as well. Let σ_n denote the average sum of active repeaters over all slots in the cycle. It is expressed as

$$\sigma_n = (\bar{I}_n + T + 1)n + (T+1)\Pr\{J_n \geq Y_m'\} \cdot \frac{(N-n)a\lambda'}{1-(1-\nu)^n(1-a\lambda')^{N-n}}. \quad (8)$$

By arguments from the theory of regenerative processes, we write the stationary system throughput S and the stationary average number of active repeaters $\bar{n}$, respectively, as

$$S = \frac{\sum_{n=0}^{N} \pi_n S_n T}{\sum_{n=0}^{N} \pi_n (\bar{I}_n + T + 1)} \quad (9)$$

$$\bar{n} = \frac{\sum_{n=0}^{N} \pi_n \sigma_n}{\sum_{n=0}^{N} \pi_n (\bar{I}_n + T + 1)}. \quad (10)$$

By Little's result, the average network delay D_n is given by $\bar{n}/S$. As for the access delay D_a, we estimate it here by

$$D_a = \frac{1}{1-B} D_{\text{NPCSMA}}(\lambda) + \frac{B}{1-B}\delta(\lambda) \quad (11)$$

where $D_{\text{NPCSMA}}(\lambda)$ is the average packet delay of an infinite population employing the nonpersistent CSMA protocol and whose output is λ; $\delta(\lambda)$ is the optimum average retransmission delay minimizing $D_{\text{NPCSMA}}(\lambda)$; B is the probability that a T_i-successful packet gets blocked at the receiving repeater and is expressed as

$$B = 1 - \frac{S}{N\lambda}. \quad (12)$$

III. DISCUSSION OF NUMERICAL RESULTS

We show in Table I numerical results obtained for various values of N, λ, and ν. These numerical results show that 1) the performance is not too sensitive to variations in ν (however a very small value of ν ($\nu \leq 0.001$) may induce degradation in performance); 2) the network delay is not much larger than one; and 3) the access delay is the predominant component of packet delay as the throughput increases due to an important increase in blocking. The large values of blocking experienced are mostly due to the lack of synchronization in transmissions between the inner hop and the outer hop, rather than to an inefficient behavior of the inner hop. These results are explained by the fact that with the nonpersistent CSMA, as long as N is not too large ($N \leq 10$), the probability that a transmission is successful is very close to 1. With the IFT protocol used here the repeater is guaranteed that the channel is idle at the end of a correct reception since, given the system connectivity, all repeaters must have been quiet during the entire reception time of the packet. (With a network delay as small as this, there was no need to consider larger values of M, or protocols other than IFT.)

TABLE I
NUMERICAL RESULTS FOR VARIOUS VALUES OF N, λ AND ν.

N	λ	ν	$(D_n)_{min}$	$(D_n)_{max}$	S_{min}	S_{max}	B_{min}	B_{max}
2	0.1	0.001	1.023	1.023	0.1534	0.1512	0.233	0.244
		0.01	1.013	1.013	0.1535	0.1513	0.232	0.243
		0.1	1.012	1.012	0.1535	0.1513	0.232	0.243
		0.5	1.012	1.012	0.1535	0.1513	0.232	0.243
2	0.8	0.001	1.528	1.528	0.4009	0.3544	0.749	0.778
		0.01	1.116	1.116	0.4198	0.3645	0.737	0.772
		0.1	1.077	1.077	0.4232	0.3668	0.735	0.770
		0.5	1.092	1.092	0.4220	0.3659	0.736	0.771
5	0.1	0.001	1.072	1.072	0.2618	0.2476	0.476	0.504
		0.01	1.022	1.022	0.2623	0.2479	0.475	0.504
		0.1	1.017	1.017	0.2624	0.2481	0.475	0.504
		0.5	1.019	1.019	0.2624	0.2480	0.475	0.504
5	0.7	0.001	2.513	2.371	0.4535	0.3461	0.870	0.901
		0.01	1.270	1.252	0.4620	0.3477	0.868	0.900
		0.1	1.163	1.163	0.4685	0.3525	0.866	0.899
		0.5	1.200	1.200	0.4650	0.3505	0.867	0.900

Examining closely the intermediate numerical results, we observe that the stationary distributions Π_{min} and Π_{max} are "identical"[3] for the optimum ν ($\nu \cong 0.1$), and the probability of success $[S_n]_{min}$ and $[S_n]_{max}$ are also very close to each other and close to 1; the average idle periods $[I_n]_{min}$ and $[I_n]_{max}$, on the contrary, show important differences affecting significantly the performance evaluation. To overcome this difficulty we resort to Monte Carlo simulation to estimate S_n and I_n for $n = 0, 1, \cdots, N$ (a much simpler task than a complete simulation of the system); then using equivalently Π_{min} or Π_{max} we derive the performance measures. Let $n^{te} = n$. The algorithm used to generate one sample of I_n, S_n and σ_n is as follows.

1) Generate $N-n$ random variables $\{Y_j'\}_{j=1}^{N-n}$ according to the distribution given in (4). Without loss of generality, we continue to assume that

$$0 = Y_0' \leq Y_1' \leq Y_2' - \cdots \leq Y_{N-n}' \leq Y_{N-n+1}' = \infty.$$

2) $j \leftarrow 0$.

3). Generate a random variable J_n^j such that

$$\Pr\{J_n^j = k\} = [(1-\nu)^n(1-a\lambda')^j]^{(k-1)} \cdot [1-(1-\nu)^n(1-a\lambda')^j] \quad (13)$$

If

$J_n^j < Y_{j+1}' - Y_j'$ then do:

$$I_n = Y_j' + J_n^j \quad (14)$$

[3] Accurate within four decimals (the accuracy used in printing the results).

$$S_n = \frac{n\nu(1-\nu)^{n-1}(1-a\lambda'\bar{y}+ja\lambda'(1-a\lambda'\bar{y}^{-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda'\bar{y}} \quad (15)$$

$$\sigma_n = (I_n + T + 1)n + \frac{a\lambda'(T+1)}{1-(1-\nu)^n(1-a\lambda'\bar{y}} \quad (16)$$

stop;

else $j \leftarrow j+1$; repeat this step.

If L samples are needed, the algorithm is repeated L times. The estimates of $\bar{I}_n$, S_n, σ_n, denoted by $(\bar{I}_n)_{sim}$, $(S_n)_{sim}$, $(\sigma_n)_{sim}$, respectively, are obtained by just taking the average over the L samples. The estimates for the performance measures S and D_n are obtained by using (9), (10), and Little's result in which we substitute $(\bar{I}_n)_{sim}$, $(S_n)_{sim}$, $(\sigma_n)_{sim}$ for I_n, S_n, and σ_n, respectively.

A. The Throughput-Delay Tradeoff

The system capacity is displayed in Fig. 5, and the throughput-delay tradeoff for $N = 2$, 5, and 10, is plotted in Fig. 6. We note a slight improvement in performance as N increases. We also compare in Figs. 5 and 7 the performance of CSMA to that of slotted ALOHA as obtained in Part I[1]. Contrary to the slotted ALOHA case in which we noted that for $N \geq 3$, the inner hop constitutes practically the bottleneck, with CSMA the inner hop is extremely efficient and the terminal-to-repeater hop becomes more critical. As N increases, the input rate λ required at each repeater to produce a given throughput S is smaller, and therefore the "wasted" time on the time lines T_i represented by the variables Y_i' is less important; accordingly it is possible to have a larger number of simultaneous receptions at various repeaters, and therefore to achieve a higher system capacity; moreover, packet delay is lower since the access delay D_a is also smaller with smaller λ. In comparison to slotted ALOHA we note that CSMA offers an improvement which becomes more significant as N increases.

IV. CONCLUSION

We pursued in this paper the analysis of centralized two-hop packet radio systems by considering that devices throughout the system use carrier sense multiple access.

It was shown that, with CSMA, the performance is not too sensitive to the transmission probability as is the case with slotted ALOHA [1]. The network delay is close to one packet transmission time rendering the access delay the predominant component of packet delay. The high levels of blocking experienced are due to the lack of synchronization in transmission between the inner hop and the terminal-access hop rather than to an inefficient behavior of the repeater hop. The results on throughput-delay tradeoffs have shown that in this most unsynchronized transmission mode between inner and outer hops, CSMA manages to provide improved performance over slotted ALOHA, especially when the number of repeaters around the station increases. This improvement, however, is not of the same magnitude as in single-hop systems [2], due to the multihop interference effect. The excellent performance achieved at the repeater hop substantiates the need to consider a buffer size of only

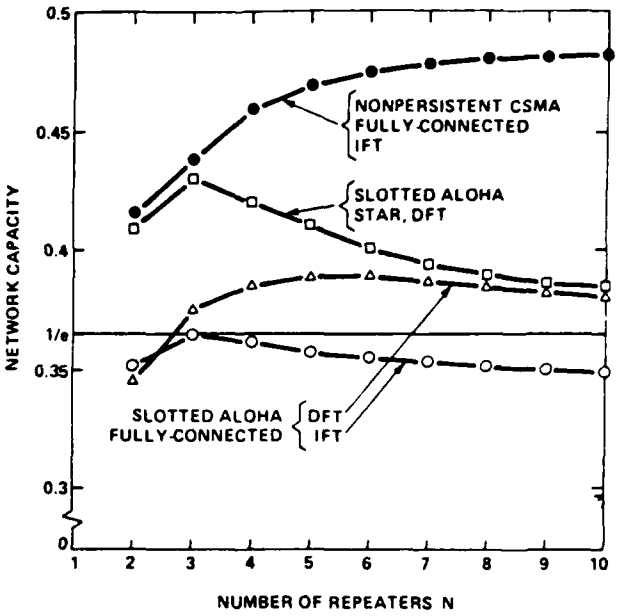


Figure 5. Network capacity versus N for slotted ALOHA and nonpersistent CSMA networks ($a = 0.01$).

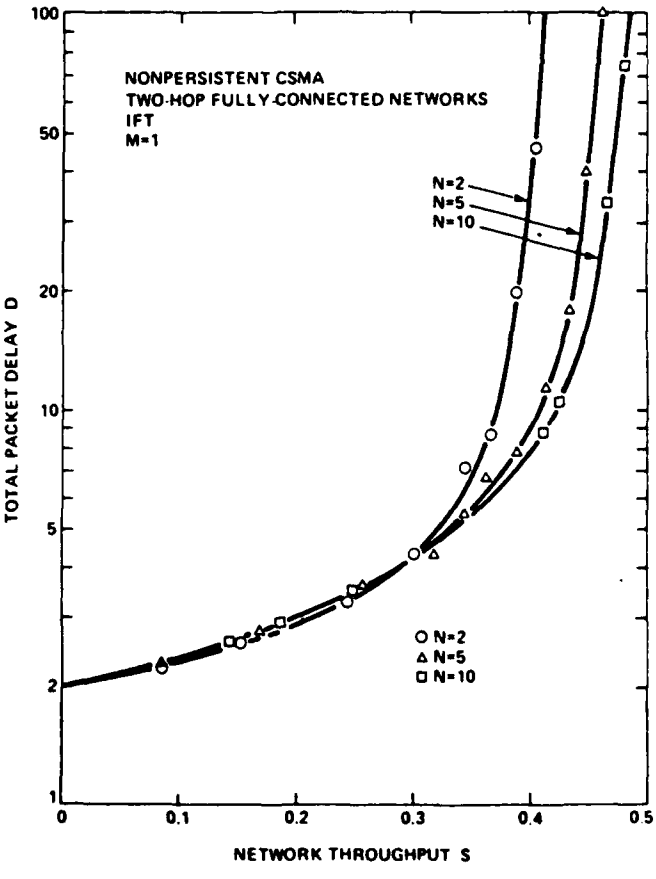


Fig. 6. Throughput-delay tradeoffs in nonpersistent CSMA two-hop fully connected networks ($a = 0.01$).

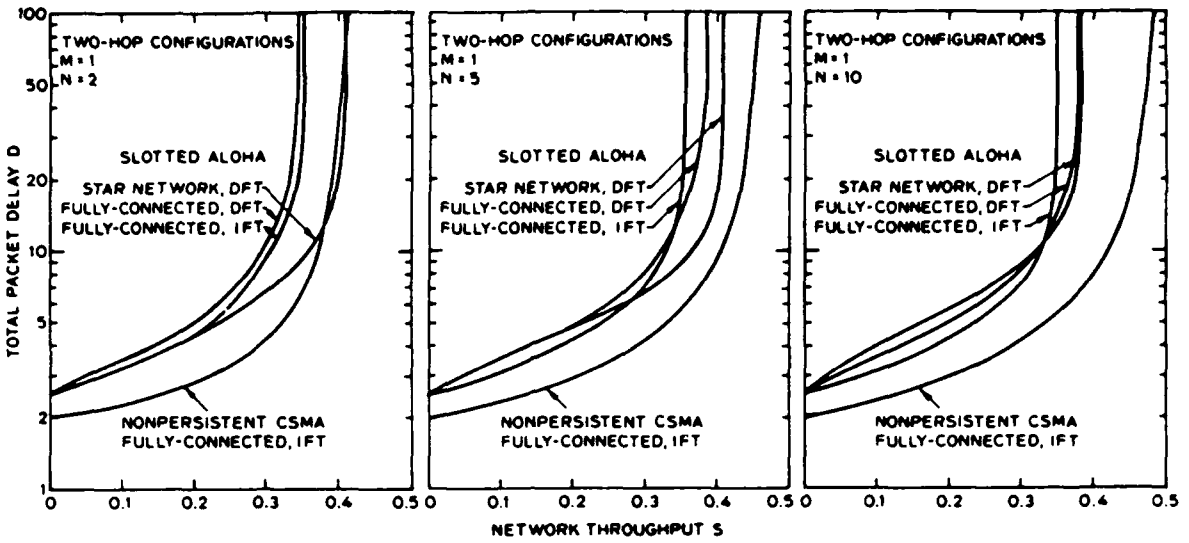


Fig. 7. Comparison between slotted ALOHA and nonpersistent CSMA ($a = 0.01$) for various values of N.

one packet. Moreover, it indicates that with CSMA, contrary to slotted ALOHA, a dynamic control procedure at the repeater-hop would have insignificant effect on the overall system performance.

Finally, we conclude by pointing out that, in order to achieve better performance in these multihop environments, we need more clever schemes which guarantee a higher level of synchronization between terminals and repeaters transmissions; one such solution may be offered by combining min-slotted alternating priorities (MSAP) [5] with a clever use of the busy tone concept.

APPENDIX

TRANSITION MATRIX FOR THE CSMA IFT PROTOCOL

The transition probabilities $p_{n,j}$ between consecutive imbedded points are given by

$$p_{0,j} = \begin{cases} \dfrac{N(a\lambda')(1-a\lambda')^{N-1}}{1-(1-a\lambda')^N} & j=0 \\ 0 & j=1 \\ \dfrac{\binom{N}{k}(a\lambda')^k(1-a\lambda')^{N-k}}{1-(1-a\lambda')^N} & j>1 \end{cases} \tag{A.1}$$

$$p_{N,j} = \begin{cases} 0 & j<N-1 \\ \dfrac{N\nu(1-\nu)^{N-1}}{1-(1-\nu)^N} & j=N-1 \\ 1-\dfrac{N\nu(1-\nu)^{N-1}}{1-(1-\nu)^N} & j=N \end{cases} \tag{A.2}$$

and for $1 \leqslant n \leqslant N-1$

$$p_{n,j} = \begin{cases} 0 & j<n-1 \\ \Pr\{J_n < Y_m'\}\dfrac{n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n} + \Pr\{J_n \geqslant Y_m'\}\dfrac{(1-a\lambda')^{N-n}n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n-1 \\ \Pr\{J_n < Y_m'\}\dfrac{1-n\nu(1-\nu)^{n-1}-(1-\nu)^n}{1-(1-\nu)^n} & \\ \quad + \Pr\{J_n \geqslant Y_m'\}\dfrac{(1-a\lambda')^{N-n}[1-n\nu(1-\nu)^{n-1}-(1-\nu)^n]+(N-n)a\lambda'(1-a\lambda')^{N-n-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n \\ \Pr\{J_n \geqslant Y_m'\}\dfrac{(N-n)a\lambda'(1-a\lambda')^{N-n-1}[1-(1-\nu)^n]}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n+1 \\ \Pr\{J_n \geqslant Y_m'\}\dfrac{\binom{N-n}{j-n}(a\lambda')^{j-n}(1-a\lambda')^{N-j}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j>n+1. \end{cases} \tag{A.3}$$

We now derive the expressions for $\Pr\{J_n \geq Y_{\min}'\}$ and $\Pr\{J_n \geq Y_{\max}'\}$. Given $n'e = n$, and the distribution for Y_j given in (4), we have

$$\Pr\{Y_{\max}' \leq T+y\} = \left[\frac{\lambda}{G} + \left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-n} \qquad 0 \leq y \leq T \tag{A.4}$$

$$\Pr\{Y_{\min}' > T+y\} = \left[\left(1-\frac{\lambda}{G}\right)\left(1-\frac{y}{T}\right)\right]^{N-n} \qquad 0 \leq y \leq T. \tag{A.5}$$

From the distribution of J_n given in (3) we note that

$$\Pr\{J_n \geq k\} = (1-\nu)^{n(k-1)}. \tag{A.6}$$

Using (A.4) through (A.6) we have

$$\Pr\{J_n \geq Y_{\max}'\} = (1-\nu)^{n(T-1)}\left[\left(\frac{\lambda}{G}\right)^{N-n} + (N-n)\cdot\left(1-\frac{\lambda}{G}\right)\frac{1}{T}\int_0^T (1-\nu)^{ny}\cdot\left[\frac{\lambda}{G} + \left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-n-1} dy\right] \tag{A.7}$$

$$\Pr\{J_n \geq Y_{\min}'\} = (1-\nu)^{n(T-1)}\left[1-\left(1-\frac{\lambda}{G}\right)^n + (N-n)\left(1-\frac{\lambda}{G}\right)\frac{1}{T}\int_0^T (1-\nu)^{ny}\cdot\left[1-\frac{\lambda}{G}-\left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-n-1} dy\right] \tag{A.8}$$

We note that the integrals are of a known form, namely

$$\int x^m e^{ax}\,dx = e^{ax}\left[\frac{x^m}{\alpha} + \sum_{k=1}^{m}(-1)^k \cdot \frac{m(m-1)\cdots(m-k+1)}{\alpha^{k+1}}x^{m-k}\right]. \tag{A.9}$$

After some algebra, we get the following expressions:

$$\Pr\{J_n \geq Y_{\max}'\} = (1-\nu)^{n(T-1)}\left(\frac{\lambda}{G}\right)^{N-n} + (1-\nu)^{n(2T-1)}(N-n)\cdot\left[\frac{1}{\alpha} + \sum_{k=1}^{N-n-1}(-1)^k\frac{(N-n-1)!}{(N-n-1-k)!}\frac{1}{\alpha^{k+1}}\right]$$

$$-(1-\nu)^{n(T-1)}(N-n)\left[\frac{(\lambda/G)^{N-n-1}}{\alpha} + \sum_{k=1}^{N-n-1}(-1)^k\frac{(N-n-1)!}{(N-n-1-k)!}\frac{(\lambda/G)^{N-n-1-k}}{\alpha^{k+1}}\right]. \tag{A.10}$$

$$\Pr\{J_n \geq Y_{\min}'\} = (1-\nu)^{n(T-1)}\left[1-\left(1-\frac{\lambda}{G}\right)^{N-n}\right] + (1-\nu)^{n(T-1)}(N-n)\left[(1-\nu)^{nT}\cdot\frac{(N-n-1)!}{\alpha^{N-n}} - \frac{(1-\lambda/G)^{N-n-1}}{\alpha} - \sum_{k=1}^{N-n-1}\frac{(N-n-1)!}{(N-n-1-k)!}\cdot\frac{(1-\lambda/G)^{N-n-1-k}}{\alpha^{k+1}}\right] \tag{A.11}$$

where

$$\alpha = -\frac{T\log[(1-\nu)^n]}{\left(1-\frac{\lambda}{G}\right)}.$$

We are now left with the determination of $\bar{I}_n$. Given that $Y_m' = y$, the average idle period is given by

$$\bar{I}_{n\,|\,Y_m'=y} = \Pr\{J_n < y\}\bar{I}_{n\,|\,J_n<y,\,Y_m'=y} + \Pr\{J_n \geq y\}\left[y + \frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}}\right]. \tag{A.12}$$

Let $n \neq 0, N$; for $0 \leq k \leq y-1$ we have

$$\Pr\{I_n = k \mid Y_m' = y, J_n < y\} = \frac{(1-\nu)^{n(k-1)}[1-(1-\nu)^n]}{1-(1-\nu)^{n(y-1)}}. \tag{A.13}$$

The average idle period in this case is

$$\bar{I}_{n\,|\,Y_m'=y,J_n<y} = \sum_{k=0}^{y-1}\frac{k(1-\nu)^{n(k-1)}[1-(1-\nu)^n]}{1-(1-\nu)^{n(y-1)}} = \frac{1-(1-\nu)^{ny}-y(1-\nu)^{n(y-1)}[1-(1-\nu)^n]}{[1-(1-\nu)^n][1-(1-\nu)^{n(y-1)}]}. \tag{A.14}$$

Thus, for $n \neq 0, N$,

$$\bar{I}_{n|Y_m'=y} = \frac{1-(1-\nu)^n - y(1-\nu)^{n(y-1)}[1-(1-\nu)^n]}{1-(1-\nu)^n} + y(1-\nu)^{n(y-1)} + \frac{(1-\nu)^{n(y-1)}}{1-(1-\nu)^n(1-a\lambda')^{N-n}}$$

$$= \frac{1}{1-(1-\nu)^n} + (1-\nu)^{n(y-1)} \cdot \left[\frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}} - \frac{(1-\nu)^n}{1-(1-\nu)^n}\right]. \quad \text{(A.15)}$$

Removing the condition on Y_m', we finally have

$$\bar{I}_n = \frac{1}{1-(1-\nu)^n} + \Pr\{J_n > Y_m'\} \cdot \left[\frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}} - \frac{(1-\nu)^n}{1-(1-\nu)^n}\right] \quad \text{(A.16)}$$

where Y_m' can be replaced by either $Y_{\min}'$ or $Y_{\max}'$.

When $n = 0$, $\Pr\{J_n < y\} = 0$ and (A.12) is written as

$$\bar{I}_{0/Y_m'=y} = y + \frac{1}{1-(1-a\lambda')^N}. \quad \text{(A.17)}$$

Removing the condition on Y_m', we get for the lower-bound case

$$\bar{I}_{0,\max} = \frac{1}{1-(1-a\lambda')^N} + T + N\left(1-\frac{\lambda}{G}\right)\frac{1}{T} \cdot \int_0^T y\left[\frac{\lambda}{G} + \left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-1} dy$$

$$= \frac{1}{1-(1-a\lambda')^N} + T + N\left(1-\frac{\lambda}{G}\right) \cdot T\left[\sum_{k=0}^{N-1}\binom{N-1}{k}\frac{(\lambda/G)^k(1-\lambda/G)^{N-1-k}}{N-1-k+2}\right] \quad \text{(A.18)}$$

and for the uper bound case

$$\bar{I}_{0,\min} = \frac{T}{1-(1-a\lambda')^N} + T + N\left(1-\frac{\lambda}{G}\right)\frac{1}{T} \cdot \int_0^T y\left[\left(1-\frac{\lambda}{G}\right) - \left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-1} dy$$

$$= \frac{1}{1-(1-a\lambda')^N} + T + N\left(1-\frac{\lambda}{G}\right)^N \cdot T\left[\sum_{k=0}^{N-1}\binom{N-1}{k}\frac{(-1)^{N-1-k}}{N-1-k+2}\right]. \quad \text{(A.19)}$$

For the case $n = N$, we simply have

$$\bar{I}_N = \frac{1}{1-(1-\nu)^N}. \quad \text{(A.20)}$$

REFERENCES

[1] F. Tobagi, "Analysis of a two-hop centralized packet radio network—Part I: Slotted ALOHA," this issue, pp. 196–207.

[2] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I—carrier sense multiple access modes and their throughput delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, pp. 1400–1416, Dec. 1975.

[3] F. 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.*, vol. COM-23, pp. 1417–1433, Dec. 1975.

[4] ——, "Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-25, pp. 1103–1120, Oct. 1977.

[5] M. Scholl, "Multiplexing techniques for data transmission over packet switched radio systems," Comput. Sci. Dep., Univ. California, Los Angeles, UCLA-ENG-76123, Dec. 1976.

★

Fouad A. Tobagi (M'77), for a photograph and biography, see this issue, page 207.

Proceedings of INFOCOM'83
San Diego, April 19-21, 1983

THROUGHPUT ANALYSIS OF MULTIHOP PACKET RADIO NETWORKS UNDER VARIOUS CHANNEL ACCESS SCHEMES*

Fouad A. Tobagi and José M. Brázio

Computer Systems Laboratory
Departments of Computer Science and Electrical Engineering
Stanford University
Stanford, CA 94305
(415) 497-1708

ABSTRACT

This paper presents the analysis of throughput for multihop packet radio networks with zero propagation delay, employing any of a number of channel access schemes. The model considered is Markovian in nature and is an extension of that used by R. Boorstyn and A. Kershenbaum for CSMA. The protocols examined are pure ALOHA, carrier sense multiple access (CSMA), busy tone multiple access (BTMA), and code division multiple access (CDMA). The general model is presented, and its underlying assumptions are outlined. Simple network configurations are considered to compare the performance of these access schemes, and in addition a discussion of numerical methods for the application of the analysis to general configurations is given.

1. INTRODUCTION

Until recently, the work done on the performance of multiaccess schemes focused mainly on the single-hop case, leading to a good understanding of the behavior of one-hop networks. Several access schemes designed specifically for single-hop networks or shown to perform particularly well in single hop networks may suffer severe degradation in performance in the multihop environment. One such example is carrier sense multiple access (CSMA) with its "hidden terminal" problem [1]. Several schemes have been proposed for multihop networks in view of providing improved performance, but no analysis has yet been performed to evaluate these schemes. Recently, a model has been developed by Boorstyn and Kershenbaum [2] to analyze CSMA in a multihop environment. In the present paper, we extend the model used for CSMA to evaluate a few other multiaccess schemes and compare their performance. In Section 2 we present the model and its assumptions. Then in Section 3, we describe the access schemes considered in this paper. The details of the analysis are given in Section 4, and applied to specific examples in Section 5. Finally, in Section 6 we discuss some computational aspects of this approach.

*This work was supported in part by the Defense Advanced Research Projects Agency under contract MDA903-79-C-0201, order A03717, monitored by the Office of Naval Research. José Brázio is on a fellowship from the Instituto Nacional de Investigação Cientifica, Lisbon, Portugal.

2. GENERAL MODEL

We consider a network with N nodes, numbered 1, 2, ..., N, and a "hearing matrix" $H = [h_{ij}]$ where

$$h_{ij} = \begin{cases} 1 & \text{if } j \text{ can hear } i \\ 0 & \text{otherwise} \end{cases}$$

and $h_{ij} = h_{ji}$. As in [2], we define $N^*(i)$ to be the set of neighbors of i, not including i, and $N(i)$ to be $N^*(i) \bigcup \{i\}$. Given the external traffic requirements and a static routing function, we can compute the rate of the messages to be successfully transmitted between any pair of neighbors. We denote by S_{ij} the throughput from i to j; S_{ij} is the average number of messages successfully transmitted by node i to neighboring node j during the average duration of a message transmitted by i. Each message may have to be transmitted several times before a success takes place. We assume the message lengths to be exponentially distributed with mean $1/\mu_i$ for messages transmitted by node i, and to be redrawn independently from the corresponding distribution each time the message is transmitted. We also assume infinite buffer space for each node, instantaneous and perfect acknowledgements, and zero propagation and processing delays.

Typically in a packet radio network, a node considers a packet in its queue (if any) for transmission at some scheduled point in time. Transmission of the packet may or may not take place depending on the protocol in use. If transmission is inhibited or unsuccessful (due to a collision at the intended neighbor) then the packet is rescheduled for a later time, chosen according to a random rescheduling interval. If the packet is transmitted successfully, then it is removed from the queue and a new scheduling point is chosen, at which time another packet (if any) is taken for consideration. In this study we consider the process of scheduling points for messages from node i to node j to be Poisson with rate λ_{ij}, and independent of any other scheduling point process in the network. Furthermore, we assume that at every scheduling point there is a message in the queue for consideration. With these assumptions the global process of scheduling points for node i is also Poisson with rate $\nu_i = \sum_{j \in N^*(i)} \lambda_{ij}$.

3. DESCRIPTION OF THE CHANNEL ACCESS SCHEMES

Among the many multihop access schemes which can be conceived today using such features as carrier sensing, code division, etc., we have selected a few for consideration in this paper, which lend themselves to simple solutions (particularly *product form solutions*). Although other schemes can be handled by the model described above, the analysis becomes more complex, and at this early stage of this research, we restricted ourselves to those described below.

The access schemes define the conditions under which a scheduling point results in a transmission. We divide them into two major groups: the carrier sense type schemes, and the ALOHA-type schemes. In addition, we define capture as the ability for a receiver to correctly receive a packet despite the presence of other time-overlapping transmissions. Perfect capture refers to the ability of receiving correctly the first message to reach the receiver regardless of the number of future overlapping messages; zero capture refers to the situation where any overlap in transmission results in complete destruction of all overlapping transmissions.

A. Carrier Sense Type Schemes

(a) Carrier Sense Multiple Access (CSMA):

A node always senses the channel before transmission. A scheduling point in time results in a transmission if neither the node nor any of its neighbors are transmitting. (This is the scheme studied in [2].) As discussed in the following section, the analysis of this scheme is tractable only if we assume perfect capture. Note that with zero propagation delay, as soon as a node initiates a transmission, all its neighbors are inhibited. The neighbors of its intended receiver which are not neighbors of the transmitter, and referred to as the *hidden nodes* with respect to this transmission, are not inhibited by this transmission, and may very well initiate a transmission any time following the start of the one in question. Due to the perfect capture assumption, these later transmissions do not affect the correct reception of the earlier message (provided that it was indeed the first to reach the intended receiver). If these overlapping transmissions are intended to neighbors of the original transmitter then they are wasted and, in such a situation, the period of time occupied by them could be better used for receiving packets from neighbors which are not already inhibited. Moreover, the presence of these "useless transmissions" blocks other nodes from transmitting because of carrier sensing, nodes which could have originated successful transmissions coexisting with the original one. If these additional transmissions are intended for nodes other than neighbors of the original transmitter, then they may be successful depending on the situation encountered at their intended receiver. Clearly, if zero capture is in effect, then regardless of which the intended receivers of the additional transmissions are, their presence causes destruction of the earlier packet.

(b) Busy Tone Multiple Access (BTMA) [1, 3]:

With the help of a busy tone transmitted on a separate frequency by the receiver, one can guarantee that the hidden nodes are inhibited from transmitting by sensing the busy tone. Under the assumptions of zero propagation delay we have the following properties: (i) capture effects are of *no* relevance; (ii) as compared to CSMA with zero-capture, the busy tone guarantees correct reception of the original packet; (iii) as compared to CSMA with perfect capture, it prevents the additional overlapping transmissions and the associated "wasted time" alluded to in the previous paragraph.

Several BTMA schemes exist depending on which nodes transmit the busy tone [3]. In the conservative BTMA scheme (C-BTMA), any node which senses carrier emits a busy tone. Thus, if node A transmits a packet to node B, B and all other neighbors of A transmit the busy tone and block all nodes in a region within twice the hearing radius from node A. In the so-called Ideal BTMA (I-BTMA) only node B transmits the busy tone blocking only its neighbors. (Clearly this requires a node to know a priori that it is the intended receiver, and may not be easily implementable.) A more realistic scheme is to use C-BTMA (or no busy tone at all) during the reception of the header portion of a message, which allows to decode the intended receiver's address, followed by I-BTMA during the remainder of the message transmission. In the following analysis, only C-BTMA will be considered.

(c) A Directional CSMA* (D-CSMA):

It is assumed here that a node wishing to transmit a packet knows the state (busy transmitting, or idle not transmitting) of the intended receiver. The transmission takes place if the node itself is neither transmitting nor receiving a packet intended for it and the intended receiver is not already transmitting.

The implementation of such a scheme may be possible if the carrier sensing operation is made function of individual nodes, such as for example by directional carrier sensing, or via the use of a uniquely *coded* "busy tone" transmitted by a node which is busy transmitting.

B. ALOHA-Type Schemes

In this type of schemes, a node does not sense the channel before transmitting. Two protocols are considered for analysis here:

(a) Pure ALOHA:

A node is allowed to transmit if and only if it is not already transmitting. This implies that if the node is busy receiving (either a valid packet or just carrier) then it aborts

*For comparative purposes, and to isolate the various elementary effects on the overall performance, we considered some hypothetical situations which may not necessarily correspond to realistic implementations. However, one constraint in the selection of these situations is again the fact that they should lend themselves to a tractable solution.

that operation and initiates transmission of the packet. For pure ALOHA both perfect capture and zero capture can be accommodated by the model.

(b) CDMA-ALOHA:

In addition to the above rules, it is considered here that a code division scheme (CDMA) is used whereby each node is assigned a unique code for reception. Nodes wishing to transmit to a particular intended node must use the code assigned to that node. Given the operation of CDMA, perfect capture can be assumed. (This is the case since whenever a receiver locks on to a particular packet, then all future overlapping transmissions can be ignored; clearly this requires that the preamble of the first packet, or a portion of it, be received free of collision; but we ignore this effect here.) However, we are making here the *pessimistic* assumption that an ongoing reception is aborted if a scheduling point for transmission is encountered (hence the name CDMA-ALOHA).

Moreover, given the operation of CDMA, a node which has completed the reception of a packet onto which its receiver was locked is ready to receive another packet, regardless of the state of the channel, i.e., despite the presence of other transmissions, with the same code, but onto which the receiver is not already locked. However, for tractability of analysis at the present stage of this research, we make the second *pessimistic* assumption that the receiver cannot lock onto a new packet until it has encountered an idle period. Such assumptions are definitely restrictive, and the results obtained under them are certainly not a good representation of the performance of CDMA schemes.

4. ANALYSIS

A. The State Space

It can be seen from the above description of the schemes under consideration that with the exception of D-CSMA, in defining a state description for the system we only need to take into account the transmit status of each node, which can be busy transmitting or idle not transmitting. Indeed, in the ALOHA schemes, the receive status of a node is completely ignored, while in the CSMA schemes (other than D-CSMA) it is implied from the transmit status of the neighbors.

We define the state of the system at time t, $X(t)$, to be the set of nodes busy transmitting at that time. Since, depending on the particular protocol, not all subsets of nodes may be busy transmitting simultaneously, the state space S is function of the protocol. For example, in a 4-node chain as shown in Fig. 1, the state space S for the various schemes is:

CSMA: $S = \{\phi, \{1\}, \{2\}, \{3\}, \{4\}, \{1,3\}, \{1,4\}, \{2,4\}\}$

C-BTMA: $S = \{\phi, \{1\}, \{2\}, \{3\}, \{4\}, \{1,4\}\}$

pure and CDMA-ALOHA: $S =$ power set of $\{1,2,3,4\}$

Let $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$, and $D \in S$. We define

$$P(D) \triangleq \left\{ i \text{ such that } i \in \mathcal{N}, i \notin D, D \bigcup \{i\} \in S \right\}$$

that is, $P(D)$ is the set of nodes which are not blocked when the state of the system is D. We note that for any $j \in D$ we have $D - \{j\} \in S$ and $j \in P(D - \{j\})$.

Fig. 1. A 4-node chain.

B. The Markov Chain

Given that $X(t) = D \in S$, and due to the assumptions of exponential message lengths and Poisson nature of the scheduling point process, the state at time $t + \Delta t$ is given by

$$X(t+\Delta t) = \begin{cases} D \bigcup \{i\}, & i \in P(D), \\ & \text{with probability } \lambda_i \Delta t + o(\Delta t) \\ D - \{j\}, & j \in D, \\ & \text{with probability } \mu_j \Delta t + o(\Delta t) \\ D, & \text{with probability} \\ & 1 - (\lambda_i + \mu_j)\Delta t + o(\Delta t) \end{cases}$$

Thus $X(t)$ is a continuous time Markov chain. Letting $Q(D)$ denote the steady state probability of state D, the following balance equations hold:

$$Q(D)\left[\sum_{j \in D} \mu_j + \sum_{i \in P(D)} \lambda_i\right] = \sum_{j \in P(D)} \mu_j Q(D \bigcup \{j\}) + \sum_{i \in D} \lambda_i Q(D - \{i\}),$$

for all $D \in S$.

It is easy to check that the expression

$$Q(D) = \left(\prod_{i \in D} G_i\right) Q_0$$

with

$$G_i = \frac{\lambda_i}{\mu_i}$$

satisfies the balance equation and is therefore the solution. The constant Q_0 is the normalization factor, given by

$$Q_0 = \left[\sum_{D \in S} \left(\prod_{i \in D} G_i\right)\right]^{-1}$$

and equals the probability of finding all the nodes idle.

Reversibility

A Markov chain is reversible [4] if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$ for all real $t_1, t_2, \ldots, t_n, \tau$. ($X(\cdot)$ is assumed to be defined from $-\infty$ to $+\infty$.) This is equivalent to the condition that, for any finite sequence of states $D_1, D_2, \ldots, D_n \in S$,

$$q(D_1, D_2)q(D_2, D_3)\cdots q(D_n, D_1) = q(D_1, D_n)q(D_n, D_{n-1})\cdots q(D_3, D_2)q(D_2, D_1)$$

where $q(i, j)$ is the rate of transitions from i to j. An equivalent condition for reversibility is the existence of a stationary distribution $Q(D)$ such that

$$Q(D_1) \cdot q(D_1, D_2) = Q(D_2) \cdot q(D_2, D_1) \quad \text{for all } D_1, D_2 \in S.$$

This last characterization implies that the steady state distribution has a very simple product form, as illustrated in the previous paragraph. In this paper we selected for analysis only schemes whose Markov chains are reversible, having used the two above results to check for reversibility and to derive the steady-state distributions.

C. The Throughput Equations

Given that the link traffic is Poisson with rate λ_{ij} for the link from i to j, each scheduling point is a random look in time. If we define $G_{ij} = \frac{\lambda_{ij}}{\mu_i}$, then the ratio S_{ij}/G_{ij} is merely the probability that a scheduling point from node i results in a successful transmission. Thus for nodes i and j such that $h_{ij} = 1$ we can write

$$\frac{S_{ij}}{G_{ij}} = \Pr\{\text{scheduling point from node } i \text{ results in a transmission and the transmission is successful at } j\}.$$

Note also that

$$G_i = \sum_{j \in N^*(i)} G_{ij}.$$

Given a protocol and the steady-state solution for its Markov chain, we can evaluate the above probability in a straightforward manner.

(a) CSMA and C-BTMA:

CSMA:

In the remainder of this paper, we assume CSMA to have the perfect capture property. Then

$$\frac{S_{ij}}{G_{ij}} = \Pr\{\text{no nodes in } N(i) \text{ or } N(j) \text{ busy transmitting}\}$$

C-BTMA:

$$\frac{S_{ij}}{G_{ij}} = \Pr\{\text{no nodes within 2 hops of } i \text{ busy transmitting}\}$$

Example: Consider again the 4-node chain of Figure 1. Given the state space as defined above, we can write the following equations:

CSMA [2]:

$$Q_0 = [1 + G_1 + G_2 + G_3 + G_4 + G_1G_3 + G_1G_4 + G_2G_4]^{-1}$$

$$\frac{S_{12}}{G_{12}} = \frac{S_{21}}{G_{21}} = \Pr\{1, 2, \text{and } 3 \text{ not busy transmitting}\} = (1 + G_4)Q_0$$

$$\frac{S_{23}}{G_{23}} = \frac{S_{32}}{G_{32}} = \Pr\{1, 2, 3, \text{and } 4 \text{ not busy transmitting}\} = Q_0$$

$$\frac{S_{34}}{G_{34}} = \frac{S_{43}}{G_{43}} = \Pr\{2, 3, \text{and } 4 \text{ not busy transmitting}\} = (1 + G_1)Q_0$$

C-BTMA:

$$Q_0 = [1 + G_1 + G_2 + G_3 + G_4 + G_1G_4]^{-1}$$

$$\frac{S_{12}}{G_{12}} = \Pr\{1, 2, \text{and } 3 \text{ not busy transmitting}\} = (1 + G_4)Q_0$$

$$\frac{S_{21}}{G_{21}} = \frac{S_{23}}{G_{23}} = \frac{S_{32}}{G_{32}} = \frac{S_{34}}{G_{34}} = \Pr\{1, 2, 3, \text{and } 4 \text{ not busy transmitting}\} = Q_0$$

$$\frac{S_{43}}{G_{43}} = \Pr\{2, 3, \text{and } 4 \text{ not busy transmitting}\} = (1 + G_1)Q_0$$

If in addition we specify a traffic pattern, then we can reduce the above expressions and write them in terms of only one independent variable G_{ij}. For example, requiring the link throughput pattern to be uniform, i.e.,

$$S_{12} = S_{21} = S_{23} = S_{32} = S_{34} = S_{43} = S$$

and taking G_1 as the independent variable, we get

CSMA: $S = \frac{G_1}{1+5G_1+2G_1^2}$ with $S_{\max} = 0.128$

C-BTMA: $S = \frac{1}{5}\left[1 - \frac{1}{5G_1+1}\right]$ with $S_{\max} = 0.2$

In Figure 2, we plot S versus G_1 for the above examples.

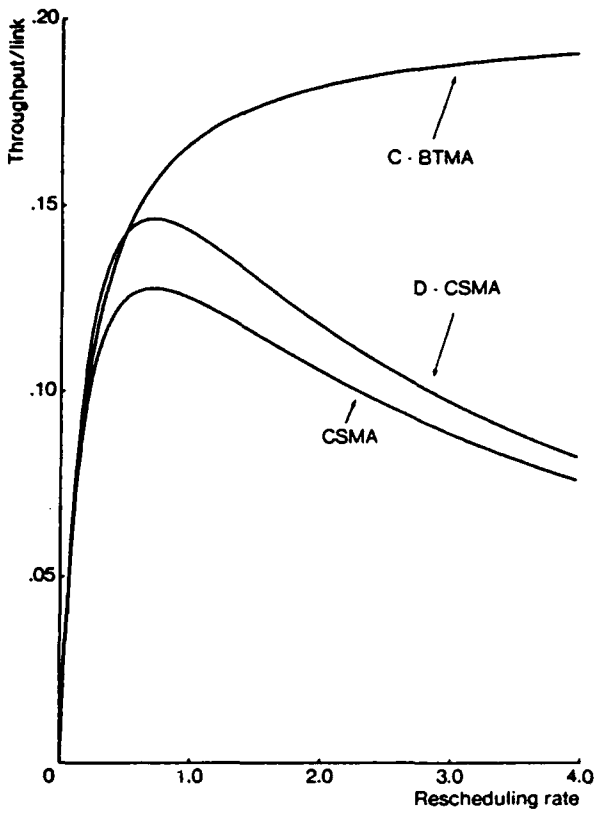


Fig. 2. S vs. G_1 for CSMA, D-CSMA, and C-BTMA for a 4-node chain.

(b) ALOHA-Type Schemes:

Here $S = 2^N$, and all nodes act independently. For every node i, we simply have

$$\Pr\{i \text{ not busy transmitting}\} = \frac{1}{1+G_i}$$

Thus $Q(D)$ is given by

$$Q(D) = \prod_{i \in D}\left(\frac{G_i}{1+G_i}\right)\prod_{j \notin D}\left(\frac{1}{1+G_j}\right).$$

For the throughput we also need the probability of node j not starting a transmission during node i's transmission, given that j is not transmitting when i starts transmitting. This probability is just

$$\mu_i(\mu_i + \lambda_j)^{-1} = \left(1 + \frac{\mu_j}{\mu_i}G_j\right)^{-1}.$$

The probability of success is given by:

Pure ALOHA with zero capture:

$$\frac{S_{ij}}{G_{ij}} = \Pr\{i,j \text{ and } N^*(j) \text{ not busy transmitting, and no node in } N(j) \text{ transmits during the entire transmission from } i \text{ to } j\}$$

$$= \prod_{k \in N(j)}\left(\frac{1}{1+G_k}\right)\prod_{\substack{l \in N(j) \\ l \neq i}}\left(\frac{1}{1+\frac{\mu_l}{\mu_i}G_l}\right)$$

Pure ALOHA with perfect capture:

$$\frac{S_{ij}}{G_{ij}} = \Pr\{N(j) \text{ not busy transmitting, and } j \text{ does not start a transmission during } i\text{'s transmission to it}\}$$

$$= \left(\frac{1}{1+\frac{\mu_j}{\mu_i}G_j}\right)\prod_{k \in N(j)}\left(\frac{1}{1+G_k}\right).$$

CDMA-ALOHA:

Noting that

$$\Pr\{k \text{ is transmitting a packet to } j\} = \Pr\{k \text{ is busy transmitting}\} \cdot \Pr\{k \text{ is transmitting to } j \mid k \text{ is busy transmitting}\}$$

$$= \frac{G_k}{1+G_k}\cdot\frac{G_{kj}}{G_k} = \frac{G_{kj}}{1+G_k},$$

we have that

$$\frac{S_{ij}}{G_{ij}} = \Pr\{i,j \text{ not busy transmitting, } j \text{ does not start a transmission during } i\text{'s transmission to it, no other neighbor of } j \text{ busy transmitting to } j\}$$

$$= \frac{1}{1+G_i}\frac{1}{1+G_j}\frac{1}{1+\frac{\mu_j}{\mu_i}G_j}\prod_{\substack{k \in N^*(j) \\ k \neq i}}\left[1 - \frac{G_{kj}}{1+G_k}\right]$$

D. Extension of the Model to D-CSMA

In order to handle D-CSMA, the state of a node (busy transmitting or idle not transmitting) is not sufficient, and must be augmented to include the intended destination. If we denote by (i,j) a transmission from node i to node j, the state of the system at time t becomes the set of ongoing link transmissions (i,j) at time t. With such a system state definition, we again obtain a continuous time Markov chain with steady-state probabilities of the form

$$Q(D) = \left[\prod_{(i,j) \in D} G_{ij}\right]Q_0$$

with

$$Q_0 = \left[\sum_{D \in S}\left(\prod_{(i,j) \in D} G_{ij}\right)\right]^{-1},$$

where $D \in S$ is a set of allowable simultaneous link transmissions.

Example: For the 4-node chain with uniform link throughput pattern, we get the following equations:

$$\begin{aligned} Q_0 &= [1 + G_{12} + G_{21} + G_{23} + G_{32} + G_{34} \\ &\quad + G_{43} + G_{12}G_{32} + G_{12}G_{34} + G_{12}G_{43} \\ &\quad + G_{21}G_{34} + G_{21}G_{43} + G_{23}G_{43}]^{-1} \\ &= [G + G_1 + G_2 + G_3 + G_4 + G_1G_3 \\ &\quad + G_1G_4 + G_2G_4 + G_{21}G_{34}]^{-1} \end{aligned}$$

and

$$S = \frac{G_1}{1 + 4G_1 + 2G_1^2}$$

leading to a maximum throughput of $S_{\max} = 0.146$. S versus G_1 is plotted in Figure 2.

E. Model Limitations

A number of other schemes of interest for multihop packet radio networks do not lend themselves to a reversible Markov chain, and thus do not lead to a simple product form solution. This was observed to be the case for example in the ALOHA schemes if a node is inhibited from initiating a transmission if it is receiving a packet (and thus requiring the state of a node to be one of three possibilities: either transmitting or receiving or idle). This was also the case for I-BTMA where only the intended receiving node transmits a busy tone. The solution for such Markov chains require different numerical methods than those discussed in this paper and is the subject of an ongoing investigation.

5. APPLICATIONS

We have already discussed in the previous sections the example of a 4-node chain. In this section we examine a few other simple configurations to compare the performance of the access schemes. In all cases we assume the mean packet length to be the same for all nodes in the network.

First we consider a two-node network. This is a fully connected network, for which all CSMA-type schemes achieve full utilization in the case of zero propagation delay. Also in this simple case all ALOHA-type schemes perform identically, for both the cases of zero capture and perfect capture. Figure 3 shows, on the (S_1, S_2) plane, the bounds of the feasible regions for these schemes. Also included, for reference purposes, is the curve $\sqrt{S_1} + \sqrt{S_2} = 1$, corresponding to slotted ALOHA [5].

The second configuration we consider, shown in Figure 4 and for which we will be especially interested in comparing the performance of CSMA and C-BTMA, consists of a fully connected network of N nodes (1 through N), plus an additional node (numbered 0) connected only to node 1. For this network we could expect CSMA to perform better than C-BTMA. Indeed, with C-BTMA, only one transmission in the entire network can take place. With CSMA, it is always possible to have one node in the set

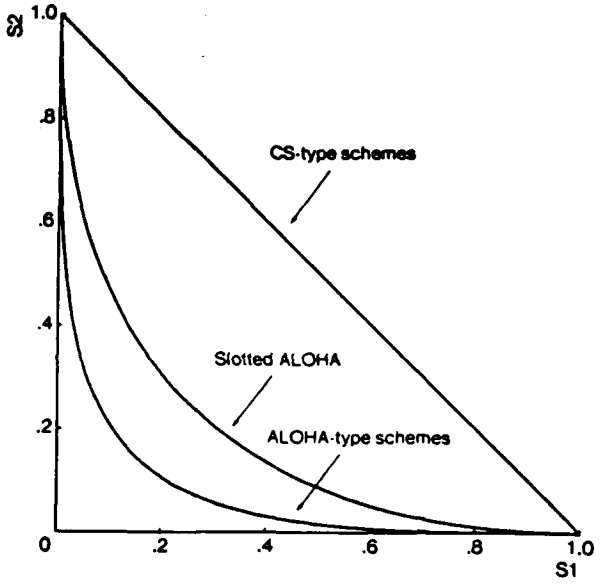


Fig. 3. Feasible regions for a 2-node chain.

$\{2, 3, \ldots, N\}$ transmit simultaneously with node 0. If such a transmission is destined to node 1, then either this transmission or node 0's transmission is wasted but not both (due to the perfect capture assumption in CSMA). If such a transmission is destined to some node other than node 1, then it is conceivable to have two simultaneous successful transmissions. However, using the analytical approach, our expectations do not come true. If we define the traffic pattern by a collection of numbers $\{\alpha_{ij}\}$ such that $\sum_{\text{all pairs of neighbors}} \alpha_{ij} = 1$ and $S_{ij} = \alpha_{ij} S$ for some $S \geq 0$, and determine the maximum S, we find that for C-BTMA we have $S_{\max} = 1$, but that for CSMA we get

$$S_{\max} = \frac{1}{1 + 2\sqrt{\alpha_{01}a_1}},$$

with

$$a_1 = \sum_{j=2}^{N} \alpha_{j1}.$$

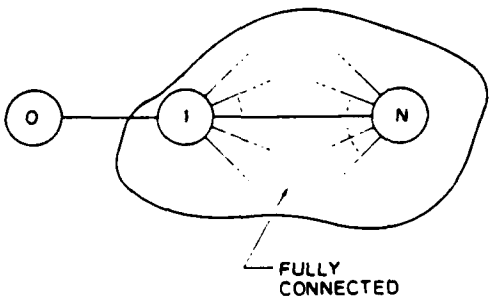


Fig. 4. An "almost fully connected" network.

The performances of CSMA and C-BTMA are equal when either $\alpha_{01} = 0$ or $a_1 = 0$. Otherwise, C-BTMA always performs better than CSMA; this is due to the time wasted in useless transmissions alluded to in Section 3. It is interesting to see that when $a_1 = 0$ the transmissions by nodes 1, 2, ..., N are always successful, but the transmissions by node 0 may have to be repeated because of a transmission by some node j, $2 \leq j \leq N$, in progress when the transmission by node 0 starts. The useless transmissions of 0 have no effect on the throughput of the rest of the network, and so we do not have the decrease in performance that occurs when, for some $j = 2, \ldots, N$, $\alpha_{j1} > 0$. At the maximum S and as $a_1 \to 0$, both G_0 and G_1 go to infinity, but $G_2, \ldots, G_N$ remain finite, so that node 0 can blindly try to "sneak in" the start of a transmission between the transmissions of $2, \ldots, N$ (sort of keeping "shooting in the dark"), achieving $S_{max} = 1$. Thus the golden rule seems to be "say it once and for all", which can be achieved with the help of the busy tone. As a numerical example, we consider the case of a uniform traffic pattern

$$\alpha_{01} = \alpha_{10} = \alpha_{ij} = (N^2 - N + 2)^{-1},$$
$$i, j \in \{1, 2, \ldots, N\}, i \neq j.$$

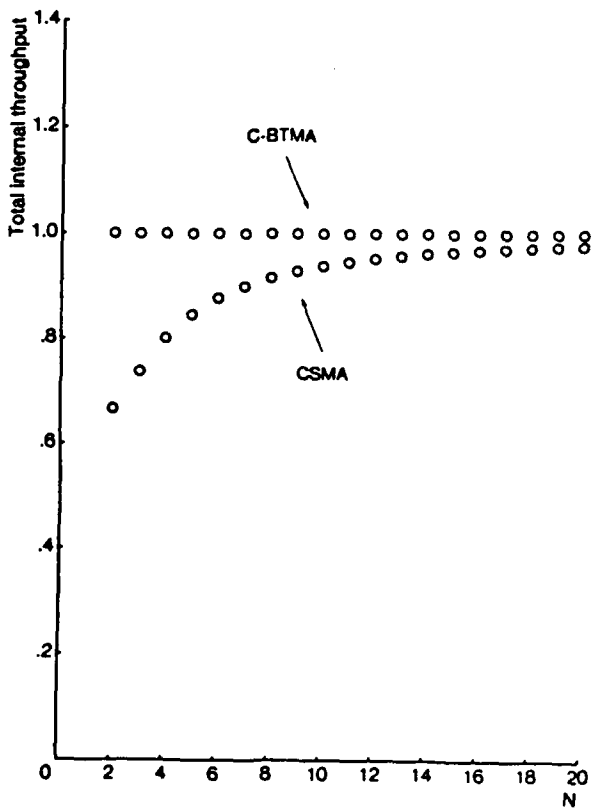


Fig. 5. S_{max} vs. N for the "almost fully connected" network.

In Figure 5 we plot S_{max} versus N for both CSMA and C-BTMA. As N increases the effects discussed above become insignificant, and CSMA and C-BTMA converge in performance.

Next we consider a ring of N nodes ($N \geq 3$), as in Figure 6, in which all nodes behave identically, i.e., $S_{i,(i+1) \bmod N} = \alpha S$, $S_{(i+1) \bmod N, i} = (1 - \alpha) S$, $i = 1, \ldots, N$, $0 \leq \alpha \leq 1$. For all protocols but CDMA-ALOHA the maximum value of S is independent of α. For CDMA-ALOHA, we show in Figure 7 the maximum value of S as a function of α. This value does not depend on the number of nodes in the ring. We see that the maximum S occurs for either $\alpha = 0$ or $\alpha = 1$, thus suggesting that a unidirectional ring would perform better under CDMA-ALOHA than a bidirectional ring. However, and depending on the exogenous traffic pattern, it may happen that the increase in throughput for the unidirectional ring does not compensate for the increase in the average path length. For an example of the relative performance of the different schemes on rings of various sizes we consider the case $\alpha = 1/2$, corresponding to all link throughputs being equal. This can be obtained, in particular in the following ways: (i) if source destination pairs are neighboring nodes and their traffic requirement is uniform, and (ii) if all pairs of nodes are source-destinations with equal end-to-end throughput requirements, and the routing procedure is one which balances the link traffic. Figure 8 shows the maximum link throughput $S_{max}/2$ as a function of N for all schemes described. This is also the maximum end-to-end throughput for each source-destination pair in case (i). Figure 9 shows the sum of all end-to-end throughputs in case (ii).

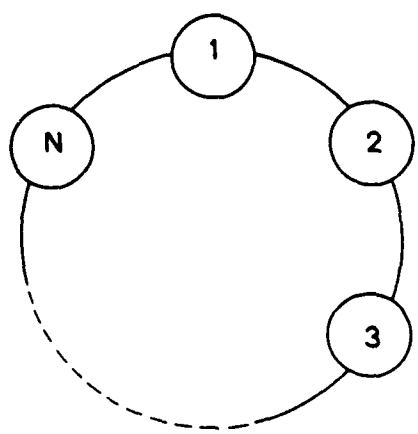


Fig. 6. A ring with N nodes.

As expected, under the ALOHA protocols the throughput is independent of the number of nodes. Under CSMA there is a different behavior for rings with even and odd number of elements: the throughput for rings with an odd number of nodes decreases as the number of nodes increases, and is always higher than the throughput for rings with

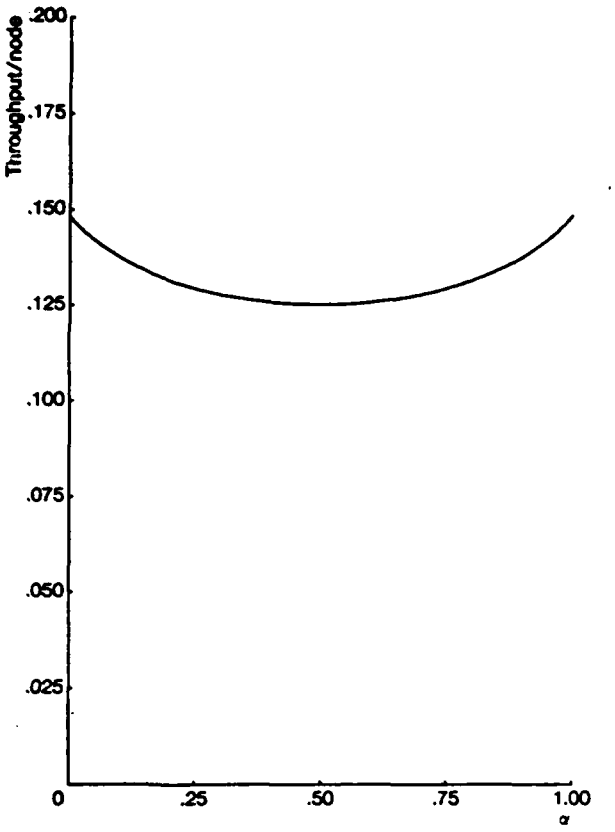


Fig. 7. S_{max} vs. α for a ring under CDMA-ALOHA.

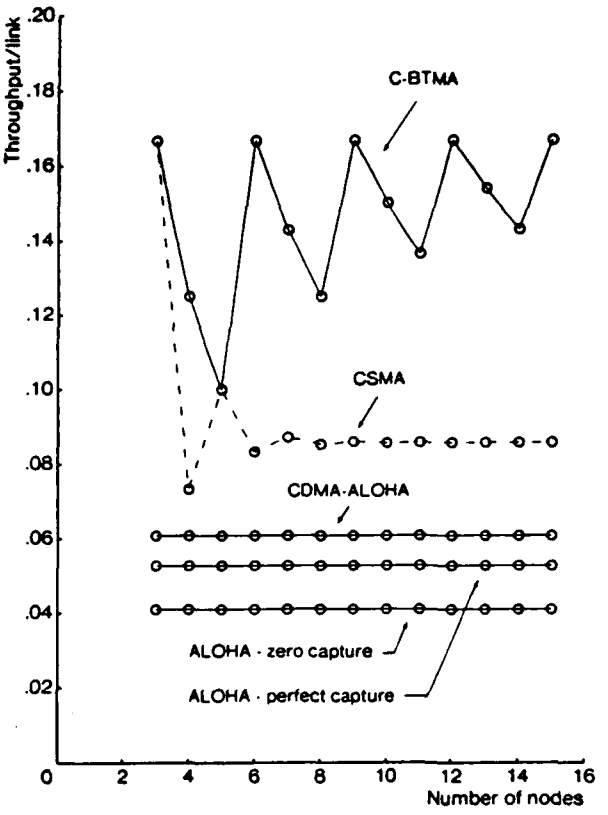


Fig. 8. Maximum link throughput for a ring.

an even number of nodes. For rings with an even number of nodes the throughput increases as the number of nodes increases. For C-BTMA the throughput exhibits a quasi-periodicity of period 3. All rings with a number of nodes which is multiple of 3 have a maximum throughput per link of 1/6, obtained when all $G_i \to \infty$, and corresponding to the situation when one in every three nodes is transmitting, and the intermediate nodes are blocked.

Finally, we consider other regular topologies than the ring, with higher degrees of connectivity. These consist of the five regular polyhedra (listed in Table 1) with the vertices representing the nodes and the edges representing the links. In Table 1 we give the maximum throughput per node for each of these topologies under a uniform traffic pattern.

Table 1

Configuration	Number of nodes	Number of neighbors	Throughput per Node				
			ALOHA 0-capture	ALOHA perf. capture	CDMA ALOHA	CSMA	C-BTMA
Tetrahedron	4	3	.0567	.0819	.1193	.2500	.2500
Cube	8	3	.0567	.0819	.1193	.0989	.2500
Octahedron	4	4	.0433	.0670	.1168	.1057	.1667
Dodecahedron	20	3	.0567	.0819	.1193	.1193	.2000
Icosahedron	12	5	.0350	.0567	.1153	.0885	.1667

As expected, the throughput for the ALOHA-type schemes decreases as the number of neighbors increases. The throughput of C-BTMA is just the ratio of the maximum number of possible simultaneous transmissions to the number of nodes in the network. It is interesting to see that CDMA-ALOHA performs consistently better than CSMA in all but the first case (a fully connected network, for which the hidden terminal problem does not exist), in spite of the pessimistic assumptions made in its analysis. This result suggests that CDMA-ALOHA might be preferable over CSMA in environments where each node has a large number of neighbors and the number of hidden terminals is high. Consistently with the previous examples, the best performing scheme was C-BTMA.

6. COMPUTATIONAL ASPECTS

In general, given an internal throughput matrix $S = [S_{ij}]$, it is difficult to solve analytically for the G_i's attaining S, if they exist, in order to determine the maximum throughput. This entails the solution of systems of non-linear equations of the form

$$\frac{S_{ij}}{G_{ij}} = \frac{P_{ij}(G_1,\ldots,G_N)}{\Delta(G_1,\ldots,G_N)}, \quad i,j \ni h_{ij}=1$$

$$G_i = \sum_{j=1}^{N} G_{ij}h_{ij}, \quad i=1,\ldots,N,$$

and where P_{ij} and Δ are polynomials in the G_i's. A numerical scheme for solution, for which we do not have proof of convergence but which has never failed to converge in all cases we have seen so far, is given by the following recursions as described in [1]

$$G_{ij}^{(k+1)} = S_{ij}\frac{\Delta(G_1^{(k)},\ldots,G_N^{(k)})}{P_{ij}(G_1^{(k)},G_N^{(k)})}, \quad i,j \ni h_{ij}=1$$

$$G_i^{(k)} = \sum_{j=1}^{N} G_{ij}^{(k)}h_{ij}, \quad i=1,\ldots,N$$

with the initial conditions $G_{ij}^{(0)} = S_{ij}$. It is believed that this algorithm will converge if S is feasible and diverge otherwise. One of the most serious limitation of this numerical method, however, (and also of the analytical solution) is the combinatorial explosion as the number of nodes increases, from which polynomials with a very large number of terms result.

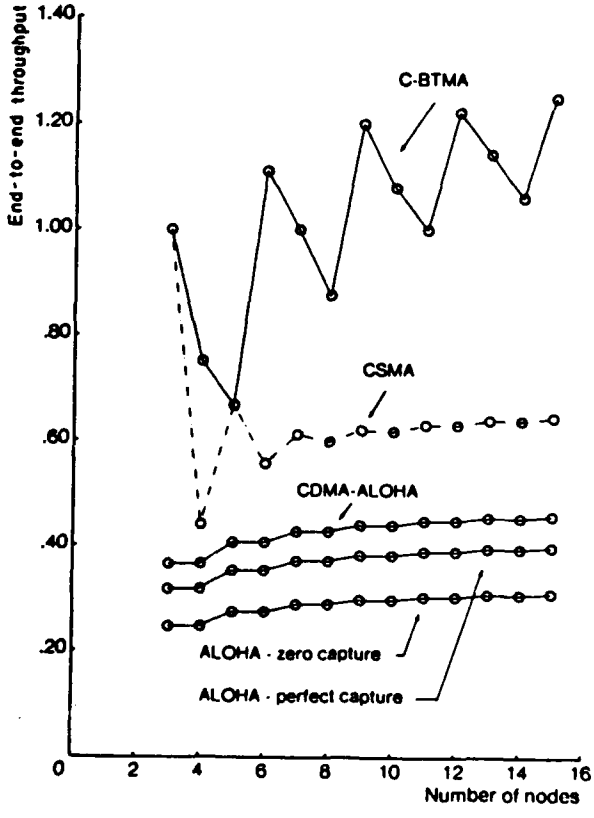


Fig. 9. Maximum end-to-end throughput for a ring.

REFERENCES

[1] F. 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. on Comm.*, Dec. 1975, pp. 1417–1433.

[2] R. R. Boorstyn and A. Kershenbaum, "Throughput Analysis of Multihop Packet Radio," *Proceedings of ICC*, Seattle, June 1980, pp. 13.6.1 -13.6.6.

[3] P. Spilling and F. Tobagi, *Activity Signalling and Improved Hop Acknowledgements in Packet Radio Systems*, Packet Radio Temporary Note #283, SRI International, January 1980.

[4] F. Kelly, *Reversibility and Stochastic Networks*, Wiley Series in Probability and Mathematical Statistics, John Wiley, 1979.

[5] L. Kleinrock, *Queuing Systems, Vol. II—Computer Applications*, Wiley Interscience, New York, 1976.

ICC'84
Amsterdam, May 14-17, 1984

THEORETICAL RESULTS IN THROUGHPUT ANALYSIS OF MULTIHOP PACKET RADIO NETWORKS*

José M. Brásio and Fouad A. Tobagi

Computer Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
(415) 497-1708

ABSTRACT

The focus of this paper is on the throughput analysis of multihop packet radio networks. Two major contributions are presented. First, the technique introduced by Boorstyn and Kershenbaum for networks operating under nonpersistent CSMA with perfect capture and zero propagation delay, and later applied by the authors to other access schemes, is justified on theoretical grounds. Secondly, new results are presented consisting of the following: (i) a characterization of access protocols which lead to a product form solution, (ii) general throughput expressions independent of capture assumptions, and (iii) the analytical solution of the throughput equations for the case of zero capture.

1. INTRODUCTION

Packet radio networks have for a long time been studied for their operational properties and potential for computer communications [1,2]. A number of papers have appeared in the literature dealing with the analysis of such systems [3,4,5,12]. However, only recently have there been significant advances in the development of analytical techniques for the evaluation of the performance of packet radio networks with a multihop nature. The difficulty in analysis has been mainly due to the fact that the access protocols introduce dependencies between the activity of different nodes. This is particularly true when delay results are desired, since the situation we then face is that of a network of queues where the service time at each server depends on the global state of the system ([6]-[8]). When only throughput results are desired, a Markovian model can be defined leading to an analytical product form solution applicable to a number of access protocols operating under perfect capture ([9]-[11]). In particular, in [9] and [11] this technique is introduced and applied to the nonpersistent CSMA protocol. In [10] other protocols are considered. In [10] it is also stated that not all protocols can be analysed by this technique, however no characterization is given for those that can. Moreover, the analysis presented in these papers accommodates exclusively perfect capture operation in the context of zero propagation and processing delays. No analytical solution exists allowing us to deal with the cases of zero and partial capture or nonzero delays. In the present study, some of these issues are addressed. In particular, the material presented in [10] is extended to include: (i) a more precise formulation of the analytical technique, (ii) a characterization of the protocols for which the technique is applicable, and (iii) the solution for the non-perfect capture case. The formulation of a model for the case of nonzero propagation delay will be treated in a forthcoming paper.

In Section 2 we present the general model and its assumptions. In Section 3 we formulate the Markov chain describing the system, present conditions for the existence of a product form solution, and characterize the protocols that lead to the product form solution. Finally, in Section 4 we give a general expression for the throughput, and then particularize it for the cases of zero and perfect capture.

*This work was supported in part by the Defense Advanced Research Projects Agency under contract MDA 903-79-C-0201, order A03717, monitored by the Office of Naval Research. José Brásio is on a fellowship from the Instituto Nacional de Investigação Científica, Lisbon, Portugal.

2. GENERAL MODEL

We consider a network with N nodes, numbered 1, 2, ..., N, and whose topology is given by a hearing matrix $H = [h_{ij}]$, where

$$h_{ij} = \begin{cases} 1 & \text{if } j \text{ can hear } i \\ 0 & \text{otherwise} \end{cases}$$

Thus each nonzero entry h_{ij} in the hearing matrix corresponds to a directed radio link in the network from node i to node j, and vice-versa. We call node i the *source* and node j the *destination* for that particular link. Due to the broadcast nature of the channel, a message sent over a given link will reach nodes other than its intended receiver, eventually colliding with messages destined to these nodes. The traffic requirements for each link are assumed to be dictated by the end-to-end traffic requirements together with a static routing function. It may happen that for some links the required traffic is zero. We refer to these links as *unused links*, and all other links as *used links*. We say that a used link is *active* whenever a transmission is taking place over that link, i.e., whenever the source node is transmitting a message intended to the destination node on that link. The activity of the links of the network is conditioned by the access protocol in use. An access protocol is a set of rules which, given the current set of active links in the network, determines whether or not a given inactive link can become active. Throughout the paper we consider all used links to be numbered $1, 2, \ldots, L$, and we let $\mathcal{L} \triangleq \{1, 2, \ldots, L\}$. For link i, $i \in \mathcal{L}$, we denote by s_i its source node, and by d_i its destination node. Alternatively we represent link i by the ordered pair (s_i, d_i).

Since the entire packet radio network operates using a single radio frequency, each node in the network has one transmitter, but can in general have more than one outgoing link. We consider that each outgoing link at a node has a separate queue for the packets to be transmitted on it and that the transmitter is shared among all queues at that node. To avoid repeated interference between transmissions in the network, transmission requests for the various queues at a node are scheduled according to random point processes, one for each queue. In this study, we consider the point process for link i, $i \in \mathcal{L}$, to be Poisson with rate λ_i ($\lambda_i > 0$), independent of all other such processes in the network.

Consider a point in time defined by the point process for some link i. If the queue is empty, this scheduling point is ignored. If the queue is nonempty then a packet in the queue is considered for transmission. The transmission may or may not take place depending on the status of the transmitter at the source node (busy or

idle), the priority structure (if any) among the queues at the source node, the channel access protocol in use, and the current activity on the network. If the transmission is inhibited, or if the transmission is undertaken but unsuccessfully (due to a collision at the intended destination or to a preemption by another transmission at the source), then the packet in question (or any other packet in the queue, for that matter) is reconsidered at the next point in time. Otherwise (i.e., the transmission is successful), the packet is removed from the queue, and the same process is repeated at the next scheduling point for that link.

It is assumed in this study that at each scheduling point of the point process there is a packet in the queue for consideration. It is also assumed that neither preemption nor priority functions are supported at the nodes. In addition, we assume the transmission time of the messages transmitted over link i to be exponentially distributed with mean $1/\mu_i$ ($\mu_i > 0$), and to be redrawn independently from this distribution each time the message is transmitted. We also assume infinite buffer space for each link, and instantaneous and perfect acknowledgments, providing immediate feedback regarding the success or failure of each transmission.

Let $X(t)$ denote the set of all active links at time t. Given that the period of time that a link remains active is exponential, and given that the scheduling point processes which determine the points in time at which links can become active (as determined by the access protocol) are Poisson, $X(t)$ is a continuous time Markov chain. The precise formulation of the Markov chain varies with the access protocol in use and is given in the following section. Also given in the next section are the conditions under which the Markov chain leads to a product form solution.

3. THE MARKOV CHAIN

3.1. Definitions

Given an access protocol, we say that link $i \in L$ blocks link j if, whenever link i is active, the protocol used does not allow a scheduling point for link j to result in an actual transmission. It is to be noted that if link i blocks link j, it does not necessarily follow that link j blocks link i.

Let D be a set of links in L. We say that D blocks link $j \in L - D$ if there exists some link $i \in D$ which blocks j. We define $U(D)$ to be the set of all links in $L - D$ which are not blocked by D.

In later treatments, the following two protocols are used as examples:

(i) Nonpersistent Carrier Sense Multiple Access (CSMA) [12]: under CSMA, a link will be blocked whenever its source node detects a transmission by any other source node that it can hear; i.e., link (s_j, d_j) is blocked by (s_i, d_i) whenever $h_{s_i s_j} = 1$, or $s_i = s_j$;

(ii) Idealistic Busy Tone Multiple Access (I-BTMA)[13]: this protocol assumes the existence of a separate channel for a busy tone. The destination of a link emits a busy tone whenever that link is active. A link is blocked if its source node hears either a transmission or a busy tone; i.e., link (s_j, d_j) is blocked by (s_i, d_i) if either $h_{s_i s_j} = 1$, $h_{d_i s_j} = 1$, or $s_i = s_j$.

3.2 State Space

We now define the state space S for the Markov chain $X(t)$. Since $X(t)$ is the set of all links that are active at time t, $S \subseteq 2^L$. Given an access protocol and its blocking properties, not all subsets of L may be in S.

Definition 1 S is the collection of subsets of L that the system can reach starting from the idle state ϕ (i.e., all links inactive) by any sequence of *link activations* and *deactivations*.

Definition 2 A subset $D = \{l_1, l_2, \ldots, l_n\}$ of L is said to be *directly reachable* if there exists some permutation $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ of D such that l_{i_j} is not blocked by $(l_{i_1}, l_{i_2}, \ldots, l_{i_{j-1}})$, $j = 2, \ldots, n$. That is, D is directly reachable if it can be reached by *only* activating the links in it, in some order, starting from the idle state ϕ.

Lemma 1 If a subset $D = \{l_1, l_2, \ldots, l_n\}$ is directly reachable, then any subset $D' \subseteq D$ is also directly reachable.

Proof: Let $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ be an ordered sequence of activations which allows D to be reached. The ordered subsequence in $(l_{i_1}, l_{i_2}, \ldots, l_{i_n})$ corresponding to links in D' is a sequence of activations which allows D' to be reached directly. ∎

Proposition 1 The state space S consists of ϕ and all subsets $D \subseteq L$ that are directly reachable.

Proof: Clearly a set D which is directly reachable belongs to S. To prove the converse, we let $D \in S$ be some subset that is reached via some sequence of states $D_0, D_1, \ldots, D_m$, with $D_0 = \phi$ and $D_m = D$, due to link activations and deactivations. (Note that since the process $X(t)$ is such that no two events can occur at the same instant, then $|D_k| = |D_{k-1}| \pm 1$ for all $k = 1, 2, \ldots, m$). Since the first transition out of $D_0 = \phi$ must be an activation, there is some index $r \leq m$ such that D_r is reached directly. Consider D_{r+1}. If $D_{r+1} = D_r \cup \{i\}$ for some i, then D_{r+1} is clearly directly reachable. If $D_{r+1} = D_r - \{j\}$ for some j, then D_{r+1} is also directly reachable, by Lemma 1. Applying the same argument to the remaining steps, we guarantee that D is directly reachable. ∎

According to Proposition 1, one can generate the state space by the following algorithm, which is not necessarily claimed to be the most efficient for this purpose:

```
begin
  S := {φ};
  L := {1, 2, ..., L};
  for k := 0 to L − 1 do
    for every D ∈ S s. t. |D| = k do
      for every l ∈ L − D do
        if l is not blocked by D,
            then add D ∪ {l} to S;
end.
```

Remark 1 Given an access protocol and some state $D \in S$, it should be noted that not all sequences of activations of its elements will necessarily allow D to be reached from ϕ. For example, consider the 4-node chain of Figure 1 with nonzero traffic requirement over links 1 and 2, and the I-BTMA access protocol. State $\{1, 2\}$ is an example of a state for which the order of activation is relevant. This state is reachable by the permutation $(1, 2)$, but not by the permutation $(2, 1)$.

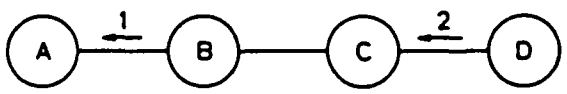


Figure 1 A 4-node chain with nonzero traffic requirements over links numbered 1 and 2

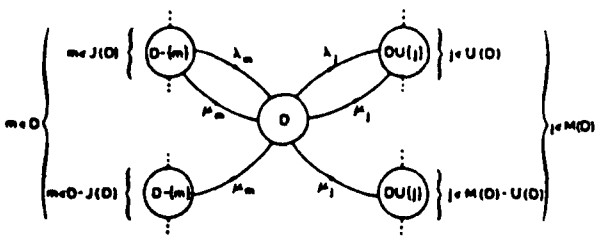


Figure 2 Typical transitions to and from a node

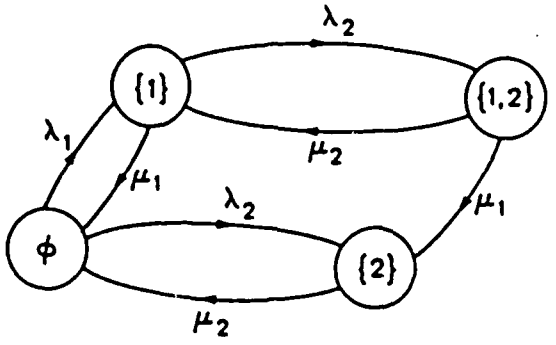


Figure 3 State space for the Markov chain of Example 1

Remark 2 Recall that L is the set of all used links and thus $\lambda_i > 0$ for all $i \in L$. Accordingly every state can be reached from the empty state in a nonzero period of time with nonzero probability. Similarly, the empty state can be reached from any other state in a nonzero period of time with nonzero probability (since $\mu_i > 0$ for all $i \in L$). It then follows that all states communicate and the resulting Markov chain is irreducible.

3.3 The Equilibrium Equations

As noted above, the Markov chain $X(t)$ is irreducible. Since the state space is finite, the chain is then positive recurrent and ergodic. Thus the existence of a strictly positive stationary distribution is ensured. We denote by $\{Q(D), D \in S\}$ the stationary probability distribution.

Let the state of the system at time t be $D \in S$, let i be any link not blocked by D, and let $j \in D$. Given the assumptions in Section 2, the time to the next scheduling point of i is exponentially distributed with parameter λ_i, and the time to the end of the transmission over link j is also exponentially distributed with parameter μ_j. Given that $X(t) = D$, the state of the system at time $t + \Delta t$ is given by (recall the definition of $U(D)$ in Sec. 3.1)

$$X(t+\Delta t) = \begin{cases} D \cup \{i\}, & i \in U(D), \text{ with probability } \lambda_i \Delta t \\ D - \{j\}, & j \in D, \text{ with probability } \mu_j \Delta t \\ D, & \text{with probability } 1 - \left(\sum_{i \in U(D)} \lambda_i + \sum_{j \in D} \mu_j\right) \Delta t, \end{cases}$$

having neglected terms of order higher than Δt. This equation defines the transition rates which we need for writing the equilibrium equations. Before doing so we have to introduce some further notation. For each $D \in S$, let $M(D)$ be the set of all links $i \notin D$ such that $D \cup \{i\} \in S$. Clearly $M(D) \supseteq U(D)$. Note however that it is not necessarily true that $M(D) = U(D)$. (See Example 1 below). Let $J(D)$ to be the set of all links $j \in D$ such that j is not blocked by $D - \{j\}$, i.e., such that $j \in U(D - \{j\})$. Clearly, $J(D) \subseteq D$. Here too, in general we have $J(D) \neq D$, as is also illustrated in Example 1. With these definitions, a sketch of the state-transition-rate diagram for state D and the transitions to and from its neighbors can be seen in Figure 2. The equilibrium equations ([14]) take then the form

$$Q(D)\left[\sum_{i \in U(D)} \lambda_i + \sum_{j \in D} \mu_j\right] = \sum_{j \in J(D)} Q(D - \{j\})\, \lambda_j + \sum_{i \in M(D)} Q(D \cup \{i\})\, \mu_i, \quad D \in S \tag{1}$$

Example 1 Consider the 4-node chain of Figure 1 with nonzero traffic requirement over links 1 and 2 only, and the 1-BTMA protocol. The corresponding state-transition-rate diagram is shown in Figure 3. From the definitions we have that $J(\{1,2\}) = \{2\}$, $U(\{2\}) = \phi$, and $M(\{2\}) = \{1\}$. These are examples of states D for which $M(D) \neq U(D)$, or $J(D) \neq D$.

3.4 Reversible Markov Chains and Product Form Solutions [15]

Definition 3 A continuous time stochastic process $X(t)$ defined on $I = (-\infty, +\infty)$ is said to be *reversible* if for any $\tau \in I$, integer n, and $t_1 \leq t_2 \leq \cdots \leq t_n$ in I, $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$.

For the particular case of Markov chains, reversibility has a simple characterisation in terms of the transition rates and steady-state distribution, as given in the following proposition, whose proof can be found in [15].

Proposition 2 A stationary continuous time Markov chain is reversible if and only if there exists a collection of positive numbers $\{\eta(D), D \in S\}$, summing to unity, such that

$$\eta(D_1) \cdot q(D_1, D_2) = \eta(D_2) \cdot q(D_2, D_1) \tag{2}$$

for all $D_1, D_2 \in S$, and where $q(D_i, D_j)$ is the rate of transitions from D_i to D_j. When such a collection exists, it is the stationary probability distribution.

An equivalent necessary and sufficient condition for reversibility (called Kolmogorov's criterion) is that, for any finite sequence of states $D_1, D_2, \ldots, D_n \in S$, the transition rates satisfy

$$q(D_1, D_2)\, q(D_2, D_3) \cdots q(D_n, D_1) = q(D_1, D_n)\, q(D_n, D_{n-1}) \cdots q(D_2, D_1). \tag{3}$$

Suppose we are given a reversible Markov chain with state space S. Let D_0 be a fixed state and D a generic state in S. Let $D_0, D_1, \ldots, D_m$ be any sequence of states in S, with $D_m = D$, such that between any two consecutive states of the sequence there exist nonzero transition rates. By repeated application of (2) it is easy to see that the steady state probability distribution for such a Markov chain satisfies

$$Q(D) = Q(D_0) \prod_{k=1}^{m} \frac{q(D_{k-1}, D_k)}{q(D_k, D_{k-1})} \tag{4}$$

A solution with the form of (4) is called a *product form* solution. It is immediately seen that if the steady state

solution satisfies (4) then (2) is automatically satisfied for all $D_1, D_2 \in S$. Thus

Proposition 3 A stationary continuous time Markov chain $X(t)$ possesses a product form solution for the steady state probability distribution if and only if it is reversible.

3.5 Criterion for the Existence of a Product Form

We use here the results of the previous section to determine the conditions on the access protocol, network topology, and traffic requirements under which the resulting Markov chain, defined in Sections 3.2 and 3.3, is reversible and hence the global balance equations (1) have a product form solution.

Lemma 2 $U(D) = M(D)$ for all $D \in S$ if and only if $J(D) = D$ for all $D \in S$.

Proof: We know already that $J(D) \subseteq D$ and $U(D) \subseteq M(D)$. To prove the desired equalities we only need to prove the reverse inclusions. Assume that $U(D') = M(D')$ for all $D' \in S$. It is evident that $J(\phi) = \phi$. Consider now any $D \in S, D \neq \phi$. For each $j \in D$, by definition $j \in M(D - \{j\})$. Since by hypothesis $U(D - \{j\}) = M(D - \{j\})$, then $j \in U(D - \{j\})$. But this just means that $j \in J(D)$. Thus $D \subseteq J(D)$, for all $D \in S$. Conversely, assume that $D' = J(D')$ for all $D' \in S$. Call a state *maximal* if $M(D) = \phi$. Since $U(D) \subseteq M(D)$, for maximal states it is true that $U(D) = M(D)$. Let now $D \in S$ be a non-maximal state, and $j \in M(D)$. By hypothesis $J(D \cup \{j\}) = D \cup \{j\}$, which in particular implies that j is not blocked by D, and thus that $j \in U(D)$. Hence $M(D) \subseteq U(D)$. ∎

Proposition 4 The Markov chain $X(t)$ is reversible if and only if

$$D = J(D) \tag{5}$$

for all $D \in S$ (or, equivalently, $U(D) = M(D)$ for all $D \in S$).

Proof: Assume that the Markov chain $X(t)$ is reversible. Clearly (5) holds for $D = \phi$. Consider now $D \in S, D \neq \phi$, and $j \in D$. From (2) we have that

$$Q(D) \cdot q(D, D - \{j\}) = Q(D - \{j\}) \cdot q(D - \{j\}, D).$$

Since $q(D, D - \{j\}) = \mu_j > 0$ and $Q(D) > 0$ for all $D \in S$, this last equation implies that $q(D - \{j\}, D) > 0$. But since $q(D - \{j\}, D)$ can only be either 0 (if $j \notin J(D)$) or λ_j (if $j \in J(D)$), we necessarily conclude that $q(D - \{j\}, D) = \lambda_j$ and $j \in J(D)$. Then $D \subseteq J(D)$ for all $D \in S$, and consequently $D = J(D)$ for all $D \in S$. Conversely, assume that $J(D) = D$ for all $D \in S$. We now show that $\{\eta(D) : \eta(D) = \eta_0 \prod_{i \in D} \frac{\lambda_i}{\mu_i}, D \in S\}$, with η_0 chosen so that $\sum_{D \in S} \eta(D) = 1$, is a collection of numbers that satisfies the conditions of Proposition 2. Let D_1, D_2 be any two states in S. Assume first that they are of either the form $D_1 = D$, $D_2 = D - \{j\}$, or the form $D_1 = D - \{j\}$, $D_2 = D$, for some $D \in S$ and $j \in D$. From the choice of $\eta(D)$ we have

$$\eta(D) = \frac{\lambda_j}{\mu_j} \eta(D - \{j\}) \quad .$$

The transition rates between these two states are $q(D, D - \{j\}) = \mu_j$ and, from the assumption $J(D) = D$, $q(D - \{j\}, D) = \lambda_j$. Thus, in this case,

$$\eta(D_1) q(D_1, D_2) = \eta(D_2) q(D_2, D_1) \quad .$$

For any other choice of D_1 and D_2, $q(D_1, D_2) = q(D_2, D_1) = 0$, and

$$\eta(D_1) q(D_1, D_2) = \eta(D_2) q(D_2, D_1)$$

is trivially verified. Thus (2) holds for all $D_1, D_2 \in S$, and $X(t)$ is reversible, by Proposition 2. ∎

Proposition 5 (Criterion for the existence of a product form) A necessary and sufficient condition for a channel access protocol, together with a given network topology and traffic requirements, to have a product form solution is that, for all pairs of *used* links i and j, link j blocks link i whenever link i blocks link j.

Proof: Since the existence of a product form solution is equivalent to the Markov chain being reversible, we will prove the equivalence between reversibility and the condition stated in the above criterion.

(a) The Markov chain is reversible.
Assume that link i blocks link j. If j does not block i, we will have the situation depicted in Figure 4 in which $\{j\} \in M(\{i\})$ but $\{j\} \notin U(\{i\})$, providing an instance of a state D for which $U(D) \neq M(D)$. But this contradicts our assumption that $X(t)$ is reversible, given the result contained in Proposition 4. Thus j blocks i.

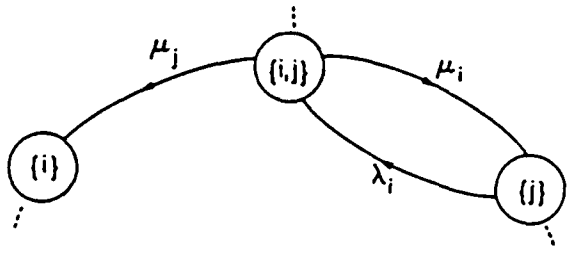


Figure 4 Portion of a non-reversible chain

(b) The Markov chain is not reversible.
From Proposition 4 we know that there exists a state D for which $D \neq J(D)$, i.e., for which there exists $j \in D$ such that j is blocked by $D - \{j\}$. Let $i \in D - \{j\}$ be some link blocking j and define $D' = \{i, j\}$. Since $D' \subseteq D$ then, by Lemma 1 and Proposition 1, $D' \in S$, and D' can be directly reached by activating links i and j in some order. By hypothesis i blocks j, and so D' has to be directly reachable from $\{j\}$. Thus j does not block i. ∎

Proposition 5 implies that, in a reversible chain and for any state $D \in S$, any order of activation of the links in D allows D to be reached directly from state ϕ, and thus the situation depicted in Remark 1 does never occur.

For a reversible chain the stationary probability distribution is given by (4). From the particular form of the transition rates we have

$$Q(D) = Q(\phi) \prod_{i \in D} \frac{\lambda_i}{\mu_i} \tag{6}$$

for all $D \in S$. We can ask if there can exist protocols for which the corresponding Markov chain $X(t)$ is not reversible, and yet the steady-state probabilities have the form (6).

Proposition 6 (6) is a solution of the global balance equations (1) if and only if

$$D = J(D) \tag{5}$$

for all $D \in S$ (or, equivalently, $U(D) = M(D)$ for all $D \in S$).

Proof: Assume that $D = J(D)$ for all $D \in S$. By Proposition 4, $X(t)$ is reversible and thus the steady-state probabilities have the form (6). Conversely, assume that (6) is a solution of (1). By substitution of (6) in (1) and

simplification we obtain

$$\sum_{i \in M(D)-U(D)} \lambda_i = \sum_{j \in D-J(D)} \mu_j .$$

We now see under which conditions this equality can hold. Recall that a state D of the Markov chain is said to be *maximal* if $M(D) = \phi$. Given a generic state D, define a *maximal path* starting at D to be a finite sequence of states $D_0, D_1, \ldots, D_k$ such that $D_0 = D$, $D_{l+1} = D_l \cup \{i\}$ for some $i \in M(D_l)$, $l = 0, 1, \ldots, k-1$, and D_k is a maximal state. Define the length of the maximal path to be k, and let $l(D)$ be the maximum of the lengths of the maximal paths starting at D. We shall now prove (5) by induction on $l(D)$. For $l(D) = 0$ we have that D is a maximal state, for which $M(D) = U(D) = \phi$. Then

$$\sum_{j \in D-J(D)} \mu_j = 0.$$

Since, by assumption, $\mu_j > 0$, we obtain that $D = J(D)$. Assume now that, for n a positive integer, (5) holds for all states D' for which $l(D') \leq n$. Let D be a state for which $l(D) = n+1$. For all $j \in M(D)$, $D \cup \{j\}$ is a state for which $l(D) \leq n$. By the induction hypothesis we then have $J(D \cup \{j\}) = D \cup \{j\}$, which means in particular that j is not blocked by D or, in other words, that $j \in U(D)$. Then $U(D) = M(D)$ and

$$\sum_{j \in D-J(D)} \mu_j = 0.$$

Again, as all $\mu_j > 0$, it follows that $D = J(D)$. ∎

Example 2: As an application of Proposition 5, we can now prove that, with a symmetric hearing matrix, non-persistent CSMA always leads to a product form solution. Consider any two used links i and j, and represent them as (s_i, d_i) and (s_j, d_j), respectively. Under CSMA, if i blocks j, then either $s_i = s_j$ or $h_{s_i s_j} = 1$. The symmetry of the hearing matrix then implies that j blocks i, and thus by Proposition 5 the stationary distribution will have a product form. If the hearing matrix is not symmetric we will not get a product form solution, except when all pairs of nodes s_i and s_j for which $h_{s_i s_j} = 1$ and $h_{s_j s_i} = 0$ are such that at least one element of the pair is the source of no used links.

Example 3: The I-BTMA protocol will not, in general, lead to a product form solution. Indeed, if the network under consideration contains the subnetwork and traffic pattern of Figure 1, we can find links i and j such that i blocks j but j does not block i. For some specific topologies and traffic patterns, however, I-BTMA will have a product form solution. Examples of these are a star network with arms of length 1 and arbitrary traffic pattern, or a 4-node chain in which the outer nodes generate no traffic.

The product form (6) is especially convenient for computation. In the cases where it does not hold, in order to determine the stationary probability distribution we have to resort to the numerical solution of the global balance equations (2). The coefficient matrix for this system is sparse, and this fact suggests the use of an iterative solution method.

4. THROUGHPUT ANALYSIS

Given a Markov chain describing the activity of a packet radio network, we wish to find an expression for the throughput of each link as a function of the transition rates and the steady-state probability distribution of the Markov chain. We do not necessarily assume reversible Markov chains; the material in this section is effectively independent of the previous section. We start in this section by presenting general expressions for the link throughput, without specific assumptions on capture. Later the general results are particularized for the cases of zero capture and perfect capture. Capture refers to success in the reception of a given message at its destination even when there is overlap with interfering messages. Perfect capture refers to the ability of receiving correctly the first message to reach the receiver regardless of the number of future overlapping messages; zero capture refers to the situation where any overlap results in complete destruction of all overlapping transmissions.

By definition, the throughput of link i, S_i, is the long-run fraction of time that link i is engaged in successful transmissions. We restrict the analysis to protocols and modes of operation such that the success of a transmission does not depend on the behavior of the system after the termination of the transmission in question.

4.1 General Case

Let $U(i)$ be the collection of states $D \in S$ that do not block link i. Define $S(D,i)$, $D \in U(i)$, to be the fraction of time that link i is engaged in successful transmissions and the state just prior to the start of those transmissions is D. $S(D,i)$ accounts for all successful transmissions on link i that are initiated by a jump of the Markov chain from state D into state $D \cup \{i\}$. Summing over D we obtain

$$S_i = \sum_{D \in U(i)} S(D,i) \quad . \tag{7}$$

For a fixed D and by the strong Markov property ([16]), the times of the successive transitions from state D to state $D \cup \{i\}$ are regeneration points for the Markov chain $X(t)$. We now consider the cycles defined by the time intervals between two successive regeneration points. Let $C_k(D,i)$ denote the length of the k-th cycle, and $T_k(D,i)$ be the total time in cycle k that the channel was used successfully by a transmission over link i. We can think of $T_k(D,i)$ as the "reward" (for the purpose of calculating the link's throughput) earned during the k-th cycle. With our assumptions on the protocols and modes of operation of the network, $\{(C_k(D,i), T_k(D,i)) : k \geq 1\}$ is a sequence of independent and identically distributed pairs of random variables. In general the elements of each pair may be correlated. In the following we will omit the subscripts in these variables whenever we refer to a generic one. If we let $N(t)$ be the number of completed cycles by time t, then

$$S(D,i) = \lim_{t \to \infty} \frac{1}{t} \sum_{k=1}^{N(t)} T_k(D,i) \quad .$$

Let $E[C(D,i)]$ and $E[T(D,i)] \equiv \overline{T}(D,i)$ denote the expected cycle length and expected reward, respectively. Standard theorems in the theory of renewal processes ([17]) assert that, with probability 1,

$$S(D,i) = \frac{E[T(D,i)]}{E[C(D,i)]} \quad . \tag{8}$$

The quantities on the right-hand side of the last equation can be computed in terms of the parameters of the system.

Proposition 6 The expected cycle length $E[C(D,i)]$ is given by

$$E[C(D,i)] = \frac{1}{\lambda_i Q(D)} \quad , \quad D \in U(i) \; . \tag{9}$$

Proof: Let $n(t)$ denote the number of state transitions of the Markov chain $X(t)$ in $(0,t]$, and define the embedded Markov chain $\{Y_k : k \geq 0\}$ by

$$Y_k = X(t), \quad \text{for any } t \text{ such that } \quad n(t) = k.$$

Y_k is the state the Markov chain $X(t)$ is in after the k-th transition. The irreducibility and positive recurrence of $X(t)$ implies the same properties for the embedded chain Y_k, which then possesses a stationary distribution $\{\pi(D) : D \in S\}$, with

$$\pi(D) = \lim_{k \to \infty} P[Y_k = D] \quad .$$

The relation between $\pi(D)$ and $Q(D)$ is given by ([17])

$$\frac{Q(D)}{Q(D')} = \frac{\pi(D)H(D)}{\pi(D')H(D')}, \quad D, D' \in S \qquad (10)$$

where $H(D)$ is the expected sojourn time of $X(t)$ in state D, given by

$$H(D) = \frac{1}{\sum_{D' \in S} q(D, D')} \quad . \qquad (11)$$

Given that $X(t)$ is a continuous time Markov chain (and hence the next state after state $D \in S$ is determined by the minimum of an independent collection of exponential random variables with parameters $q(D, D')$, $D' \in S$), the transition probabilities for the embedded chain Y_k are easily computed; in particular, those relevant to this proof are given by

$$P[Y_{k+1} = D \cup \{i\} \mid Y_k = D] = \frac{\lambda_i}{\sum_{D' \in S} q(D, D')} \quad , \quad D \in U(i) \quad . \qquad (12)$$

In order to compute the average cycle length we note that ([17]),

$$\lim_{t \to \infty} P[X(t) = D \cup \{i\}, Y_{n(t)-1} = D] = \frac{H(D \cup \{i\})}{E(C(D,i))} \quad . \qquad (13)$$

Developing the left-hand side of (13) gives us

$$\begin{aligned} &\lim_{t \to \infty} P[X(t) = D \cup \{i\}, Y_{n(t)-1} = D] \\ &\quad = Q(D \cup \{i\}) \\ &\quad \cdot \lim_{t \to \infty} P[Y_{n(t)-1} = D \mid Y_{n(t)} = D \cup \{i\}] \\ &\quad = Q(D \cup \{i\})\, P[Y_{k+1} = D \cup \{i\} \mid Y_k = D] \frac{\pi(D)}{\pi(D \cup \{i\})} \end{aligned}$$

which, when substituted in (13) after using (10)-(12), gives equation (9). ∎

From (7), (8) and Proposition 6 we then obtain

Proposition 7 The throughput of link i, S_i, is given by

$$S_i = \lambda_i \sum_{D \in U(i)} Q(D) \overline{T}(D, i) \qquad (14)$$

or, defining the normalized rescheduling rate $G_i \triangleq \frac{\lambda_i}{\mu_i}$,

$$S_i = G_i \sum_{D \in U(i)} Q(D)\, \mu_i \overline{T}(D, i) \quad . \qquad (15)$$

4.2 Perfect Capture

Under perfect capture we assume that for each link i there exists a set of links $C(i)$ not containing i, such that transmission over link i is successful if and only if at the time it starts no other link in $C(i)$ is active, irrespective of what happens after the start of the transmission over link i.

Let $U_s(i)$ be the subset of $U(i)$ formed by those states that do not contain any link in $C(i)$. For $D \in U(i) - U_s(i)$, we have $\overline{T}(D, i) = 0$; for $D \in U_s(i)$, we have $\overline{T}(D, i) = 1/\mu_i$. We thus have

Proposition 8 The throughput of link i under perfect capture is given by

$$S_i = G_i \sum_{D \in U_s(i)} Q(D) \quad . \qquad (16)$$

Equation (16) was first derived by Boorstyn and Kershenbaum in [9] for nonpersistent CSMA, using a heuristic argument.

4.3 Zero Capture

Under zero capture we assume that for each link i there exists a set of links $C(i)$ not containing i, such that a transmission over link i is successful if and only if at the time it starts and during the whole duration of the transmission no link in $C(i)$ is ever active.

As in the case of perfect capture, for all states D in $U(i) - U_s(i)$, $\overline{T}(D, i) = 0$, and thus

$$S_i = G_i \sum_{D \in U_s(i)} Q(D)\, \mu_i \overline{T}(D, i) \, .$$

However, in this case the average transmission time of a successful message is not $1/\mu_i$, due to the dependency that exists between the message length and its success. The computation of $\overline{T}(D, i)$ involves the construction of an auxiliary Markov chain. In the original chain, let $A_s(i)$ be the collection of states in which i is active and no element of $C(i)$ is active, let $A_c(i)$ be the collection of states in which i is active and some element of $C(i)$ is active, and let $J(i)$ be the set of states obtained from $A_s(i)$ by deactivating link i. With respect to these definitions, the start of a transmission over link i which does not suffer a collision at its very start corresponds to a transition of the Markov chain $X(t)$ from a state $D \in U_s(i)$ into state $D \cup \{i\} \in A_s(i)$. $X(t)$ will remain in $A_s(i)$ as long as i is active and not collided with. A possible later collision of i with a transmission over some other link in $C(i)$ corresponds to a transition from some state in $A_s(i)$ into a state in $A_c(i)$. The successful completion of link i's transmission corresponds to a transition from some state in $A_s(i)$ into a state in $J(i)$ without having previously visited any state in $A_c(i)$.

The structures of $U_s(i)$ and $A_s(i)$ are related. Any state of the form $D \cup \{i\}$, with $D \in U_s(i)$, is in $A_s(i)$. However, if $X(t)$ is not reversible, $A_s(i)$ will contain other states. These states are the ones that contain some link $j \notin C(i)$ that blocks link i but is not blocked by i. Any state containing such a j cannot clearly be of the form $D \cup \{i\}$, with $D \in U_s(i)$, since then we would have $j \in D$ and thus i would be blocked by D, contrary to the definition of $U_s(i)$; but nevertheless there will be states in $A_s(i)$ containing such links j, namely the state $\{i, j\}$. Any state in $A_s(i)$ not containing any such j will be of the form $D \cup \{i\}$, for some $D \in U_s(i)$.

The auxiliary Markov chain is now constructed by grouping all states in $J(i)$ into one absorbing state denoted

again $J(i)$, grouping all states in $A_c(i)$ into another absorbing state of the same name, and deleting all states not in $A_s(i) \cup A_c(i) \cup J(i)$. When deleting a state, all arrows incident to that state are deleted. In this new chain, the states in $A_s(i)$ are transient and, with probability 1, $X(t)$ will be absorbed in either $A_c(i)$ or $J(i)$. From what was said above, we see that a transmission over link i, initiated successfully by a jump of $X(t)$ from some $D \in U_s(i)$ into $D \cup \{i\}$, will terminate successfully if $X(t)$ is absorbed in $J(i)$, and will suffer a collision if absorption occurs in $A_c(i)$. Thus, for $D \in U_s(i)$, $T(D,i)$ equals the length of the time interval between the first entrance to $D \cup \{i\}$ and absorption in $J(i)$, if absorption occurs in $J(i)$, and 0, otherwise.

Let k be the cardinality of $A_s(i)$. By suitable reordering of the states of the modified chain, its transition rate matrix is

$$\mathbf{R}^*(i) = \begin{bmatrix} \mathbf{R}_s(i) & \mu_i \mathbf{e} & \varphi \\ \mathbf{s} & 0 & 0 \\ \mathbf{s} & 0 & 0 \end{bmatrix}$$

where $\mathbf{R}_s(i)$ is the $(k \times k)$ matrix of the transition rates between states in $A_s(i)$, $\mathbf{e} \triangleq [1 \ldots 1]^T$ is of dimension $(k \times 1)$, $\mu_i \mathbf{e}$ is the vector of the transition rates from $A_s(i)$ into $J(i)$, φ is the $(k \times 1)$ vector of the transition rates from $A_s(i)$ into $A_c(i)$, and $\mathbf{s} = [0 \ldots 0]$ is of dimension $(1 \times k)$. With these definitions, we have

Proposition 9 The throughput of link i under zero capture is given by

$$S_i = G_i \sum_{D \in U_s(i)} Q(D)\, \mu_i \overline{\mathbf{T}}_{D \cup \{i\}} \quad , \qquad (17)$$

where $\overline{\mathbf{T}}_{D \cup \{i\}}$ is the component with index $D \cup \{i\}$ of the column vector

$$\overline{\mathbf{T}} = \mu_i \, \mathbf{R}_s^{-2}(i)\mathbf{e} \quad .$$

Proof: As we saw above, $\overline{T}(D,i)$ is the average time to absorption in $J(i)$ for a chain started in state $D \cup \{i\}$, over the set of sample paths for which absorption in $J(i)$ occurs. Thus $\overline{T}(D,i)$ can be determined from the probability transition matrix, $\mathbf{P}^*(t)$, of the modified chain, defined by

$$\mathbf{P}^*_{D,D'}(t) = P[X(t) = D' \mid X(0) = D]$$

for D, $D' \in A_s(i) \cup A_c(i) \cup J(i)$. The transition probability matrix corresponding to $\mathbf{R}^*(i)$ has the form

$$\mathbf{P}^*(t) = \begin{bmatrix} \mathbf{P}_s(t) & \mathbf{P}_J(t) & \mathbf{P}_c(t) \\ \mathbf{s} & 1 & 0 \\ \mathbf{s} & 0 & 1 \end{bmatrix}$$

and is determined by the forward Kolmogorov equation

$$\frac{d}{dt}\mathbf{P}^*(t) = \mathbf{P}^*(t) \cdot \mathbf{R}^*(i) \qquad , \quad \mathbf{P}^*(0) = \mathbf{I} \quad .$$

Given the structure of $\mathbf{R}^*(i)$ the forward Kolmogorov equation takes the form

$$\begin{aligned} \frac{d}{dt}\mathbf{P}_s(t) &= \mathbf{P}_s(t)\,\mathbf{R}_s(i) && , \quad \mathbf{P}_s(0) = \mathbf{I} \\ \frac{d}{dt}\mathbf{P}_J(t) &= \mu_i \mathbf{P}_s(t)\mathbf{e} && , \quad \mathbf{P}_J(0) = 0 \\ \frac{d}{dt}\mathbf{P}_c(t) &= \mathbf{P}_s(t)\varphi && , \quad \mathbf{P}_c(0) = 0 \end{aligned}$$

with solution

$$\begin{aligned} \mathbf{P}_s(t) &= e^{\mathbf{R}_s(i)t} && , \quad t \geq 0 \\ \mathbf{P}_J(t) &= \mu_i \left(e^{\mathbf{R}_s(i)t} - \mathbf{I}\right)\mathbf{R}_s^{-1}(i)\mathbf{e} && , \quad t \geq 0 \\ \mathbf{P}_c(t) &= \left(e^{\mathbf{R}_s(i)t} - \mathbf{I}\right)\mathbf{R}_s^{-1}(i)\varphi && , \quad t \geq 0 . \end{aligned}$$

Note that, since the states in $A_s(i)$ are transient, $e^{\mathbf{R}_s(i)t} \to 0$ as $t \to \infty$. Let now $\mathbf{T}$ be a column vector with rows indexed by the states in $A_s(i)$ in the same order as the rows of $\mathbf{R}_s(i)$ and where, for $D' \in A_s(i)$, the component with index D', $\mathbf{T}_{D'}$, is the random variable giving the time to absorption in $J(i)$ when the chain is started in state D'. Then

$$P\{\mathbf{T} \leq t\,\mathbf{e}\} = P_J(t) \quad , \qquad t \geq 0$$

and

$$\begin{aligned} \overline{\mathbf{T}} \triangleq E\{\mathbf{T}; \mathbf{T} < \infty\} &= \int_0^\infty [\mathbf{P}_J(\infty) - \mathbf{P}_J(t)]dt \\ &= -\mu_i \int_0^\infty e^{\mathbf{R}_s(i)t}\mathbf{R}_s^{-1}(i)\,\mathbf{e}\; dt = \mu_i \mathbf{R}_s^{-2}(i)\mathbf{e} \quad . \end{aligned}$$

Since $\overline{T}(D,i) = \overline{\mathbf{T}}_{D \cup \{i\}}$, we obtain equation (17). ∎

5. CONCLUSION

We presented in this paper a Markovian model for the throughput analysis of multihop packet radio networks, and which is applicable to a large class of access protocols. In the first part of the paper we described the structure of the Markov chain, and studied the existence of a product form solution for the stationary probabilities. We showed that the existence of a product form solution is equivalent to the property known as *reversibility*, and we gave a criterion which allows the existence of this property to be easily determined from the specification of the access protocol, the network topology, and the traffic pattern. In the second part of the paper we derived general expressions for the link throughputs of such a system, which we then particularized for the cases of perfect capture and zero capture.

REFERENCES

[1] R. Kahn, "The Organization of Computer Resource into a Packet Radio Network", *IEEE Trans. Comm.*, COM-25, January 1977.

[2] R. Kahn *et al.*, "Advances in Packet Radio Technology", *Proc. IEEE*, vol. 66, no. 11, November 1978

[3] F. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network Part I : Slotted ALOHA", *IEEE Trans. Comm.*, COM-28, February 1980

[4] F. Tobagi, "Analysis of a Two-Hop Centralized Packet Radio Network - Part II : Carrier Sense Multiple Access", *IEEE Trans. Comm.*, COM-28, February 1980

[5] I. Gitman, "On the Capacity of Slotted ALOHA Networks and Some Design Problems", *IEEE Trans. Comm.*, COM-23, March 1975

[6] M. Sidi and A. Segall, "A Three-Node Packet-Radio Network", *Proc. INFOCOM'83*, San Diego, California, May 1983.

[7] M. Sidi and A. Segall, "Two Interfering Queues in Packet-Radio Networks", *IEEE Trans. on Comm.*, COM-31, January 1983.

[8] G. Fayolle and R. Iasnogorodski, "Two Coupled Processors – The Reduction to a Riemann-Hilbert Problem", *Wahrscheinlichkeitstheorie*, pp. 1-27, 1979.

[9] R. Boorstyn and A. Kershenbaum, "Throughput Analysis of Multihop Packet Radio", *Proc. ICC'80*, Seattle, Washington, June 1980.

[10] F. Tobagi and J. Brásio, "Throughput Analysis of Multihop Packet Radio Networks Under Various Channel Access Schemes", *Proc. INFOCOM'83*, San Diego, California, May 1983.

[11] B. Maglaris, R. Boorstyn and A. Kershenbaum, "Extensions to the Analysis of Multihop Packet Radio Networks", *Proc. INFOCOM'83*, San Diego, California, May 1983.

[12] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics", *IEEE Trans. Comm.*, COM-23, December 1975.

[13] P. Spilling and F. Tobagi, "Activity Signalling and Improved Hop Acknowledgements in Packet Radio Systems", *Packet Radio Temporary Note* #283, SRI International, January 1980

[14] L. Kleinrock, *Queueing Systems*, Vol. 1, Wiley, New York, 1975.

[15] F. Kelly, *Reversibility and Stochastic Networks*, Wiley, New York, 1979.

[16] E. Çinlar, *Introduction to Stochastic Processes*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[17] S. Ross, *Applied Probability Models With Optimization Applications*, Holden-Day, S. Francisco, 1970.

APPENDIX IV.

M. Nassehi and F. A. Tobagi, "Performance of Gateway-to-Gateway and End-to-End Flow Control Procedures in Internet Environments" in *Proceeding of the 21st IEEE Conference on Decision and Control,* Orlando, Florida, December 1982.

PERFORMANCE OF END-TO-END AND GATEWAY-TO-GATEWAY FLOW CONTROL PROCEDURES IN INTERNET ENVIRONMENTS*

Mehdi Nasschi and Fouad Tobagi

Computer Systems Laboratory
Stanford University
Stanford, CA 94305

Abstract

A performance comparison between end-to-end flow control (EEFC) and gateway-to-gateway flow control (GGFC) in internet environments is presented. The performance is measured in terms of average delivery delay of packets. First, a new technique for computing the average delivery delay across a network is introduced. It is shown that, for a given input rate to the network, there exists an optimum time-out which minimizes the average delivery delay. Then the performance is evaluated for EEFC and GGFC in an internet environment. It is observed that when the networks are in tandem, GGFC offers a better performance over that of EEFC. However, in general configurations where there is a high degree of traffic bifurcation between the networks, only under adaptive routing does GGFC result in a lower average delivery delay than that of EEFC. Finally, routing and flow control in internets are discussed.

I. Introduction

As computer communication networks multiply in number, it becomes more desirable to interconnect these networks in order to broaden their user services. The interconnection of networks is implemented through entities called gateways, which are interfaced to the individual networks as hosts. As in the case of a single network, a reliable delivery of packets between the end hosts must be provided. When there is some probability of packet loss the reliable delivery can be insured through a flow control mechanism such as windowing which incorporates an automatic-repeat-request (ARQ) feature.

In an internet environment the flow control may be implemented between the source and destination hosts, or it may be implmented across every network on the communication path, i.e., between the gateways as well as the gateways and end hosts. In this report we refer to the former case as end-to-end flow control (EEFC) and to the latter one as gateway-to-gateway flow control (GGFC). Our objective, here, is to make a performance comparison between EEFC and GGFC. Furthermore, we consider the use of routing and flow control algorithms to enhance the performance. The performance is measured in terms of packet delivery delay, i.e., the time elapsed since the packet arrives at the source host until the first correct copy of it is delivered to the destination host. This performance is a function of the end-to-end delay and probability of loss across the network as well as the input rate and retransmission frequency. As we will observe, there is an optimum retransmission frequency, or alternatively an optimum timeout, which minimizes the average delivery delay. Edge compares EEFC and GGFC in internet environments according to a number of criteria, including the average delivery delay. [1] To compute the delivery delay, he uses Sunshine's approach in which end-to-end delay distribution across the network at some total traffic rate called bandwidth is assumed to be known. [2] Then the average delivery delay is numerically calculated for different retransmission frequencies while adjusting the input rate in such a way that the total rate of traffic, comprising both input and retransmissions, be equal to the bandwidth. Consequently the tradeoff between average delivery delay and throughput is obtained. Edge concludes that the delivery delay under GGFC is lower than that of EEFC.

In addition to Sunshine, others have considered the problem of computing the average delivery delay under different conditions. Konheim analyzes the delivery delay when end-to-end delay is deterministic and the transmissions are slotted. [3] His assumption of deterministic end-to-end delay results in an implicit Negative ACK (NACK) eliminating the problem of timeout optimization. Kleinrock and Kermani compute the delivery delay when packets are lost due to the destination host buffer overflow rather than transmission error. [4] To simplify the analysis, they assume that only the ACK for the last retransmission is accepted at the host source. Fayolle et al. analyze the delivery delay when packet losses are due to transmission error, but they too assume that only the ACK corresponding to the last retransmission is acceptable. [5]

Here we present a new technique for computing the delivery delay when packet losses are independent as in the case where transmission errors are the major contributers to packet losses. In the next section, assuming that the timeouts are exponentially distributed, we derive a compact formula for delivery delay distribution in terms of the end-to-end delay distribution. In Section III, the mean and coefficient of variation of end-to-end delay are used to obtain a stepwise estimation to its distribution. Then, from this stepwise distribution the average delivery delay is found. Assuming that the mean and coefficient of variation of end-to-end delay are given as functions of the total traffic rate, we find the average delivery delay as a function of input rate. Throughout we numerically optimize the timeout to obtain the minimum average delivery delay. In the final section, the performance of EEFC and GGFC in an internet are investigated. In general, our results confirm those of Edge regarding the performance advantage of GGFC over EEFC. In addition, we show that when there is a large degree of traffic bifurcation between the networks, only under adaptive routing does GGFC offer a better performance than that of EEFC. The advantages of GGFC under adaptive routing suggest using routing and flow control algorithms in internets. Some routing algorithm have been developed based on control theory. [6, 7, 8] We discuss their application to develop a routing and flow control algorithm for internet environments.

II. Analysis of Delivery Delay

In this section we derive a formula for computing expected delivery delay across a network. The packet delivery delay is defined as the time elapsed from when a packet arrives to

*This work was supported by the Defense Advanced Research Projects Agency under Contract #MDA 903-79-C-0201, monitored by the Office of Naval Research order #A03717.

Proceedings of CDC '82
Orlando, FL, 1982

the source host until its first correct copy is delivered to the destination host. We shall call the delay undergone by each copy across the network the end-to-end delay and assume that its distribution is given. Moreover, we assume that the loss of copies are independent from each other and have a given fixed probability.

Since, in general, the delivery delay distribution of a packet depends on the end-to-end delay of every copy of that packet, the exact analysis requires the knowledge of the joint distribution of the one-way delays. However, realizing that this joint distribution is often not known, we develop a simple model to characterize the dependencies between the end-to-end delays. To motivate this model, we consider two extreme cases. The first case is that of fixed routing—all the copies take the same route through the network. Given that first-in-first-out scheduling is utilized at the nodes, the order of copies at arrival to the network is the same as that upon their departure. The other extreme case occurs when the one-way delays are independent. This situation is realized when every copy takes a different path across the network from every other copy—i.e. there exists "complete alternate routing".

Based on the above observation we may model the network as consisting of a number of identical and disjoint paths. Every copy may be transmitted over any of these paths with equal probability. Furthermore the copies that are transmitted over the same path, although their ordering is preserved, have the same marginal end-to-end delay distribution. In fact in a network where there exists a large mixing of different traffic at every node we expect that all the copies in an stream experience approximately the same delay distribution. On the other hand, the copies taking different routes to the destination source undergo independent but again identically distributed end-to-end delays. Let us denote these paths by $\pi_1, \pi_2, \ldots, \pi_n$. In the following n is assumed to be integer, however, a real value may be used for n in the final formula of the expected delivery delay. Then n may be interpreted as the inverse of the probability that two copies are transmitted over the same path. It is also clear that the extreme case of the fixed routing and the complete alternate routing correspond to $n = 1$ and $n = \infty$ respectively.

Immediately after the arrival of a packet we transmit a copy of it over some path, say π_{k_0}. Without loss of generality we can assume that the packet has arrrived at time zero. Denote the delay incurred by this copy over the network by $y_0^{(k_0)}$ and its reception time by $z_0^{(k_0)}$. Furthermore denote the time of the i^{th} retransmission that is routed over path π_k by $x_i^{(k)}$ and the delay of that copy by $y_i^{(k)}$ where $i = 1, \ldots, \infty$. Finally let $z_i^{(k)}$, $i = 1, \ldots, \infty$ denote the reception time of the i^{th} copy on the path π_k. From the description of the model we know that if $j > i$ then $z_j^{(k)} > z_i^{(k)}$, $k = 1, \ldots, n$. Let the $m_k{}^{\text{th}}$ packet on the path π_k be the first packet on that path which is not lost. Then if the delivery delay of the packet across the network is denoted by $\tilde{T}$, we have,

$$\Pr\left\{\tilde{T} > t\right\} = \prod_{k=1}^{n} \Pr\left\{z_{m_k}^{(k)} > t\right\} \tag{1}$$

We shall treat the term corresponding to the path of the first copy π_{k_0} separately.

$$\begin{aligned}\Pr\left\{z_{m_{k_0}}^{(k_0)} > t\right\} &= \sum_{i=0}^{\infty} \Pr\{m_{k_0} = i\} \Pr\left\{z_{m_{k_0}}^{(k_0)} > t \mid m_{k_0} = i\right\} \\ &= \sum_{i=0}^{\infty} (1-L) L^i \Pr\left\{z_i^{(k_0)} > t\right\} \end{aligned} \tag{2}$$

where L denotes the probability of loss of a packet. If we denote the probability density function of the $x_i^{(k_0)}$, $i = 0, \ldots, \infty$ by $p_{x_i^{(k_0)}}(x)$ then Eq. (2) can be reduced further as follows,

$$\begin{aligned}\Pr\left\{z_{m_{k_0}}^{(k_0)} > t\right\} &= (1-L) \Pr\left\{y_0^{(k_0)} > t\right\} \\ &\quad + \sum_{i=1}^{\infty} (1-L) L^i \Pr\left\{y_i^{(k_0)} + x_i^{k_0} > t\right\} \\ &= (1-L) \Pr\left\{y_0^{(k_0)} > t\right\} \\ &\quad + \sum_{i=1}^{\infty} (1-L) L^i \int_0^{\infty} \Pr\left\{y_i^{(k_0)} > t - x\right\} p_{x_i^{(k_0)}}(x)\, dx \end{aligned} \tag{3}$$

In order to get a compact formula for the expected delivery delay we assume that the time-outs are exponentially distributed. In the sequel, we compare the results based on this assumption to the exact results. Let τ denote the (average) time-out period. Since the routing of the packets is uniform, the transmission time of the i^{th} copy on path π_k, $x_i^{(k)}$, has an Erlang distribution with parameter i and mean τ. Letting $P(y)$ denote the distribution of the end-to-end delays; we get

$$\begin{aligned}\Pr\left\{z_{m_{k_0}}^{(k_0)} > t\right\} &= (1-L)[1 - P(t)] \\ &\quad + \sum_{i=1}^{\infty} (1-L) L^i \int_0^{\infty} [1 - P(t-x)] \frac{\left(\frac{1}{n\tau}\right)^i x^{i-1}}{(i-1)!} e^{-\frac{x}{n\tau}}\, dx \end{aligned} \tag{4}$$

Interchanging the order of the summation and integration we have,

$$\begin{aligned}\Pr\left\{z_{m_{k_0}}^{(k_0)} > t\right\} &= (1-L)[1 - P(t)] \\ &\quad + L \int_0^{\infty} \frac{1-L}{n\tau} [1 - P(t-x)] e^{-\frac{x}{n\tau}} \sum_{i=0}^{\infty} \frac{\left(\frac{Lx}{n\tau}\right)^{i-1}}{(i-1)!}\, dx \end{aligned} \tag{5}$$

or

$$\begin{aligned}&\Pr\left\{z_{m_{k_0}}^{(k_0)} > t\right\} \\ &\quad = 1 - (1-L) P(t) - L \int_0^t \frac{1-L}{n\tau} e^{-\frac{1-L}{n\tau} x} P(t-x)\, dx \end{aligned} \tag{6}$$

If we do a similar computation for the terms corresponding to the paths other than π_{k_0}, we get

$$\Pr\left\{z_{m_k}^{(k)} > t\right\} = 1 - \int_0^t \frac{1-L}{n\tau} e^{-\frac{1-L}{n\tau} x} P(t-x)\, dx \tag{7}$$

Combining Eq. (6) and Eq. (7) with Eq. (1) we get

$$\Pr\left\{\tilde{T} > t\right\} = \left[1 - (1-L)P(t) - L\int_0^t \frac{1-L}{n\tau} e^{-\frac{1-L}{n\tau}x} P(t-x)\,dx\right] \cdot \left[1 - \int_0^t \frac{1-L}{n\tau} e^{-\frac{1-L}{n\tau}x} P(t-x)dx\right]^{n-1} \quad (8)$$

Note that the kernel of $P(t-x)$ in Eq. (8) is the probability density function of a random variable exponentially distributed with mean $n\tau/(1-L)$. The average interarrival time of retransmitted copies that are not lost is also exponentially distributed and has the same mean. This observation suggests that we could have obtained Eq. (8) more directly by considering only correctly-received copies. Nevertheless we have chosen the above approach because it is independent of the distribution of the time-outs. In the sequel using a fixed time-out, we apply numerical methods to this approach to obtain the exact results. We then use these exact results to check the approximate results based on the exponentially-distributed time-outs.

We know that the average delivery delay, T can be found as follows,

$$T = \int_0^\infty \Pr\left\{\tilde{T} > t\right\} dt \quad (9)$$

For general n, Eq. (8) may be used in Eq. (9) in order to find T. there is no simple compact expression for this general case and we must resort to numerical evaluation of the integral. However, in the two extreme cases of fixed routing and complete alternate routing we can make some further reductions.

For fixed routing we have $n = 1$ and

$$\Pr\left\{\tilde{T} > t\right\} = 1 - (1-L)P(t) - L\int_0^t \frac{1-L}{\tau} e^{-\frac{1-L}{\tau}x} P(t-x)\,dx$$

and using Eq. (9)

$$T = D + \frac{L}{1-L}\tau \quad (10)$$

For alternate routing we have $n = \infty$. Note that

$$\lim_{n\to\infty}\left[1 - \int_0^t \frac{1-L}{n\tau} e^{-\frac{1-L}{n\tau}x} P(t-x)\,dx\right]^{n-1} = \exp\left[-\frac{1-L}{\tau}\int_0^t P(x)\,dx\right] \quad (11)$$

Applying Eq. (11) to Eq. (8) and combining the result with Eq. (9), we get

$$T = \int_0^\infty [1 - (1-L)P(t)] \exp\left[-\frac{1-L}{\tau}\int_0^t P(x)\,dx\right] dt \quad (12)$$

III. Delivery Delay in a Network

Here we use the results of the previous section to evaluate the delivery delay in a network. We start by presenting some numerical results in a network where the end-to-end delay distribution is not a function of the load. Based on some approximations, we can express the average delivery delay only in terms of the mean and coefficient of variation of the one-way delay in addition to the time-out period and the probability of packet loss. As expected we observe that, as long as the one-way delay is independent of the load, reducing the time-out period always decreases the average delivery delay. However, we know that a shorter timeout period results in a larger retransmission traffic which in turn should increase the end-to-end delay. Therefore to account for this effect, we next assume that the mean end-to-end delay is given as a function of the total load on the network. Then we get a more realistic behavior of the average delivery delay versus the time-out period. It is observed that the average delivery delay is *minimized* for some optimum time-out period.

In the derivation of the average delivery delay presented in the previous section, the degree of bifurcation of traffic was accounted for by the parameter n. Here, to simplify the numerical computations, we apply that formula only to the special cases of fixed routing ($n = 1$) and complete alternate routing ($n = \infty$). In the results obtained in this section all the other cases are in the range between these two extremes.

For the case of the fixed routing the average delivery delay is simply expressed in terms of mean one-way delay, time-out period and probability of packet loss by Eq. (10). In the case of the complete alternate routing, however, the relationship is more complicated as can be obeserved from Eq. (12). Although the integrals in Eq. (12) complicate the numerical analysis they also suggest that the average delivery delay T is not very sensitive to the exact distribution of the end-to-end delay, $P(y)$. We take advantage of this fact and approximate $P(y)$ in such a way that the integrals may be performed explicitly such as in the following:

$$P(y) = \sum_{i=1}^{N} p_i u(y - y_i) \quad (13)$$

where

$$u(x) = \begin{cases} 0, & \text{if } x < 0; \\ 1, & \text{if } x >= 0. \end{cases}$$

Furthermore to simplify the notation and expressions in the sequel, we find it advantageous to introduce $y_0 = 0$, and $y_{N+1} = 0$ along with $p_0 = p_{N+1} = 0$; then substituting $P(y)$ as given by Eq. (13) in Eq. (12) and performing the integrations we have

$$T = \frac{\tau}{1-L}\sum_{j=1}^{N+1}\frac{[L + (1-L)q_j]}{r_j} \cdot \left\{\exp\left[-\frac{1-L}{\tau}(r_j y_{j-1} - s_j)\right] - \exp\left[-\frac{1-L}{\tau}(r_j y_j - s_j)\right]\right\} \quad (14a)$$

where

$$q_j = \sum_{i=j}^{N} p_i \quad (14b)$$

$$r_j = \sum_{i=0}^{j} p_i \quad (14c)$$

$$s_j = \sum_{i=0}^{j} p_i y_i \quad (14d)$$

For most communication systems, analysis can only provide the mean and coefficient of variation of the (one-way) delay. Also, results based on measurements or computer simulation are usually more accurate for the first few moments than for the complete distribution of the delay. Therefore we require that the approximate distribution given by Eq. (13) only have its first

two moments equal to those of the true distribution.

Since the distribution given by Eq. (13) offers $2N$ degrees of freedom, $N = 2$ shall be sufficient for an approximation satisfying the requirements. Therefore if we denote the mean and

coefficient of variation of end-to-end delay by D and c_D respectively the parameters of the approximate distribution must satisfy the following conditions,

$$p_1 + p_2 = 1 \tag{15a}$$

$$p_1 y_1 + p_2 y_2 = D \tag{15b}$$

$$p_1 y_1^2 + p_2 y_2^2 = (1 + c_D^2) D^2 \tag{15c}$$

The above equations have many solutions. We restrict y_1 and y_2 to be geometric inverses with respect to 0 and D:

$$\frac{D - y_1}{y_1} = \frac{y_2 - D}{y_2} \tag{15d}$$

The above condition is justified by the accuracy of the final results.

Solving *Eq.* (15) we have,

$$p_1 = 1 - p_2 = \frac{1}{2}\left(1 + \frac{c_D}{\sqrt{1 + c_D^2}}\right) \tag{16a}$$

$$y_1 = \left(1 + c_D^2 - c_D\sqrt{1 + c_D^2}\right) D \tag{16b}$$

$$y_2 = \left(1 + c_D^2 + c_D\sqrt{1 + c_D^2}\right) D \tag{16c}$$

Now, we can use Eq. (14) in conjunction with Eq. (16) to compute the average delivery delay in the case of complete alternate routing; i.e., $T = T(D, c_D, L, \tau)$. The dashed line in Figure 1 depicts the normalized average delivery delay as a function of the normalized time-out period for the case of complete alternate routing. Both normalizations are with respect to the average end-to-end delay. The dotted curve is exact for the case of the exponentially distributed one-way delay. It is otained by numerical methods using Eq. (12). The solid curve corresponds to the case of the fixed routing.

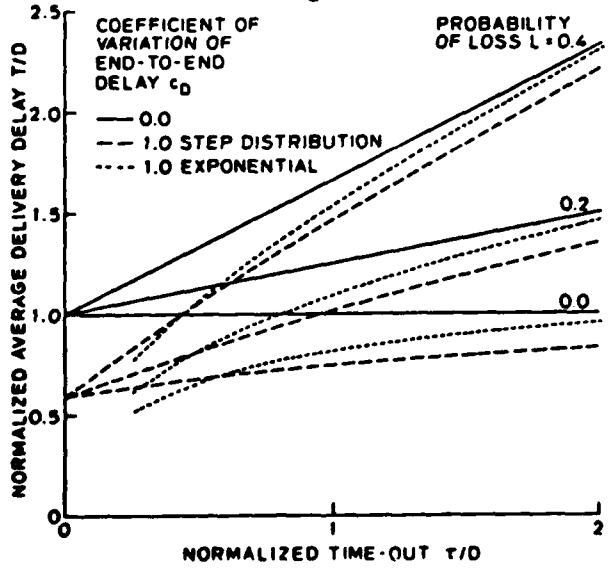


Fig. 1 *Average delivery delay in a load-insensitive network*

In the following, all the results are obtained using the approximate one-way delay distribution introduced above. Therefor we only need the mean and coefficient of variation of the one-way delay as a function of the load. To simplify the numerical computations we assume that the coefficient of variation is constant with respect to the load. This is a particulary good approximation when the traffic stream under study does not constitute a large portion of the total traffic on its path—i.e., when there is a large degree of mixing.

For the mean delay across the network, we use a one-pole function. Let r denote the average total rate of packet arrival from the traffics stream under consideration. Then the function expressing the mean end-to-end delay, D in terms of r is specified by three parameters a, b, and c as follows:

$$D(r) = a\left[1 + b\frac{\frac{r}{c}}{1 - \frac{r}{c}}\right] \tag{17}$$

Here, a denotes the mean delay across the network under no load from the commodity under consideration. b can be interpreted as the sensitivity of the mean delay to the load. Finally, c is the capacity as seen by the commodity.

If the average delay that the ACK's undergo is denoted by T', the average number of retransmissions per packet will be $(T + T')/\tau$. Therefore if we denote the rate of input traffic by λ the average delivery delay is given by the following implicit equation in T

$$T = T\left[D\left(\left(1 + \frac{T + T'}{\tau}\right)\lambda\right), c_D, L, \tau\right] \tag{18}$$

Eq. (15) may be solved numerically by an iterative method for T. In Figure 2 normalized delivery delay has been shown versus the normalized timeout. Here the normalization is with respect to the average no load delay, a. We have shown curves for both fixed routing and alternate routing for different sensitivities. Notice the sudden jump of the delivery delay when the timeout is reduced. For all the cases there is an optimum timeout which minimizes the delivery delay. As it can be observed both the minimum delivery delay and the optimum timeout increase at the same time. In Figure 3 we have drawn the *minimum* delay and the optimum time out as functions of the utilization factor, $\lambda/(1 - L)c$.

IV. Internetting

IV.1 Model

Figure 4 depicts an internet consisting of four hosts communicating across two networks. The networks are connected by three gateways. At every gateway two levels of the communication protocol are explicitly shown. The first level is that of flow control which among other things insures the reliable delivery of the packets to the next flow control level on the communication path. This level was discussed and analyzed in the previous sections. The other level is implemented for routing the packets among the netwoks.

In Figure 4 flow control exists at the gateways as well as the end hosts. To insure reliable delivery it is sufficient to implement the flow control only at the end hosts. As mentioned in the introduction, we call the former case gateway-to-gateway flow control (GGFC) and the latter one end-to-end flow control (EEFC). In EEFC and GGFC the gateways connect two networks at datagram level and Virtual-circuit level, respectively.

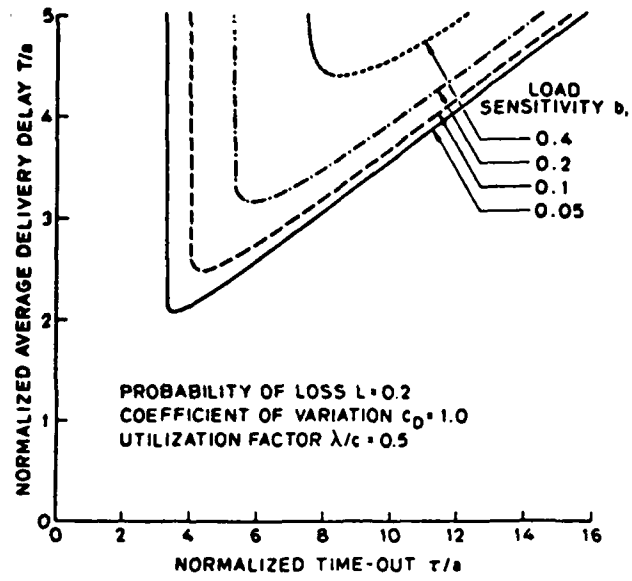


Fig. 2 *Average delivery delay when one-way delay is* $D(r) = a[1 + b(r/c)/(1 - r/c)]$

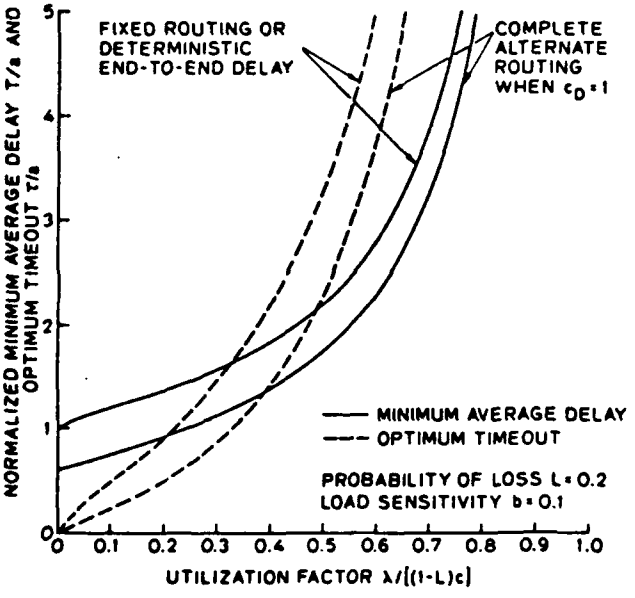


Fig. 3 *Minimum average delay and optimum timeout vs. input load.*

In the following we present a numerical comparison between the performance of these flow control procedures.

The examples we consider are those of the host pairs shown in Figure 4. The arrowed line segments indicate the connection between the hosts and gateways over the networks. Host H_S^S communicates to host H_S^D over networks 1 and 2 with gateway G_1 completing a series connection. Gateways G_2 and G_3, however, provide a parallel interconnection of the networks for communication between hosts H_P^S and H_P^D. The dotted contours between H_P^S and G_2, and G_3 and H_P^D are to suggest the possible existence of intermediate networks with negligible end-to-end delay and probability of loss—e.g., there may be a fast access link, or the logical gateway may reside in the host.

As in the previous section, in the following we shall optimize timeouts and possibly the routing parameters to obtain the minimum average delivery delay. Like in any multi-user system, there are two types of optimization in the internet under study.

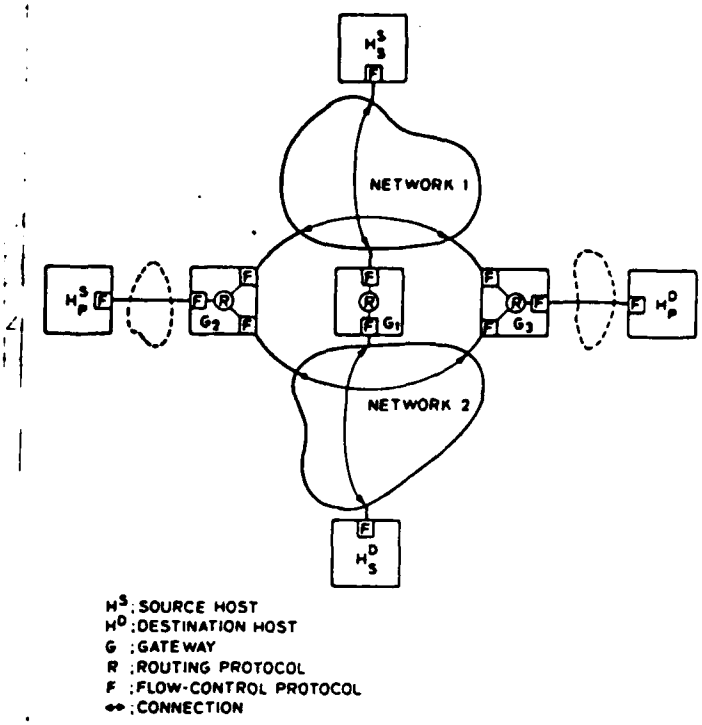


Fig. 4 *Example of a network interconnection.*

One is user optimization in which the control parameters of a host pairs are optimized to achieve the minimum delivery delay for that pair. The other is system optimization which consists of optimizing control parameters to obtain the minimum of some system cost measure, say the average delivery delay over the internet. At the end of this section, where we discuss an algorithm for routing and flow control in an internet, we address the problem of system optimization. Here, however, our objective is solely to compare the EEFC and GGFC in terms of their performance. Therefore it suffices to assume the characteristic of the networks and find the average delivery of the host pairs separately.

IV.2 Series Interconnection

In Section III we showed how the average delivery delay is computed in terms of the arrival rate of new packets into the network λ and the transmission timeout τ. As in that section, we assume that in addition to the probability of packet loss L_i, $i = 1, 2$, we have a discrete approximation to the end-to-end delay p.d.f. across network i, $i = 1, 2$, denoted by $p_i(t, s)$. Let us denote the average delivery delay over network i and its minimum with respect to the timeout by T_i and $\bar{T}_i$ respectively. Now consider the connection between the hosts H_S^S and H_S^D in Figure 4. Under EEFC, G_1 simply routes the packets, as does any other nodes in networks 1 and 2. Therefore G_1 and networks 1 and 2 may be replaced by a single equivalent network having the end-to-end delay p.d.f.,

$$p_S(t, \lambda) = p_1(t, \lambda) * p_2(t, \lambda) \tag{19}$$

and probability of packet loss

$$L_S = 1 - (1 - L_1)(1 - L_2) \tag{20}$$

Here, $*$ denotes convolution with respect to t. Since both $p_1(t, \lambda)$ and $p_2(t, \lambda)$ consist of impulses, $p_S(t, \lambda)$ will too only consists of impulses. Therefore we may use the techniques presented in the previous sections to compute the average delivery delay

across the equivalent network, $T_S^{EE}(\lambda, \tau)$. As in the case of the single network we assume that the ACK delay is the smallest possible end-to-end delay; i.e., the position of the lowest impulse under no load.

When GGFC is implemented a packet is transmitted across network 1 every τ_1 units of time from H_S^S to G_1 until an ACK is received by H_S^S. Similarly after G_1 receives a correct and unduplicated copy it transmits it to H_S^D every τ_2 units of time until it receives an ACK. Therefore the total delivery delay T_S^{GG} and its minimum with respect to the timeouts $\mathcal{T}_S^{GG}$ will be

$$T_S^{GG}(\lambda, \tau) = T_1(\lambda, \tau_1) + T_2(\lambda, \tau_2) \tag{21}$$

$$\mathcal{T}_S^{GG}(\lambda) = \mathcal{T}_1(\lambda) + \mathcal{T}_2(\lambda) \tag{22}$$

In Figure 5 we have plotted $\mathcal{T}_S^{EE}$ and $\mathcal{T}_S^{GG}$ versus the utilization factor. Here, utilization factor is defined as the ratio of input rate λ over the minimum of the effective capacities of networks 1 and 2. In this figure we have assumed that networks 1 and 2 are identical and have the parameters given in the figure. As it can be observed GGFC results in a lower average delay—EEFC inefficiently loads network 1 with the packets that are lost over network 2 and vice versa. The difference in performance between these control strategies may become even larger if some of the network parameters are greatly different; as is the case in Figure 6 where the probabilities of loss are different.

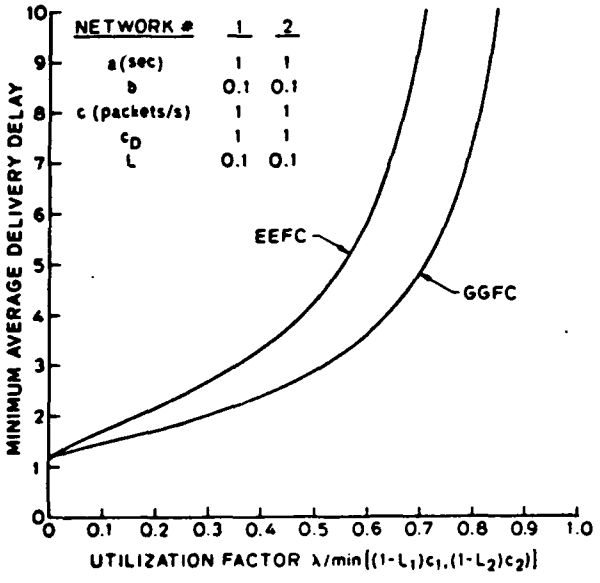


Fig. 5 *Performance of EEFC and GGFC in an internet consisting of two identical networks in tandem.*

IV.3 Parallel Interconnection

Now we consider the performance of GGFC and EEFC for the hosts H_P^S and H_P^D in Figure 4. Here, in addition to the timeout period there exists another control parameter, namely the routing parameter ρ. The fractions of the traffic routed over networks 1 and 2 are ρ and $\bar{\rho} = 1 - \rho$ respectively.

For EEFC, as in the case of the series interconnection, we may replace G_2, G_3, and networks 1 and 2 by an equivalent network. The p.d.f. of the end-to-end delay across this equivalent network will be

$$p_P(t, \lambda) = \rho p_1(t, \rho\lambda) + \bar{\rho} p_2(t, \bar{\rho}\lambda) \tag{23}$$

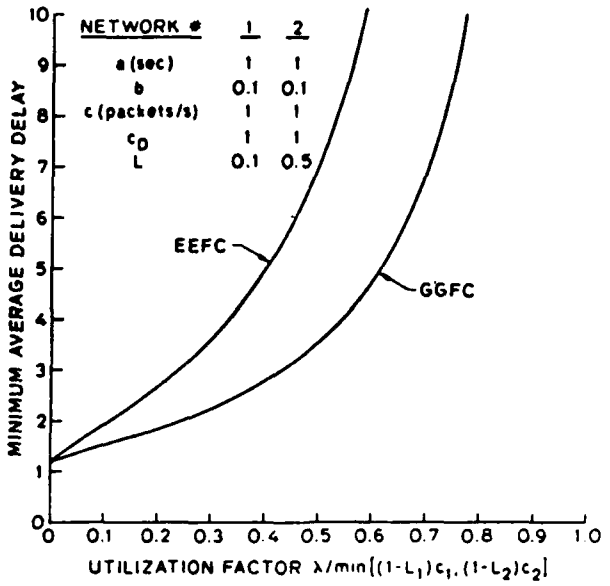


Fig. 6 *Performance of EEFC and GGFC in an internet consisting of two networks connected in tandum, with different probabilities of packet loss.*

and the corresponding probability of loss

$$L_P = \rho L_1 + \bar{\rho} L_2 \tag{24}$$

Then we proceed to compute the average delivery delay $T_P^{EE}(\lambda, \tau, \rho)$ as before.

We shall consider two routing strategies. One is proportional routing (PR) in which the traffic routed over each network is proportional to its effective capacity:

$$T_{PR}^{EE}(\lambda, \tau) = T_P^{EE}(\lambda, \tau, \rho_{PR}) \tag{25a}$$

where

$$\rho_{PR} = \frac{(1 - L_1)c_1}{(1 - L_1)c_1 + (1 - L_2)c_2} \tag{25b}$$

The other one is optimized routing (OR) in which for given λ and τ the delivery delay is minimized over ρ:

$$T_{OR}^{EE}(\tau, \lambda) = \min_{\rho} T_P^{EE}(\lambda, \tau, \rho) \tag{26}$$

From Eqs. (25) and (26) their corresponding timeout-optimized average delivery delays are numerically computed.

Now we turn our attention to the case of GGFC in parallel interconnection. As shown in Figure 4, the flow control levels for transmitting packets over different networks are logically distinguishable—if the initial copy of a packet is transmitted over some network the subsequent copies are transmitted over the same network. Therefore, the average delivery delay between the hosts will be

$$T_P^{GG}(\lambda, \rho) = \rho \mathcal{T}_1(\rho\lambda) + \bar{\rho} \mathcal{T}_2(\bar{\rho}\lambda)$$

Depending on whether routing is proportional or optimized we have respectively

$$T_{PR}^{GG}(\lambda) = T_P^{GG}(\lambda, \rho_{PR})$$

and

$$T_{OR}^{GG}(\lambda) = \min_{\rho} T_P^{GG}(\lambda, \rho)$$

When the networks are connected in parallel, we may use PR and OR as well as EEFC or GGFC which results in four cases: EEPR, EEOR, GGPR, and GGOR. When all the parameters of the networks other than the capacities are equal, the average delivery in all these four cases will be the same for the following reason. The conditions under which minimum delay is achieved are proportional routing and equal timeouts across the networks. It is clear that these conditions, which are achievable in all the four cases, are necessary for minimizing the delay. From Eq. (17) we observe that under these conditions absolute and marginal end-to-end delays are the same for both networks. Since the timeouts and probabilities of loss are also the same, the marginal average delivery delays will be the same. The equality of the marginal average delivery delays, however, is a necessary condition for minimality of the average delivery delay.

Figure 7 gives the performance when one of the networks has a larger noload end-to-end delay than the other one. In the case of GGPR, half of the packets are routed over the network with the large delay and since all the subsequent copies of the packets are also routed over the same network, they undergo a large delay resulting in inferior performance. The GGOR is not as degraded because it routes a larger fraction of the packets over network 1. In the case of EEFC subsequent packets are not necessarily routed over the network with large delay resulting in a better performance. Note that by assumption the ACK delays for EEPR and EEOR are equal to the minimum of the end-to-end delays across networks 1 and 2 and therefore less than the ACK delays for either GGPR or GGOR. This difference in ACK delays results in a lower average delivery delays for EEPR and EEOR at high utilization rates than those of GGPR or GGOR.

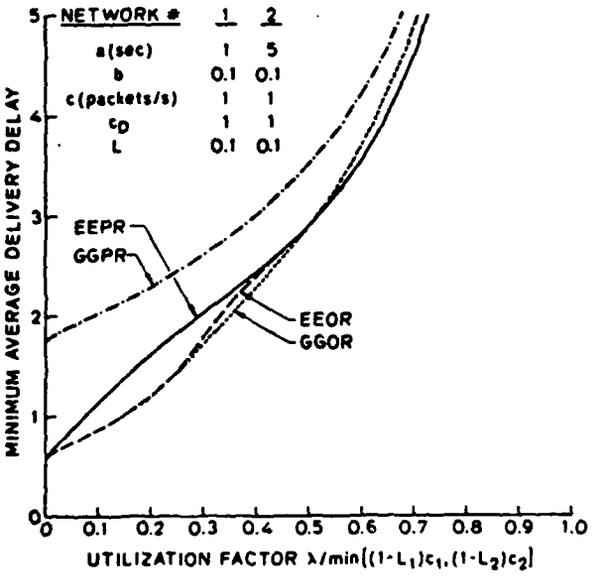


Fig. 7 *Performance of EEFC and GGFC in an internet consisting of two networks with different no-load end-to-end delays connected in parallel.*

Figure 8 shows the case where load sensitivities are different. As can be expected OR results in a better performance by routing the packets over the net with the lower delay. The difference in performance where the coefficient of variations were different was not significant.

In Figure 9 we show the performance results for networks with different probabilities of loss. The results are similar to those of series interconnection and for the same reason, under GGFC the timeouts are made longer for packets routed over the network with larger probability of loss, but under EEFC the

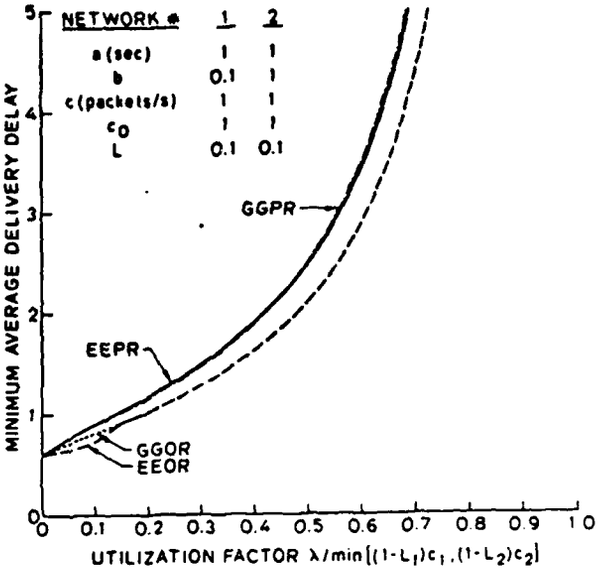


Fig. 8 *Performance of EEFC and GGFC in an internet consisting of two networks with different end-to-end delay load sensitivities connected in parallel.*

timeouts are the same for the packets transmitted over either network. Again OR results in a somewhat lower delay by routing the packets over the network with smaller probability of loss.

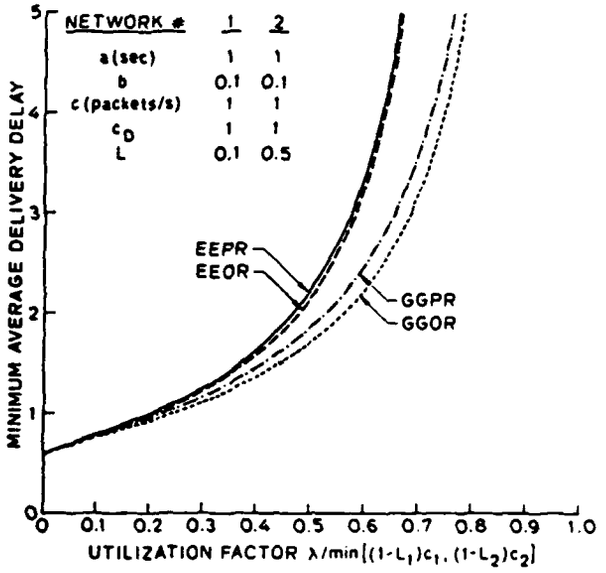


Fig. 9 *Performance of EEFC and GGFC in an internet consisting of two networks with different probabilities of packet loss connected in parallel.*

IV.4 On Routing and Flow Control in Internets

In the previous sections, we alluded to the problem of system optimization regarding routing and flow control in internets. Here, we address some of the issues involved in this problem. First, we shall categorize the networks comprising an internet into two classes. One class consists of those networks in which local traffic constitutes the main portion of the total traffic. The second class of networks, on the other hand, includes those in which a significant fraction of their total traffic is internet traffic. To be able to make the routing and flow control in an internet amenable to the analysis techniques developed in the previous sections, we assume that the internet consists of only the networks in the first class. In view of the fact that the majority of networks carry mainly local traffic at the present, this assumption is not very restrictive.

We observed the advantages in performance of GGOR in an internet. Consequently, we consider here only those routing and flow control algorithms that implement GGOR. As before, we are only concerned with that part of the flow control protocol which insures reliable delivery of packets. Golestaani demonstrates how the main function of flow control—i.e., regulation of the input traffics at the source for purposes of congestion minimization—can be incorporated into the routing algorithm. [9] The algorithm which we discuss consists of two segments: an inner segment which optimizes the timeouts to achieve the minimum average delivery delay across every network in the internet; an outer segment which optimizes the routing parameters at every gateway in order to minimize some cost function defined on the average delivery delays across the networks.

Let HG denote the set of all hosts and gateways in the internet. Let $i, j \in HG$. If there is a direct connection from i to j across some network, we denote it by c_{ij}. Let C be the set of all those connections. Let λ_{ij}, r_{ij}, and T_{ij} denote the traffic rate, timeout, and average delivery delay corresponding to c_{ij}, respectively. Let D_{ij} and r_{ij} denote the mean end-to-end delay and total load on c_{ij} respectively. The inner segment of the algorithm which is implemented at every i, $\forall i$, $\ni j \in HG \bigwedge c_{ij} \in C$, numerically optimizes r_{ij} to achieve the minimum average delivery delay, $T_{ij}(\lambda_{ij})$. The numerical optimization is likely to require $T_{ij}(\lambda_{ij})$ and $\frac{\partial}{\lambda_{ij}} T_{ij}(\lambda_{ij})$ at every iteration. $T_{ij}(\lambda_{ij})$ and $\frac{\partial}{\lambda_{ij}} T_{ij}(\lambda_{ij})$ may be measured from the delivery delay of packets that are received between the iterations. Alternatively $D_{ij}(r_{ij})$ and $\frac{\partial}{\partial r_{ij}} D_{ij}(r_{ij})$ may be measured and Eq. (18) used to compute $T_{ij}(\lambda_{ij})$ and $\frac{\partial}{\lambda_{ij}} T_{ij}(\lambda_{ij})$. The latter method clearly results in better estimations of $T_{ij}(\lambda_{ij})$ and $\frac{\partial}{\lambda_{ij}} T_{ij}(\lambda_{ij})$ than those of the former method. Given that, on c_{ij}, the probability of packet loss L_{ij} is not negligible, only a fraction of the copies transmitted are successfully received making the measurements of $D_{ij}(r_{ij})$ and $\frac{\partial}{\partial r_{ij}} D_{ij}(r_{ij})$ more reliable than $T_{ij}(\lambda_{ij})$ and $\frac{\partial}{\lambda_{ij}} T_{ij}(\lambda_{ij})$. Of course, if $D_{ij}(r_{ij})$ and $\frac{\partial}{\partial r_{ij}} D_{ij}(r_{ij})$ can be calculated analytically, e.g. using a queueing model of the network, the evaluation of the timeouts may be performed more rapidly.

If the cost function is defined properly, the outer segment of the algorithm which optimizes the routing parameters at gateways can be identical to the algorithms developed for optimum routing of traffic in a single network. Since most of these algorithms are based on convex optimization techniques, we must define the cost function in such a way that it is a convex function in the λ_{ij}, $\forall_{ij}$. As shown in Figure 3, $T_{ij}(\lambda_{ij})$ for low values of λ_{ij} is not convex. However, we may define the function $\hat{T}_{ij}(\lambda_{ij})$ in such a way that $\hat{T}_{ij}(\lambda_{ij})$ is convex for all values of λ_{ij}. Then the total cost may be defined as

$$c = \sum_{i,j \ni c_{ij} \in C} \hat{T}_{ij}(\lambda_{ij})$$

Now, finding the optimum routing in the internet environment is equivalent to finding the optimum routing in a network. The set of nodes of this network is HG. For every connection c_{ij} in the internet we have a link, l_{ij} from node i to j. Now if we interpret λ_{ij} and $T_{ij}(\lambda_{ij})$ as the rate of traffic and the delay function on l_{ij}, then c will represent the total delay over the network.

There are several quasi-static routing algorithms that can be employed as the outer segment of the routing and flow control algorithm in internets. [6, 7, 8] The quasi-static property of the algorithm implies that the variations in the input rates to the source hosts and the variations in the characteristics of the networks comprising the internet must be slow relative to the updating frequency of the routing parameters. Moreover, we require the frequency of updating the inner segment be much larger than that of the outer segment. Otherwise, since $T_{ij}(\lambda_{ij}, r_{ij})$ is not necessarily convex, neither of the segments may converge.

Conclusion

A new technique for computing the delivery delay in a network was presented. The main assumption in the underlying model was that the timeouts are random and exponentially distributed. It was shown that the results based on this assumption are close to the exact values. Then the average delivery delay across the network was computed in terms of mean and coefficient of variation of end-to-end delay. Using this technique, we evaluated the performance of end-to-end and gateway-to-gateway flow controls in an internet. We observed that when the networks are in tandem, GGFC offers better performance than that of EEFC. However, when there is a high degree of traffic bifuraction between the networks, only under adaptive routing GGFC results in a lower average delivery delay than that of EEFC. When GGFC is employed, the optimum timeouts may be computed at gateways and hosts using numerical methods. Then any routing algorithm which minimizes the average delay in a network can be used to minimize a cost function of the average delivery delays across the internet. When developing the routing and than control algorithm, we only considered those internets which consists of networks carrying mainly local traffic. To develop an algorithm for routing and flow control in the general case requires further research. It seems that techniques based on control theory are most promising in devising such an algorithm.

References

[1] S. W. Edge, "Comparison of the Hop-by-Hop and Endpoint Approaches to Network Interconnection," *Proceedings of the International Symposium on Flow Control in Computer Networks*, Versailles, France, Feb. 12-14, 1979.

[2] C. A. Sunshine, *"Interprocess Communication Protocols for Computer Networks,"* Stanford Electronics Laboratories, Technical Report 105, 1975.

[3] A. G. Konheim, "A Queueing Analysis of two ARQ Protocols," *IEEE Trans. on Comm., Vol. COM-28, No. 7, pp. 1015-29,* 1980.

[4] L. Kleinrock and P. Kermani, "Static Flow Control in Store-and-Forward Computer Networks," *IEEE Trans. on Comm., Vol. COM-28,* pp. 271-279, 1980.

[5] G. Fayolle, E. Gelenbe, and G. Pujolle, "An Analytical Evaluation of the Performance of the Send and Wait Protocol," *IEEE Trans. on Comm., Vol. COM-26,* pp. 313-319, 1978.

[6] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," *IEEE Trans. on Comm., Vol. COM-25,* pp. 73-83, 1977.

[7] A. Segall, "The Modelling of Adaptive Routing in Data-Communications Networks," *IEEE Trans. on Comm., Vol. COM-25,* pp. 85-95, 1977.

[8] D. P. Bertsekas, *"A Class of Optimal Routing Algorithms for Communication Networks,"* Report LIDS-P-1012, 1980, Laboratory of Information and Decision Systems, MIT, Cambridge, MA 02139.

[9] S. I. Golestaani, *"A Unified Theory of Flow Control and Routing in Data Communication Networks,"* Ph.D thesis, Department of Electrical Engineering and Computer Science, MIT, 1980.