1

UCLA–ENG–8003
March 1980

DARPA CONTRACT NO. MDA 903–77–C–0272

MEASUREMENT AND EVALUATION STUDIES OF PACKET RADIO TELECOMMUNICATIONS SYSTEMS FINAL TECHNICAL REPORT

Principal Investigator: Wesley W. Chu
Co-Principal Investigator: Leonard Kleinrock

March 19, 1980

COMPUTER SCIENCE DEPARTMENT
School of Engineering and Applied Science
University of California
Los Angeles

UCLA SCHOOL OF ENGINEERING AND APPLIED SCIENCE

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER UCLA-ENG-8003	2. GOVT ACCESSION NO. AD-A157696	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Measurement and Evaluation Studies of Packet Radio Telecommunications Systems Final Technical Report		5. TYPE OF REPORT & PERIOD COVERED 7/1/77 - 1/30/80
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Wesley W. Chu Zaw-Sing Su		8. CONTRACT OR GRANT NUMBER(s) MDA 903-77-C-0272
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Engineering and Applied Science University of California Los Angeles, California 90024		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DARPA Order No. 2496
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE March 19, 1980
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED (A)

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 is the final report for the DARPA Contract MDA 903-77-C-0272 for Measurement and Evaluation Studies of Packet Radio Telecommunications Systems by UCLA Computer Science Department covering the period from July 1, 1977 to January 30, 1980. The main purpose of this contract is to develop tools for, and to perform, evaluation of Packet Radio Network in support of the evolution of packet radio technology. The
(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) activities include:

(1) Measurement Activities: We have continued guiding the development of measurement tools and performed measurement experiments. Progress in this contract period was slow due to the lack of testbed network availability and the occupation of implementors in other priority tasks, such as the deployment of Ft. Bragg packet radio network.

(2) Simulation Activities: Two simulation packages have been developed, a detailed simulation which closely simulates the radio subnet, and a system-level simulation which aims at high level investigation of pertinent design issues.

(3) Analytical Investigation: It has progressed from studies of channel access schemes to packet hop transport system for multihop networks. Employing mathematical analysis and simulation, we have conducted studies in support of the PRNET design and implementation efforts.

Significant contributions have been made by the PRNET investigation efforts in support of the continuing evolution of packet radio technology. The reliance on simulation and measurement will likely expand with the increasing complexity of the subject model. A consolidation of the measurement and simulation facilities is thus considered crucial to the continuing investigation efforts.

UCLA-ENG-8003
March 1980

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

MEASUREMENT AND EVALUATION STUDIES OF PACKET RADIO COMMUNICATIONS SYSTEMS

Final Technical Report

March 19, 1980

DARPA Contract: MDA 903-77-C-0272
DARPA Order No.: 2495

Principal Investigator: Wesley W. Chu
(213) 825-2047

Co-Principal Investigator: Leonard Kleinrock

Report Period: July 1, 1977 - January 30, 1980

Computer Science Department
School of Engineering and Applied Science
University of California, Los Angeles

Acknowledgement

During the period from July 1, 1977 to June 23, 1978 the Project was lead by Dr. Fouad A. Tobagi, and from August 15, 1978 to January 30, 1980 by Dr. Zaw-Sing Su. Stanley E. Lieberson and Medy Elsadanini played key roles in measurement and simulation efforts. Others technically involved in this Project include Joseph Hellerstein, Nancy Ishii, Robert Lincoln, Jeffrey P. Schaffer, and Judi Uttal.

Contents

I. INTRODUCTION

This final report documents activities and accomplishments of UCLA Network Measurement Group for a continuing contract under the DARPA Contract Number MDA 903-77-C-0272. This contract, covering the period from July 1, 1977 to January 30, 1980, is for Measurement and Evaluation Studies of Packet Radio Communications Systems.

Under the auspices of Defense Advanced Research Projects Agency, the Packet Radio Network (PRNET) Project was established to investigate the feasibility of packet switching employing broadcasting transmission over radio channels and to explore the potentials of such technology. The PRNET Project has involved the efforts of a number of contracting institution. As one of the major participants, this UCLA Group has been responsible for PRNET performance analysis in support of the continuing evolution of PRNET. Through earlier efforts, an experimental PRNET was constructed and installed in the San Francisco Bay Area. In its development process, we defined the PRNET measurement functions, designed testbed facilities to support these measurement functions; generated a preliminary measurement plan, completed preliminary design of measurement data reduction programs as well as that of a detailed simulation program. At the beginning of this contract period, the implementation of the measurement facilities was partially completed. The main purpose of this contract is to continue quiding the implementation of testbed measurement facilities, developing measurement reduction programs and simulation programs, and to coordinate the employment of these tools as well as analytical techniques to continue carrying out PRNET performance analysis in support of the continuing exploration of packet radio technology.

We model the PRNET as composed of a packet radio subnet consisting of a number of switching nodes, the PRUs, connected together via common channel radio links, and other network devices, the TIUs and a station, connected to the subnet via wire links to the PRUs. The PRUs and the radio channel are considered comprising the basic resources of the subnet.

The central subject of our performance analysis efforts is PRNET throughput-delay behavior. Other measures, e.g., reliability, vulnerability, fairness, mobilty, are considered as constraints of our investigation. The PRNET throughput-delay behavior is a function of many system attributes:

A. Channel Usage -- The design of dynamic allocation and management schemes for the radio channel assures its efficient utilization. These schemes include: bandwidth management, channel access policy, and modulation scheme.

B. PRU Design -- Processing efficiency and buffer availability of the PRU, the packet switch of the radio subnet, directly affect network performance.

C. Operational Protocols -- Network protocols at various level govern different operational aspects of PRNET. They include PRU transmission scheduling, hop acknowledgement and retransmission, end-to-end acknowledgement and retransmission, flow control, routing, congestion control, and monitoring functions. The operational protocols constitute a major part of the network functions. Their importance to the network design is self-evidence.

D. Network Topology -- Topology of a radio network, owing to its time-variant nature, plays a much different role from that of a point-to-point wired network.

There are a number of possible directions for extending PRNET capabilities. Multistation and stationless operations extend the current PRNET design to accommodate the requirements for a larger geographical coverage and less vulnerable packet radio network. Internetworking offers further extension of PRNET capabilities through coordination with other packet switching networks.

We take a bottom-up general approach to our investigation. Beginning with the basic network elements, e.g., the radio channel and the PRUs, forming a single-hop packet radio subnet, our investigation has progressed toward the analysis and evaluation of multi-hop networks. An understanding of the current PRNET will provide a foundation for investigation efforts to support the design and development for internetworking, and multi-station and stationless operations.

For investigation technique, we coordinate efforts employing analysis, measurement, and simulation. Mathematical modelling and analysis have been used to establish basic principles and to guide the investigations. Measurement experiments have been used for extracting system parameters, calibrating and verifying investigation results. Simulation extends capabilities of measurement tools and analytical techniques to extensive enumerative studies, the study of large scale networks, and the study of design alternatives.

In the next section, we summarize our activities and accomplishments. We suggest areas for future efforts in the final section. Each of the publications issued under the support of this contract and other internal documents cited in this report is reproduced and attached in the Appendix.

II. ACTIVITIES AND ACCOMPLISHMENTS

The major activities during this contract period involve the development of simulation programs and continuing guidance of testbed measurement facility implementation. For PRNET investigation, we have continued the efforts employing analytical techniques. Near the conclusion of this contract, simulation programs became available for use. The implementation of measurement facilities remains incomplete. The specific activities and accomplishments are discussed below.

1. Development of Measurement Tools

We have continued guiding the construction, testing, and updating of testbed measurement facilities, and have also implemented a series of measurement data reduction programs. The measurement facilities for PRNET testbed [1] include:

-- Cumulative Statistics Collection Facilities: Cumulative statistics collection facilities are implemented in both the PRUs (for hop statistics) and the TIUs (for end-to-end statistics). They collect data regarding a variety of events accumulated over a specified collection period, and provided in the form of sums, frequencies, and histograms. Minor modification and updating according to the evolution in protocol design have been made.

-- Tracing Facility. Design of the tracing mechanism calls for a special type of packets, the Pick-Up Packets. These packets gather routing and delay statistics while passing through the network obeying the transport protocols. In PRTN#237 [2], we have documented its specification. It can be a useful tool for statistics collection and system debugging. Its implementation has yet to be completed.

-- Snapshot Statistics Collection Facility: Snapshot statistics are collected in the PRUs and the station. The snapshots describe instantaneous states of the PRUs. The implementation of snapshot collection facility has been completed. Through experimentation, a statistics biasing problem was identified [3]. Changes were implemented to eliminate a portion of this bias.

-- Experimental Traffic Generators: They allow the generation of artificial traffic streams from designated TIUs for pre-defined destinations and packet length. For arrival process, we have recommended the implementation of constant, feedback dependent, and Poisson arrival processes. The presently existing traffic generation is feedback dependent. An algorithm for the efficient generation of Poisson traffic in

TIU has been specified [4].

-- Station Measurement Process: For single station PRNET operation, the station provides central control for the entire network. It was therefore logical to embed the control of the execution of network measurement functions in the station. It controls, for example, initiating and terminating experiments, enabling and disabling statistics collection facilities. It is to the station all measurement data destined, time-stamped, and stored for off-line reduction. It is also through the station the measurement parameters are set. A number of updates to the station measurement process have been incorporated to conform to the evolution of network protocols.

In addition to the measurement facilities implemented on PRNET testbed, we have designed reduction programs for the planned experiments. A number of them have been constructed in support of measurement activities. An annotated list of the existing programs [5] is included in the Appendix.

2. Development of Simulation Packages

Following the preliminary design, efforts have been made to construct a detailed PRNET simulation. A brief description of its design is documented in PRTN#244 [6]. Documentation for its current state is included in the Appendix [7]. It has recently been used in assisting the evaluation of Ft. Bragg PRNET configuration.

We have undertaken another simulation effort to complement the functions of the detailed simulation. Aiming at this objective, we have developed a system-level PRNET simulation package following a building-block approach described in PRTN#268 [8]. This package consists of three programs, each simulates PRNET at a different level of details. Its architecture allows flexibilities for PRNET evaluation, studying design alternatives, extensive enumerations, and the isolation of individual design issues. A description of this simulation package is documented [9]. We have used this package for evaluating design alternatives for routing strategy and power control issues [10], performance estimation for Ft. Bragg PRNET [11], comparison of different transmisrion scheduling schemes [12], etc.

3. Measurement Experiments

During the initial period of this contract, a measurement plan was finalized and documented [13]. The testbed was not readily available for performaning experiments during the second half of the contract period. The execution of measurement experiments has thus been substantially varied from the plan. Orientation of the measurement program has also been re-directed during the course of this contract towards calibration and verification of the simulation packages. The experiments carried out include the following:

-- Transceiver Activities: The objective of this experiment was to assess the efficiency of the radio transceiver. Being the first experiment, it was also used for testing and exercising the measurement facilities. The fraction of time it is busy, transmitting, or receiving; and the fraction of time its receiver is enabled and disabled were measured and reported. A bias in PRU statistical collection process was recognized [3] and partially corrected.

-- Hop-By-Hop Acknowledgement Protocol: The objective of this experiment was to evaluate the currently implemented echo hop acknowledgement scheme. Results were obtained and documented [14]. This experiment was conducted using feedback-dependent traffic generator which behaved as a dynamic flow-control mechanism. This built-in flow-control mechanism prevented the injection of heavy input traffic and allowed only partial conclusions.

-- Exporting Gateway/Ft. Bragg Configuration Evaluation: This experiment was partially completed due to testbed network component failures and the network unavailability. The collected results were reported [15]. The feedback-dependent traffic generator remained to be severely limiting experimentation at the critical traffic rates.

-- Oscilloscope Measurement: Hands-on measurement using oscilloscope was conducted to access system parameters such as PRU processing times for various tasks, minimum hop delay, acknowledgement time, etc. [16].

4. PRNET Performance Investigations

The broadcasting nature of packet radio technology is the most significant departure of PRNET from classic packet switching technology. Prior to this contract period, we had devoted extensive efforts to the exploration of various channel access schemes for achieving efficient channel utilization [17,18]. These initial studies were based on a one-hop network model. This model consists of a population of PRUs in line-of-sight of each other. Each PRU is assumed to have one packet buffer. A number of channel access schemes, including random access schemes, e.g., variations of slotted ALOHA, and channel sensing multiple access (CSMA) schemes, as well as centrally controled assignment schemes, e.g., polling and reservation schemes, have been devised, analyzed, and evaluated via analytical models. Based on these initial studies, a number of extensions have been made to include the consideration of multiple PRU buffers [19], acknowledgement traffic [20], or the effect of hidden traffic [21]. Both analytical and simulation techniques had been applied in these studies. Measurement experiments were also planned to verify some of these results.

During this contract period we have also investigated packet transport systems. The central issues studied concern multihop packet transport. In the investigation of multihop networks, the issue of hidden traffic is compounded with the complication of varying network topology. Initially, models for specific configurations were formed and analyzed [22,23,24,25].

The performance - slotted-ALOHA access scheme has been evaluated for a star-configured (or tree-structured) two-hop network [23,26,28]. The star-configured two-hop network is a centralized network with a station (and its PRU) at the root of the tree, and N isolated repeaters (stand-alone PRUs not in line-of-sight of each other) surrounding the station. To each repeater is connected a terminal population. Relating to the interests in hierarchical routing at the time, we considered only traffic destined to the station from the terminals. Finite buffer space and FIFO transmission order are assumed at the repeaters. When there is a packet in its buffer for transmission, the repeater transmits the packet at the head of its transmit queue with probability p. When the packet is successfully transported, i.e., the transmission is free of interference and storage is available at the receiving PRU, the packet is deleted from the transmit queue. It otherwise incurs a retransmission after a geometrically distributed delay with mean 1/p. Also assumed is that there is neither acknowledgement traffic nor time spent waiting for acknowledgement. That is, the acknowledgement is assumed free and instantaneous after the packet transmission. It was observed in this study that when PRU processing is assumed very efficient, the system appears to be channel-bound rather than storage-bound. (It should be noted that this conclusion is drawn with the assumption that acknowledgment is free and instantaneous.) Further investigation, under the same assumption, indicates that no significant improvement should be expected by increasing PRU buffer size. On the other hand, the performance of slotted-ALOHA for such configuration can be improved by employing a dynamically controlled transmission protocol maximizing instantaneous throughput with respect to transmission probability p. Such a maximizing process requires each repeater to have an exact knowledge of the instantaneous state of the entired network. Since such knowledge cannot be available to each repeater in practical situations, the results obtained should be used as a theoretical bound on the performance of such a system.

Similar studies were conducted for evaluating and comparing the performance of slotted-ALOHA [26,28] and CSMA access schemes

[27,29] for fully-connected two-hop networks. A fully-connected network differs from a star-configured network in that the repeaters are not isolated from each other. Instead, all repeaters are in line-of-sight of each other. In addition to the variation of channel access protocol, the impact of immediate first transmission (IFT) protocol is also considered. The design of IFT protocol merely eliminates the initial delay before the first transmission. This study concludes with the observation that for fully-connected two-hop networks, non-persistant CSMA with IFT can achieve much better performance than other schemes studied in terms of throughput-delay tradeoff and network capacity.

In an attempt to generalize the above studies, a Markovian model is formulated to include hop acknowledgement traffic [30]. Taking into consideration of a change from hierarchical to point-to-point routing scheme in PRNET development, we assumed a different two-hop star-configuration. This network configuration consists of a repeater as a hub relaying traffic among N surrounding terminal PRUs. The terminal PRUs are not in line-of-sight of each other. This study may be viewed as the first step in an attempt to understand the general case for a repeater, which is usually surrounded by other PRUs. Each terminal PRU here can be viewed as modelling one of a number of isolated groups of PRUs in the neighborhood of a repeater. To simplify the model, the following assumptions are also made:

-- Slotted-ALOHA channel access mode is assumed. When PRUs are hidden from each other, CSMA should behave qualitatively similar to slotted-ALOHA access scheme.

-- Symmetrical traffic pattern is assumed among all pairs of terminal PRUs. The traffic volumn is assumed at a level such that there is always a packet ready for transmission at each terminal. This assumption quarantees a heavy traffic condition.

-- Single buffer is assumed at each PRU. This assumption permits simplicity of analytical modelling. Attempts will be made to relaxed this assumption in subsequent studies. Implication of this assumption is also accessed.

The major conclusion which can be drawn from this study is that when acknowledgement is neither free nor instantaneous, single buffer in each PRU becomes a restrictive network resource. With the 'single-buffer' assumption, network capacity of such a model is determined at 6% of the network bandwidth. This capacity can be improved to 8% by giving priority to acknowledgement traffic. An network capacity of 10.6% results from splitting the channel into two halves: one for data traffic and the other for acknowledgement traffic.

Using the same model we have investigated other network configurations, namely, two- or three-hop linear configurations [30]. Results obtained are similar to, and therefore confirm, those for star-configured networks.

To further generalize the model of a repeater, we have followed a building-block approach [31]. In this approach, we isolated a PRU as the building-block of a PRNET. The PRU can then be characterized with its environment (the remaining network) which is defined by a set of varying parameters. Thus, many issues involving hop transport mechanism and PRU design may be studied without the constraints of a specific network configuration. With the increasing model complexity, analytical approach gives way to simulation. The system-level simulation package has been constructed following this approach [9]. This package includes a building-block simulator which simulates a PRU and the remaining network as its environment. To investigate network (end-to-end) behavior, other simulators included in this package simulate the PRNET by integrating multiple incidents of the building-block. These simulators vary from each other in their forms of integrating the building-blocks.

Using the building-block simulator, we have evaluated the advantage of cyclic transmission scheduling algorithm (CAP4.9) against the FIFO algorithm in previous CAP protocol specifications [12]. The impact of network configuration is embedded in the specification of success transmission probabilities and the intensity of unintended traffic. Due to the cyclic nature of CAP4.9 transmission algorithm, for example, we found the throughput-delay relation varies significantly over the number of neighboring PRUs. We also observed that variation in success transmission probability, which reflects network configuration, traffic intensity and distribution, does not significantly impact throughput-delay performance of the cyclic scheduling scheme. In this study, we also noticed the benefit of reducing initial transmission delay (the delay prior to the transmission of a newly arrived packet).

Using simulation, we have also supported a number of network design and deployment activities. They include an estimate of Ft. Bragg PRNET performance [11], an evaluation of design alternatives in routing strategy and power level adjustment [10], etc.

We conclude this contract period by a review of packet hop transport system [31]. The basic observation we have made in this review is that the conservation of available resources: broadcasting channel bandwidth, PRU processing capability, buffer space, etc., can be effectively achieved by conservation: reducing the rate of redundant transmissions. A retransmission may take place due to either collision of transmissions or non-collision reasons, such as expiration of waiting for an acknowledgement. We recommend a modification to the current transmission scheduling scheme to reduce retransmission rate due to non-collision reasons. It thus reduces the consumption of network resources, and in turn improves performance of the network transport mechanism. A modification to the current design for achieving this objective may be to first transmit the waiting packet with the least number of previous transmissions. A simpler scheme is to assign priority to the transmission of a newly arrived packet at a PRU.

III. SUGGESTED AREAS FOR FUTURE EFFORTS

In support of the on-going exploration of packet radio technology, the PRNET performance analysis and characterization effort has made important contributions. Its important role is anticipated to continue in the future. The effectiveness of such effort can be enhenced through close cooperation with the implementors.

The PRNET investigation effort of this contract has progressed from a concentrated effort in a series of channel access studies to the study of hop transport mechanism for multihop networks. The model for our studies has also been generalized from that for specific network configurations to one which is relatively independent from the constraints imposed by network configuration. With increasing complexity of the model, the investigation becomes necessarily more dependent on measurement and simulation. Mathematical analysis should continue to be used for establishing principles and for guiding the investigation. For the investigation efforts to effectively support the continuing evolution of packet radio technology it is crucial, therefore, to consolidate the presently existing measurement and simulation facilities.

The consolidation effort of currently existing measurement facilities may include the following:

-- reviewing and testing the available facilities;

-- implementing the tracing facility (the pick-up packets), and feedback independent traffic generators;

-- relocating measurement process to be independent from the station.

For quantitative evaluation of PRNET using the simulation packages, they need to be further verified and calibrated via experiments.

For PRNET investigations, effort is needed for continuing the study of hop transport system. Specific subjects need be addressed may include:

-- hop acknowledgement protocol: an understanding of the tradeoffs between active and echo acknowledgement schemes.

-- PRU transmission scheduling algorithm evaluation: quantitative evaluation of the alternatives, e.g., CAP5 cyclic transmission, 'least previous transmissions first', and 'newly arrive packet first' algorithms.

-- a quantitative understanding of the effect of various system parameter values, such as those for transmission delays, processing times for various packet types, buffer size, packet length, etc.

A continuing study in this direction is to investigate the end-to-end packet transport system, such as end-to-end acknowledgement and retransmission, flow control, routing.

In parallel to the investigation of packet transport systems, another important subject for investigation is congestion control in PRNET. Since the implementation of the currently used alternate routing scheme, some difficulties and efficiency issues have been encountered. A number of alternative schemes have also been suggested. In support of the design efforts in search of a congestion control mechanism, it appears to be in need of a systematic investigation of the integrity of alternative congestion control schemes and their respective efficiency.

Further investigations in the areas mentioned above can serve as a foundation in understanding and quiding the planning and development of PRNET internetworking capability, multistation and stationless operations.

BIBLIOGRAPHY

Each of the items listed below marked with an '*' is reproduced and attached in the Appendix.

1.* Tobagi, Fouad A., Mario Gerla, Richard W. Peebles, Eric G. Manning, Proc. of IEEE, vol.66, no.11, pp.1423-1447, Nov. 1978.

2.* Lieberson, Stanley E., "Pickup Packet Format, Handling and Generation," Packet Radio Temporary Note #237, rev. 1, Comput. Sci. Dept., Univ. California, Los Angeles, Sept. 1978.

3.* Lieberson, Stanley E., and Fouad A. Tobagi, Network Message (to Tim Quilici of Rockwell International), PRNET Measurement Group, Univ. California, Los Angeles, January 18, 1978.

4.* Lieberson, Stanley E., "Proposed Procedures for TIU Generation of Poisson Traffic Streams," Internal Memorandum (attached to a network message to Keith Klemba of SRI International), PRNET Measurement Group, Univ. California, Los Angeles, Oct. 1978.

5.* Lieberson, Stanley E., "Packet Radio Files Archived at ISIE <STANL> as of 11-Dec-79," Internal Memorandum, PRNET Measurement Group, Univ. California, Los Angeles, Dec. 1979.

6.* Elsanadidi, Medy, Stanley E. Lieberson, and Fouad A. Tobagi, "A Brief Description of the UCLA Packet Radio Network Simulation," Packet Radio Temporary Note #244, Comput. Sci. Dept., Univ. California, Los Angeles, Feb. 1978.

7.* Elsanadidi, Medy, "PRUNET_SIM: A Simlation of the Radio Subnet of PRNET," Internal Memorandum, PRNET Measurement Group, Univ. California, Los Angeles, Jan. 1980.

8.* Su, Zaw-Sing, "The Building-Block Methodology Used for Studying PRNET," Packet Radio Temporary Note #268, Comput. Sci. Dept., Univ. California, Los Angeles, March 1979.

9.* Su, Zaw-Sing, "Documentation for PRNET System-Level Simulation," Internal Memoramdum, PRNET Measurement Group, Univ. California, Los Angeles, Jan. 1980.

10.* Su, Zaw-Sing, "PTP Route Assignment & PRU Power Control," Network Message (to Jil Westcott of BBN), PRNET Measurement Group, Univ. California, Los Angeles, Sept. 1979.

11.* Su, Zaw-Sing, "Simulation Study of Ft. Bragg Configuration," Network Message (to Michael Frankel of SRI International), PRNET Measurement Group, Univ. California, Los Angeles, Sept. 1979.

12.* Su, Zaw-Sing, "Some Results on a Simulation Study of CAP4.9 Transmission Scheme," Packet Radio Temporary Note #273, Comput. Sci. Dept., Univ. California, Los Angeles, July 1979.

13.* Lieberson, Stanley E., and Fouad A. Tobagi, "Current PR Measurement Plan," Packet Radio Temporary Note #253, Comput. Sci. Dept., Univ. California, Los Angeles, March 1978.

14.* Su, Zaw-Sing, "Current Measurement Results for Hop-by-Hop Acknowledgement Scheme Experiment," Internal Memorandum, PRNET Measurement Group, Univ. California, Los Angeles, Oct. 1978.

15.* Lieberson, Stanley, E., and Zaw-Sing Su, "Measurement Results from the Exporting Gateway Experiment," Packet Radio Temporary Note #275, Comput. Sci. Dept., Univ. California, Los Angeles, Sept. 1979.

16. Kunzelman, Ronald C., et al, "Progress Report on Packet Radio Experimental Network," Quarterly Technical Report No. 14, SRI International, Menlo Park, California, pp.10-14 (Nov. 1978).

17. Kleinrock, Leonard and Fouad A. Tobagi, "Random Access Techniques for Data Transmission over Packet-Switched Radio Channels," Proc. of Nat'l Comput. Conf., 1975.

18. Kleinrock, Leonard and Michel Scholl, "Packet Switching in Radio Channels - New Conflict-Free Multiple Access Schemes," Packet Radio Temporary Note #225, June 1977.

19. Tobagi, Fouad A. and Leonard Kleinrock, "On the Analysis and Simulation of Buffered Packet Radio Systems," Proc. 8th HICSS, Honolulu, HI, pp.42-45, Jan. 1976.

20. Tobagi, Fouad A. and Leonard Kleinrock, "The Effect of Acknowledgement Traffic on the Capacity of Packet-Switched Radio Channels," IEEE Trans. Commun., vol.COM-26, no.6, pp.815-826, June 1978.

21. Tobagi, Fouad A. and Leonard 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, no.12, Dec. 1975.

22.* Tobagi, Fouad A., "On the Performance Analysis of Multihop

Packet Radio Systems: Part I - the Design Problem," Packet Radio Temporary Note #246, Comput. Sci. Dept., Univ. California, Los Angeles, Jan. 1978.

23.* Tobagi, Fouad A., "On the Performance Analysis of Multihop Packet Radio Systems: Part II - Star Configuration Employing Slotted ALOHA," Packet Radio Temporary Note #247, Comput. Sci. Dept., Univ. California, Los Angeles, Jan. 1978.

24.* Tobagi, Fouad A., "On the Performance Analysis of Multihop Packet Radio Systems: Part III - Fully Connected Configurations Employing Slotted ALOHA," Packet Radio Temporary Note #248, Comput. Sci. Dept., Univ. California, Los Angeles, Jan. 1978.

25.* Tobagi, Fouad A., "On the Performance Analysis of Multihop Packet Radio Systems: Part IV - Fully Connected Configurations Employing CSMA," Packet Radio Temporary Note #249, Comput. Sci. Dept., Univ. California, Los Angeles, Jan. 1978.

26.* Tobagi, Fouad A., "Performance Analysis of Packet Radio Communication Systems," Proc. IEEE Nat. Telecom. Conf., Los Angeles, Calif., pp. 12:6-1 to 12:6-7, Dec. 1977.

27.* Tobagi, Fouad A., "Analysis of Slotted ALOHA in a Centralized two-Hop Packet Radio Network," Proc. Comput. Conf. (COMPCON) Fall 1978.

28.* Tobagi, Fouad A., "Analysis of a Two-Hop Packet Radio Network: Part I - Slotted ALOHA," IEEE Trans. Commun., vol. COM-28, no. 2, pp. 196-207, Feb. 1980.

29.* Tobagi, Fouad A., "Analysis of a Two-Hop Packet Radio Network: Part II - Carrier Sense Multiple Access," IEEE Trans. Commun., vol. COM-28, no. 2, pp. 208-216, Feb. 1980.

30. Elsanadidi, Medy, Comput. Sci. Dept., Univ. California, Los Angeles (in preparation).

31. Su, Zaw-Sing, "A Discussion of PRNET Hop Transport System," Comput. Sci. Dept., Univ. California, Los Angeles (in preparation).

APPENDIX

MODELING AND MEASUREMENT TECHNIQUES IN PACKET COMMUNICATION NETWORKS

Fouad A. Tobagi, Mario Gerla, Richard W. Peebles, and Eric G. Manning

Reprinted from Proceedings of the IEEE, Vol. 66, No. 11, November 1978
0018-9219/78/1100-1423$00.75 © 1978 IEEE

Modeling and Measurement Techniques in Packet Communication Networks

FOUAD A. TOBAGI, MEMBER, IEEE, MARIO GERLA, MEMBER, IEEE, RICHARD W. PEEBLES, AND ERIC G. MANNING, MEMBER, IEEE

Invited Paper

Abstract—Considerable advances in the modeling and measurements of packet-switched networks have been made since this concept emerged in the late sixties. In this paper, we first review the modeling techniques that are most frequently used to study these packet transport networks; for each technique we provide a brief introduction, a discussion of its capabilities and limitations, and one or more representative applications. Next we review the basic measurement tools, their capabilities, their limitations, and their applicability to and implementation in different networks, namely land based wire networks, satellite networks, and ground packet radio networks; we also show the importance of well-designed experiments in satisfying the many measurement goals. Finally we discuss briefly some open problems for future research.

I. Introduction

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 either the channel bandwidth or the time frame to each of several subscribers, and are very successful for voice traffic. But the advent of computers has led to an explosive growth in data traffic and the old multiplexing techniques are not nearly so successful. We measure success by the degree of utilization of the transmission channels and switches, which is reasonable since the lower the utilization the more of these expensive resources we need to support a given level of subscriber demand. Data traffic is less effectively supported by a fixed subchannel allocation because it is more "bursty" in nature than voice traffic. This simply means that if one were to watch data traffic over a long period of time then there would be no activity at all for a while, then a flurry of transmission, no activity for another long while, and so on. That is, there is inherently large peak to average transmission rate. Burstiness is a statistical property, both the time between messages and the message length are usually random variables. Fixed subchannel allocation schemes must assign enough capacity to each subscriber to meet his peak transmission rates with the consequence that the resulting channel utilization is low.

Manuscript received April 6, 1978; revised July 3, 1978. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-77-C-0272.

F. A. Tobagi is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

M. Gerla is with the Computer Science Department, University of California, Los Angeles, CA 90024.

R. W. Peebles is with the Research and Development Group, Digital Equipment Corporation, Maynard, MA 01754.

E. G. Manning is with the Department of Computer Science, University of Waterloo, Waterloo, Ont., Canada.

Packet-transmission networks have been developed over the past ten years in an attempt to solve this problem [16], [18], [69], [70], [73], [76], [79]. The basic idea is to allocate some or all of the system capacity (along some path between subscribers) to one customer at a time; but only for a very short period of time. Customers are required to divide their messages into small units (packets) to be transmitted one at a time. Each packet is accompanied by the identity of its intended recipient. In packet-switched networks each packet is passed from one packet switch to another until it arrives at one connected to that recipient, whereupon it is delivered. Packets arriving at a switch may need to be held temporarily until the transmission line that they need is free. The resulting queues require that packets be stored in the switches and it is not unusual that all packet buffers are occupied in a given switch. Thus both the switch capacity (processing and storage) and transmission capacity between switches is statistically multiplexed by subscribers. The designers of such packet networks are faced with the problem of choosing line capacities and topologies that will result in relatively high utilization without excessive congestion.

Another type of packet-transmission network is the "multiaccess/broadcast" network typified by the ALOHA network [2], ring network [70], and the ETHERNET [73]. Here, a single transmission medium is shared by all subscribers. Again, each subscriber is given the whole resource but only for the time required to transmit a single packet. In these networks each subscriber is interfaced through a smart port that listens to all transmissions and absorbs any packets addressed to itself. These multiaccess/broadcast networks also statistically multiplex the communications channel. Queueing does not occur within the system but at the ports to it. Packet-switching networks on the contrary exhibit queues both within the system and at the ports.

All of the packet transmission schemes are designed to share a resource that is inadequate to meet the simultaneous peak demand of all subscribers. It is a reasonable assumption that such simultaneous demands are exceedingly rare events and that average traffic can be adequately supported. In the design of these systems we require that we achieve a desired balance between high average utilization and acceptable levels of congestion under peak loads. The difficulty of such a task leads us to the need for applying modeling techniques to assist us in this design.

We should observe that packet switching networks offer another potential advantage in addition to improved resource utilization. They offer greater reliability. In fact, that was the original motivation for suggesting the idea, [4]. One of the original design goals for the ARPA network was to insure network connectivity between any pair of nodes with a downtime of less than 30 s/yr [79].

A. Modeling

A broad spectrum of mathematical tools has been applied to the design of packet transport networks. Various aspects of the theory of stochastic processes have been used to understand their behavior as average system utilization increases. Typically we are interested in developing estimates of transit delay, packet loss probability, line and buffer utilizations, network throughput and so forth. Queueing models and models of networks of queues have been used to predict the behavior of packet-switching networks with a high degree of success [40], [47], [49], [65], [94]. A recent survey paper by Kobayashi and Konheim [60] examines the application of queueing theory to communications systems in general and serves as a useful companion paper to this one. Comprehensive introductions to queueing theory can be found in many texts including [7], [13], [20], [53], [57], [78]. The theory of networks of queues was heavily influenced by the fundamental papers of Jackson [41], and Gordon and Newell [34]. Recently significant extensions of that work have been made and are reported in [5]. Markovian models and Markov decision processes have been applied to the study of multiaccess networks and flow control procedures [48], [54], [56], [61], [64], [88].

A separate discipline of mathematics has been applied to the design of packet switching networks, the theory of network optimization. Early work by Frank *et al.* [24], was used in the design of the ARPA network. The complexity of the problem has led to many enhancements of their model. Gerla [28] has applied multicommodity flow optimization to the problem of determining optimal routing algorithms. The general problem of determining the optimal structure of a packet-switching network (topology, link capacity, and routing algorithms) is NP complete (i.e., no polynomial time-bounded algorithm can be found to solve it [3]) and for networks of size above ten nodes, say, we are obliged to turn to heuristic solution techniques. The general idea of these schemes is to "guess" a good topology, apply optimization tools to determine the link capacities for that topology which minimizes the cost subject to delay constraints (or which minimizes the delay subject to cost constraints). The algorithm then makes small perturbations to the first topology and recomputes the optimal link assignments, comparing the resulting cost to previously computed values. These methods usually will converge to a "local minimum" but there is no guarantee that the best solution will be found. The heuristics are often proprietary in nature since a good one is highly marketable; some published work has been presented by Gerla [30] and Lavia [66].

There are numerous occasions where analytical techniques may also have to be abandoned. This happens because the theory is not currently able to deal with some of the "real" properties of packet networks such as state dependent transition probabilities and correlations between interarrival times and service time requirements. We move then to simulation. The application of simulation techniques to packet transport networks has frequently led to useful insights, [39], [49], [62], [77]. Most of the major systems that have been implemented have also been studied through simulation prior to construction. Great care must be taken, however, in the use of this tool since it is often the case that the results of an experiment are based on correlated samples and the underlying assumptions of statistics are violated. A great deal of work has been done lately to resolve this problem. For example, Crane and Iglehart [14], [15] have introduced the notion of regeneration points for defining intervals in which independent samples can be taken. Unfortunately many experiments have taken no account of these problems and have reported simulation results with experimental data points supported just by single samples, without mentioning the applicable confidence intervals. This was mainly motivated by simple cost considerations; there are so many sample points to consider that it is impossible to run the simulation a large number of times for each point! However, computing costs are dramatically decreasing so that the trend is for more accurate and more reliable simulation models to support future network studies.

B. Measurements

We will not know the true operating characteristics of packet transport networks for some years. All of the modeling work is based upon assumptions about the traffic characteristics and the subnet behavior. It is inherent both in analytic and simulation models that fairly gross assumptions are made. If this is not done, the models become intractable. We expect to find an iterative procedure where initial models guide first implementation; measurements of real characteristics then are fed back into refined models and so forth. This has already occurred, and is important for improving the value of simulations.

In the remaining sections of this paper, we will examine both modeling and measurement in more detail. Section II focuses on modeling. We first consider the application of queueing theory and Markov models, then turn to the theory of optimization and to simulation techniques. In Section III we address the measurement problem: we define the measurement functions and the performance measures; we describe the tools necessary to perform the measurement tasks; and we review some of the significant results obtained in some recent experiments. Finally, in Section IV, we describe some of the important open problems.

II. Modeling Techniques

This section reviews the modeling techniques that are most frequently used to study packet transport networks. For each technique we provide a brief introduction, a discussion of its capabilities and limitations, and one or more representative applications.

We start with *queueing theory*. In Section II-A, we consider the (simple) single queue server model; we state two widely applicable results, namely, Little's result and the Pollaczek-Khinchin formula for the mean number of customers in an $M/G/1$ queueing system, and discuss their limitations. Two applications are treated: i) the analysis of an allocation strategy of buffers in a packet switch known as the complete partitioning strategy, and ii) Kleinrock's model for message delay in a packet-switched network. Next, we turn to a discussion of "network of queues" models and their application to packet network modeling in Section II-B. We state the condition under which a network of queues has a tractable solution and treat two applications: i) a derivation of the distribution of end-to-end delay for source–destination pairs in a packet-switched network, and ii) the analysis of another allocation strategy of buffers in a packet switch known as the complete sharing strategy.

A second set of powerful analytical tools is provided by the *theory of stochastic processes.* This includes renewal theory, Markov chain theory, semi-Markov and regenerative processes, and Markov decision theory. Numerous applications exist to illustrate the usefulness of these techniques, but due to the great interest we see today in multiaccess/broadcast networks, in this presentation we shall limit ourselves to examples drawn from radio communications systems. We first start, in Section II-C, by the consideration of renewal theory. We give a brief account on the (relatively recent) use of radio for data transmission and discuss the related issues, in particular, the so-called random access schemes. We then show how the assumption of an infinite population of users in conjunction with renewal theory arguments have allowed the determination of the (radio-multiaccess) channel capacity and other performance measures under various access schemes. Following that, we briefly discuss the limitations of the infinite-population/renewal-theory model for these access schemes and emphasize the need for a more accurate performance evaluation. The latter is obtained in Section II-D via Markov and semi-Markov chain models which are used to analyze slotted ALOHA and Carrier Sense Multiple Access respectively [54], [81]. These models greatly improved our understanding of the behavior of random access schemes under "static" conditions. It is important, however, to design systems which can dynamically adapt to time-varying inputs and to changes in the system state. We discuss this issue in Section II-E, in which we give a brief introduction to Markov decision theory and its most relevant results, and then proceed with a discussion of various practical control schemes and their analysis.

Some concepts of network optimization relevant to the design of packet networks are then introduced. Linear, nonlinear, and integer programming techniques are briefly reviewed, and a nonlinear programming technique, namely, the method of Lagrangian multipliers, is illustrated in an optimal capacity assignment problem in Section II-F. In the following section, the routing problem, i.e., the problem of optimally routing packets in the network is formulated and solved using a multicommodity flow approach.

Unfortunately, mathematical programming has its limitations, and heuristic approaches are often required to obtain practical solutions to network optimization problems, as discussed in Section II-H. Similarly, many of the performance models are analytically intractable, and require simulation for their solution, as discussed in Section II-I.

A. Queueing Theory

The single server queue is perhaps the simplest of all the mathematical modeling tools and has been widely applied. The model assumes that "customers" arrive at a service facility, that we know the service time distribution, the interarrival time distribution, and the order in which customers are served. The models tell us the distributions for the number of customers in the system, their waiting time, the server busy period distributions, and so forth. Single server queues are categorized according to the interarrival time and service time distributions and we speak of $M/M/1$ systems, $G/G/1$ systems, and others, where the first leter denotes the interarrival time distribution, the second denotes the service time distribution, and the third element denotes the number of servers in the system. The letters used in this paper have the following meaning, by convention: M–exponential, D–fixed, and G–general. There have been many contributors to the theory of single server queues and readers looking for an introduction should turn to one of the many basic texts [13], [53], [57].

There are some basic formulas that we will state without derivation which are broadly applicable. The first is Little's result [68]. The result is stated as a simple formula:

$$\bar{n} = \lambda T. \tag{1}$$

Here, $\bar{n}$ is the average number of customers in the system (both queue and server). λ is the customer arrival rate, and T is the average time that a customer spends in the system (including service). The formal proof makes no assumptions about the arrival process distribution, the service time distribution, the number of servers, nor the service discipline which can be first-come-first-serve (FCFS), last-come-first-serve (LCFS), round-robin (RR), etc.

A second widely applicable result is the Pollaczek–Khinchin formula for the mean number of customers in an $M/G/1$ queueing system. This is expressed as

$$\bar{n} = \rho + \rho^2 \frac{(1 + C_b^2)}{2(1 - \rho)} \tag{2}$$

where ρ is the traffic intensity (also referred to as the server utilization if $\rho < 1$) and is defined as λ/μ with μ denoting the service rate, and C_b^2 is the squared coefficient of variance for the service time and is defined as σ_b^2/μ^2, with σ_b^2 denoting the variance of the service time. For the $M/D/1$ queueing system this formula holds with $C_b^2 = 0$.

We observe that this formula only gives us the mean value for the number of customers in the system. This is certainly useful, but it is dangerous to design systems based on mean-value estimates only. The variance of the number in the system is also of great interest: in fact one would ideally like to know the exact distribution for the number of customers in the system. This would then allow one to compute other statistics of interest. Specifically, it is often desirable to design to such criteria as: 90 percent of messages will be transmitted within 2 s and the average time will be 0.9 s. The general approach taken to the analysis of $M/G/1$ systems is to develop the Laplace transform for the distribution of the number in the system (or the waiting time). But then it is often very difficult to invert the result and the detailed distribution cannot be obtained. Nevertheless it is easy to derive the moments of the distribution of the number of customers by evaluating the derivatives of the transform at $s = 0$ [53] where s denotes the argument of the transform.

For an $M/M/1$ queue, $C_b^2 = 1$, and the expression for $\bar{n}$ reduces to

$$\bar{n} = \frac{\rho}{1 - \rho}. \tag{3}$$

In this case, the detailed state probabilities can easily be obtained and they are expressed as

$$P(j) = (1 - \rho)\rho^j. \tag{3a}$$

The $M/M/1$ model has been applied to a wide variety of problems for three reasons. First, it is so very simple that it is convenient to work with. Second, it is a very good approximation to many real systems. Third, even when the approximation is not good, it provides upper bounds to systems with $C_b^2 < 1$. Practitioners must beware, however, if the service time distribution of the real system is known to have wide variance.

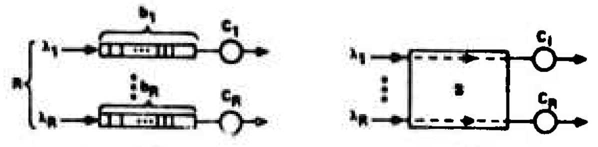


Fig. 1. Buffer management at the switch. (a) Complete partitioning (CP). (b) Complete sharing (CS).

Let us now consider two problems in the design of packet-transport networks that have been attacked with elementary queueing theory: first, the problem of buffer allocation in a packet switch and secondly, the problem of estimating packet transit delay in a packet-switched network.

a) Buffer Allocation: Each packet switch has several outgoing lines; incoming packets will queue for these lines awaiting retransmission. If there is no available storage space an incoming packet will be dropped. How can one allocate the storage of the switch in such a way that packets will not be refused entry when space is available and that no single outgoing line is able to capture all of the switch buffers?

Kamoun [46], [47] has analyzed several strategies, the simplest of which is an $M/M/1$ model with finite customer waiting room. Each of the output lines has a fixed protion of the buffer pool assigned to it, b_i for line i. The rate of external arrivals destined for line i is λ_i. The service rate is μC_i, where C_i is the capacity of line i and $1/\mu$ is the average length of the message, the latter assumed to be exponentially distributed. If the buffer assigned to line i is full, an arriving packet destined for line i is dropped and does not return. Service at a line is FCFS and there are R output lines, as shown in Fig. 1(a). In this model all of the R subsystems are independent and we can write down the equations for each separately The finite waiting room variant of the $M/M/1$ queue also has a simple solution: (see [13] or [53].) Let n_i denote the number in subsystem i. Its distribution is given by

$$\Pr\{n_i = k\} = \begin{cases} \dfrac{1-\rho_i}{(1-\rho_i^{b_i+1})}\rho_i^k, & 0 \leq k \leq b_i \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\rho_i = \lambda_i/\mu C_i$.

The probability that a packet destined for line i is dropped is the probability that there are exactly b_i packets in subpool i at the time it arrives. Let PB_i denote that probability. We have

$$PB_i = \frac{1-\rho_i}{1-\rho_i^{b_i+1}}\rho_i^{b_i}. \tag{5}$$

The independence of the subsystems also means that the global state has a simple product form. Let $\tilde{n} \triangleq (n_1, n_2, \cdots, n_R)$ and let $P(\tilde{n})$ denote the probability of state $\tilde{n}$. We have:

$$P(\tilde{n}) = \left(\prod_{i=1}^{R} \frac{1-\rho_i}{1-\rho_i^{b_i+1}}\right)\left(\prod_{i=1}^{R} \rho_i^{n_i}\right). \tag{6}$$

The average time spent at the switch by type i packets, denoted by $\bar{t}_i$, is obtained from Little's result:

$$\begin{aligned} \lambda_i' &= (1 - PB_i)\lambda_i \\ \bar{t}_i &= \bar{n}_i/\lambda_i' \end{aligned} \tag{7}$$

where

$$\bar{n}_i = \sum_{j=1}^{b_i} j \Pr\{n_i = j\}.$$

This simple example shows how even very modest queueing theory allows for the determination of important quantities and thus can give us useful insight into the design of packet networks. We would like to compare this partitioned storage strategy to other strategies. In particular, we like to compare it to the case where the entire buffer pool is shared by all lines. However, this other alternative is more difficult to analyze and we defer the discussion until Section II-B after we have discussed networks of queues.

b) Transit Delays: Another useful application of $M/M/1$ queues was made by Kleinrock [57] in predicting the delay of messages flowing through a network. The delay is expressed in terms of the line capacity and the traffic between ports on the network. With the aid of this expression it is then possible to adjust the line capacity to meet certain delay constraints. It was precisely this technique which was applied in the design of the ARPANET (see Sections II-F and G). The model is developed as follows.

The network is assumed to contain M links between switches of infinite storage capacity and we seek to determine T, the expected message delay in the net. This is just the average delay over all mesages flowing through the net. Let N be the average number of packets in the system as a whole and let n_i be the average number of packets in the link i subsystem. Clearly

$$N = \sum_{i=1}^{M} n_i. \tag{8}$$

Then, if γ is the aggregate packet arrival rate from all sources, λ_i is the arrival rate at link i, and T_i is the expected time spent at link i, Little's result yields:

$$T = \sum_{i=1}^{M} \frac{\lambda_i T_i}{\gamma} \tag{9}$$

It remains to compute the values of λ_i and T_i. The former are obtained by considering the traffic between all source–destination pairs and the routing rules (which can be fixed or random but not adaptive). The T_i's are then obtained by treating each link as an independent $M/M/1$ queueing system. (We defer the discussion concerning the validity of this assumption to the next paragraph.) Given that link i has capacity C_i bits per seconds and the average packet size is $1/\mu$ bits, T_i is expressed as

$$T_i = \frac{1}{\mu C_i - \lambda_i} \tag{10}$$

so the expected packet delay is

$$T = \sum_{i=1}^{M} \frac{\lambda_i}{\gamma}\left[\frac{1}{\mu C_i - \lambda_i}\right]. \tag{11}$$

This simplified model can be enhanced by introducing terms that express propagation delay, processing delay, and multiple customer types. The more elaborate the model the more accurate its predictions. The reader is referred to [57] for a detailed discussion.

Why should we believe that each of the links behaves as a separate $M/M/1$ queue? There are many reasons to suspect that it may not. For example, although traffic entering the first switch of a path may be Poisson it may not be so when it leaves the switch. Furthermore, the interarrival time and service time for packets are supposed to be independent random variables but at the second server on a chain this is not the case since packets preserve their length! It is, in fact, not true that the two variables are independent in packet-switched networks but IF THERE IS SUFFICIENT MIXING AT A NODE (i.e., packets joining the queue arrive from several different input lines) then the switch behaves AS THOUGH they were independent. This is Kleinrock's celebrated "independence assumption." It is reasonable if the network topology and routing algorithms are such that the traffic from many preceding switches is mixed at any successive switch. This is true under remarkably loose constraints: a "fan in" of two or three lines seems to be sufficient for the approximation to be accurate [48a].

We turn now to a discussion of networks of queues and their application to packet network modeling.

B. Networks of Queues

Most computer-communications systems are most naturally represented as networks of queues. We have seen in the last section that a simple model of delay in a packet-switched network can be developed if we are free to treat each switch link as an independent $M/M/1$ queue. This depends upon whether or not it is reasonable to believe that the process of passing through a switch does not alter the basic Poisson nature of the traffic. Pioneering work was done in this area by Burke [9] and by Jackson [41].

Burke showed that the output of an $M/M/1$ queue is Poisson. (Limited details regarding Burke's output theorem can be found in Inose and Saito [38].) Jackson extended this work to include feedback networks of N servers as well. If $\tilde{n} \triangleq (n_1, n_2, \cdots, n_N)$ is the global state variable denoting n_i customers at server i then the equilibrium probability distribution has a simple product form:

$$P(n_1, n_2, \cdots, n_N) = P(n_1)P(n_2)\cdots P(n_N) \tag{12}$$

where $P(n_i)$ is the marginal probability of finding n_i customers at server i, and is given by the simple $M/M/1$ formula. To apply Jackson's result we must know the actual traffic arriving at server i. This is easily computed if we know the external arrival rate a_i and the customer branching probabilities, b_{ij}. This yields the set of equations:

$$\lambda_j = a_j + \sum_{i=1}^{N} \lambda_i b_{ij}, \quad j = 1, \cdots, N. \tag{13}$$

The network will reach an equilibrium state provided that none of the servers is overloaded. The interesting point in this result is that the network of queues behave as though the traffic remained Poisson in that the equilibrium state probabilities factor into the product of the marginal probabilities despite the fact that, in truth, it is not Poisson.

Jackson considered also more elaborate models, but the most general results have recently been derived by Baskett *et al.* [5]. They assume that there are N nodes, L classes of customers (such that each class may have different routing through the network and possibly different service time at a node), and four allowed node types which satisfy the Poisson output property and thus guarantee a product form solution. These are: type 1–FCFS, $M/M/1$; type 2–RR, $M/G/1$; type 3–processor sharing $M/G/\infty$; type 4–LCFS, $M/G/1$. For the general service time distributions, we require that they have a rational Laplace transform; in types 2, 3 and 4 the service time distribution can vary with the customer class. Each customer class travels through the network according to a probabilistic routing (which can be fixed) specified by $b_{ij;mn}$ where the implication is that customers can also change class $(m \to n)$ during a transition from one node to another $(i \to j)$. The network can be open for some classes of customers and closed for others. (A closed network has a fixed number of customers and none leave or enter the network; an open network allows for external arrivals and departures and the number of customers in the network may vary). Again the customer arrival rate at each node is computed by using the external arrival rates and the routing information; in particular for the case where there are no class changes, we have

$$\lambda_j(c) = a_j(c) + \sum_{i=1}^{N} \lambda_i(c) b_{ij} \tag{14}$$

where c denotes the customer class. The solutions for the global state distributions are given in [5]. They are of the form

$$P(n_1, n_2, \cdots, n_N) = C \prod_{j=1}^{N} f_j(n_j) \tag{15}$$

where the f_j depends on the node type. For the type 1 nodes, for example, we have

$$f_j(n_j) = \left(\sum_c \frac{\lambda_j(c)}{\mu_{jn_j}}\right)^{n_j} \tag{16}$$

where μ_{jn_j} is the service rate of node j when there are currently n_j customers and where the summation over customer classes is assumed to be over those that are routed through the node. The constant C is a normalizing constant (i.e., is determined by the condition that $\Sigma_{\tilde{n}} P(\tilde{n}) = 1$). The case of a completely open system is of special interest; it can be shown that

$$P(\tilde{n}) = \prod_{j=1}^{N} P_j(n_j) \tag{17}$$

where

$$P_j(n_j) = \begin{cases} (1-\rho_j)\rho_j^{n_j}, & \text{for nodes of types 1, 2 and 4} \\ \dfrac{\rho_j^{n_j}}{n_j!} e^{-\rho_j}, & \text{for nodes of type 3.} \end{cases}$$

That is, these rather complex systems (node types 1, 2, and 4) behave just like a set of connected but independent $M/M/1$ queues! And type 3 nodes behave like isolated $M/G/\infty$ servers. This gives us a few very powerful tools for analytic modeling.

Before we consider examples of the application of these models it is worthwhile to consider some of the important problems that cannot be handled: dynamic storage allocation at a switch which allows for variable sized blocks (this is a situation where the allowed number of packets at a node depends on their total required storage volume); the flow of multiple customer classes through a FCFS node in which the classes have different service time distributions; state dependent customer routing (thus representing adaptive routing algorithms in packet-switched networks); priorities. As a final comment we must warn potential users to consider carefully whether or not their network will support the independence assumption.

An important application of this theory has recently been developed by Wong [94]. He uses the results of Basket *et al.* as a starting point and develops the distribution of the end-to-end delay for source-destination pairs in a packet switched network. Thus Kleinrock's earlier result for the mean delay in a net (described above) has been extended in an important way. The fact that the detailed distributions are known means that we can compute variance and percentile information. The model allows for both fixed and random routing. The independence assumption is still required. The switches use a FCFS discipline on their communications links; infinite storage is assumed, and all packets are assumed to have the same length distribution, namely exponential with mean $1/\mu$. Channel i has the capacity C_i, so the service rate is μC_i. If c denotes a customer class then $a(c)$ represents its path through the network and $\gamma(c)$, its traffic rate. (For random routing this becomes a set of paths, each with a known probability of use.) We will present the formulas for the fixed routing case only. We let λ_{ic} be the mean arrival rate of class c customers to channel i. For fixed routing:

$$\lambda_{ic} = \begin{cases} \gamma(c), & \text{if } i \text{ is in } a(c) \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

We let ρ_{ic} be the utilization channel i by class c customers, so,

$$\rho_{ic} = \frac{\lambda_{ic}}{\mu C_i} \tag{19}$$

and the total utilization of channel i, $i = 1, \cdots, M$, must satisfy

$$\rho_i = \sum_{c=1}^{R} \rho_{ic} < 1. \tag{20}$$

Let the probability density function of the end-to-end delay for class c customers be denoted by $t_c(x)$, and its Laplace transform by $T_c^*(s)$. Wong shows for this model of a packet-switched network with fixed routing that

$$T_c^*(s) = \prod_{i \in a(c)} \frac{\mu C_i(1-\rho_i)}{s + \mu C_i(1-\rho_i)}. \tag{21}$$

Notice that each term in the product is the Laplace transform of the time spent in each queue calculated as if the queue was independent of the rest of the network (while in reality it is not). This implies that the end-to-end delay can be interpreted as the sum of independent delays along the path! The Laplace transform expression can be inverted using partial fractions (see [53]) to obtain $t_c(x)$. The mean and variance can be obtained by taking derivatives and are given by

$$\bar{T}_c = \sum_{i \in a(c)} \frac{1}{\mu C_i(1-\rho_i)}$$

$$\sigma_c^2 = \sum_{i \in a(c)} \frac{1}{[\mu C_i(1-\rho_i)]^2}. \tag{22}$$

This result allows us to greatly inhance our understanding of packet networks; we note, however, that the case of finite storage capacity in the switches is still not modeled. In fact, it is still an open problem.

An important tool in solving for the system state probabilities is the set of "local balance" equations. We have not as yet written down any of the system state equations because we wished to expose answers, not derivations. But when the time comes to derive similar models to those described here, these state equations will immediately arise. State equations for stable systems come in two types: global balance equations and local balance equations. The former states that, in equilibrium, the total rate of flow into any given state must equal the total rate of flow out of that state. "Flow," here, means probabilistic flow (state transitions over time). This is not surprising: if it were not true then some states would have increasing (or decreasing) probability of occurrence as time passed. The local balance equations are useful in studying networks of queues and assume that in equilibrium the flow into a state due to arrivals at server i can be equated to the flow out of that state due to departures from server i. It is known that if a solution to the local balance equations can be found then it will also satisfy the global balance equations. The local balance equations are generally much easier to solve and are thus the preferred route. What is not known are the necessary conditions for the local balance equations to have solutions. All of the conditions listed in the paper by Baskett *et al.* are sufficient conditions.

We will now look at an example of global and local balance equations: the second part of our switch buffer allocation example [46]. Recall that the problem is to allocate buffer space in a packet switch to a set of R communication links, and that we have considered a partitioned allocation where each link is assigned a fixed subpool of size b_i. This clearly has the disadvantage of blocking packets on a busy link when buffer space is available in the switch but is dedicated to other channels. Another alternative is *complete sharing*; any buffer can be used for any outgoing link; (see Fig. 1(b)). The total available storage will accommodate B packets (here each packet takes a full buffer even if it is not of full length). A state of the switch is described by the R-tuple: $\tilde{n} \triangleq (n_1, n_2, \cdots, n_R)$. Let J be the set of states where $n_i = 0$. In Fig. 2, we show a portion of the state transition diagram. Each node in the figure is a possible state. The edges indicate possible transitions and are weighted by the rate of flow conditioned upon starting in the state at the tail of the arrow. The global balance equations are obtained by drawing a circle around state $(n_1, n_2, \cdots, n_R)$ and equating flows across that boundary; that is, the stochastic flows in and out of that state. Here we get two sets of equations depending on whether or not the total number of customers is less than or equal to B. For states with $\sum_i n_i < B$ we have

$$\sum_{\substack{i=1 \\ i \notin J}}^{R} \lambda_i P(n_1, \cdots, n_i - 1, \cdots, n_R) + \sum_{i=1}^{R} \mu_i P(n_1, \cdots, n_i + 1, \cdots, n_R) = \left(\sum_{i=1}^{R} \lambda_i + \sum_{\substack{i=1 \\ i \notin J}}^{R} \mu_i \right) P(n_1, \cdots, n_i, \cdots, n_R). \tag{23}$$

For all states with $\sum_i n_i = B$ the equations are

$$\sum_{\substack{i=1 \\ i \notin J}}^{R} \lambda_i P(n_1, \cdots, n_i - 1, \cdots, n_R) = \sum_{\substack{i=1 \\ i \notin J}}^{R} \mu_i P(n_1, \cdots, n_i, \cdots, n_R). \tag{24}$$

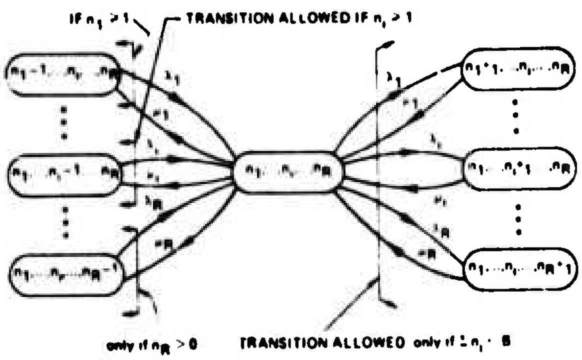


Fig. 2. State transitions for the R-link buffer pool.

The local balance equations for this system equate the flow rates due to arrivals and departures from a given channel. Specifically, we get

$$\lambda_i P(n_1, \cdots, n_i - 1, \cdots, n_R) = \mu_i P(n_1, \cdots, n_i, \cdots, n_R). \tag{25}$$

The traditional method of solving these equations is to guess the answer and try it out. With a little practice the guessing is not too hard. Here the solution is

$$P(\tilde{n}) = \begin{cases} P_0 \rho_1^{n_1} \rho_2^{n_2} \cdots \rho_R^{n_R}, & \text{for all } \tilde{n} \text{ such that } \sum_i n_i \leq B \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

To prove this we need only to substitute this expression back into the local balance equations. The evaluation of P_0 again follows from the normalization requirement (that the $P(n)$ be valid probabilities) so

$$P_0^{-1} = \sum_{\tilde{n}} \rho_1^{n_1} \cdots \rho_R^{n_R} \tag{27}$$

where the summation is taken over the set of all feasible states. Efficient algorithms for the evaluation of such constants can be found in [9a], [74], [93].

The complete sharing scheme has a lower probability of dropping a packet if traffic is reasonably well-balanced, but under highly assymetrical loads it tends to be unfair, i.e., favors heavily utilized channels far too much. The delay on links with low utilization becomes exorbitant. This suggests that some of the buffers should be permanently allocated to each link; but how many? Kamoun goes on to explore several other sharing strategies and concludes that no one scheme is always optimal. It is desirable to select a scheme whose delay and packet loss behavior best suit the operational constraints. When this is not possible, a scheme that dedicates some buffers to each channel and leaves some in a general pool is preferred.

The problem of buffer allocation in a packet switch has also been studied by Irland [40]. He considers a scheme that bounds each channel queue subject to the constraints that all space can be used and no queue can have more than the total space. He then develops a queueing model for the state distributions and uses this to drive a Linear Programming model that seeks the optimum assignment of queue bounds.

We have shown so far in this section that queueing theory is a powerful tool for the study of packet transport network behavior and a very broad class of related problems. We have also identified several limitations in the queueing theory approach, which force us in many cases to turn to simulation techniques. We shall defer the discussion on the use of simulation to Section II-I below. In the following sections we review the (more basic) theory of stochastic processes and discuss its applications.

C. Renewal Theory

Computer communication systems, as pointed out earlier in the paper, are characterized by unpredictable sequences of random demands on the available resources. The theory of stochastic processes (which generally includes renewal theory, Markov chain theory, semi-Markov processes and regenerative processes) thus also provides a large and effective set of analytical tools particularly suited for the modeling of these probabilistic systems. This body of theory is well known and has been established throughout many years; its applications are numerous in very many different areas. In fact, carefully examining the queueing systems for which some solution is obtainable, we realize that virtually in each of them there exists an underlying Markov or semi-Markov process. Queueing theory is a very powerful tool and was shown in the above sections to be extremely effective in the design of computer networks. However, there are situations in which queueing theory does not provide the appropriate model. The latter has to be drawn from the more basic theory of stochastic processes, thus allowing for the determination of the system's steady-state performance. Another problem which also is of great practical importance is the optimization of these probabilistic communication systems. By viewing the models from a probabilistic point of view, dynamic programming and applied probability theory have been combined to give rise to a simple and precise treatment of sequential decision theory, a result of which is the well known Markov decision theory [82]. This is found particularly useful in the design and analysis of efficient procedures for the (optimum) control of communication subsystems.

We intend here to briefly review these tools and illustrate their usefulness by calling on examples from the (relatively recent) packet radio communication systems in both satellite and ground environments. Although as pointed out in the introduction, the latter are not the only examples of applications one can give for these tools, we restrict ourselves to these here for the sake of a unified presentation.

The advantages of using radio communication for data transmission have been extensively discussed in the literature [1], [33], [43], [44], [45], [54], [63], [84]. In essence, satellite transponders in a geostationary orbit above the earth provide long-haul communication capabilities, while broadcast ground radio communications provide us with easy access to central computer installations and computer networks. The topic of interest to this discussion, common to most of these radio systems, is the sharing of a *single* radio channel by users. The difficulty in controlling a multiaccessed channel of this sort, which has to carry its own control information, gave rise to the so-called *random access* techniques. In the event of transmission overlap, these techniques suffer from destructive interference (unless a spread spectrum modulation scheme is used); acknowledgement procedures are devised to recover from errors and overlapping transmissions.

A simple scheme, known as "pure-ALOHA," permits users to transmit any time they desire [1]. Another method, referred to as "slotted-ALOHA," requires each user to start his packets only at the beginning of a slot (whose duration is equal to the transmission time of a packet) [50], [81]. These two ALOHA schemes are suitable for both satellite and ground environments. In ground radio environments, where the channel can further be characterized as a wide-band channel with a small propagation delay between any source-destination pair as compared to the packet transmission time,[1] a third scheme has proven to be efficient: it is the carrier sense multiple access (CSMA) mode. In this scheme one attempts to avoid collisions by listening to the carrier due to another user's transmission; a terminal never transmits when it senses the channel is busy [54], [84]. In the (simple) nonpersistent CSMA protocol, a terminal with a packet ready for transmission transmits the packet if the channel is sensed idle, or reschedules the (re)transmission of the packet to some later time if the channel is sensed busy. A slotted version of this scheme is also considered in which the time axis is slotted and the (mini-) slot size is τ seconds (the propagation among pairs of devices is assumed to be the same [54]). All terminals are synchronized and start transmission only at the beginning of a slot, according to the protocol described above. In addition to these and other CSMA protocols, a number of clever schemes have also appeared in the literature offering improved performance under various specific conditions such as heavy traffic, large users, etc. For more details the reader is referred to [8], [42], [43], [80].

The remainder of this subsection will be devoted to renewal theory and its application to the analysis of packet switching in radio channels. Markov chain models and Markov decision models will be treated in the following two subsections.

The Infinite Population Model and Renewal Theory: The focus here is to show how the assumption of an infinite population in conjunction with renewal theory arguments have allowed the determination of the channel capacity under the various schemes. In this presentation, we shall make sure not to overlook the importance of simulation techniques and simulation results whenever they have proven useful, be it for the validation of a model, or the determination of some performance measure hard to obtain analytically, or just the gain of insight into the behavior of the system under specific conditions.

The model assumes that the traffic source consists of an infinite number of users who collectively form an independent Poisson source with an aggregate mean packet generation rate of S packets per packet transmission time T. (We assume here that each packet is of constant length requiring T seconds for transmission.) This is an approximation of a large but finite population in which each user generates packets infrequently and each packet can be successfully transmitted in a time interval much less than the average time between successive packets generated by a given user. Each user in the infinite population is assumed to have at most one packet requiring transmission at any time (including any previously blocked packet). Under equilibrium conditions, S is also the channel throughput. Because of packet interference, the achievable throughput will always be less than 1. The traffic offered to the channel from our collection of users consists not only of new packets but also of previously collided packets: this increases the mean offered traffic rate which we denote by G (packets per transmission time T) where $G \geq S$. To avoid repeated conflicts, each user delays the transmission of a previously collided packet by some random time whose mean is $\overline{X}$ (chosen, for example, uniformly between 0 and $X_{max} = 2\overline{X}$). Two additional assumptions are introduced here.

Assumption 1: The average retransmission delay X is large compared to T.

Assumption 2: The interarrival times of the point process defined by the start times of all the packets plus retransmissions (and reschedulings) are independent and exponentially distributed.[2]

If we use $T = 1$ (for normalization), then we express τ as $a = \tau/T$ and $\overline{X}$ as $\delta = \overline{X}/T$. The throughput analysis consists of solving for S in terms of G and other system parameters (namely a). The channel capacity is then found by maximizing S with respect of G.

Renewal theory, and the theory of regenerative processes in general, relate to systems in which there exists an underlying process which probabilistically restarts itself. Perhaps the result in renewal theory which proves most useful here is the one corresponding to alternating renewal processes.[3] An alternating renewal process is one which describes a system which can be in one of two states, say on or off. Starting in the on state, the system alternates between these two states. The periods of time it spends in each are random variables which follow a common distribution for each of the two states. Let EX be the average time the system remains in the on state, and EY be the average time it remains in the off state. Let $P(t)$ be the probability that the system is on at time t; we have the following simple result: $P(t) = EX/(EX + EY)$. This result is easily extendable to any number of states that the system may be cycling through. The key element in such an analysis is to identify points in time at which the system regenerates itself: the interval of time separating two consecutive regenerative points is called a cycle; the ratio of the average time that the system spends in a given state to the average cycle time is precisely the fraction of time that the system spends in that state.

Consider for example the slotted ALOHA scheme. By the infinite population assumption and the Poisson assumption on the channel traffic, each slot boundary is a regenerative point. It is clear that e^{-G} is the probability that the slot is empty and this is also the fraction of time that the channel is idle. The probability that a slot is carrying a successful packet is clearly Ge^{-G}, the probability that a single packet is transmitted in that slot; by the same argument this is also the fraction of time that the channel is carrying successful information and thus constitutes the average throughput S.

A slightly different approach using the same type of argument can also lead to the result derived above. Considering the (slotted) time axis, it is clear that we observe a number of consecutive nonempty slots (which we refer to as a busy period (B)), followed by a number of consecutive empty slots (which we refer to as an idle period (I)). A busy period and the following idle period constitute a cycle. The idle period is geometrically distributed with mean $\overline{I} = 1/(1 - e^{-G})$. The busy period is also geometrically distributed with mean $\overline{B} =$

[1] Ratio on the order of 0.01 [54].

[2] It is clear that Assumption 2 is violated and that it has been introduced for analytic simplicity. However some simulation results are discussed below which show that performance results based on this assumption are excellent approximations. Moreover, in the context of slotted-ALOHA it was analytically shown that, in the limit as $X \to \infty$, Assumption 2 is satisfied [63].

[3] It will be clear from the discussion below that this constitutes a special case of the more general result obtained with regenerative processes.

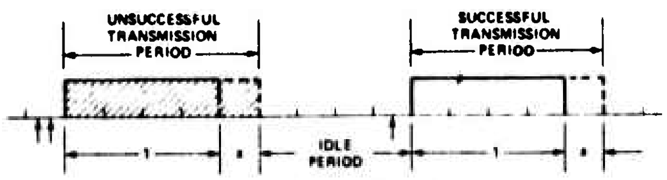


Fig. 3. Slotted nonpersistent CSMA: Transmission and idle periods.

$1/e^{-G}$. Thus the fraction of time that the channel is idle is equal to

$$\frac{\bar{I}}{\bar{I}+\bar{B}} = \frac{(1-e^{-G})^{-1}}{(1-e^{-G})^{-1}+e^{G}} = e^{-G} \tag{28}$$

Let $\bar{U}$ denote the average time during a cycle that the channel is carrying successful packets. Given that a slot is nonempty, the probability that it is successful is simply $Ge^{-G}/(1-e^{-G})$. $\bar{U}$ is, therefore, given by

$$\bar{U} = \bar{B}Ge^{-G}/(1-e^{-G}) = G/(1-e^{-G}). \tag{29}$$

Taking the ratio of $\bar{U}$ to $\bar{B}+\bar{I}$, we find again that the channel throughput is precisely $S = Ge^{-G}$.

Let us consider now, as another example, the slotted nonpersistent CSMA protocol, analyzed by Kleinrock and Tobagi [54], [84]. Considering the time axis, we define a transmission period (TP) to be the period of time required for transmission and reception of a packet and its (possible) overlapping packets. Thus we observe on the time axis transmission periods separated by idle periods, as depicted in Fig. 3. The length of a TP is $1+a$. A TP is successful if only one packet is transmitted; the probability of this occurring is

$$P_s = \frac{aGe^{-aG}}{1-e^{-aG}}. \tag{30}$$

Due to the memoryless property of the Poisson process, the average idle period (normalized to T) is simply

$$\bar{I} = \frac{ae^{-aG}}{1-e^{-aG}}. \tag{31}$$

Using the same renewal theory argument as above, we find that the average channel utilization is given by

$$S = \frac{P_s}{\bar{I}+1+a} \tag{32}$$

Substituting for P_s and $\bar{I}$ the expressions found above, we get

$$S = \frac{aGe^{-aG}}{1+a-e^{-aG}}. \tag{33}$$

This relatively simple argument has been applied in numerous occasions to analyze the throughput and channel capacity of many other (more complex) protocols as well as the effect on system capacity of the overhead created by various acknowledgment schemes. For this the reader is referred to the work by Tobagi and Kleinrock [54], [84], [85], [90]. We illustrate such results here by plotting in Fig. 4 S versus G for various random access schemes. An important question remains: what about packet delay analysis? Kleinrock and Lam [50] formulated an analytic model for a slotted-ALOHA channel using a uniform retransmission randomization scheme, and assuming that the channel is in equilibrium. Such a task proved more difficult for CSMA, and simulation techniques appeared then to be the only recourse. A brief discussion of some simulation results and of the validity of the equilibrium assumption follows.

Discussion and Delay Analysis [54], [84]: The above analysis is based on renewal theory and probabilistic arguments requiring independence of the random variables provided by Assumption 2. Steady state conditions are also assumed to exist. However, from the (S, G) relationships derived above (see Fig. 4) and the throughput-delay performance derived in [50] for slotted-ALOHA, one can see that steady state may not exist because of the inherent instability of these random-access techniques. This instability is simply explained by the fact that when statistical fluctuations in G increase the level of mutual interference among transmissions, then the positive feedback causes the throughput to decrease to 0. Extensive simulation runs performed on a slotted-ALOHA channel with an infinite population [63] have indeed shown that the assumption of channel equilibrium is not strictly speaking valid; in fact, after some finite time period of quasi-stationarity conditions, the channel will drift into saturation with probability one.[4]

In the simulation models considered, [54], [63] Assumptions 1 and 2 concerning the retransmission delay and the independence of arrivals for the offered channel traffic are relaxed. That is, only the newly generated packets are derived independently from a Poisson distribution. In general, simulation results obtained with moderate length runs indicate the following. For each value of the input rate λ, there is a minimum value δ_0 for the average retransmission delay variable such that below that value it is impossible to achieve a throughput equal to the input rate. The higher λ is, the larger δ_0 must be to prevent a constantly increasing backlog, i.e., to prevent the channel from saturating. Simulation also shows that for finite values of δ, $\delta > \delta_c$, but not large compared to 1, the system already "reaches" the asymptotic results ($\delta \to \infty$). That is, for some finite values of δ, Assumption 2 is excellent and delays are acceptable. Moreover, the comparison of the (S, G) relationship as obtained from simulation and the results obtained from the analytic model exhibits an excellent match. Thus we consider the results derived above under the assumption of channel equilibrium useful since they are meaningful for these finite (and possibly long) periods of time. Also they provide an accurate assessment of the channel capacity. In [54] simulation experiments were also conducted to find the CSMA "optimal" delay; that is, the value of $\delta(S)$ which allows one to achieve the indicated throughput with the minimum delay. "Delay" here refers to the average over all samples collected in the period of time which represents the length of the simulation runs.

D. Markov Chain and Semi-Markov Chain Models

It is clear from the above discussion that the (assumed) equilibrium throughput-delay results are not sufficient to characterize the performance of the infinite population model. A more accurate measure of channel performance must reflect the trading relations among stability, throughput and delay. The intent here is to show how this can be done by formulating a Markovian model for a population of M users, where M

[4] It is interesting to point out here that it was more difficult to observe this behavior of saturation with the CSMA simulator because CSMA, as shown in [88] and as will be discussed later, is relatively speaking, less unstable than slotted-ALOHA; it will require an extremely long run before one can observe the unstable behavior.

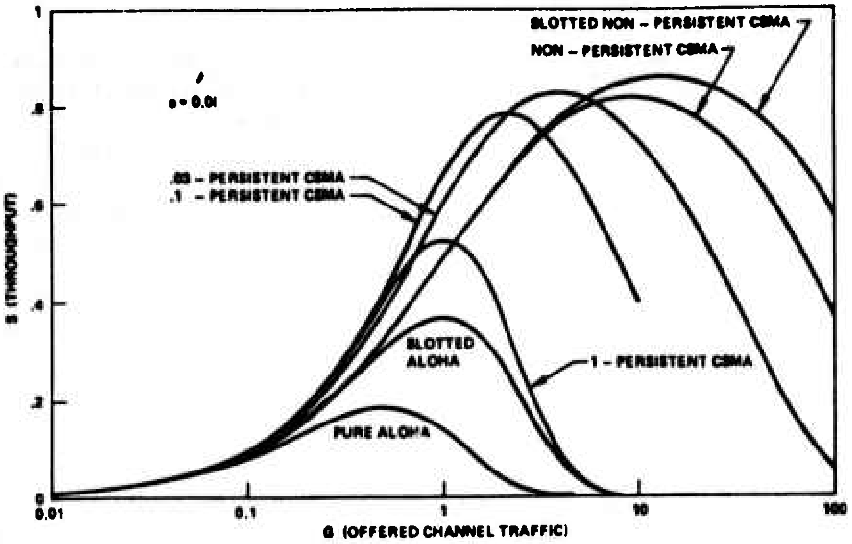


Fig. 4. Throughput versus channel traffic for various random-access schemes.

can be infinite as well. In summary, in this section we first give the definition of a Markov chain and state the Limit theorem which is most relevant to the present discussion. Then we proceed with the Markovian model for a slotted-ALOHA channel and derive its steady-state throughput-delay performance. Next, a discussion concerning non persistent CSMA under similar conditions follows which shows that, due to the dependence of the system evolution on the state of the channel (busy or idle), a simple Markov Process is not sufficient to model the system; instead, results from the theory of semi-Markov chains and regenerative processes are required. We give a brief account on this theory, and the resulting model.

1) Markov Chains: A Markov chain is a stochastic process $\{X_n, n = 0, 1, 2, \cdots\}$ with a finite or countable state-space such that for all states $i_0, i_1, \cdots, i_{n-1}, i, j$ and all $n \geq 0$

$$\Pr\{X_{n+1} = j \mid X_0 = i_0, X_1 = i_1, \cdots, X_{n-1} = i_{n-1}, X_n = i\} = \Pr\{X_{n+1} = j \mid X_n = i\}. \quad (34)$$

If $\Pr\{X_{n+1} = j \mid X_n = i\}$ is independent of n, then the Markov chain is said to possess stationary transition probabilities. In this case we let

$$p_{ij} = \Pr\{X_{n+1} = j \mid X_n = i\}. \quad (35)$$

Perhaps the major results in the theory of Markov chains consist of the Limit Theorems (as $n \to \infty$), and in particular the following [82]

"*Theorem:* An irreducible aperiodic Markov chain belongs to one of the following two classes:

(a) either the states are all transient or all null recurrent; in this case $p_{ij}^{(n)} \to 0$ as $n \to \infty$ for all i, j and there exists no stationary distribution.

(b) or else, all states are positive recurrent, that is,

$$\pi_j = \lim_{n \to \infty} p_{ij}^{(n)} > 0. \quad (36)$$

In this case, $\{\pi_j, j = 0, 1, 2, \cdots\}$ is a stationary distribution and there exists no other stationary distribution where $p_{ij}^{(n)} \triangleq \Pr\{X_{n+m} = j / X_m = i\}$ (the probability of reaching state j from state i in n steps)."

Consider again the slotted-ALOHA scheme and let the channel user population consist of M independent users. Each such user can be in one of two states: blocked or thinking [56], [63], [72]. In the thinking state, a user generates and transmits a new packet in a time slot with probability σ. A packet which had a collision and is waiting for retransmission is said to be backlogged. A backlogged packet retransmits in the current slot with probability p; thus a backlogged packet incurs a retransmission delay which is geometrically distributed. Let N^t denote the total number of backlogged packets at time t. Given the memoryless property of both the generation process and the retransmission process, N^t is a Markov chain with stationary transition probabilities. The state space consists of the set of integers $\{0, 1, \cdots, M\}$. The one-step state transition probabilities of N^t are easily derived (see [56]). For finite M the Markov chain is finite, irreducible and aperiodic and all states are positive recurrent; there exists a stationary distribution $\{\pi_j\}_{j=0}^{M}$. The channel input rate at time t is $S^t = (M - N^t)\sigma$. The average stationary throughput is then simply given by $S = (M - \overline{N})\sigma$ where $\overline{N} = \Sigma_j j\pi_j$. The average packet delay is equal to the average backlog time plus the transmission time of the packet (which equals one slot); the average backlog time, by Little's result, is simply $\overline{N}/S$. The (true) steady-state throughput-delay performance of a slotted-ALOHA system with finite population is thus obtained.

Consider now the slotted nonpersistent CSMA protocol in which the time axis is (mini-) slotted and the slot size is τ seconds, the propagation delay. Packets, assumed to be of fixed length, require a transmission time of T slots. Just as with slotted-ALOHA above, we consider here a user population consisting of M users (terminals), all in line of sight and within range of each other. Again each such user can be in one of two states: backlogged or thinking. In the thinking state, a user generates and transmits (if the channel is sensed idle) a new packet in a (mini-) slot with probability σ. A user whose packet either had a channel collision or was blocked because of a busy channel is said to be backlogged. A backlogged user remains in that state until he successfully transmits the packet at which time he switches to the thinking state. The rescheduling delay of a backlogged packet is also assumed to be geometrically distributed, i.e., each backlogged user is scheduled to resense the channel in the current slot with a probability ν; as specified in the description of the protocol, a retransmission would result only if the channel is sensed idle.

The memoryless property of the geometrically distributed retransmission delay will permit here again a simple state description for the mathematical model. The terminal stays in the backlogged state during the transmission period.

Let again N^t be a random variable representing the number of backlogged users at the beginning of (mini-) slot t. The channel input rate at time t, defined as the average number of new packets generated by the thinking users at time t, is denoted by S^t. Assume M, σ and ν to be time invariant. In slotted-ALOHA, the action that a terminal takes pertaining to the transmission of a packet (either newly generated or rescheduled) is completely independent of the state of the channel (busy or idle). In CSMA, on the contrary, the action taken by a terminal does depend on the state of the channel. One could increase the state description to include an indicator $\mathcal{I}$ for the channel state (busy or idle), however, with *fixed length* packets, $(N^t, \mathcal{I})$ is still not a Markov chain. Tobagi and Kleinrock [88] show that a simpler analysis can be obtained by using results from semi-Markov and regenerative processes. But before we proceed with the model, we give here a brief account on the notion of semi-Markov and regenerative processes and present results most relevant to this discussion.

2) Semi-Markov Chains: A stochastic process which makes transitions from state to state in accordance with a Markov chain but in which the amount of time spent in each state before a transition occurs is random, is called a *semi-Markov* process. Basic results similar to those obtained for Markov chains exist for semi-Markov chains. In particular, the process $X(t_e)$ taken at times t_e defined as immediately before or after a transition is a Markov chain and is referred to as the *imbedded* Markov chain.

If a stochastic process $\{X(t), t \geq 0\}$ with state space $\{0, 1, 2, \cdots\}$ has the property that there exists time points at which the process probabilistically restarts itself, then it is called a *regenerative* process. A cycle is said to be completed every time a renewal occurs. Let $E[.]$ denote the expected value of the random variable following the letter E. A main result relating to regenerative processes is the following

$$p_j \triangleq \lim_{t \to \infty} \Pr\{X(t) = j\}$$

$$= \frac{E[\text{amount of time in state } j \text{ during one cycle}]}{E[\text{time of one cycle}]} \tag{37}$$

for all $j \geq 0$; a very simple and pleasing result.

Furthermore, we can impose a reward structure on the process in the following manner. Suppose that when the process is in state j we earn a reward at a rate $f(j), j \geq 0$. Because of the regenerative nature of the process, it follows that with probability one

$$\lim_{t \to \infty} \int_0^t \frac{f[X(s)]\,ds}{t} = \lim_{t \to \infty} E \int_0^t \frac{f[X(s)]\,ds}{t}$$

$$= \frac{E \int_{\text{cycle}} f[X(s)]\,ds}{E[\text{cycle length}]}. \tag{38}$$

Moreover,

$$\lim_{t \to \infty} E \int_0^t \frac{f[X(s)]\,ds}{t} = \sum_{j=0}^{\infty} p_j f(j). \tag{39}$$

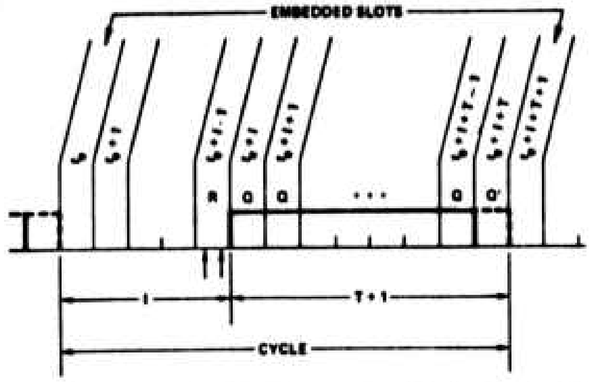


Fig. 5. The imbedded Markov-chain in slotted nonpersistent CSMA.

We now proceed with the semi-Markov model for the nonpersistent CSMA scheme. Referring to Fig. 5, which depicts transmissions and idle periods, consider the imbedded slots to be the first slot of each idle period. The intervals of time between two consecutive imbedded slots are defined as subcycles. These subcycles, of course, are of random length. Consider one such subcycle and let t_e denote the first slot. N^{t_e} denotes the state of the system at t_e. N^{t_e} is an imbedded Markov chain. Let R denote the one-step transition matrices over slot $t_e + I - 1$; Q, over slots $t_e + I$ through $t_e + I + T - 1$; and Q', over slot $t_e + I + T$. The determination of these matrices is easily done based on simple probabilistic arguments (for details see [88].) Given R, Q, and Q', the transition matrix P corresponding to the imbedded Markov chain N^{t_e} is easily calculated. For finite M, the stationary distribution $\Pi = \{\pi_0, \pi_1, \cdots, \pi_M\}$ where $\pi_i = \lim_{t_e \to \infty} \Pr\{N^{t_e} = i\}$ is obtained by solving recursively $\Pi = \Pi P$. This distribution exists for finite values of M.

It is clear that our process N^{t_e} is actually a regenerative process. Thus we can apply the above stated results. Consider an imbedded point t_e and its corresponding subcycle. Since the subcycle is entirely determined by N^{t_e} we consider N^{t_e} to be the representative state for the subcycle which is of random length $I + T + 1$. The success of a transmission in a subcycle N^{t_e} is also entirely determined by the value of N^{t_e}. Given that $N^{t_e} = n$, it is expressed as [88]:

$$P_s(n)$$

$$= \frac{(1-\nu)^n (M-n)\sigma(1-\sigma)^{M-n-1} + n\nu(1-\nu)^{n-1}(1-\sigma)^{M-n}}{1-(1-\nu)^n(1-\sigma)^{M-n}}. \tag{40}$$

Let t_e' and t_e'' denote two successive regenerative points; they are such that $N^{t_e'} = N^{t_e''}$ and such that for any $t_e' < t_e < t_e''$, $N^{t_e} \neq N^{t_e'}$. The interval of time separating t_e' and t_e'' constitutes a cycle. π_j represents the fraction of subcycles such that $N^{t_e} = j$. We now derive the channel throughput. In each subcycle, we define a reward function $f(N^{t_e}) = P_s(N^{t_e})T$, representing the expected time during the subcycle that the channel is used without conflicts; then the throughput S can be written as (s denoting a generic imbedded point)

$$S = \lim_{t_e \to \infty} \frac{\sum_{s_e=0}^{t_e} f(N^{s_e})}{t_e} = \frac{E\left[\sum_{t_e=t_e'}^{t_e''} f(N^{t_e})\right]}{E[t_e'' - t_e']}. \tag{41}$$

The numerator is given by $\Sigma_{j=0}^{M} \pi_j T P_s(j)$; while the denominator is given by $\Sigma_{j=0}^{M} \pi_j [\bar{I}_j + T + 1]$; where $\bar{I}_j$ is the average idle period, given that $N^{t_e} = j$.

If we now define a cost function $f(N^t) = N^t$, representing the backlog at time t, then the average number of backlogged packets is given by (s denoting a generic minislot)

$$\bar{N} = \lim_{t \to \infty} \frac{\sum_{s=0}^{t} f(N^s)}{t} = \frac{E\left[\sum_{s=t_e'}^{t_e''} f(N^s)\right]}{E[t_e'' - t_e']}. \tag{42}$$

Given $N^{t_e} = j$, the knowledge of transition matrices R, Q, and G' allows the determination of [88]

$$n_j = E\left[\sum_{t=t_e}^{t_e+T+I+1} f(N^t) \middle| N^{t_e} = j\right].$$

The numerator is then expressed as $\Sigma_{j=0}^{M} \pi_j n_j$. By Little's result, the average packet delay is $\bar{N}/S$. The (true) steady-state throughput-delay performance of a CSMA channel with finite population is obtained (analytically!).

So far, we have shown how a model based on Markov chain theory and regenerative process has permitted us to derive analytic expressions for the average throughput-delay performance of some important random access schemes under a finite population constraint. This has constituted a considerable progress in our attempt to understand the behavior of these systems.

In particular, a very important observation of the results obtained from the above analysis has been the following. Even in a finite population environment (thus guaranteeing the existence of equilibrium), if the retransmission delay is not sufficiently large (i.e., the retransmission or rescheduling probability is not sufficiently small), then the stationary performance attained is significantly degraded (low throughput, very high delay), such that, for all practical purposes, the channel is said to have failed; it is then called an *unstable* channel. With an infinite population, the Markov chain is not ergodic and stationary conditions just do not exist [11]; the channel is always unstable thus confirming the results obtained from simulation as discussed in Section II-C above. For unstable channels Kleinrock and Lam [56] defined a stability measure which consists of the average time the system takes, starting from an empty state, 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. The stability measure is the average first exit time (FET) into the unsafe region (again a concept borrowed from Markov chain theory!). As long as the system operates in the safe region, the channel performance is acceptable; but then, of course, it is only valid 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 [56], [88].

E. Markov Decision Models

In the Markovian models discussed above it was assumed that the system parameters were all fixed, time invariant and state-independent. These models are referred to as *static*. Clearly, 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 is to find the control functions which provide the best system performance. If the system is Markovian in nature, then the theory known as Markov decision theory provides a basis for analysis. We first give in this section a brief introduction to Markov decision theory and the most relevant results. We then proceed with our example of packet-switching in radio channels and discuss various practical control schemes and their analysis.

Consider the process $X(t)$ and its state space S labeled by the nonnegative integers $\{0, 1, 2, \cdots, M\}$. Let $\mathcal{Q}$ be a finite set of possible actions such that corresponding to each action $\alpha \in \mathcal{Q}$, a set of state transition probabilities $\{p_{ij}(\alpha)\}$ is specified and a cost $C_i(\alpha)$ is incurred. A policy f is a rule for choosing actions. Let $\mathcal{P}$ be the class of all policies. An important subclass is the class of stationary policies $\mathcal{P}_s$. A stationary policy is defined to be one which chooses an action at time t depending on the state of the process at time t. It easily follows that if a stationary policy f is employed, then the sequence of states $\{X_t, t = 0, 1, 2, \cdots\}$ forms a Markov chain with transition probabilities $p_{ij}[f(i)]$. It is thus called a Markov decision process, and it possesses stationary transition probabilities. Let us define the expected cost per unit time for $X(t)$ which was initially in state i as

$$\phi_i(f) \triangleq \lim_{T \to \infty} \frac{1}{T+1} E_f\left[\sum_{t=0}^{T} C_{X(t)}[f(X(t))]/X(0) = i\right]. \tag{43}$$

For a stationary policy f such that $X(t)$ is irreducible we have the following result: $\phi_i(f)$ is simply expressed as

$$\phi_i(f) = \sum_{j=0}^{M} \pi_j(f) C_j(f) \triangleq g(f), \quad \forall i = 0, 1, \cdots, M \tag{44}$$

where $\{\pi_i(f)\}$ is the (unique) stationary distribution of $X(t)$ such that

$$\begin{aligned} \pi_j(f) &= \sum_{i=0}^{M} \pi_i(f) p_{ij}(f) \\ \pi_j(f) &\geq 0 \\ \Sigma \pi_i(f) &= 1 \end{aligned} \tag{45}$$

$g(f)$ is also called the cost rate.

Another important result in the theory of Markov decision processes states that if every stationary policy gives rise to an irreducible Markov chain, then there exists a stationary policy f^* which is optimal over the class of all policies such that

$$g(f^*) = \min_{f \in \mathcal{P}} \Phi_i(f), \quad \forall i = 0, 1, \cdots, M. \tag{46}$$

This pleasing result means that, in most cases of interest, one can limit the attention to the class of stationary policies. A very efficient computational algorithm known as the Howard's policy iteration method [36], [37] exists which allows the evaluation of the cost rate $g(f)$ and which leads to an optimal stationary policy.

We continue here with our now familiar example of random access techniques. It follows from the discussion in the previous section that, if M is finite, a stable channel can be achieved by using a sufficiently small retransmission or rescheduling probability. But a smaller retransmission probability implies a larger backlog size and hence a larger packet delay! Moreover, it is noted that, for stable channels and for a given total throughput, the packet delay increases with increasing population size [56], [88]. Basically, the reason for this behavior is that the appropriate constant retransmission probability has to be small enough to overcome the degrading effect resulting from those statistical fluctuations which otherwise would drive the system into an "unsafe region." Dynamic control provides an effective solution to this problem; it enables an originally unstable channel to achieve a much improved throughput-delay performance, and conversely it improves the (otherwise high) delay performance of a stable channel with large M. With dynamic control we also allow the channel to accommodate varying input load without any disastrous effect.

Markov Decision theory has successfully been applied by Lam and Kleinrock [64] to the design and analysis of control procedures suitable to slotted-ALOHA in particular and random-access techniques in general. More precisely the objective is to decide on some practical control scheme and to derive the optimal stationary policy. Two main types of control are proposed: an *input* control procedure (ICP) corresponds to an action space consisting of either accepting or rejecting all new packets arriving in the current slot; a *retransmission* control procedure (RCP) corresponds to an action space $\{f(i)\}$ where $f(i)$ denotes the retransmission probability in a slot in which the backlog is i. A more general description of the action space, of which ICP and RCP are special cases, is as follows [64]. Let $\mathfrak{a}_1 = \{\beta_1, \beta_2, \cdots, \beta_m\}$ where $0 \leq \beta_1 < \beta_2 < \cdots < \beta_m \leq 1$ and $\mathfrak{a}_2 = \{\gamma_1, \gamma_2, \cdots, \gamma_k\}$ where $0 < \gamma_1 < \cdots < \gamma_k < 1$. Let $\mathfrak{a} = \mathfrak{a}_1 \times \mathfrak{a}_2$. A stationary policy f maps the state space $\{0, 1, \cdots, M\}$ into $\mathfrak{a}$ such that $f(i) = (\beta, \gamma)$ means that whenever the system state at slot t is $N^t = i$, then each new packet is accepted with probability β and rejected with probability $1 - \beta$ and each backlogged packet is retransmitted in that slot with probability γ.

Given a policy f, it is easy to write the one step transition probabilities $\{p_{ij}[f(i)]\}$ for the Markov decision process N^t. Under the condition that N^t is irreducible, the stationary distribution $\{\pi_i[f]\}$ exists. It remains to define the cost rates $C_i(f)$ and to determine the performance measures. For the sake of simplicity, let us restrict ourselves here to just RCP. The more general treatment can be found in [64]. Let $f = \{f(i)\}$ denote the policy. Supposing $N^t = i$, define the immediate reward $C_i(f)$ to be the expected channel throughput in the t^{th} time slot, $\overline{S}_{out}(i, f)$. Then by (44), the reward rate becomes

$$g_s(f) = \sum_{i=0}^{M} \pi_i(f)\,\overline{S}_{out}(i, f). \tag{47}$$

This is also the channel throughput rate. Consider now the stationary average packet delay. Supposing $N^t = i$, define the expected immediate cost to be $C_i(f) = i$ thus accounting for the waiting cost of i packets incurred in the t^{th} time slot. By (44), the cost rate is

$$g_d(f) = \sum_{i=0}^{M} i\pi_i(f). \tag{48}$$

Applying Little's result, the average "wasted" time of a packet is simply $D_w = g_d(f)/g_s(f)$. The main objective here is to find a policy which optimizes channel performance, that is, which minimizes the delay for a given stationary throughput. Fortunately, this task is simple since it can easily be shown that for any stationary policy $f = \mathcal{S} \rightarrow \mathfrak{a}$, we have

$$g_d(f) = -\frac{g_s(f)}{\sigma} + M \tag{49}$$

meaning that if there exists a stationary policy which minimizes the cost rate $g_d(f)$, this policy will also maximize the reward rate $g_s(f)$ [64]. Having decided upon a channel control procedure, the optimal policy is determined via Howard's policy-iteration algorithm. This will also allow the determination of the associated optimum performance [64].

Independently, Fayolle *et al.* [19] give yet another treatment of the instability of slotted ALOHA channels with infinite populations and propose similar control procedures to recover stability. In particular, it was shown that, with retransmission control procedures, only policies which assure a rate of retransmission f from each blocked terminal which is inversely proportional to the number of backlogged terminals, will lead to a stable channel. An expression for the optimal policy $\hat{f}$ was also given. In their paper, Lam and Kleinrock [64] suggest that a good control policy must be of the control limit type. Their intuition was confirmed by the numerical solutions obtained from Howard's policy iteration algorithm; however, there is no rigorous proof of optimality. In [88] Tobagi and Kleinrock similarly addressed themselves to the dynamic control of the nonpersistent CSMA protocol. In essence, it was shown that one can improve the channel performance by selecting the retransmission probability which maximizes the "instantaneous" throughput, that is, the average throughput over a subcycle. The resulting overall channel performance was further shown to be then insensitive to the population size.

It is apparent to the careful reader that the preceding (exact) analysis is based on a major system assumption, namely that each user knows the exact current state of the system. Clearly, this assumption does not hold in practical situations! The channel users have no means of communication among themselves other than the multiaccess broadcast channel itself. But each channel user may individually estimate the channel state by observing the channel outcome over some period of time, and apply a control action based upon the estimate. In the context of slotted-ALOHA, Lam and Kleinrock [64] give some heuristic control-estimation algorithms which prove to be very satisfactory. With appropriate modifications and extensions, these algorithms can be applied to CSMA channels as well. The difficulty in incorporating the estimation algorithms into the mathematical model incites us again to the use of simulation techniques. The results obtained by the mathematical model assuming full knowledge of the system state represent the ultimate performance; a bound on the performance obtained via any heuristic estimation algorithms. The goodness of the latter is evaluated by comparing simulation results to these bounds. (For numerical results, the reader is referred to [64].)

F. Mathematical Programming

So far we discussed modeling tools which have been mainly used to evaluate throughput and delay performance of communications systems without paying much attention to optimization issues (except, perhaps, in the development of Markov decision models). Here, we address the optimization problem more directly, and review the mathematical programming tools commonly used in network design.

The typical problem consists of optimizing a performance measure (e.g., cost, delay, throughput), over a set of variables, meeting given performance constraints. The ability to solve the design problem depends critically on our success in expressing both objective function and constraints in analytically manageable form as a function of the design variables. Thus network models are an essential prerequisite to design.

Unfortunately, most performance expressions are rather complex, requiring approximations in the model or relaxation of the constraints in order to formulate the design problem in convenient terms and solve it with powerful mathematical programming techniques. A typical example of this approach is the linearization of discrete line costs in the minimum cost design of land based packet networks [57]. In some cases, the nature of the variables and of the constraints is so complex that a mathematically manageable formulation of the problem is not possible. An example is the topological network design to satisfy two-connectivity constraints [83]. In these cases only heuristic approaches can be of help (see Section II-H).

Design variables may be continuous (e.g., link data rates, packet length) or discrete (e.g., topology, number of switch sites).

Examples of methods commonly used in continuous variable computer network designs are: 1) linear programming; 2) Lagrangian optimization; 3) multicommodity flow optimization (discussed in the following section); 4) gradient projection method. Examples of methods used in discrete optimization are: 1) dynamic programming; 2) Lagrangian decomposition; 3) branch and bound [32].

In this section, as an example, we discuss the Lagrangian method as it applies to the Capacity Assignment (CA) problem in land based packet networks [57].

The CA problem can be formulated as follows.

Given: topology, average link data flows, $f = (f_1, f_2, \cdots, f_M)$ (where M is the number of links and f_i is the bit rate on link i and is given by $f_i = \lambda_i/\mu$),

minimize link costs: $D = \sum_{i=1}^{M} d_i(C_i)$

over the selection of link capacities $C = (C_1, C_2, \cdots, C_M)$

subject to: $C > f$

$$T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i} < T_{MAX}.$$

Line capacity options are in general discrete. To simplify the problem we may approximate the discrete capacities with continuous values; furthermore, we may approximate the cost function with a linear function, i.e.:

$$d_i(C_i) = d_i C_i + d_{i_0} \tag{50}$$

To solve the linearized problem we use the method of Lagrange multipliers. To this end we construct the Lagrangian L, defined as the sum of the objective function plus the constraint function multiplied by the multiplier β:

$$L = D + \beta(T - T_{MAX})$$

$$= \sum_{i=1}^{M} d_i C_i + d_{i_0} + \beta\left(\frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i} - T_{MAX}\right). \tag{51}$$

By setting the partial derivatives $\partial L/\partial C_i$ to zero, and choosing β so that the delay constraint is satisfied, we obtain the optimal expressions for C_i:

$$C_i = f_i + \frac{\sum_{k=1}^{M} \sqrt{d_k f_k}}{\gamma T_{MAX}} \sqrt{\frac{f_i}{d_i}}. \tag{52}$$

G. Multicommodity Flow Optimization

In earlier sections we showed how to evaluate delay performance in land based networks, assuming static routing. In most networks, however, route selection is adaptive to load pattern and to network conditions. Since the delay performance is critically dependent on the routing policy, we wish to develop models that predict delay performance also in a dynamic routing environment.

Unfortunately, the general dynamic routing problem is very complex. Network of queues theory may be used. However, the fact that transition probabilities depend on network load precludes the derivation of "product form" solutions (see Section II-B). The system may still be represented as a very general Markovian system and solved using numerical techniques. This type of solution, however, is computationally very cumbersome and offers little insight into the dependence of network performance on dynamic routing parameters, let alone their optimization.

To overcome this problem, we approximate the dynamic routing solution with the optimal static solution. More precisely, we first find the optimal static routing strategy using mathematical programming techniques. Then we verify that the dynamic strategy performs almost as well as the static strategy at steady state. This verification may be carried out using Markovian models in simple cases, and simulation in more complex situations.

The advantage of this two-stage approach is that the verification stage although computationally cumbersome needs to be carried out only for a few representative benchmarks, while the static optimization stage (which must be solved an endless number of times in a typical network design problem with several topological alternatives) is performed very efficiently using multicommodity flow techniques.

Multicommodity flow techniques are mathematical programming techniques used to optimize the distribution of "commodities" (in our case, packet flows) throughout a network, between several origin–destination pairs. The problem constraints are generally the line capacities. The objective may be the total throughput, or the delay, or another appropriate function of the flows in the network.

The multicommodity flow matrix F for a data network is an $N \times L$ matrix (N = number of nodes; L = number of links) whose entry $F(i, k)$ represents the average data flow (bits per second) on link k with final destination i. Each row of F represents the "single commodity" flow to a distinct destination.

The matrix F uniquely identifies the steady-state routing scheme. In order to find the optimal routing solution (at steady state) we just need to optimize $g(F)$, where $g(F)$ is the desired performance measure. This optimization can be carried out very efficiently using a decomposition approach. We decompose each single commodity flow into the convex sum of the flows on all possible paths to a given destination. Clearly, the number of possible paths can grow very large, but one can show that only at most N paths are included in the optimal solution. Using an iterative procedure (called flow-deviation method [26], we introduce at each step a new path that can improve performance. The selected path is the shortest path evaluated using as weight for link i the partial derivative $W_i = \partial g(F)/\partial f_i$, where f_i is the total flow (sum of all commodities) in link i. An appropriate amount of traffic is then "deviated" from the previous paths to the new shortest path. The iterative procedure terminates when the relative improvement obtained by the deviation falls below a specified tolerance, at which point we have reached a local minimum.

Multicommodity flow techniques can be used to solve a variety of problems in data network designs. Here we consider, as a specific example, the problem of finding the minimum delay routing in a land-based packet-switched network. The problem is formulated as follows.

Given:
- Topology
- Channel capacities
- Traffic requirement matrix R.

Minimize:

$$T = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{f_i}{C_i - f_i}.$$

Over the design variable:

$$f = (f_1, f_2, \cdots, f_M).$$

Subject to

a) f is a multicommodity flow induced by requirement matrix R

b) $f_i \leq C_i$.

It is easy to verify that the objective function $T(f)$ is convex, and, therefore, the locally optimal solution found by the Flow Deviation Algorithm is also globally optimal. The link weights used at each iteration are:

$$W_i = \frac{\partial T}{\partial f_i} = \frac{C_i}{(C_i - f_i)^2}, \quad \forall i = 1, \cdots, M. \tag{53}$$

In general, the key requirements for the successful solution of the routing problem are the capability of expressing the delay as a function of link flows, and the convexity of such a function. For land-based networks, we have just shown that such requirements are satisfied. One may show that the requirements are also met for mixed terrestrial and satellite networks [29]. Distributed packet radio networks, on the other hand, the delay formulation is much more complex because of the interference existing between neighboring nodes.

H. Heuristic Techniques

Heuristic and approximate solutions are often the only feasible approach to some of the more complex analysis and design problems related to large packet networks. A classification of the various heuristic methods is certainly beyond the scope of this paper. Interested readers are referred to [67] for an overview. One particularly important class of heuristics, however, will be discussed here: namely, the class of topological design algorithms.

The topological optimization of a packet network is a formidable mathematical programming problem made difficult by the combination of integer type variables (switch number and location, topology, line speeds) and multicommodity flow variables describing the routing of packets in the network. Therefore, the only practical way of approaching medium and large network designs is via heuristics. Several procedures have been proposed, which are based on different concepts. The common philosophy of these procedures consists of identifying a condition which is necessary (although generally not sufficient) for optimality, and of achieving this condition by means of repeated topological transformations until a locally optimal solution is found. Starting from randomly chosen initial configurations, a large number of local minima is explored to enhance the probability of success of the heuristic method.

One of the most popular topological design heuristics is the Branch X-Change (BXC) method [24]. The local condition for optimality is the condition that the cost be not reduced by any BXC (i.e., the elimination of one or more old links and the insertion of one or more new links). Thus, the BXC algorithm explores exhaustively all the feasible BXC's, accepting only the X-changes that lead to cost reduction, until no more improvements are possible.

Another, more recent algorithm, the Cut Saturation (C-S) algorithm, can be viewed as a refinement of the BXC Algorithm in that only a selected subset of the possible X-changes is explored [25], [30]. More precisely, only the X-changes involving the insertion of links in the Saturated Cut (i.e., the minimal set of most utilized links that, when removed, disconnects the network) and the deletion of lightly utilized links are considered.

A third algorithm, the concave branch elimination (CBE) is based on flow optimization concepts [32]. The CBE algorithm can be applied whenever the discrete line costs can be approximated with continuous concave curves of cost versus capacity. Under these assumptions the total network cost for a given topology can be expressed as a concave function D of the link flows, namely:

$$D(f) = \sum_{i=1}^{M} \left[d_{i_0} + d_i f_i + \frac{\left(\sum_{k=1}^{M} \sqrt{d_k f_k} \right)^2}{\gamma T_{MAX}} \right]. \tag{54}$$

We can then apply the flow-deviation algorithm to obtain a minimum. This minimum is only a local minimum since the function $D(f)$ is concave (instead of convex, as in the routing problem). In the process of finding a local minimum, it can be shown that the algorithm eliminates uneconomical links (i.e., reduces their flow to zero), and, therefore, strongly reduces the initial topology. Exploiting this locally optimal property, several local solutions are investigated starting from different initial configurations.

Several other heuristic methods have been proposed and applied with more or less success to various types of networks. The common element of all the methods is the existence of a local optimality principle and the need to randomly explore several solutions in order to improve the accuracy of the design.

I. Simulation

As we have noted at several points in our earlier discussion, analytic modeling techniques are inadequate to deal with many of the details of a system to be modeled. Simulation is then used.

Basically, simulation has two main purposes: a) the performance evaluation of network protocols that are analytically intractable; and b) the validation of analytical models based on simplifying assumptions. These purposes, however, are adequately served only if the simulation model itself is valid; and thus techniques that can be used to guarantee the validity of the model and its results are required. We discuss each of these issues in more details and supply the reader with some typical examples.

Numerous examples exist where simulation is used as a tool to evaluate alternative protocols. Unfortunately, there is no particular work that serves as a basis for a discussion here as was the case with analytic modeling. We refer the reader to applications which have appeared in the literature on the ARPANET, the Atlantic Packet Satellite Network, the NPL network and the Cigale Network [27], [39], [33], [49], [77].

A recent example of simulation applied to validate analytical models is offered by Lam's study on network congestion control [65]. In this model, each nodal switch is represented by a network of queues. The nodes in turn are interconnected by a higher level network, which is, in fact, the network topology. Due to buffer constraints and, consequently, nodal blocking, the global network of queues does not have a convenient "product form" solution, as discussed in Section II-C. However, by postulating static (i.e., state independent) blocking probabilities for the nodes, the global model was reduced to a collection of independent submodels (one for each packet switch), which were solved separately, using the product form approach. Clearly, the assumption of static blocking probabilities is of critical importance for this study. Thus it was thoroughly tested using a very accurate simulation program. Experiments showed good agreement between simulation and analytical results.

Analytic queueing models and simulation queueing models go hand in glove. We have pointed out that, when we are trying some new technique of analysis it is useful to cross check that analysis with the results of simulation runs. Also very important is the use of analytic models to validate large simulation programs. It is often the case that a simulation program will be thousands of lines of code (in Simscript, say) and the modeler is faced with the question of whether or not this enormous program actually models the system in question. Now it is often the case that the logic of the program can be tested by comparing its results with an analytic model when the program parameters are set to correspond. That is, much of the complexity that cannot be handled in analytic models may appear in the simulation as parameter settings. When the parameters are set to certain values the simulation may be modeling a Markovian network of queues for which we can analytically predict the behavior. If the simulation produces correct results for this case then one is inclined to believe that the results for other parameter settings are also correct. An example of this approach can be found in [62].

When we perform a simulation we usually gather statistics on certain variables that we wish to measure (queue lengths, transit times, $\cdots$). But these statistics are not very accurate since they are derived from highly correlated samples. Their proper interpretation requires the use of time series analysis rather than classical statistics. It is all too common, however, to find that simulation results are quoted for a single run of each desired experimental point. Simulation languages often gather statistics using models based on classical statistical theory. However, a correct approach requires the experimenter to obtain a large number of independent samples (40 to 100 samples, say) if classical analysis is to be used to estimate the mean of some measured quantity and to place a confidence interval around that estimate. This would involve thousands of computer runs each of which may be several hours long. Few people can afford this; and the typical strategy is to take the one observation as correct. There are very many problems where this will work, but it is fortuitous and modelers must beware of extrapolating such successes.

An alternative scheme is to break the simulation run down into several "subruns" in which statistics are separately gathered. These runs hopefully are independent. The problem is to be sure of this. Recent work by Crane and Iglehart [14], [15] and by Fishman [21], [22], [23] helps us here. They carefully explain the problem and suggest the use of "regeneration points" to obtain independent samples. The basic idea is that systems return periodically to certain configurations and that the behavior of the system after reaching such a regeneration state is independent of its behavior prior to entering that state. Thus by taking separate samples during the cycles between entry into the regeneration state one obtains the required independent samples that permit the use of the simple classical statistical tools. The most recent work by Fishman [23] offers practical guidance here.

Regeneration points are not a panacea. The problem is that the cycle time between reentry may be too long for practical application. Consider a system with 10 queues each with finite waiting room for 10 customers. There are 10^{10} possible states! Nevertheless, the above referenced material is important reading since it offers useful insight into the problem of establishing confidence in our simulation experiments.

These problems are one reason that analytic models are to be preferred to simulation if they can be used. They give results much more cheaply, even if several weeks are required to develop the model. Furthermore, they aid our intuition. It is much easier to comprehend the implications of even the most complex formulas than it is to comprehend the meaning of 8000 lines of Simscript code and a basketful of output tables! Both tools are needed.

Finally, a novel approach known as *hybrid simulation*, is worth mentioning. Basically, the idea is to combine both discrete-event simulation and mathematical modeling in an attempt to achieve good agreement with the results of an equivalent complete simulation model, but at a significantly lower cost. This approach is possible if the frequency of state transitions of some portions of the system is much higher than those of other portions. Then the high-frequency events are accounted for in a computationally efficient analytic submodel while the relatively infrequent (and often more complex) events are accurately simulated. Parameters are exchanged back and forth between the various submodels. Hybrid models have been successfully applied for the analysis of computer systems [10a], but have not yet been widely used in computer communications systems.

III. Measurements

Manfred Eigen wrote: "A theory has only the alternative of being right or wrong. A model has a third possibility: it might be right but irrelevant" [17]. Indeed, most if not all of the modeling work is based on simplifying assumptions without which the analysis becomes intractable; and with these assumptions we run the risk of providing results which do not exactly conform to the real situation. "Irrelevant" is perhaps too strong a word: in the absence of a real system (that is, in the early design phase) analytic and simulation models are the only tools available to guide us in first implementations. But once the system is built, measurements allow us to gain valuable insight regarding the network usage and behavior. They provide a means to evaluate the performance of the operational protocols employed and the identification of their key parameters; they allow for the detection of system inefficiencies and the identification of design flaws; when used to improve network design, they are a valuable feedback process by which existing analytical models are validated and/or improved, and in which design deficiencies are detected and subsequently corrected. Thus contrary to early designed computer systems which did not allow sufficient freedom in experimentation, and in line with Hamming's observation that "it is difficult to have a science without measurement," elaborate measurement facilities constitute an integral part of all experimental and many operational computer communications systems of today. Basically, the measurement task consists of identifying the measurement functions with respect to the system and devising the measurement facilities required to support those functions under the constraints that the system imposes. In this section, we shall first review the basic measurement tools, their capabilities, their limitations, and their applicability to and implementation in different network environments, namely, land-based wire networks, satellite networks, and ground packet radio networks; next, we shall show the importance of well designed experiments in satisfying the many measurement goals.

A. Measurement Facilities

Although the objective of measurements is basically the same for all types of networks (wire, ground radio, or satellite), several factors exist which do not allow for a simple transfer of measurement facilities from one to the other. The techniques may very well be the same, but the implementations of these tools will have to be compatible with the particular system's design and comply with its limitations.

Most of us are now familiar with land-based wire packet-switched networks and their structure; the switches are minicomputers which provide the store-and-forward function and handle routing and error control; typical examples are the ARPANET [79] (in which the nodal switch is called the IMP), the Cigale Network [76a], TELENET [75a], DATAPAC [10b]. Satellite and ground radio networks, however, are far less common than wire networks and a brief introduction here is in order. In a satellite system (an example of which is the SATNET [33], [43], a node is basically a minicomputer switch similar to the ARPANET IMP interfacing with the satellite channel by means of satellite radio equipments. All nodes share a common satellite channel via some access scheme. In a ground radio environment, the issues are somewhat more complex. Besides the original but simple one-hop ALOHA system at the University of Hawaii, the only and prominent example of a fully distributed radio network is the ARPA Packet Radio Network (PRNET), a prototype of which has already been deployed in the San Francisco Bay Area [44], [45]. There are three basic functional components in the PRNET: (i) the packet radio terminals which are the sources and sinks of traffic, (ii) the packet radio repeaters whose function is to extend the effective radio range by acting as store-and-forward relays; and (iii) the packet radio stations which provide global control for the network of repeaters and act as interfaces between the broadcast system and other computers or networks. The repeater is called a packet radio unit (PRU) and consists of a radio transceiver and a microprocessor. It receives and transmits packets according to dynamic routing and control information provided by the stations. For simplicity and uniformity of design the PRU is also used as the front-end of terminals and stations interfacing them with the radio net.

Although the PRNET utilizes the technique of packet-switching, the packet radio measurement facilities are unique with respect to the system constraints [87]. These constraints are in large part due to the desire to keep the components small and portable, to the limited speed of the microprocessor (which in turn is due to the assumed limited power supply available in some military applications) and to the limited available storage at each PRU. The overhead placed on the PRU is also of utmost importance in evaluating the feasibility of a measurement tool and of the collection of data in support of a measurement function. In particular, due to the broadcast nature of transmissions, the transmission of collected measurement data not only introduces overhead over its own path, but causes interference at all neighboring repeaters within hearing distance and creates additional overhead on those PRU's activities. The development of the measurement tools in the PR Net is an excellent illustration of the iterative design process involved whereby a balance is sought between supporting the measurement functions and satisfying the system constraints, thus making sure that the network communication protocols allow the implementation and proper functioning of those tools. Specific examples will surface in the sequel substantiating this statement.

We now describe the various types of statistics used, and the artificial traffic (and noise) generators needed in measurement experiments and the various techniques available for data collection.

1) Cumulative Statistics (CUMSTATS): Cumulative statistics consist of data regarding a variety of events, accumulated over a given period of time. These are provided in the form of sums, frequencies, and histograms. In the ARPANET, for example, cumstats are collected in each IMP and include a summary of the sizes of messages entering and exiting the network [12], [51], the number of IMP words, the number of control messages, etc. "Global" traffic data are also collected; they are referred to as round-trip CUMSTATS; they include the number of round-trips per possible destination and their delays.

In the PRNET, distinction is made between cumstat data collected at the PRU's (which provide information about the local environment and behavior such as traffic load, channel access, routing performance and repeaters' activity) and those collected at the end devices (which reflect more global network behavior such as user delays and network throughput). A detailed list of the data items of interest in PRNET experiments can be found in [87].

The implementer may add the flexibility of tailoring to some degree, the content of the CUMSTAT to the requirements of a specific experiment. In SATNET, for example, most of the items in the CUMSTAT message are optional and may be requested by the experimenter at the beginning of the experiments. The advantage of this solution is to make available a very large set of measurements, without the line and processor overhead usually required to construct and transmit long CUMSTAT messages.

2) Trace Statistics: The trace mechanism allows one to literally follow a packet as it flows through a sequence of nodes and thus trace the route it takes and the delays it encounters at each hop. In the ARPANET, selected IMP's whose trace parameter has been set gather data (at the IMP's themselves) on each (trace marked) packet and send this data to the experimenter at the collection point as a new packet. Trace data consists of time-stamps related to the time of arrival, the time of transmission, and the time that the acknowledgment is received [12]. While the above implementation has been possible in the ARPANET, in a PRNET, the collection of trace data at the repeaters is prohibited by the limited size of storage in the PRU. To overcome this problem, a new type of packet called the pick-up packet has been introduced [87]. Pick-up packets are generated with an empty text field by traffic generators at end devices. As these packets flow normally in the network according to the transport protocols, selected repeaters gather the trace statistics and store them within the text field of the pick-up packets themselves.

3) Snapshot Statistics: A Snapshot provides an instantaneous look at a device (IMP, PRU) showing its state with regard to various queue lengths and buffer allocation. Although this information can be obtained by a time sequence of state changes, the snapshot technique is preferable in that it reduces considerably the overhead and artifact caused by collecting and sending the statistics too frequently. In the ARPANET, snapshots also include the IMP routing table and its delay table. The correlation of these with other collected statistics will help explain abnormal or unexpected behavior. In the PRNET, the stations play a central role in providing some degree of global control for the operation of the entire network. They contain tables describing the network connectivity, the repeaters states and their parameter values. Changes in appropriate tables are time stamped and collected as the stations snapshot functions.

4) Artificial Traffic Generators: Artificial traffic generation is clearly a requirement of any experimental system. In the absence of real user traffic, the experimenter is thus given the ability to create streams of packets between given points in the net, with specified durations, inter-packet gaps, packet lenghts, and other appropriate characteristics. In the ARPANET, the IMP's message generator can send fixed length messages to one destination; in the PRNET a higher level of flexibility is implemented: each traffic source (at terminals and stations) can provide one or more streams of both "information" packets and/or pick-up packets, each with a specified packet length, frequency, destination and duration. In SATNET, each station may generate up to 10 independent streams of artificial traffic, each stream having its own characteristics (generation rate, packet length, priority, etc.). Furthermore, the generation rate of a stream may be changed to preset values at preset times during an experiment. For example, the generation rate of a station may be set to 0 in the interval $(0, T_0)$; to 1 in $(T_0, T_0 + \Delta T)$; and again to 0 in $(T_0 + \Delta T, \infty)$. The result, in this case, is a pulse of amplitude 1 and duration ΔT, at time T.

5) Emulation: In most initial experiments, the system under investigation consists of a limited number of elements, thus placing severe constraints on the experimenter in his attempt to understand the system behavior in future and more realistic environments. This makes it desirable to emulate in a single element the traffic that would be generated by several separate sources. An interesting example of multiple node emulation is offered by SATNET. The physical configuration of SATNET consists of 4 stations, a number too small to carry out any meaningful type of stability experiments. The experimental capabilities in this direction were considerably expanded by implementing in each real station 10 "fake" stations equipped with all the essential protocols to permit their independent operation.

6) Network Measurement Center–Control, Collection, and Analysis: The need for controlling the measurement facilities, and collecting and analyzing measurement data, gives rise to the notion of a network measurement center (NMC). For the three above mentioned networks (ARPANET, PRNET, and SATNET), for example, the University of California at Los Angeles (UCLA) has been successfully playing the role of NMC. For the French Cigale network, the measurement task is being undertaken by IRIA [35].

In the ARPANET, messages are sent to a background program in the IMP to trigger the necessary parameter change; in fact, once an experiment is specified, these messages are automatically formatted and sent by a set of programs constructed at the NMC. Conversely, measurement data is gathered at the various sites, formatted and routed to the UCLA-NMC where it is stored and analyzed. Similar techniques are employed for the PRNET and SATNET while using the ARPANET to communicate with the measurement facilities and to transport the data back. In the PRNET, in particular, it is through the station that the initiation and termination of measurement experiments is controlled. When provided with the appropriate commands, the station enables and disables the CUMSTAT and Pick-up packet functions at the PRU's and assigns to the various elements the intervals for CUMSTAT collections, and to the artificial traffic generators their corresponding parameters. At the present time, it is also to the station that all measurement data is destined; upon arrival at the station, the data is time-stamped and stored in a single measurement file for shipment to UCLA-NMC where off-line reduction and analysis is performed.

It is possible that if enough care is not taken, the collection of measurement data at a network measurement center will create overhead traffic on the network, and serious considerations have to be made as to the techniques used. For an illustration, let us limit the discussion here to the PRNET context, and consider specifically the collection of cumulative statistics at the station using the PRNET itself. Two ways can be thought of to achieve the transport of the data. One method, called the Automatic method, consists of having the PRU form, at the end of each CUMSTAT interval, a measurement CUMSTAT packet which is time-stamped and transmitted to the station. The second method consists of having the station issue at regular intervals executable packets (control packets with code to be executed at the destination PRU) called Examine packets, to PRU's; these collect the time-stamped CUMSTAT data and return to the station. Both methods are certainly valid; but since for analysis purposes, it is strongly desirable for the CUMSTAT data received at the station to correspond to equal length time intervals at the originating device, the automatic method in conjunction with a reliable transmission protocol between the PRU's and the station is preferred. With the Examine method, variable length intervals will occur since Examine packets, even though sent at regular intervals from the station, are subject to network random delays en route to the PRU and subject to the possibility of loss in either direction. Analysis of these collection methods and the overhead they impose appeared in [86]. Also, to alleviate the congestion around the station due to the convergence of (i) traffic normally flowing through the station, (ii) control traffic carrying information needed at the station, and (iii) measurement traffic, it is not unlikely that the future design will use a separate station remote from control stations for the sole purpose of measurements and other less vital system functions. In fact, in recent experimentations with the CIGALE network, to prevent any interference with the operational CIGALE traffic, the IRIA measurement group went to the extreme of installing a measurement laboratory isolated from the network [35]. Such a setup was perfectly adequate for their need, namely measuring the performance of line protocols in an environment of somewhat limited complexity.

7) Network Control Center–Status Report, Monitoring, and Control: In addition to the measurement effort which is primarily for experimentation and performance evaluation, certain measurement data can be of vital importance to the proper operation of a network. With the notion of network control emerges also the concept of a network control center (NCC). In the ARPANET, each IMP sends periodically a status report to the NCC containing various data such as the up-down status of each HOST and channel, a count of the number of packets entering each IMP, and other statistics regarding each channel and the traffic it carries. Data is processed at the NCC creating summary statistics and advising the operator of failures. Network monitoring and control is even more significant in a PRNET where, in order to satisfy the constraints imposed by the repeaters, one or more station have the responsibility for distributed control over an entire region of the network (reliability is achieved through redundancy). To carry out this responsibility, the station requires various indications of network activity and performance. Some of this information is acquired from incoming traffic, but much of it is specifically obtained by having monitoring procedures collect, from the various devices, a subset of the measurement items. These will assist the control and routing algorithms implemented at the stations in taking the proper actions.

B. Measurement Experiments

After having described the types of measurement tools, available in packet networks, we proceed to show how these tools are selected and coordinated to carry out a specific experiment. We first define the object of our experiment, i.e., the specific aspect of the network that we want to investigate (e.g., channel access protocol in a ground radio network), along with the goals of the experiment (e.g., verification of correct protocol implementation). Then we select the performance measures which best characterize the aspect of the network under study (e.g., in a random access experiment we may choose to monitor the number of retransmissions until success). Finally, we select the appropriate subset of measurement tools (e.g., CUMSTATS; Pick-up packets etc.) which permit us to monitor the desired performance measures and, if necessary, to follow step by step the various network operations. Thus four ingredients come together to form an experiment: 1) the measured object of the measurement; 2) the goals; 3) the performance measures; and 4) the tools used in the experiment.

The complete list of experiments proposed for a given system constitutes the so-called experiment plan. A preliminary experiment plan is generally prepared before the implementation of the network. Based on this plan, the network designer and the experimenter agree on a set of tools that are adequate to carry out the desired functions, and yet do not overtax the system. Naturally, after network implementation, the experiment plan is frequently revised using as feedback the results of previous experiments. A new set of experimental requirements may arise which were not anticipated during the preliminary plans and for which there is no adequate measurement software support. While this occurrence may be minimized by careful preplanning (possibly assisted by analytical and simulation models), a more general solution is the flexible design of the measurement software to permit modifications and expansions of existing tools to meet new demands.

An example of production interaction between measurement planning, interpretation of the results, and new measurement software design is offered by SATNET, the Atlantic experimental satellite network. In SATNET the software was developed in stages, each stage having a different set of operational capabilities and measurement tools. This allowed the upgrading of the measurement software at each step, based on previous experience. In particular, the traffic generator, a very critical element in the testing of channel access performance, underwent a remarkable evolution throughout the duration of the experiments. In the original implementation the artificial messages were subject to the ARPANET, RFNM based end-to-end protocols [33]. Early measurement results showed that this protocol structure was too restrictive in the presence of large satellite delays and would actually trigger undesired "capture" effects. A new generation, therefore, was developed to provide a separate source of packets at constant rate. Later on, during the experimentation of S-ALOHA channel access protocols, the need was identified for a time varying packet generation rate. This feature was also introduced making the evaluation of channel stability properties much more effective. More recently, the generator in each station was upgraded to generate a certain number of parallel packet streams, each stream having different rate, priority and message length characteristics. This feature was used to simulate a diversified user environment, necessary to exercise the very sophisticated, priority oriented, demand assignment protocols implemented in SATNET [42].

1) Objects of the Measurements: Let us return to the basic ingredients that define an experiment. First, we focus on the object, i.e., the aspect of the network that we want to test. Here the following classification may be found useful.

a) Experiments on communications subnet protocols: These experiments may be directed to the study of specific network protocols (e.g., routing protocol; acknowledgment protocol; source-destination node protocol [52]) or may involve the interaction of all the protocols (global performance evaluation). Generally, only artificial traffic is used in this phase to better isolate the behavior of the protocols from user related effects.

b) Experiments on user behavior: We are interested in determining user traffic characteristics (traffic pattern; message length distribution, etc.) which are independent of the subnet itself. An excellent example of this type of experiment is offered by the investigation of ARPANET user behavior by Kleinrock and Naylor [51].

c) Experiments on user performance in presence of subnet protocols: Here we want to study the performance of a Host-to-Host (more generally, user to user) connection across a packet network. Typical examples of such experiments include the evaluation of the quality of digitized speech transmitted through a packet network, and the evaluation of Host-to-Host protocols [6].

d) Experiments on nodal processor performance: We are interested in determining throughput, delay and reliability characteristics of a nodal processor which supports given protocols. This set of experiments differs from the subnet protocol experiments in a) in that we want to assess the limitations of hardware and software implementation of the protocols, rather than the intrinsic limitations of the protocols themselves. Clearly, in an operational network, the subnet performance will depend both on protocol design and hardware and software implementation. An example of experiments aimed at the nodal processor evaluation is the series of throughput tests performed on the ARPANET IMP [71].

Undoubtedly, the majority of packet network experiments reported so far in the literature fall under category a). This is explained by the fact that the packet technology is still young (especially in the satellite and ground radio areas) and there are many performance issues to be clarified at the internal network protocol level before considering higher level protocols. Generally, internal subnet performance is better studied in a controlled traffic environment (i.e., artificial traffic generators) rather than in a real traffic environment. There is a growing interest, however, in the experimental evaluation of end-to-end performance as seen by the user and of the interaction of subnet protocols and higher level protocols (i.e., type c) experiments). The recent experimental networks are promoting both subnet and end-to-end experiments. In SATNET, for example, one of the important experimental issues is the performance of the TCP protocol (a Host-to-Host level protocol) [10] in presence of different channel access schemes. In this case, TCP experiments are carried out using specialized measurement facilities in the Hosts.

User behavior (type b)) and nodal processor performance (type d)) are generally given more attention in an operational environment than in an experimental environment. In fact, the main goal of an experimental network is to demonstrate the validity of a new communications concept. This evaluation is usually carried out independent of the actual hardware implementation of the nodal processor, and assumes very general user traffic characteristics. (Clearly, after having verified the validity of the communications scheme, some analysis of the actual user behavior, and a rigorous testing of the communications hardware and software is necessary before committing to a full scale operational net).

In contrast to experimental networks, the main goal of an operational network is to provide reliable service to its users. Therefore, the network manager must constantly monitor the trends in the user load profile and must be aware of the hardware and software limitations of the existing equipment in order to plan appropriate system expansions/modifications.

2) Measurement Goals: As we mentioned earlier in this section, measurements in an experimental network are motivated by one or more of the following goals:

a) Software verification: Although the implementor generally performs a systematic checkout of each software component before release, it is advisable that the experimenter test the software after field installation, to make sure that it operates according to the specifications.

b) Performance evaluation and verification: System performance is evaluated to determine whether the system meets the original design goals, and eventually to identify the applications for which the system can be effectively used. Measured performance parameters are compared with modeling results to verify the validity of the models.

c) Feedback for system design iterations: In a complex system, some of the parameters which affect performance may be properly tuned only using experiments on the real system, since analytical and simulation models are too limiting and do not possess the required accuracy.

d) Study of user behavior and characteristics: In their simplest form, user behavior experiments are intended to assess important user attributes (e.g., traffic pattern; degree of burstiness etc.) that may impact network design. In some cases, user behavior may also be monitored to determine the effect of network protocols on user characteristics. In the SATNET packet speech experiment, for example, one of the issues currently being investigated is the effect of long satellite delays (which vary according to the access scheme) on speech statistics and, more generally, on user behavior in dialogues as well as conferencing situations [43].

3) Performance Measures: The performance measures of a system (or part of a system, or of a procedure) are those parameters, or sets of parameters which best characterize and quantify the performance of the system in a real operating environment, and which permit comparison of the system with other systems performing similar functions.

The traditional performance measures used in most data communications models and experiments are the average delay and the average throughput (under a given offered traffic pattern). These measures are always available, in one form or another, from the CUMSTATS messages and probably offer the most straightforward means of evaluation and comparison of different systems, without requiring the detailed knowledge of the internal mechanisms. For these reasons we may refer to throughput and delay as the basic measures in the set of possible performance criteria. The average delay measure may of course be refined by introducing histograms (in addition to mean values). This feature, however, often proves to be very costly in terms of nodal processor requirements, and is rarely implemented. An exception to the general rule is offered by SATNET, in which the capability to measure delay distributions was deemed essential for the thorough evaluation and understanding of priority and delay class disciplines [42]. In SATNET, however, to limit storage and processor overhead, the experimenter is required to specify for each experiment the appropriate range and quantization subintervals in which histograms must be collected.

Delay and throughput, although conceptually simple to define, are not always so straightforward to measure. For example, end-to-end delay cannot in general be measured in a distributed network due to the lack of clock synchronization among the nodes. The common substitute for end-to-end delay is the round-trip delay, i.e., the interval between transmission of the data packet and reception of the acknowledgment from the destination. This approach is currently used in ARPANET. Clearly, the round-trip measurement divided by two provides only an approximation to the one-way delay measurement, since data packets have different length and (possibly) priority than acknowledgment packets, and some additional delay may be incurred at the destination before returning the acknowledgment.

A more accurate measurement of one-way delay may be obtained with the Pick-up packet. Each packet is stamped with entry and exit time at each intermediate node. The total delay is then the sum of the times spent at each node (which are reported in the pick up packet) and the transmission times on each hop (which can be accurately computed). Some imprecision, however, may still exist due to time gaps between the actual arrival/departure of a packet and the time stamps.

An exact measure of one-way delay is possible only with time-synchronized nodes. This is generally the case of the nodes in a packet satellite network, which must be synchronized in order to properly schedule packet transmissions to the multiple access channel (an exception, of course, is the case of the pure-ALOHA satellite system, in which no synchronization is required). In particular, the nodes in SATNET have synchronous clocks and therefore permit exact one-way delay measurements [33].

Some complications may also arise during throughput measurements, especially if throughput is measured as the sum of all packets successfully received at the destination. In this case, the measured throughput will include also duplicate packets (generated by the network because of missed acknowledgments, and therefore will be higher than the actual throughput seen by the user. Duplicate detection is, of course, performed by user level protocols and, in some cases, by network protocols (e.g., ARPANET). However, artificial traffic generators and sinks in the network are generally not equipped with duplicate detection protocols. Thus the experimenter has to carefully filter out duplicates from the measured throughput using some additional information (e.g., measured channel errors etc.).

The basic delay and throughput measures have some limitations, especially in those experiments aimed at gaining insight into a specific procedure. In such cases, the basic measures must be complemented with specialized measures. A typical example of specialized measure is the count of the number of times a loop is detected on a path from origin to destination: this measure is an essential complement to average throughput and delay in the evaluation of adaptive routing policies [75]. Similarly, in the PRNET the number of additional transmissions beyond success incurred by a packet is used to measure the efficiency of the Echo Acknowledgment protoc ' [87].

Besides basic and specialized measures, the is another category of measures which reflect some global system properties not easily characterized solely by total throughput and delay. These measures generally require the collection of a set of different measurements during a properly designed experiment and therefore may be referred to as composite measures. A typical example of composite measure is "fairness." Fairness is the property of allocating network capacity among an arbitrary population of large and small users (i.e., users with large and small traffic requirements) in a fair manner, without favoring one class of users over another. One of the possible definitions of fairness, proposed in [59], verifies the condition that small users get a share of capacity equal to their demand, while large users are all given the same allocation (which is larger than the allocation of any small user). The boundary between large and small users is determined by the condition that the sum of the individual allocations be equal to the maximum network capacity. Clearly, the total throughput measure alone is not a sufficient representation of fairness, since a protocol may be efficient and yet unfair. For a better appraisal of fairness one has to investigate the ratios between offered rate and effective throughput for each user in a carefully designed experiment. Other examples of composite performance measures are: congestion protection, stability, robustness of network algorithms to line errors, and reliability of a network configuration with respect to component failures.

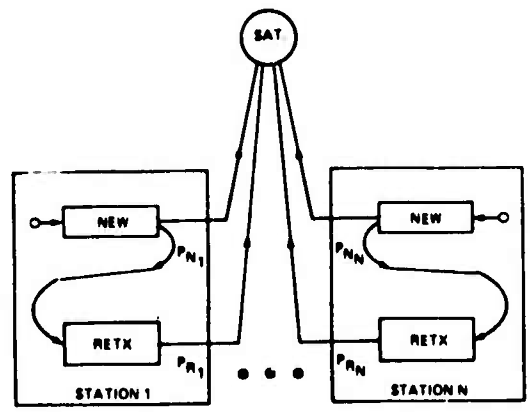


Fig. 6. S-ALOHA access protocol.

4) Designing an Experiment: The design of the experiment is probably the most critical and delicate phase of the experimentation process. Once the measurement object and the goals are defined, we must identify a meaningful set of performance measures consistent with our goals and with the tools at our disposal. Experience shows that a bad choice of performance measures and measurement facilities (e.g., inadequate traffic pattern; system parameters chosen outside of the range of interest) may produce results irrelevant to the original goals and cause a frustrating waste of efforts and resources. The design of experiments should be carefully conducted and, if possible, guided by analytical and simulation models.

To illustrate the various phases of the experiment design process we report here the highlights of the S-ALOHA control experiment carried out on SATNET in mid-1977 [59]. First, a brief description of the S-ALOHA access protocol is in order.

In the S-ALOHA channel access protocol each station maintains two output queues as shown in Fig. 6. The new queue (for new packets); and the retransmit queue (for packets that need to be transmitted because of previous conflict). All stations follow the same rules for transmission: at the beginning of a slot the station will transmit a packet from the retransmit queue with probability *PR* (retransmit gate). Only if the retransmit queue is empty, will the station then transmit a packet from the new queue with probability *PN* (new gate). If two or more stations transmit in the same slot, their packets will collide and will mutually destroy each other. The senders detect the conflict by monitoring the channel after a round-trip time and promptly return a copy of the collided packet to the retransmit queue.

It can be easily shown that if several stations have data to send at the same time and the ALOHA gate values are not properly adjusted, the system may become congested, i.e., total throughput in the system may reduce to zero [56]. The congestion situation will persist for a prolonged period of time even if the external traffic sources are removed. To overcome this congestion problem, a distributed stability control algorithm was implemented in SATNET [31]. The algorithm dynamically controls the ALOHA gates based on channel load observations, and implements a set of gate values which are optimal for the current traffic problem. One of the objectives of the S-ALOHA experiment in SATNET was to evaluate the performance of the controls, namely their performance at steady state and their ability to prevent congestion. Regarding

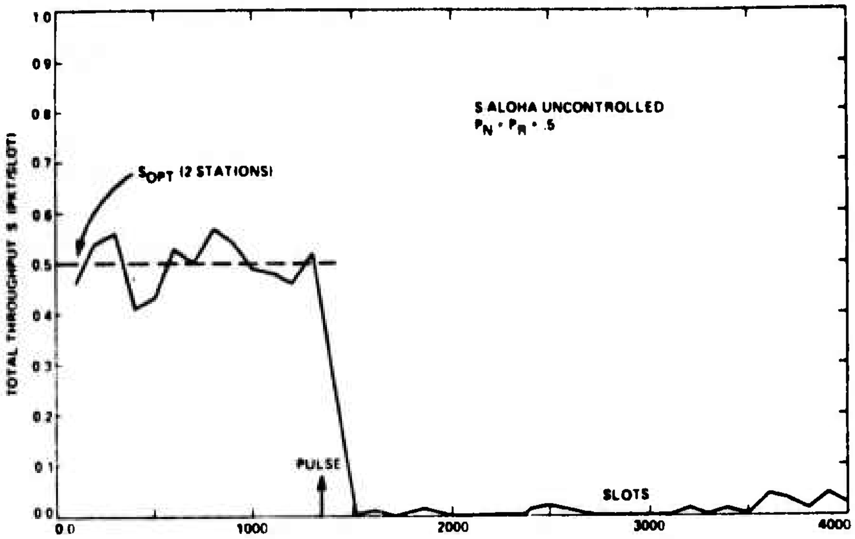


Fig. 7. S-ALOHA (uncontrolled) measurements. 10 stations.

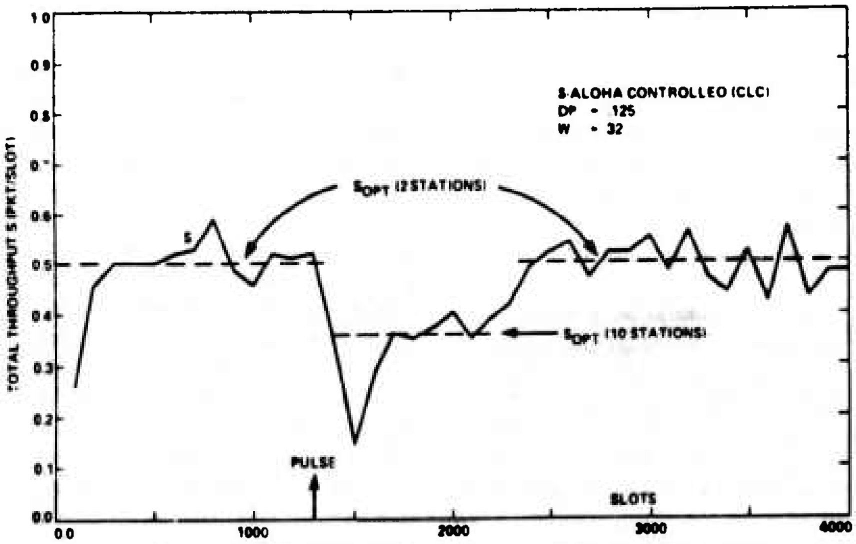


Fig. 8. S-ALOHA (controlled) measurements. 10 stations.

the network configuration, it was soon recognized that a large user population was essential in a stability experiment. Therefore, a variety of test configurations with number of stations varying from 5 to 30 were defined. As explained in Section III-A, only a few of these stations were real stations, while the remaining stations were emulated in the software of the real stations (fake stations). The best traffic environment to probe stability was found to be the superposition of a stable background pattern (involving only a small number of stations) and a traffic pulse of short duration, during which all stations are active. This traffic pattern would induce a sharp degradation of throughput immediately after the pulse, followed by a slow recovery to preexisting performance values.

In order to characterize stability, the following measures were chosen: average throughput before the pulse, time to recover after the pulse, and average throughput during the recovery period. The rationale behind this choice was the need to monitor performance both at steady state and during the recovery period, since both situations are of great concern to the user. Clearly, a simple measure of throughput averaged over the entire duration of the experiment would be grossly inadequate.

Another important set of decisions in the planning of the experiment involved the selection of the stability control parameters. These were chosen on the basis of existing analytical and simulation results [31] as well as previous experiments carried out in steady traffic conditions.

The main results of the experiments are summarized in Figs. 7 and 8 [59]. Fig. 7 shows the measured throughput performance S (packets/slot) as a function of time t (in slots) for a 10 station, uncontrolled S-ALOHA system, with fixed gates $PN = PR = 0.5$. The traffic pattern is the sum of a steady load consisting of two stations active all the time with rate $R = 1$ pkt/slot, and a pulse pattern of 20 slot duration, in which all stations are active. From Fig. 7, we notice that the average throughput before the pulse is $S = 0.5$, i.e., the theoretical optimum in a 2 station system. The introduction of the pulse, however, causes the throughput to collapse to zero, confirming the well known tendency of uncontrolled systems to become unstable.

Fig. 8 shows the measured throughput performance for a controlled S-ALOHA system using the same traffic environment as in the previous experiment. Recovery from the effect of the pulse is completed in the controlled system in 1000 time slots (analytical calculations also show that recovery in the controlled system would have required about 1000 time slots, while recovery in the uncontrolled system would have required on the order of 100 000 slots!). Performance before the pulse and after recovery (steady state) is nearly optimal.

The above experiments, and similar ones, led to important quantitative conclusions regarding the need for controls in S-ALOHA systems and the performance of specific control procedures implemented in SATNET. The success was not in little part due to the careful design of the experiments, namely, the appropriate definition of the performance measures, and the perfect matching of tools and measures to achieve the proposed goals.

We mentioned earlier that a successful network experiment is usually the result of carefully conducted modeling, simulation, and measurement activities. Analytic models assist in the specification of the experiment and the selection of parameter ranges; simulation is an extension of analysis in that it validates analytic models and permits the evaluation of analytically intractable protocols; measurements complete the experimental cycle.

The Atlantic Pachet Satellite Experiment was no exception. Analytical models for various channel access protocols were derived [33]. When analysis failed, simulation was used. In particular, the closed loop control mechanism proposed for the stabilization of the S-ALOHA protocol was tested extensively via simulation before implementation and before measurement planning. Finally, measurements were performed on the system. By this time, the experimenter already had a good idea of the results that he should expect, since the same experimental configuration had been previously evaluated via analysis or simulation. This proved to be very rewarding, and permitted tracing many of the discrepancies which arose between models and measurements back to their origin—typically, a wrong setting of the parameters, or a bug in the measurement software. More rarely, the discrepancies were found to stem from a different interpretation of the protocols by the implementer and the experimenter, respectively. This generally stimulated a productive discussion and reevaluation of the protocol among the working groups involved, leading to the adoption of the best alternative.

IV. Open Areas and Conclusion

Many advances in modeling and measurements of packet-switched networks have been made since this concept emerged in the late sixties. We have described in this paper many of the basic techniques and illustrated their use by calling on specific examples. It is clear, however, that we are far from having answered all our needs. In the following we briefly discuss some open areas, just to name a few, where more work is in order.

Random access has been thoroughly studied in single-hop environments. The performance analysis of multiaccess schemes in (the more interesting) packet radio multihop environments has proven to be a rather difficult task; no simple model exists yet for this more general problem, nor is there any obvious way to translate the results obtained in single-hop systems to multihop environments. The only analysis work relating to this that has already appeared can be found in [89], [91], [92].

In addition to the many design problems (topological, capacity assignment, routing, etc.—), network behavior is believed to be greatly affected by the flow and congestion control algorithms in use; the modeling and analysis of these techniques are still in their infancy. Perhaps the most elaborate work so far achieved consists of the analysis of a dynamic decision process relating the number W of messages outstanding in the network to the destination buffer occupancy in an environment where the changes in W do not affect the network response time [48]. Unfortunately, such limitations in the model render it only a first approximation. In sum, there is a definite shortage of analytic work in this area. The measurement of end-to-end protocols has been also extremely limited due to the difficulty of synchronizing end devices and their measurement facilities, and the difficulty in interpreting the results which depend not only on the particular protocol implementation, but also on the characteristics of the communication subnet [6].

Although in this paper we have uniquely focused on packet switched networks suitable mainly for computer-to-computer communications, we observe today an important trend towards the design of integrated packet-circuit switching communications to satisfy a broader class of users with a large variety of traffic characteristics (interactive data, long files or facsimile, real time data such as digitized voice, etc.—). The design and analysis of such systems are only at their start.

Finally, we get to the problem of network interconnection. It is all too evident that the behavior of a communication network varies with its type (land based, ground radio, satellite) as well as with the specific implementation and control techniques used. The interconnection of networks exhibits the need for a simple and accurate characterization and classification of these networks and for the development of analytic tools which help predict the performance of various interconnection topologies. Moreover, measurement facilities which allow for coordination, control and collection of simultaneous measurements in several interconnected networks in view of future internetworking experiments will be of utmost importance. These experiments include among others the evaluation of internetwork protocols and the end-to-end user performance in a multinetwork environment.

Thus we conclude here that in this exciting area of modeling and measurements of data communication networks, we are still faced with many problems of the most challenging kind.

References

[1] 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.

[2] N. Abramson, "The Aloha system," *Computer Commun. Networks*, N. Abramson and F. Kuo, Eds. Englewood Cliffs, NJ: Prentice Hall, 1973.

[3] A. V. Aho, J. E. Hopcroft, and J. D. Ullmen, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison Wesley, 1974.

[4] P. Baren, "On distributed communications," Rand Corporation, Santa Monica, CA, Rand Series Reports, Aug. 1964.

[5] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. Assoc. Comp. Mach.*, vol. 22, no. 2, pp. 248-260, April 1975.

[6] C. J. Bennett and A. J. Hinchley, "Measurements of the transmission control protocol," in *Proc. Workshop on Computer Network Protocols* (Liege, Belgium), Feb. 1978.

[7] U. N. Bhat, "Sixty years of queueing theory," *Management Sci.*, vol. 15, pp. B280-B294, 1969.

[8] R. Binder, "A dynamic packet-switching system for satellite broadcast channels," in *Proc. Int. Communications Conf.*, pp. 41-1-41-5, June 1975.

[9] P. J. Burke, "The output of a queueing system," *Operations Res.*, vol. 4, pp. 699-704, 1956.

a) J. Buzen, "Queueing network models of multiprogramming," Ph.D. dissertation, Division of Engineering and Applied Science, Harvard Univ., Cambridge, MA, 1971.

[10] V. Cerf and R. E. Kahn, "A protocol for packet network intercommunication," *IEEE Trans. Commun.*, vol. COM-22, pp. 637-648, May 1974.

a) W. W. Chiu and W. Chow, "A hybrid hierarchical model of a multiple virtual storage (MVS) operating system," IBM T. J. Watson Research Center, Yorktown Heights, NY, Research Rep. R. C. 6947, Jan. 1978.
b) W. W. Clipshaw and F. Glave, "Datapac network review," in *Int. Comput. Commun. Conf. Proc.*, pp. 131-136, Aug. 1976.

[11] J. W. Cohen, *The Single Server Queue*. New York: Wiley, 1969.

[12] G. D. Cole, "Performance measurements on the ARPA computer network," *IEEE Trans. Commun.*, vol. COM-20, pp. 630-636, June 1972.

[13] D. R. Cox and W. L. Smith, *Queues*. Chapman and Hall Monographs on Applied Probability and Statistics, 1961.

[14] M. A. Crane and D. I. Iglehart, "Simulating stable stochastic systems, I: General multiserver queues," *J. Assoc. Comp. Mach.*, vol. 21, no. 1, pp. 103-113, Jan. 1974.

[15] —, "Simulating stable stochastic systems, II: Markov chains," *J. Assoc. Comp. Mach.*, vol. 21, no. 1, pp. 114-123, Jan. 1974.

[16] 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 Sym. Operating System Principles. Gatlinburg, TEN, Oct. 1-4, 1967.

[17] *A Selection of Scientific Quotations*. Collected by A. L. Mackay (M. Ebison, Ed.).

[18] 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.

[19] G. Fayolle, E. Gelembe, and J. Labetoulle, "Stability and optimal control of the packet-switching broadcast channels," *J. Assoc. Comp. Mach.*, vol. 24, no. 3, pp. 375-386, July 1977.

[20] W. Feller, *An Introduction to Probability Theory and Its Applications, Volume 2*. New York: Wiley, 1966.

[21] G. S. Fishman, "Statistical analysis for queueing simulations," *Management Sci.*, vol. 20, no. 3, pp. 363-369, 1973.

[22] —, "Estimation in multiserver queueing simulations," *Operations Res.*, vol. 22, no. 1, pp. 72-78, 1974.

[23] —, "Achieving specific accuracy in simulation output analysis," *Commun. Assoc. Comp. Mach.* vol. 20, no. 5, pp. 310-314, May 1977.

[24] H. Frank, I. T. Frisch, and W. Chou, "Topological considerations in the design of the ARPA network," in *Proc. AFIPS Conf. (SJCC)*, vol. 36, pp. 581-587, 1970.

[25] H. Frank, R. E. Kahn, and L. Kleinrock, "Computer communication network design: Experience with theory and practice," *Networks*, vol. 2, no. 2, 1972.

[26] L. Fratta *et al.*, "The flow deviation method—An approach to store and forward communications network design," *Networks*, vol. 3, 1973.

[27] G. L. Fultz, "Adaptive routing techniques for message switching computer communications networks," UCLA School of Engineering, Rep. No. 7252, July 1972.

[28] M. Gerla, "The design of store-and-forward (S/F) networks for computer communications," Computer Science Dep., School of Engineering and Applied Science, Univ. California, Los Angeles, Tech. Rep. UCLA-ENG-7319, Jan., 1973. (Also published as Ph.D. dissertation.)

[29] M. Gerla, W. Chou, and H. Frank, "Cost-throughput trends in computer networks using satellite communications," presented at ICC 74, Minneapolis, MN, June 1974.

[30] M. Gerla, H. Frank, W. Chou, and J. Eckly, "A Cut-Saturation Algorithm for Topological Design of Packet Switched Communications Networks," in *Proc. Nat. Telecommunications Conf.* (San Diego, CA), Dec. 1974.

[31] M. Gerla and L. Kleinrock, "Closed loop stability controls for S-ALOHA satellite communications," *Data Communications Symp.*, Snowbird, UT, Sept. 1977.

[32] —, "On the topological design of distributed computer networks," *IEEE Trans. Commun.*, COM-25, Jan. 1977.

[33] M. Gerla *et al.*, "Packet satellite multiple access: Models and measurements," in *NTC Conf. Proc.* (Los Angeles, CA), Dec. 1977.

[34] W. J. Gordon and G. F. Newell, "Closed queuing systems with exponential servers," *Operations Res.*, vol. 15, pp. 254-265, 1967.

[35] J. L. Grange and P. Mussard, "Performance measurements of line control protocols in the Cigale network," in *Proc. Workshop on Computer Network Protocols* (Liege, Belgium), Feb. 1978.

[36] R. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: M.I.T. Press, 1960.

[37] —, *Dynamic Probabilistic Systems, Vol. I: Markov Models, Vol. 2: Semi-Markov and Decision Processes*. New York: Wiley, 1971.

[38] H. Inose and T. Saito, "Theoretical aspects in the analysis and synthesis of packet communication networks," this issue, pp. 1409-1422.

[39] M. Irland, "Simulation of Cigale 74," in *Proc. 4th Data Communications Symp.* (Quebec City, P.Q., Canada), Oct. 1975.

[40] —, "Queueing analysis of a buffer allocation scheme for a packet switch," in *Proc. IEEE National Telecommunications Conf.* (New Orleans, LA), Dec. 1975.

[41] J. R. Jackson, "Job shop-like queueing systems," *Management Sci.*, vol. 10, pp. 131-142, 1963.

[42] I. Jacobs *et al.*, "C-PODA—A demand assignment protocol for SATNET," presented at *5th Data Communication Symp.*, Snowbird, UT, Sept. 1977.

[43] I. M. Jacobs, R. Binder, and E. V. Hoversten, "General purpose packet satellite networks," this issue, pp. 1448-1467.

[44] 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.*, COM-25, Jan. 1977.

[45] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," this issue, pp. 1468-1496.

[46] F. Kamoun, "Design considerations for large computer communication networks," Ph.D. dissertation, Computer Science Dep., Univ. California, Los Angeles, Apr. 1976.

[47] F. Kamoun and L. Kleinrock, "An analysis of shared storage in a computer network environment," in *Proc. 8th Hawaii Int. Conf. System Science* (Honolulu, HI), pp. 89-92, Jan. 1976.

[48] P. Kermani, "Switching and flow control techniques in computer communication networks," Ph.D. dissertation, Dep. Computer Science, Univ. California, Los Angeles, 1977.
a) L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*. (New York: McGraw-Hill, 1964 (out of print). Reprinted by Dover Publications, 1972).

[49] L. Kleinrock, "Analytic and simulation methods in computer network design," in *AFIPS Conf. Proc. (SJCC)*, vol. 36, pp. 569-579, 1970.

[50] L. Kleinrock and S. Lam, "Packet-switching in a slotted satellite channel," in *Nat. Computer Conf. AFIPS Conf. Proc.*, vol. 42. Montvale, NJ: AFIPS Press, 1973, pp. 703-710.

[51] L. Kleinrock and W. E. Naylor, "On measured behavior of the ARPA network," in *AFIPS Conf. Proc. 1974, National Computer Conf.*, vol. 43, pp. 767-780, 1974.

[52] L. Kleinrock and H. Opderbeck, "Throughput in the ARPANET—Protocols and measurements," in *Proc. 4th Data Communications Symp.* (Quebec City, P.Q., Canada), Oct. 1975.

[53] L. Kleinrock, *Queueing Systems, Volume I: Theory*. New York: Wiley Interscience, 1975.

[54] 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.

[55] 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, Dec. 1975.

[56] 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.

[57] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*. New York: Wiley Interscience, 1976.

[58] L. Kleinrock, W. E. Naylor, and H. Opderbeck, "A study of line overhead in the ARPANET," *Commun. Assoc. Comp. Mach.*, vol. 19, no. 1, pp. 3-12, 1976.

[59] L. Kleinrock and M. Gerla, "On the measured performance of packet satellite access schemes," presented at *Int. Conf. on Computer Communications*, Kyoto, Japan, Sept. 1978.

[60] H. Kobayashi and A. G. Konheim, "Queueing models for computer communications system analysis," *IEEE Trans. Commun.*, vol. COM-25, pp. 2-29, Jan. 1977.

[61] A. G. Konheim and B. Meister, "Service in a loop system," *J. Assoc. Comp. Mach.*, vol. 19, no. 1, pp. 92-108, Jan. 1972.

[62] J. Labetoulle, E. Manning, and R. W. Peebles, "A homogeneous computer network: Analysis and simulation," *Computer Networks*, vol. 1, no. 4, pp. 225-240, 1977.

[63] S. S. Lam, "Packet switching in a multi-access broadcast channel with application to satellite communication in a computer network," Ph.D. dissertation, Dep. Computer Science, Univ. California, Los Angeles, Mar. 1974; also in Univ. California, Los Angeles, Tech. Rep. UCLA-ENG-7429, Apr. 1974.

[64] 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.

[65] S. Lam and M. Reiser, "Congestion control of store-and-forward networks by input buffer limits," in *NTC 77 Conf. Proc.* (Los Angeles, CA), Dec. 1977.

[66] A. Lavia and E. Manning, "Topological optimization of packet switched data communications networks," *CCNG Tech. Rep. No. E-35*, Oct. 1975, CCNG, Univ. Waterloo, Waterloo, Ont., Canada.

[67] S. Lin, "Heuristic programming as an aid to network design," *Networks*, Jan. 1975.

[68] J. Little, "A proof of the queueing formula $L = \lambda W$," *Operations Res.*, vol. 9, pp. 383-387, Mar.-Apr. 1961.

[69] M. T. Liu and C. C. Reames, "Communication protocol and network operating system design for the distributed loop computer

network (DLCN)," *Proc. 4th Annu. Symp. Computer Architecture*, pp. 193-200, Mar. 1977.
[70] M. T. Liu, "Distributed loop computer networks," (to appear in *Advances in Computer Networks*, M. Rubinoff and M. C. Yovits, Eds. New York: Academic Press, 1978.)
[71] J. M. McQuillan, W. R. Crowther, P. P. Cossell, D. C. Walden, and F. E. Heart, "Improvements in the design and performance of the ARPA network," in *AFIPS Conf. Proc. 1972 Fall Joint Computer Conf.*, vol. 41, pp. 741-754, 1972.
[72] R. M. Metcalfe, "Steady-state analysis of a slotted and controlled ALOHA system with blocking," in *Proc. 6th Hawaii Int. Conf. System Sciences*, Univ. Hawaii, Honolulu, Jan. 1973.
[73] R. M. Metcalfe and D. R. Boggs, "ETHERNET: distributed packet switching for local computer networks," *Commun. Assoc. Comp. Mach.*, vol. 19, no. 7, pp. 395-403, 1976.
[74] F. R. Moore, "Computational model of a closed queueing network with exponential servers," *IBM J. Res. Dev.*, vol. 16, pp. 567-572, 1972.
[75] W. E. Naylor, "A loop-free adaptive routing algorithm for packet switched networks," in *Proc. 4th Data Communications Symp.* (Quebec City, P. W., Canada), pp. 7-9 to 7-14, Oct. 1975.
a) H. Opderbeck and R. B. Hovey, "Telnet-Network features and interface protocols," in *Proc. NTG-Conf. Data Networks* (Baden-Baden, West Germany), Feb. 1976.
[76] L. Pouzin, "Presentation and major design aspects of the cyclades computer network," presented at Datacom 73, ACM/IEEE, 3rd Data Communications Symposium. St. Petersburg, FL, pp. 80-87, Nov. 1973.
a) L. Pouzin, "CIGALE, the packet switching machine of the CYCLADES computer network," presented at IFIP Congress (Stockholm), pp. 155-159, Aug. 1974.
[77] W. L. Price, "Data network simulation: Experiments at the national physical laboratory 1968-1976," *Computer Networks*, vol. 1, no. 4, pp. 199-210, 1977.
[78] J. Riordan, *Stochastic Service Systems*. New York: Wiley, 1962.
[79] L. G. Roberts and B. D. Wessler, "Computer network developments to achieve resource sharing," in *Proc. AFIPS Conf.* (*SJCC*), vol. 36, pp. 543-549, 1970.
[80] L. G. Roberts, "Dynamic allocation of satellite capacity through packet reservation," in *1973 Nat. Comput. Conf., AFIPS Conf. Proc.*, Vol. 42. New York: AFIPS Press, 1973, pp. 711-716.
[81] L. G. Roberts, "ALOHA packet system with and without slots and capture," *Computer Communication Rev.*, vol. 5, no. 2, pp. 28-42, Apr. 1975.
[82] S. M. Ross, *Applied Probability Models with Optimization Applications*. San Francisco, CA: Holden Day, 1970.
[83] Steiphitz *et al.*, "The design of minimum cost survivable networks," *IEEE Trans. Circuit Theory*, Nov. 1969.
[84] F. Tobagi, "Random access techniques for data transmission over packet switched radio networks," Ph.D. dissertation, Comput. Sci. Dep., School Eng. and Appl. Sci., Univ. California, Los Angeles, rep. UCLA-ENG 7499, Dec. 1974.
[85] 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.
[86] F. Tobagi and S. E. Lieberson, "Measurements in packet radio systems: Methods for collection of cumstat data," *Packet Radio Temporary Note* #175, Comput. Sci. Dept., Univ. California, Los Angeles, Mar. 1976.
[87] F. Tobagi *et al.*, "On measurement facilities in packet radio systems," in *Nat. Computer Conf. Proc.* (New York), June 1976.
[88] 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.
[89] F. Tobagi, "Performance analysis of packet radio communication systems," in *Proc. IEEE Nat. Telecom. Conf.* (Los Angeles, CA), pp. 12:6-1 to 12:6-7.
[90] 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.
[91] F. Tobagi, "On the performance analysis of multihop packet radio systems: Parts I-IV," Packet Radio Temporary Notes #246-249, Comput. Sci. Dept., Univ. California, Los Angeles, 1978.
[92] F. Tobagi, "Analysis of slotted ALOHA in a centralized two-hop packet radio network," in *Proc. 17th IEEE Computer Society Int. Conf.* (CompCon, Fall 78), Washington, DC, Sept. 1978.
[93] A. C. Williams and R. A. Bhandiwad, "A generating function approach to queueing network analysis of multiprogrammed computers," Mobil Oil Corp., Princeton, NJ, Apr. 1974.
[94] J. W. Wong, "Distribution of end-to-end delay in message-switched networks," submitted for publication, Dep. Computer Science, Univ. Waterloo, Waterloo, Ont., Canada.

PRTN # 237, Rev. 1 September, 1978

Pickup Packet

Format, Handling and Generation

Stan Lieberson, UCLA

This note describes the generation, format and handling of the Pickup Packet by TIUs, PRUs and the station.

Introduction

The Pickup Packet [PP] is a part of the overall measurement facility in the PRNET. The function of pickup packets is to gather at selected nodes information regarding their passage through that node. The information collected at the nodes is stored in the text area of the pickup packets themselves; when the packets have completed their journey, they are then sent on a measurement connection to the station to be entered into the measurement file. As a result, a pickup packet can exist in one of two forms: as a pickup packet in the process of collecting data about its source-to-destination journey, and as an ex-PP that is on its way to the station from its original destination and no longer collecting nodal information. Ex-PPs will be sent on the TIU-station measurement process connection and the type code in their header will be the same as any measurement packet from a TIU to the station measurement process. Previous to that, however, while it is an active PP collecting nodal data, the type code in the header will indicate that it is a pickup packet. The three-bit type code to be used to identify a pickup packet is to be selected by the implementors.

Contents of the Pickup Packet

The Pickup Packet has three parts: header, initial text, and nodal text. The <u>header</u> contains the type code identifying it as a pickup packet. When the pickup packet has reached its destination

TIU and is being rerouted to the station by that destination TIU, its type will be changed by the TIU to 011 (information packet) and sent on the TIU-station measurement connection.

The initial text constitutes the first six text words and is defined at the end-device that generated the pickup packet. The first word is a unique pickup packet sequence number, set by the TIU. The next word is a pointer, for use by the PRUs, indicating the next free area of the PP text available for storage of nodal data; the next four words are for two time-stamps of two words each: the time the PP left its source TIU, and the time it arrived at its destination TIU. The remainder of the packet text is available for the nodal data itself. The following information is to be recorded at each PRU, if the PP function is enabled at that PRU, and stored into the text of each PP that is received:

- the PRU "hardwire" id
- 2-word PRU clock time of packet rx
- one word with fields to indicate the number of packets in each of the following tx queues: active ack; radio tx; S/T tx
- the previous PP's sequence number
- 2-word clock time of recognition (not rx) of prev. PP's HBH ack
- time this PP placed on the tx queue
- number of tx's for this PP
- clock time for tx # 1
- (clock time for tx # 2)

(etc)

Several notes must be made at this point.

1. The queue time and tx clock times of the pickup packet being processed are not specified above as being two words long. While they may be two words, they may also be a one-word displacement from the 2-word pickup packet rx clock time already captured. A one-word displacement will reduce the pickup packet text space per node, resulting in a longer life for pickup packets. The cost of this savings is the necessity to perform the subtractions in the PRUs. Unless these 2-word subtractions can be done quite efficiently, the cost is probably not worth the text space.

2. Because a PP cannot contain within itself the time its own HBH ack was received at a PRU, this information is temporarily stored in the PRU, as data from the "previous PP". This "prev. PP" data will be stored in the next PP encountered.

3. Likewise, a pickup packet cannot contain within itself the time of its last transmission. Two solutions can be considered: One is to store a PP's number of tx's, and the associated tx times, in the same area as its HBH ack time (the "prev. PP" buffer), so that a PP will carry its predecessor's HBH time AND tx times. An advantage of this method is that when a pickup packet arrives, it can immediately know if sufficient room exists within its

remaining text area to store the data from this node. This is because the data to be stored is of known length, i.e. it will not vary with the number of retransmissions of this PP, since retransmission data is stored in the "prev. PP" buffer in the PRU. Another solution is to save only the latest tx time in the "prev. PP" buffer, and before each retransmission, include within the packet's text the time of the previous tx attempt (and update the pointer).

With this method, one must either allow space in the PP text for the maximum number of tx's (which in practice will rarely be reached), thus declaring as "insufficient" some pickup packets that could still accomodate that hop, or else one must check the remaining space before each transmission. The advantage of this method is that, by storing only the last tx clock time, the PRU's "prev. PP" buffer is several words smaller than if it had to save several tx clock times. Determination of an appropriate data storage method will be made by the implementors.

Enabling and Disabling the PRU's Pickup Packet Function

To allow for better experimenter control and to make more flexible the path over which pickup packet data may be collected, PRUs will not always store nodal data in pickup packets they receive. That is, their pickup packet function can be either enabled or disabled; when disabled, any pickup packets encountered will be treated simply as information packets. Only when the pickup packet function has been enabled will the FRU respond by collecting and storing nodal data. The enabling and disabling will be controlled by the station measurement process via an SIP control packet, in much the same manner as it controls the enabling and disabling of FRU cumulative statistics, with the distinction that enabling the PP function will not require that a connection remain open until the function is disabled. The default mode of the bit or field indicating the status of the PP function in a FRU will be the "disabled" mode.

TIU Handling of Pickup Packet Streams

TIUs as Pickup Packet Sources. A TIU will generate pickup packets on command from the station measurement process in much the same way as it generates information packets: the command from the station contains the information needed by the TIU to define the packet stream (packet length, destination, generation process, window size, etc). When generating PPs, an additional parameter need be included: the initial internal PP sequence number. And of course one must have indicated that the stream is to consist of pickup packets rather than information packets.

When generating the stream of pickup packets, the TIU must increment the PP sequence number, resetting it to 1 when it reaches 7FFF. [The high-order bit is reserved, as explained below.]

TIUs as Pickup Packet Sinks. TIUs acting as pickup packet sinks have three functions to perform, none of which involve discarding the pickup packet! The first function is to store the arrival time of the pickup packet in the two words reserved for that purpose in the "initial text" area. The second is to set the leftmost bit of the pickup packet sequence number on. This is to insure that a low sequence number, being located in the first text word of that packet, will not be mistaken for a TIU measurement packet type code when the packet is entered into the station measurement file. (This bit may equivalently be set by the source TIU.)

The third function is to reliably deliver the now expended pickup packet to the station. This is performed by altering the type code in the header of the pickup packet from "pickup packet" to "information" (011), and then readdressing the packet to the station, using the station's measurement SFP connection to the TIU for reliable delivery.

Station Interaction

The station measurement process must add one or two parameters to its set of parameters that define TIU traffic streams. It is necessary to tell the TIU (i) that the traffic stream is composed of PPs, and (ii) to give the TIU the initial pickup packet sequence number (0001 through 7FFF). Since it will probably be desirable to have uniform formats for control packets that define traffic streams, the station measurement process will probably always send the parameter "initial PP sequence number," regardless of the type of stream. If this is the case, then that parameter can double as the flag indicating whether the stream is to consist of pickup packets or information packets, a zero value indicating an information packet stream, a non-zero value indicating both a pickup packet s[illegible]m and the initial sequence number.

Finally, the station must have a means of enabling and disabling the pickup packet function in PRUs. This ability has two aspects: (i) a dialogue with the operator specifying which PRUs are to be set which way; and (ii) appropriate parameter packets or additions to existing CumStat parameter packets sent to the PRUs for controlling the parameter. If a new packet is used, then it should enter the measurement file under the same circumstances as CumStat measurement packets do.

15-Jan-76 14:30:54-PST,6569;000000000001
Date: 15 Jan 1976 1430-PST
From: STAHL
Subject: Attn: Jim Quilici
To: Garrett at ISI, Sunlin at ISI
cc: Klemba at SRI-KL, Kunzelman, Cerf at ISI, Chu, Tobagi

Jim,

As you will recall from the SRI meeting last month, we were concerned with the low incidence of 'TX-ING' relative to 'RX-ING' reported in the Transceiver Activity experiment, and we suggested that it may be the result of sampling bias resulting from the implementation. To test the first part (that it was caused by sampling bias), we ran an experiment this month involving only two PRUs (a station front-end and a terminal front-end), no ROF traffic, and transparent (non-SPP) terminal traffic destined to itself (via the station forwarder). The traffic generator was set with an intergeneration gap of 0.2 seconds, and during the five minute collection some 1500 packets were sent. Assuming no retransmissions (i.e. complete tx success), the Snapshot "capture" of the RX-ING state in one PRU should on the average equal the "capture" of the RX-ING state in the other PRU. (In fact, due to the absence of other traffic (except for the 60 CumStat packets in the outbound direction and their H-H and ETE acks), the RX-ING and TX-ING states in each of the PRUs should all equal eachother.) The measured results, however, showed a significantly higher capture of the RX-ING state over the TX-ING state. [RX-ING is defined as Disposition/Status code of 'CC', and TX-ING as code '9A'. All packets were at 100 KBPS.] The table below summarizes the results, with the numbers expressed as the probability of the transceiver being in the given state.

	RX-ING	TX-ING
Station	.025	.012
Terminal	.025	.019

Note that the assumption of perfect transmission success favors the higher RX-ING state, i.e. that there were in fact a small number of retransmissions should make the TX-ING state somewhat greater than the corresponding RX-ING state!

Further calculations show bias not just in the TX-ING state vis-a-vis RX-ING, but in the absolute probability of either state as well. This is the reasoning: the packets were 25 words long (header and text). Including three words of preamble and two for checksum, each PRU tx'ed 1500 30-word packets, and 1500 16-word active acks. At 100 KBPS, the following relationships exist:

```
1500 full pkts rx'd @ 4.80 ms/pkt   =  7.20 sec's
1500  act acks rx'd @ 2.56 ms/pkt   =  3.84 sec's    sum = 11.04 sec's.
```

11.04 sec's over 5 minutes means the probability of rx'ing is .0368, which is 50% larger than the measured results for RX-ING. (The same calculations are valid for TX-ING, i.e. probability should be .0368.)

The second part of the statement (that the sampling bias is the result of implementation) is suggested by the "incomplete loop" nature of the PRU mainline code. By incomplete loop I am referring to the loop structure of the code, and the fact that after servicing certain events the code transfers back to the beginning of the loop rather than continuin on around the loop. This not only means that the section of the loop, where a check occurs to see if its time for a Snapshot, is only reached under a somewhat limited set of circumstances (to whit, in the absence of events that cause a transfer back to the top of the loop), it also means that circumstances can collude to delay (via many transfers back to the top of the loop) the reaching of the Snapshot code. In fact, in this experiment the CumStat interval time was set to five seconds and the Snapshot interval to 0.4 seconds, which should result in 12 to 13 Snapshots taken for each CumStat packet sent. In fact, an average of 9-10 Snapshots were collected during this interval, which in itself constitutes a source of error in interpreting the results.

To overcome this bias, we propose a somewhat different way to collect Snapshots (which is largely taken from our "Automatic Collection" note of April, 1976).

In the current system, wherever the Snapshot-check part of the main-line loop is reached, the elapsed time since the last Snapshot is computed and compared with the Snapshot interval; if the elapsed time is larger than the interval, then a Snapshot is taken. Our proposal is to remove this loop-dependent check and to take advantage of the hardware clock overflow interrupt. When the 2 usec hardware clock overflows (every 131+ msec), the interrupt handler responds by calling the clock routine, which increments the 131+ msec tic software clock. We propose to add to this routine a single decrement of a counter, initially set to the Snapshot interval time. When the counter is decremented to zero, a Snapshot could be taken, and the counter reinitialized.

To demonstrate that this modification in fact remedies the bias problem, a retest of the above bias experiment would be run and analyzed. If this modification is in fact a solution, and no other problems develop, then it can become standard for CAP.

Below appears some IMP-16 code and comments; this is provided to clarify the modifications discussed above rather than as a specific design for implementation.

(Note that, instead of decrementing the Snapshot interval counter, SSINTC, from the value of the desired interval to zero, the efficiency of the ISZ command is used to increment the counter from the negative of the desired value up to zero.)

```
     <hardware clock overflow interrupt routine...>
     ISZ   SSINTC      ;decrement count of tics between Snapshots
     JMP   END         ;non-zero, thus not yet time for a Snapshot
     LD    Rx,SSINT    ;Snapshot time; reinit SSINTC counter
     CAI   Rx,1        ;two's complement, to make it negative (for the ISZ)
     ST    Rx,SSINTC   ;re-init'ed
     LD    R0,CSSTAT   ; its Snapshot time, but are Snapshots enabled?
     BOC   R0,END      ; nope, branch around next stmt
     JSR   @CSSNPA     ;yup, take a snapshot
END: <exit clock overflow interrupt routine>
```

Ron Kunzelman has informed us that Keith Klemba will be available to assist you, as necessary, in making a cassette tape for the PPUs with the modified code. Keith is familiar with our modification recommendation

We feel that debugging the measurement code is a priority item, and we will appreciate your devoting the necessary energy to this effort. Please inform us as soon as you can of your plan of action.

Regards,
Stan & Fouad

Proposed Procedures For

TIU Generation of Poisson Traffic Streams

Stan Lieberson

UCLA

October, 1978

It is usual to approximate user traffic with a Poisson generation process. Given a mean packet generation rate λ, the probability of generating a packet during interval k is defined by

$$Pr\{k,\lambda\} = \frac{\lambda^k e^{-\lambda}}{k!} \quad . \qquad (1)$$

This equation represents the various generation <u>rates</u>. Intergeneration <u>intervals</u>, of course, are the quantities desired for traffic generators. One can transform equation (1) so as to produce intergeneration intervals; the resulting function taking the form of an exponential distribution.

In the generation of artificial traffic, one would like to randomly sample intergeneration intervals from that distribution. This can be done by uniform random sampling from the interval (0,1) and applying those samples to the equation

$$\underline{I} = \lambda[-\ln(1-\underline{r})] \qquad (2)$$

where $\underline{r}$ is a random sample from (0,1) and $\underline{I}$ is the corresponding intergeneration interval.

Thus, the generation of packet intergeneration intervals is now broken down into two parts: (i) generate a uniformly distributed random number on (0,1) and (ii) use equation (2) to compute the intergeneration interval. This second part requires the use of natural logarithms. Actually computing such logarithms can be avoided by approximating the function $-\ln(1-\underline{r})$ with a series of line segments, and using linear interpolation of any point on a segment to approximate the function value with little loss of accuracy.

A further efficiency can be introduced by generating uniformly distributed integers on the interval (0,65535) as input to an appropriately modified equation (2) rather than generating floating-point numbers on (0,1).

Before one can begin the Poisson traffic generation, some initialization must occur. This is most properly done at the time a packet requesting a Poisson traffic stream arrives at the TIU from the station. First, one must store λ, the desired mean intergeneration time, as a floating-point value, TMEAN. This number may be sent either as the floating-point number of TIU clock tics (if the station and TIU share the same floating-point form), or as an integer number of time units, say milliseconds, in which case a conversion need be performed from integer to floating-point and from milliseconds to clock tics.

Next, a seed for the random number generator, NSEED, must be chosen. The seed is to be of the form

$$8\underline{t} + 3 \qquad (\underline{t} > 32) \ .$$

An aribitrary value for $\underline{t}$ might be selected by using the low-order 16 bits of the TIU's clock, modified as may be necessary to make it non-negative and greater than 32. Also, the next random number, NEXTR, (initially the "last" random number) must initially be set to NSEED.

Following this initialization, one is ready to produce uniformly distributed pseudo-random numbers and their corresponding Poisson intergeneration intervals. Note that the subroutine calls for these two steps can be set up such that the four traffic generators in a TIU are free to either share the same seed and pseudo-random number sequence or to each use their own seed, and in either case to generate traffic at their own mean intergeneration interval.

In words and FORTRAN code, we have the following:

Initialization:

```
    set TMEAN from station to floating-point TIU clock tics
    grab low-order 16 bits of clock, store in NSEED
    C TURN SIGN BIT OFF (SURELY DONE MORE EASILY IN ASSEMBLY CODE)
          IF (NSEED.LT.0)NSEED=(NSEED+32768)+32768
    C MAKE T>32 AND 8T+3 < 65536
          IF (NSEED.GT.8191) NSEED=NSEED-(NSEED/8191)*8191
    C THIS IS EASILY DONE IN ASSEMBLY CODE, AS IT IS JUST A MOD FUNCTION
          IF (NSEED.LT.32)NSEED=NSEED+32
    C SET THE SEED:
          NSEED=8*NSEED+3
    C INITIALIZE 'NEXTR'
          NEXTR=NSEED
```

Now, to determine the next intergeneration time:

```
CALL RANDOM(NSEED, NEXTR)
CALL POISS(NTICS,TMEAN,NEXTR)
```

where NTICS will be the number of TIU clock tics for the intergeneration interval. The subroutines are:

```
      SUBROUTINE RANDOM(NSEED,NEXTR)
      NEXTR=NSEED*NEXTR
C (IGNORE OVERFLOW, USE RESULTING MOD 65536 RESIDUAL)
      RETURN
```

Subroutine POISS includes the table used to approximate the natural log function. The table requires two columns by 48 rows; the first column contains the integers to compare with 'NEXTR', the second column contains the associated function values. Since column one contains integers and column two contains floating-point function values, I will EQUIVALENCE the table LOGTBL (48,2) with its REAL name TBLLOG(48,2) so as to be able to distinguish below between integer and floating-point arithmetic.

```
SUBROUTINE POISS(NTICS,TMEAN,NEXTR)
DIMENSION LOGTBL(48,2),TBLLOG(48,2)
EQUIVALENCE(LOGTBL,TBLLOG)
DATA LOGTBL / (see appendix A) /
search LOGTBL(*,1) using NEXTR. Should an exact
match be found, NEXTR=LOGTBL(L,1) for some entry
L, then:
```

```
VAL=TBLLOG(L,2)
TICS=TMEAN*VAL
NTICS=IFIX(TICS+.5)
RETURN
```

otherwise, we find the nearest interval around NEXTR:

LOGTBL(L1,1)<NEXTR<LOGTBL(L2,1), L1=L2-1.

now we use the linear interpolation

$$y = y_1 + (y_2 - y_1)(\frac{x-x_1}{x_2-x_1}) .$$

```
VAL=TBLLOG(L2,2)+(TBLLOG(L2,2)-TBLLOG(L1,2))*
     (FLOAT(NEXTR-LOGRVL(L1,1)) /
     FLOAT(LOGTBL(L2,1)-LOGTBL(L1,1)) )
TICS=TMEAN+VAL
NTICS=IFIX(TICS+.5)
RETURN
```

If one is willing to trade core for time, one can eliminate the relatively costly

FLOAT(LOGTBL(L2,1)-LOGTBL(L1,1))

computation by increasing LOGTBL to (48,3) with the third column defined as follows:

TBLLOG(n,3) = FLOAT(LOGTBL(n,1)-LOGTBL(n-1,1))

so that the term $(x_2 - x_1)$ in the linear interpolation becomes simply TBLLOG(L2,3).

In any case, a FLOAT and an IFIX routine must be written.

Appendix A

Values for LOGTBL

n	(n,1)	(n,2)	(n,3)
1	3277	0.0513	0.0513
2	6554	0.1054	0.0541
3	9830	0.1625	0.0572
4	13107	0.2231	0.0606
5	16384	0.2877	0.0645
6	19661	0.3567	0.0690
7	22938	0.4308	0.0741
8	26214	0.5108	0.0800
9	29491	0.5978	0.0870
10	32768	0.6931	0.0953
11	36045	0.7985	0.1054
12	39322	0.9163	0.1178
13	42598	1.0498	0.1335
14	45875	1.2040	0.1542
15	47514	1.2910	0.0870
16	49152	1.3863	0.0953
17	50790	1.4917	0.1054
18	52429	1.6094	0.1178
19	53740	1.7148	0.1054
20	55050	1.8326	0.1178
21	56361	1.9661	0.1335
22	57672	2.1203	0.1542
23	58327	2.2073	0.0870
24	58982	2.3026	0.0953
25	59638	2.4079	0.1054
26	60293	2.5257	0.1178
27	60948	2.6593	0.1335
28	61604	2.8134	0.1542
29	61932	2.9004	0.0870
30	62259	2.9957	0.0953
31	62587	3.1011	0.1054
32	62915	3.2189	0.1178
33	63242	3.3524	0.1335
34	63570	3.5066	0.1542
35	63898	3.6889	0.1823
36	64225	3.9120	0.2231
37	64553	4.1997	0.2877
38	64881	4.6052	0.4055
39	65044	4.8929	0.2877
40	65208	5.2983	0.4055
41	65307	5.6550	0.356
42	65405	6.2146	0.5596
43	65438	6.5023	0.2877
44	65470	6.9078	0.4055
45	65497	7.4186	0.5108
46	65523	8.5173	1.0987
47	65529	9.2102	0.6928
48	65535	11.0904	1.8802

References

IBM Reference Manual C20-8011. "Number Theory Background for for the Power Residue Method."

Schriber, Thomas J., Simulation Using GPSS. John Wiley & Sons, 1974.

Hemmerle, William J., Statistical Computations on a Digital Computer, Blaisdell Publishing Co., Waltham, Mass., 1967.

Date: 25 Jan 1980 1723-PST (Friday)
From: Stan at UCLA-ATS (Stan Lieberson)
Subject: ISIE Archivals
To: su

Packet Radio Files archived at ISIE <STANL> as of 11-Dec-79
==

Files beginning:	Refer to:
PRADIO	Stan's routines associated with the driver for reduction of raw data. (Does not include Naylor's routines associated with the unpacking of raw data from the collection facility.) User-oriented documentation (input, jcl) is available from Stan.
TDE	Jeff's data reduction routines for the reduction of Traffic Distribution and Efficiency experiment data. (On-line documentation, prepared by Jeff, is on UNIX and is available from Stan.)
TAC	Jeff's data reduction routines for the reduction of Transceiver Activity experiemnt data. (On-line documentation, prepared by Jeff, is on UNIX, and is available from Stan.)
NOS	Jeff's data reduction routines for the reduction of something, we're not sure exactly what. In any case it is quite similar to the HBH routines, and apparently consists of the HBH routines with some modifications.
HBH	Jeff's data reduction routines for the reduction of data from the Hop-By-Hop experiemnt. (On-line documentation, prepared by Jeff, is on UNIX and is available from me.)
ARPO65-	"X" and "Y" versions of the PLIX compiler version of the PRNET detailed simulation.

Note raw data and much output have been archived locally on 9-track tapes, maintained by Stan.

Following are the files archived at ISIE:

PS:<STANL>ARPO65-SLG-GENXPLIX..1 archived on tapes 3992 and 3994
PS:<STANL>ARPO65-SLG-GENYPLIX..1 archived on tapes 5104 and 5106
PS:<STANL>PRADIO.PRMAP.1 archived on tapes 4012 and 4015
PS:<STANL>PRADIO.PRINITI.1 archived on tapes 4012 and 4015

```
PS:<STANL>PRADIO.PRAM.1  archived on tapes 4012  and  4015
PS:<STANL>PRADIO.PRADIO.1  archived on tapes 4012  and  4015
PS:<STANL>PRADIO.COMMNT.1  archived on tapes 4012  and  4015
PS:<STANL>PRADIO.BADTYP.1  archived on tapes 4012  and  4015
PS:<STANL>TDE.TOUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.TCUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.TCOMP.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.OUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.INITI.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.CUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>TDE.COMP.1  archived on tapes 4015  and  4011
PS:<STANL>TAC.OUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>TAC.INITI.1  archived on tapes 4015  and  4011
PS:<STANL>TAC.CUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>TAC.COMP.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.TOUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.TCUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.TCOMP.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.OUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.INITI.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.CUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>NOS.COMP.1  archived on tapes 4015  and  4011
PS:<STANL>HBH.OUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>HBH.NOUTPUT.1  archived on tapes 4015  and  4011
PS:<STANL>HBH.NCUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>HBH.CUMMS.1  archived on tapes 4015  and  4011
PS:<STANL>HBH.COMP.1  archived on tapes 4015  and  4011
```

A Brief Description of the UCLA Packet Radio Network Simulation

by Medy Elsanadidi, Stan Lieberson, and Fouad Tobagi

Packet Radio Temporary Note # 244

February, 1978

This document offers the reader a brief introductory description of the UCLA PRNET simulation, currently in the latter stages of development, which has been designed to closely and realistically simulate the ARPA Packet Radio Network. The network configuration, program structure, simulation parameters, statistics collected, events, and data structures used are all described in moderate detail. It is assumed throughout that the reader is familiar with the Packet Radio Network in some detail.

I. Network Configuration

Topology. The simulation allows for one station, a maximum of 255 PRUs, and a maximum of 255 terminal devices. (Definitions as per PRTN 155A.) All elements are located at points on a cartesian plane. Mobile terminals may move along straight lines or along arcs of circles.

Radio Channel. Radio connectivity is a function of sender-receiver linear distance and of transmit power level. Contention on the channel and the effects of the spread spectrum technique are simulated so as to match what is known about the effects of bit rate, receive power level and time-capture on success or failure of correct receive in the PRNET. Channel access techniques are the three available in the PRNET: CSMA, Aloha, and Disciplined Aloha. The code simulating the channel connectivity and the code simulating the channel access methods are each fully contained in one routine each; this allows for their relatively easy modification for testing of alternate protocols.

Protocols. Acking, labeling, and PTP routing are all simulated as defined in the PRNET, CAP4. This includes HBH and Active acks, ROP generation by PRUs and handling by the station, label packets, and PTP routes.

Buffer Management. Buffer management occurs in PRUs, terminal devices, and in the station. In the PRUs buffers are managed according to the design implemented in CAP4. Again, programming modularity to allow for easy modification of protocols is insured by having the PRU buffer management controlled by three routines that are separate from other functions.

Terminal devices have individually selectable buffer capacities, and the station buffer capacity is also a parameter.

II. Simulation Program Sturcture

The PRNET simulation is written in PL/I, and is composed of one main procedure, whose structure and flow is indicated below:

```
PRNSIM: PROC OPTIONS(MAIN);
        <DECLARE simuation run parameters>
        <read in     "      "       "    >
 BEGIN: <DECLARE data structures, using sim. run parameters>
        <initialize free lists, etc>
        IF <this is a new run>
                  THEN <schedule station initiation as
                        first event>
                  ELSE <get net/sim status of previous run>
        <schedule end-of-simulation event>
DRIVER: <go to event code indicated by event at top of
         event list>
```

```
<EVENT1:  process type 1 event
          (e.g. 'begin radio tx') >
<EVENT2:  process type 2 event       >
<EVENT3:  ... etc ...                >
<Utility Procedures: handling linked lists,
                     scheduling events, etc  >
<Special Procedures: channel access, correct rx,
                     PRU buffer management, etc >
<Statistics Reporting>
END;
```

Upon termination of a run, the status of the simulation at that point may be saved in a file; the user may continue the simulation undisturbed at a later date by specifying that the simulation is to initialize by reading the saved file.

III. Input Parameters to the Simulation

1. Sizes - e.g. number of: PRUs, terminals, buffers per terminal, etc.
2. Aspects of the run, e.g.:
   - suppression of HBH acking (to simulate free ack analysis)
   - suppression of station routing control (for fixed-route experiments)
   - new run or continuation of old run <from file ...>

3. Default values for PRU and Terminal parameters (time-outs, etc)

4. Post-initialization events (scheduling of events to be read in from data cards: device initiation [including associated parameters], termination, changes in generated traffic rates, etc)

IV. Data Structures Used

All queues in the simulation are in the form of doubly-linked lists. There also is a doubly-linked list of free packet buffers.

Tables and queues in the station include:

- statistics
- routing table
- connectivity table
- ETE ack queue
- processing queue
- tx queue

Tables and queues in the terminal devices include:

- statistics
- tx queue
- ETE ack queue

Tables and queues in the PRU include:

- statistics

- radio channel environment
- redio tx queue
- ETE ack queue
- station/terminal tx queue
- packet processing queue

V. System Events

Terminal events:

- initiation of a terminal
- generation of traffic
- begin (wire) tx to PRU
- end (wire) tx to PRU
- move pkt on ETE queue to tx queue (after time-out)
- correct rx (wire) from PRU [and generate response]

PRU events:

- initiation of a PRU
- correct rx (wire) from station/terminal line
- enable rx from station/terminal
- end processing correctly rx'd packet
- begin (wire) tx to station/terminal
- end (wire) tx to station/terminal
- begin radio tx
- end radio tx
- enable radio rx

- begin radio rx
- end radio rx

Station events:

- initiation
- correct (wire) rx from PRU
- end processing correctly rx'd packet
- begin (wire) tx to PRU
- end (wire) tx to PRU
- move packet on ETE queue to tx queue (after time-out)
- generate traffic
- relabel PRUs needing relabeling
- reset unupdated connectivity table entries

VI. Statistics Reporting

Various types of statistics can be generated by the simulation. A means exists to limit the printing to only the desired statistics for any run. The time interval between each statistics report is a simulation parameter. Two types of information are reported:

Status reports, in which the status of devices (queue lengths, table contents) are reported, and

Statistics reports, which consist of statistics accumulated during that period regarding a device. Device statistics are listed below.

Terminal Statistics

queue lengths (min, max, mean)

queue residency time (min, max, mean)

queue service time (min, max, mean)

number of tx queue overflows

ETE statistics:

number of tx's to success (hist, min, max, mean)

number of packets discarded for lack of ETE ack

one-way time (hist, min, max, mean)

round-trip time (hist, min, max, mean)

number of unrecognized (duplicate) acks

Station Statistics

queue lengths, residency times, service times
(min, max, mean)

ETE statistics: same as for Terminal device

Throughput statistics:

total number of packets rx'd

" " of info packets rx'd

" " of acks rx'd

" " of ROPs rx'd

" " of control data packets rx'd (= acks for control pkts)

PRU Statistics

queue lengths, residency times, service times

(min, max, mean)

radio tx:

number of radio tx reschedules due to chan^nel^ busy (min, max, mean)

number tx's until HBH/Act ack (hist, min, max, mean)

number of packets discarded for lack of HBH/Act ack

interdeparture time of radio tx's [not necessarily ack'ed] (hist, min, max, mean)

radio rx:

number of packets radio rx'd with errors

number of packets correctly rx'd but not intended

number of packets correctly rx'd and intended

interarrival time of correct rx's (hist, min, max, mean)

amount of time radio is: tx-ing, rx-enb(idle), rx-disabled

neighbor table statistics:

number of packets- correctly rx'd,
incorrectly rx'd
(100, 400 KBPS)

buffer management:

number of packets discarded to allow for a radio rx buffer

number of buffer requests denied for lack of buffers

buffer utilization (use x time)

ETE: (no PRU ETE statistics currently)

PRUNET_SIM:

A Simulator of the Radio

Subnet of PRNET

by

M. Elsanadidi

Contents:

1. Introduction

The need was established for a detailed simulation of the PRNET to be used for verification of the results of the network measurement experiments, and to support studies of the network performance. Our efforts began with specifying a 'high level design' of the complete simulation, which would be developed in 'top-down' fashion. Since then, much of the program code for PRUs and TIUs has been written and effort has been spent on its testing. The station portion has remained in high level design form. The program documented here is a reduced version of the PRNET simulation. This version is capable of accurately simulating the radio subnet of a PRNET, that is, a network of PRUs hence the name PRUNET_SIM. PRUNET_SIM can be particularly useful for studying relatively small networks, with an emphasis on protocol details. Results concerning throughput, delay, and resource utilization (buffer space and radio channel utilization) can be obtained. Studies of the effect of protocol parameters and of various network configurations and channel access schemes may be carried out. Comparisons of alternative protocols may also be carried out with appropriate modification to those protocol elements where feasible. Thus, PRUNET_SIM can be an effective tool for the design and performance evaluation of the radio subnet.

Except for SPP protocol and routing, the PRU protocols of radio transmission and reception, processing, acknowledgment, and buffer management are all included in detail according to available information and CAP documents.

The simulated model of a PRU is schematically represented in Figure 1. A processing queue (PROC), a transmit and wait for acknowledgment queue (TWA), and a transmit and no wait for acknowledgment queue (TNWA) are maintained. Buffer management is implemented as in CAP 4.9 and radio transmission scheduling is cyclic as in CAP 5.0.

The radio channel is simulated by keeping track of which packets are heard on the channel at each PRU location. Packet collision is simulated with the effect of the spread spectrum technique. Such effect can be specified by input parameter. The PRUs use ALOHA, CSMA (1), and CSMA (2) as defined in CAP 5.0 documents. The radio connectivity of the network and therefore network configuration, can be set by input parameters.

Traffic entering the PRU network is received through the 1822 hardware interface, henceforth called 'wire' reception. Poisson input, as well as other input processes, can be simulated. The effect of buffer management on reception function, and "scrounging" (as in CAP 5.0) is simulated.

This is a discrete event simulation written in PL/I, and can be compiled by either PLC or PLIX compiler. The major data structures used are for PRUs, packets, events and running parameters. The program is approximately 1500 statements long, and one PLIX object occupies 200K bytes of core.

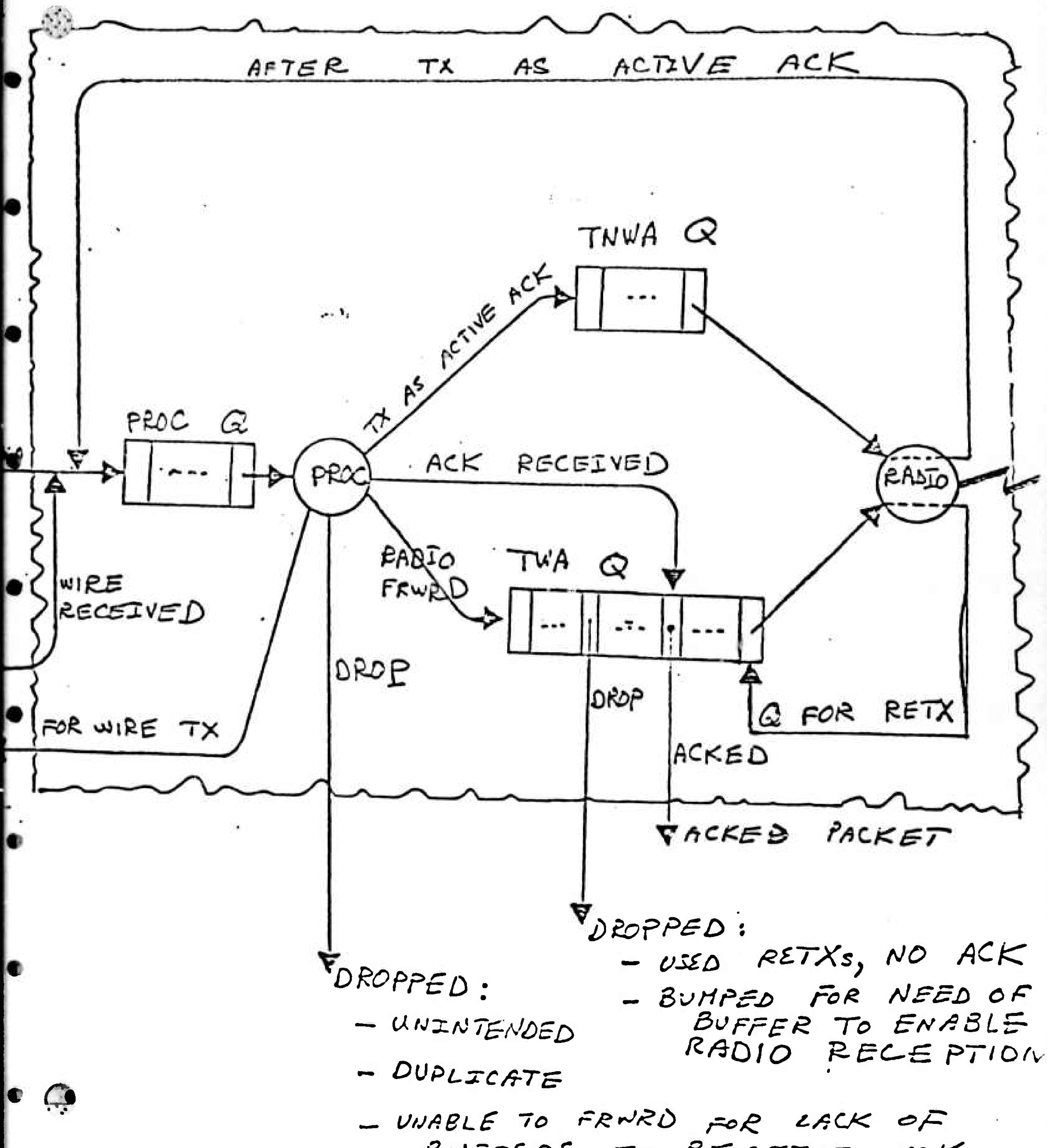

AFTER TX AS ACTIVE ACK
TNWA Q
TX AS ACTIVE ACK
PROC Q
PROC
ACK RECEIVED
RADIO
WIRE RECEIVED
RADIO FRWRD
TWA Q
DROP
DROP
Q FOR RETX
FOR WIRE TX
ACKED
ACKED PACKET
DROPPED:
- USED RETXs, NO ACK
- BUMPED FOR NEED OF BUFFER TO ENABLE RADIO RECEPTION
DROPPED:
- UNINTENDED
- DUPLICATE
- UNABLE TO FRWRD FOR LACK OF BUFFERS TO RECEIVE ACK.

Figure 1.

Because of the detail of the model simulated, the program is large and rather complex. Consequently, only small network configurations (4 to 10 PRUs, 2 to 4 hops) can be simulated with reasonable cost.

In the following sections we provide more details of the simulation model, the data structures and the logical structure of the program.

2. The Model

2.1 Traffic Generation and Wire Reception

It is assumed that traffic entering the PRU network arrives through wire reception. The traffic rate, packet lengths, routing and destination are specified by input parameters. The PRU 'wire receiver' is not enabled unless a buffer has been assigned to the receiver to accomodate an arriving packet, and the minimum time interval allowed between successive wire receptions (TRMIDY) has elapsed. TRMIDY is a PRU parameter and its value is specified for each PRU by input data. Upon reception, a packet is queued for processing by the PRU. Events representing wire reception are ENABLE WIRE RECEPTION and WIRE RECEPTION.

2.2 Radio Reception

Two radio receivers exist in each PRU. Their bit rates are specified by input data. The radio channel status is represented by maintaining information about those packets that are heard at each PRU location. This information is updated at the beginning and at the end of reception for each packet heard by a PRU. In this fashion, contention among packets and the status of the channel at the PRU location are simulated regardless of whether the PRU is disabled, receiving or listening for preamble. If a packet is received correctly, it is queued for processing and the radio receiver is disabled until an idle buffer is allocated for the affected radio receiver. Events involved in the simulation of radio reception are BEGIN RADIO RECEPTION, END RADIO RECEPTION, and ENABLE RADIO RECEPTION.

2.3 Packet Processing

Packets that are wire/radio received are queued for processing. Also, packets that were transmitted as active ack, are queued for processing. Processing time for each packet is drawn from one of two uniform distributions. The two distributions allow for differential processing types of packets, e.g., unintended (short) and active-acknowledge (long) packets. BEGIN PROC and END PROC are the two events used. At BEGIN PROC, the packet disposition is decided upon, but no action is taken until the END PROC event of that packet. The interval between BEGIN PROC and END PROC events is the processing time. It is drawn from one of the two distributions mentioned above, depending on whether, at BEGIN PROC, the packet is determined to need a long or short processing time.

2.4 Radio Transmission

Two radio transmission queues are maintained: Transmit and Wait for Ack (TWA), and Transmit and No Wait for Ack (TNWA). When it is time to transmit a packet, the TNWA queue is checked first. If the queue is not empty, radio transmission is initiated for the packet at the top. (Active acks are queued on the TNWA queue). If the TNWA queue is empty, the TWA queue is searched for a packet eligible for transmission. The algorithm determining "eligibility" is as specified in CAP 5.0.

In either case, when radio transmission is initiated, the radio transmitter first accesses the channel according to the access scheme specified by the input data. If the channel is available, the PRU "broadcasts" the packet.

Two events are used to simulate radio transmission: BEGIN RADIO TRANSMISSION and END RADIO TRANSMISSION. When a PRU decides to broadcast a packet at a BEGIN RADIO TRANSMISSION event, it schedules BEGIN RADIO RECEPTION at all PRUs in range. At END RADIO TRANSMISSION event, END RADIO RECEPTION events are scheduled for the same PRUs.

2.5 Buffer Management

The number of buffers in a PRU is an input parameter for the simulator. At initiation of a PRU, a buffer is allocated to each radio receiver, and to the wire receiver. The remaining buffers are designated idle buffers. A set of PL/I procedures are used to handle buffer management functions such as allocating or freeing a buffer, or 'scrounging' if no idle buffers exist.

2.6 Statistics

Throughput, delays, and resource utilization statistics are maintained throughout a simulation run. A user may specify the interval between statistic reports, and the time of the first report. Regardless of what is specified, a statistics report is always provided at the end of the simulation run (The length of the simulation run is also specified by the user). The contents of a statistics report can also be controlled by the user, who may select which items to be printed in the report. The following are the major statistics produced by the PRUNET_SIM.

Throughput:

Total net throughput, and throughput for each source destination pair.

Delays:

(1) end-to-end for each source-destination pair. The delay is defined as the time from reception of the packet on the wire at the source PRU until it is ready to be delivered to the destined end device by the destination PRU.

(2) 'one-hop delay' for each neighboring pair of PRUs. This delay is defined as the time from reception of the packet at the PRU until the one-hop acknowledgment is received.

Queues:

Lengths, residency times in TWA, TNWA and Processing queues.

Buffers:

Relative buffer utilization in radio reception, wire reception, TWA, TNWA, processing and queued for radio reception. Also statistics on scrounging (packets discarded to allocate buffers to radio receivers).

Radio Statistics:

(1) Transmission: histogram and mean of number of transmissions until acked, number of transmissions from TNWA queue.

(2) Reception: Number of packets received incorrectly, number received correctly, number of packets kept, etc...

(3) Time radio reception/transmission is disabled, idle, or active.

3. The Data Structures

In this section we provide an overview of the data structures used, and the management of queues and linked lists in the program.

3.1 The Run Parameters

This is a data structure containing parameters whose values are set by the user via input data. The parameters include the total simulation time, statistics report interval, time of first statistics report, the number of buffers in each PRU, the number of PRUs and array sizes that determine the number of packet structures, number of event structures, maximum route length, etc. Also, some statistics report control bits and debugging facilities control bits are included in the Run Parameters data structure.

3.2 The PRU

Each PRU data structure contains the following:

Parameters: A subset of the PRU protocol parameters relevant to the simulation model is used. It includes: radio channel access scheme, maximum number of retransmissions until packet is acked, retransmission delays (RTXDLY and RETXDY), min interval between successive wire reception (TRMIDY).

Queues: Each queue of packets (TWA, TNWA, PROC) has a pointer to the entry at the top. In addition, queue statistics are maintained in this substructure: current queue length, maximum and minimum length attained, average length, time of last insertion or deletion and average, maximum, minimum residency and service times.

Buffers: The number of buffers currently in use is maintained, as well as cumulative utilization statistics.

I/O Channels Status and Statistics: Radio receivers, radio transmitters, and wire receiver status are maintained as well as cumulative time in each state. A radio receiver may be enabled and idle, enabled and busy of disabled. The radio transmitter may be either idle, active or transmitting. The transmitter is 'active' if it has a packet ready for transmission but the transmission has not been started yet. This is the case when in CSMA the channel is sensed busy and the PRU has to defer its transmission to a later time. The wire receiver can be enabled and idle, enabled and busy, or disabled.

Radio Channel at PRU Location: To represent the radio channel at each PRU's location, the following is maintained:

- a pointer to any packet which is a candidate for correct reception
- the reception power of the 'candidate'
- the sum of the reception power of all other packets on the channel at the PRU location

A packet is a 'candidate' if its BEGIN RADIO RECEPTION at the PRU occurs when no other packets are on the channel and the radio receiver is enabled and idle. At END RADIO RECEPTION of a candidate, if its reception power has not been exceeded by the non-candidate reception power, and if any contention has been avoided by spread spectrum (simulated as a probability function), then correct reception is determined.

Traffic Generation and ETE Statistics: Generated traffic to be received on the wire at a PRU is controlled by information in a PRU substructure. ETE statistics of traffic originating at this PRU are also maintained in this substructure. For each PRU destination, arrival rate, route, generation time interval, and packet length are parameters to be specified by the simulation user.

Other PRU Statistics: Further statistics of duplicates, one-hop delays, radio transmission and radio reception are also maintained for each PRU.

3.3 The Packet Data Structure

A packet data structure contains the packet header and other pointers and counters for simulation and statistics purposes. No text is included, although a packet length is kept in the packet data structure.

The packet header includes a unique packet identification (assigned when the packet is generated at the source PRU), the ID of the source and destination PRUs, the packet route, a HOPOINTER (as in CAP), and an ACT bit that is set if the packet is an active ack.

Other information kept in the packet data structure includes link pointers for queues and lists, pointer to the event with which the packet is associated, statistics counters such as number retransmissions until ack, and time inserted in a queue.

3.4 The Event Data Structure

An event data structure contains an identification of the event, its time, pointers to the packet and the PRU associated with the event, as well as other miscellaneous flags.

3.5 Event List and Packet Queues

The lists in the programs are doubly linked circular lists. Procedures that are used to insert, delete, and search the lists can be used for any list, whether it be for packets, events, or any other entry. Link pointers are maintained in arrays, so no use is made of PLIX Pointer facilities. This makes the program compilable by both PLIX and PLC compilers.

At initialization of the simulator all event data structures are linked on a 'free events list'. Whenever an event is to be scheduled, an entry from the free list is deleted, its fields are set according to the event to be scheduled, and then it is inserted in the event list. The event list is ordered by event time. Linear search is used to determine an event's position.

A 'free packet list,' similar to the 'free events list,' is initially formed. Each PRU queue is maintained by a pointer to the top entry on the queue. A packet may exist in more than one PRU at any instant. We allow the same packet data structure to be shared by all the PRUs who are yet to process the packet after having received it, along with the PRU that transmitted it (the packet original 'holder'). If a receiving PRU decides to keep the packet, it makes a copy of that packet in a new packet data structure.

Because a packet can be on more than one processing queue at the same time, 'processing entries' are used to represent a packet on a processing queue. Each processing entry has a pointer to the packet it represents. The list of 'processing entries' are linked to form the processing queue.

Blank
page

4. Program High Level Design

In this section we provide a 'high level' description of the structure of the program.

```
PRNET:  PROC ;
Declare run parameters ;
Read in run parameters values ;
BEGIN ;
     Declare Packet, PRU, Event, and other data structures ;
     Initialize free lists, statistics, etc. ;
     Schedule first PRU INIT event ;
     Schedule END OF SIMU event ;
     DRIVER:  IF time for statistics report
                THEN call STATISRP ;
                GOTO code section to handle
                event at top of event list ;
/* PRU initiation, Traffic Generation and Wire Reception Events */
PRU INIT:  event handling code ; GO TO DRIVER ;
WIRE RX :  event handling code ; GO TO DRIVER ;
ENABLE WIRE RX:  event handling code ; GO TO DRIVER ;
     TRAFFIC:  PROC for traffic generation ;
/* RADIO RX  Events and Supporting Procedures  */
BEGIN RAD RX:  event code ; GO TO DRIVER;
END RAD RX:  event code ; GO TO DRIVER ;
ENABLE RAD RX:  event code ; GO TO DRIVER ;
```

```
CORRX:  PROC to determine whether
        a packet is received correctly.  Uses
        available information about spread spectrum;
/*PACKET PROCESSING Events and Supporting Procedures */
BEGIN PROC:  event code ; GO TO DRIVER ;
END PROC ; event code ; GO TO DRIVER ;
COPYPKT :  PROC called when PRU decides
           to keep a packet it received ;
DRADRPKT : PROC called when PRU decides
           to discard a packet it received ;
ACKREC :  PROC called when a packet
          received is an ack ;
NQACTACK :  PROC called when as a
            result of processing, the PRU needs
            to enqueue an active ack ;
PRFWD :  PROC called when the packet
         processed is to be forwarded ;
SPPRX :  PROC called when a received
         packet is found to be destined to
         the PRU ;
/* Radio Transmission events and Supporting Procedures */
BEGIN RAD TX :  event code ; GO TO DRIVER ;
END RAD TX :  event code ; GO TO DRIVER ;
     INITTX :  PROC called to initiate radio transmission when
               a packet eligible for transmission is found ;
```

CHACC : PROC called by INITTX to
access the radio channel

BRDCST : PROC called when a packet is to
be broadcast after determination that channel is available

CHSEPKT : PROC called to determine which
packet from the TWA queue, if any,
is to be transmitted now ;

NXTPOSTX : PROC to estimate the time
of next possible transmission.

/* Buffer Management */

INITPBUF : Proc called from
PRU INIT event to perform
initial buffer allocation in a PRU ;

GETBUF : PROC called when a buffer
is needed for enabling radio/wire reception ;

RETBUF : PROC called when a buffer is
freed.

SCROUNGE : PROC called to discard
a packet from TWA queue,
and grab a buffer for radio reception ;

/* Miscellaneous PRU Support Procedures */

RELEASE - PACKET : PROC called whenever
a PRU needs to discard a
packet ;

SCHBEGTX : called whenever

a BEGIN RAD RX event is to

be scheduled ;

SCHINIT : PROC called whenever a

PRU INIT is to be scheduled ;

ENT[illegible]PRC : PROC called whenever

a packet is to be enqueued

on the Processing Queue ;

/* End of Simu Event */

END OF SIMU : event code ; STOP ;

/* Statistics Procedures */

STATISRP : PROC called whenever

it is time for a statistics

report ;

UBSTATIS : PROC called to update

buffer statistics ;

UQSTATIS : PROC called to update

a queue's statistics ;

UHOPSTAT : PROC called to update onehop statistics

/* Utility Procedures */

DELAY : PROC called to compute a random

delay interval, uniformly distributed ;

RANDX : PROC for random number generation ;

INSRT : PROC called to insert an entry in

a linked list ;

SEARCHL : PROC called to search a

linked list for an entry ;

```
RELEV :  PROC called at the end of an
         event code to free event structure ;

PRINT_EVENTS :  PROC to dump  event list
         for debugging purposes ;

SCHED :  PROC called to form
         and schedule an event ;

PRUNET ;
```

Input and Output of the Program

The input data of the program specify the network configuration, traffic requirements, PRU parameters, simulation time and other simulation run parameters. We included in Appendix 1 an example of an input file. Except for arrays, the GET DATA statement is used to read in data. This makes the input file more readable to a user.

The Run Parameters data structure includes the following:

PRU# : total size of PRUs,

#PKT : size of array of packet data structures.

PRU#BUF : number of buffers in a PRU.

NUMDST : maximum number of destinations per source PRU.

MAXROUTE : maximum length of a route.

RTRATE : specify radio transmission bit rates :

0 rate will depend on PRU function (repeater or device front end)

1 all net uses lower rate

2 all net uses higher rate

TRATE(1) : lower bit rate, e.g. 100 KBit/sec.

TRATE(2) : higher bit rate, e.g. 400 KBit/sec.

SPECTRUMPROB : probability that a candidate packet is received correctly.

UBRETX : maximum number of retransmissions for all PRUs. Used for statistics array size.

SIMTIME : total simulation time.

NXTSTATISRP : time of first statistics report.

STATISRPINT : interval between statistics reports.

PROCMEAN : mean of uniformly distributed packet processing time (long processing).

PROCVAR : variance of that distribution

PROC MEAN SHORT, PROCVARSHORT : same for short packet processing time.

OSMEAN : mean of uniformly distributed delay represents the time used by operating systems.

OSVAR : variation of that distribution.

HALFOSMEAN, HALFOSVAR : same for short delay pf Operating System

SEED, MULT : used in random number generation.

Data Items for Control of Debugging Facilities :

TRACEV : 0 no tracing of events

1 print at execution of an event

2 print at both executive and scheduling of an event.

BEGTRC, ENDTRC : Beginning and End of time interval in which tracing is requested.

PKTPRNT : bit used to control printing information regarding an end to end packet arrival.

TXPRINT : bit used to control printing information regarding TWA queue at each broadcasting.

Values of the parameters mentioned above are read at the very beginning, and some are used subsequently for storage allocation according to the arrays' sizes specified by those parameters.

The next data structure read is PRUDEFPAR. It contains the default PRU parameters:

PPRID : 4-character PRU indentification.

PTXMODE : channel access scheme:

0 ALOHA, 1 CSMA(1), 2 CSMA(2)

PDMACTL : Transmission bit rate control:

0 not used, all net use same rate specified in RTRATE

1 PRU uses rate according to its function.

PRTXCNT : Maximum number of transmissions of a packet until hop acked

PRTXDLY : initial packet transmission delay (as in CAP)

PRETXDY : incremental delay (as in CAP)

PTRMIDY : minimum interval between successive receptions on 1822.

An array of 11 bits is then read in to control statistics reports contents. If a bit is set, the corresponding statistics item is printed out. These items are the following:

1 queues statistics

2 buffer statistics

3 not used

4 radio transmission protocol statistics

5 radio reception protocol statistics

6 radio transmission/reception channel statistics

7 statistics on scheduling of BEGIN RAD TX event

8 one hop statistics

9 ete statistics

10,11 not used

Following the statistics reporting bits, the connectivity matrix is to be specified in the input file.

HEAR (I,J) = 1 means PRU J may hear a packet transmitted by PRU I.

Next to the hearing matrix, data for PRU initiation events for PRUs are to be specified.

INITEV # is the type of initiation. It should always be 10 for a PRU initiation.

INITEVTIME is the time the PRU and its traffic generation is initiated.

Data for updating the parameter values of the PRU is specified next. The example shows an update to the PRU identification. Other PRU parameters take their default values specified in PRUDEFPAR data structure.

Data concerning traffic generation is specified next to the initiation event and the update of parameters of each PRU. This data includes: the route, the destinations PRU, the average interarrival time, the total number of packets to be generated, the time at which no more generation should occur, and the length of a packet to be generated for that route.

The end of the initiation events is signalled by the last two items of a data file which has

INITEV # = -1 and

INITEVTIME = 0 ;

In Appendix 2, we included an example of an output from the program.

The output of that run contained statistics of the four PRUs that were in the network; however, e included here statistics for only one PRU. These statistics are:

Queues Statistics: TWA, TNWA, PROCQ statistics are reported. SPP and WIRE TX queues are [illegible] used in this version. All times reported are in microseconds In addition to mean lengths and service times for the three queues, the sources of packets processed are printed out.

Buffer Statistics: These statistics give the relative utilization of the buffers for different purposes.

REQUESTS DENIED is an indication (not the real number) of lack of free buffers to be allocated for radio/wire reception. In the case of processing buffers it is used for a completely different purpose. It is the number of packets received at a PRU to be forwarded; but the PRU decides to discard them for lack of buffers to receive the hop acknowledgment.

RADIO TX Protocol Statistics

In this section, the histogram of number of transmissions until acknowledgments is printed, as well as the number of packets discarded after maximum transmission count was attained, and no acknowledgment was received.

RADIO RX Protocol Statistics

A breakdown of the disposition of packets received on the radio is provided.

Radio Channel Statistics

The utilization of radio channel in transmission and reception, and the time the radio is idle or disabled is printed out in this section.

Begin RAD TX Scheduling

These numbers give an idea about why the PRU delays the radio transmission, e.g. RTXDLY not passed. A breakdown (by cause) of the number of times BEGIN RAD TX event is rescheduled is provided here.

One Hop Statistics:

This section gives an average and standard deviation of the one hop delay from the PRU to each of the neighbors to which it transmits. The one hop delay is defined as the period from when the packet is received at the PRU until it is acknowledged by the immediate neighbor.

ETE Statistics

In this section, ETE statistics of packets generated at the PRU are printed for each destination.

At the end of a simulation run, ETE statistics for the whole network are printed, as well as a reprint of the ETE statistics for each source-destination pair. Also, the count of scheduled and executed events is printed for use as a monitoring aid to the simulation program performance and correction.

Appendix 1:

Example of an Input File

```
RUNPAR.TITLE                 = '3 HOP TEST, TOTAL INPUT 20PPS' ,
RUNPAR.PRU#                  = 4                              ,
RUNPAR.PRU#BUF               = 6                              ,
RUNPAR.#PKT                  = 35                             ,
RUNPAR.#EV                   = 50                             ,
RUNPAR.MAXROUTE              = 4                              ,
RUNPAR.NUMDST                = 3                              ,
RUNPAR.RTRATE                = 0                              ,
RUNPAR.TRATE(1)              = 100                            ,
RUNPAR.TRATE(2)              = 400                            ,
RUNPAR.TRACEV                = 0                              ,
RUNPAR.SPECTRUMPROB          = 95                             ,
RUNPAR.PKTPRNT               = '0'B                           ,
RUNPAR.TXPRINT               = '0'B                           ,
RUNPAR.SIMTIME               = 5000000                        ,
RUNPAR.NXTSTATISPR           = 1000000                        ,
RUNPAR.STATISPRINT           = 1000000                        ,
RUNPAR.BEGTRC                = 0                              ,
RUNPAR.ENDTRC                = 0                              ,
RUNPAR.UBRETX                = 6                              ,
RUNPAR.SEED                  = 131759                         ,
RUNPAR.MULT                  = 977241                         ,
RUNPAR.PROCMEAN              = 10000                          ,
RUNPAR.PROCVAR               = 2000                           ,
RUNPAR.PROCMEANSHORT         = 6000                           ,
RUNPAR.PROCVARSHORT          = 2000                           ,
RUNPAR.OSMEAN                = 3000                           ,
RUNPAR.OSVAR                 = 1000                           ,
RUNPAR.HALFOSMEAN            = 1500                           ,
RUNPAR.HALFOSVAR             =  500                           ,
RUNPAR.ACCESSMEAN            = 4000                           ,
RUNPAR.ACCESSVAR             = 2000                           ,
```

```
PRUDEFPAR.PPRID        = '    '                    ,
PRUDEFPAR.PTXMODE      = 1                         ,
PRUDEFPAR.PDMACTL      = 1                         ,
PRUDEFPAR.PRTXCNT      = 6                         ,
PRUDEFPAR.PRTXDLY      = 8200                      ,
PRUDEFPAR.PRETXDLY     = 10240                     ,
PRUDEFPAR.PTRMDLY      = 1500                      ;
```

```
'1'B, '1'B, '1'B, '1'B, '1'B, '1'B, '1'B, '1'B, '1'B, '1'B, '1'B,
```

```
0, 1  0, 0,
1, 0, 1, 0,
0, 1, 0, 1,
0, 0, 1, 0,
```

```
P.PPRID='A001'    ;

1 , 2, 3,  4,
   4  ,  100000,  120, 20000000,17,
 1,-1,-1,-1,
   -1 , 200000,   0,  4000000,17,
 1,-1,-1,-1,
    -1, 200000,   0,  4000000,17,


INITEV#=10,INITEVTIME=0;

 P.PPRID='A002';

 2,-1, -1 -1,
    -1,   200000,    0,    20000000,17,
 2,-1,-1,-1,
   -1 ,   000000,    0,      400000,17,
 2,-1,-1,-1,
   -1 ,   000000,    0,    2000000,17,


INITEV#=10,  INITEVTIME=0;

 P.PPRID='A003';

3 ,-1,-1,-1,
   -1 ,     25000,     0,    20000000, 17
3 ,-1,-1,-1,
   -1 ,    200000,     0,     4000000, 17,
3 ,-1,-1,-1,
   -1 ,    200000,     0,  2000000,17,


INITEV# = 10,   INITEVTIME=0;

P.PPRID='A004';

 4, 3, 2,  1,
    1 ,    100000,   120,  20000000,  17,
 4,-1,-1, -1,
    -1,    200000,     0,  20000000, 17,
 4,-1,-1,-1,
    -1,     25000,     0,  20000000, 17,


INITEV# = -1,   INITEVTIME=0;
```

Appendix 2:

Example of a Statistics Printout

RU 1 A001

PPPP QUEUES STATISTICS PPPP

	T N-A	T WA	SPP ETC	WIRE TX	PRC LIST
LENGTH: CURRENT	0	1	0	0	C
AVG	0	1	0	0	1
MAX	1	3	0	0	2
MIN	0	0	0	0	C
LAST ACTIVITY	4837177	4999731	0	0	4999731
# PASSED IN Q	47	56	0	0	265
RESIDENCY: AVG	2760	105015	0	0	14202
MAX	2760	334326	0	0	36370
MIN	2760	876	5000000	5000000	5366
SERV TIME: AVG	2760	67015	0	0	9311
MAX	2760	311723	0	0	11972
MIN	2760	14043	5000000.	5000000	4068

BREAKDOWN OF SOURCES OF PKTS IN PROCESSING

CORRECT WIRE RX	56
CORRECT RADIO RX	162
AFTER-TXED-AS-ACT-ACK-FOR-DUP	0
AFTER TXED AS ACT ACK FOR OTHER REASON	47

PPPP BUFFERS STATISTICS PPPP

	RX1	RX2	RAD OD	T WA	WIRE TX	WIRE RX	PRC LST	IDLE
CURRENT ALLOCATION	1	1	0	1	0	1	0	2
REQUESTS DENIED	0	1	0	0	0	57	0	0
LAST ACTIVITY	2576381	4845670	4718304	4999731	-1	4908032	4599731	4980022
UTILIZATION	0.99997	0.99686	0.18816	1.19922	0.00000	0.66180	0.75277	1.17526
# OF DUMPS				16				

PPPP RADIO TX PROTOCOL STATISTICS PPPP

# TX UNTIL HOP ACK	0	1	2	3	4	5	6
	0	10	13	5	10	2	5

AVG=2.91111 USCD 3

	LOW RATE	HIGH RATE
TOTAL # PKT TX'S	47	162

PPPP RAD RX PROTOCOL STATISTICS PPPP LOW RATE HIGH RATE

```
# PACKETS RECEIVED IN ERROR                                        0         0
# ABORTED PACKETS RECEIVED                                         0         0
# PACKETS RECEIVED CORRECTLY AND KEPT                              0       160
# PACKETS RXED COR-BUT USED                                        2         0
# PACKETS LOST DUE TO RX DSBD BY RADIO TX                          0         0
# PACKETS LOST DUE TO RX DSBD DUE TO NO BUFS                       0         0
# PACKETS LOST DUE TO RX DSBD BY TX AND NO BUFS                    0         0
# PACKETS LOST DUE TO RX DSBD SINCE TERM PRO                       0         0
# DUPS IN FILTER TABLE                     20
# DUPS ON TWA QUEUE                         0
# DUPS ON TWA QUEUE                         0
# ACKS RX AFTER PKT DROPPED                71

PPPP   RAC CHANNELS STATISTICS   PPPP

     **** TRANSMISSION ****
  HIGH RATE               LOW RATE
 LAST        CUMU         LAST        CUMU
4822445     110140       4836417      82720

     **** RECEPTION ****
              RECEIVING                      ENABLED                                        DISABLED
        LAST                CUMU               LAST              CUMU              LAST              CUMU
   R1        R2        R1        R2        R1        R2        R1        R2        R1        R2        R1
2574541   4844910     3520    160000    4837257   4849670   4792603   4777299   4835417   4845590    207917

CURRENTLY TX                        0
CURRENTLY PROC                     91
RTXDLY NOT PASSED                 141
TX DELAY NOT PASSED FOR ANY PKT              5
DROPPED CHOSEN PKT: 0 HAS OTHER PACKETS      0
CHANNEL BUSY                        0
```

```
PPPP    CAP FOP STATISTICS    PPPP

NEIGHBOR    # OF TX TO    AVE DELAY    STANDARD DEV
   2            57           70472         55105

PPPP    NONTERMINAL ETE STATS    PPPP

DESTINATION    # OF PKTS GEN    # OF PKTS ETE RX    ETE DELAY
     4              57                 32             161400

           ETE PKT STATISTICS

# PKTS RECEIVED ETE                    50
# PKTS LOST ETE                        50
AVG ONE WAY DELAY                  153924

SOURCE                        1

DESTINATION    # OF PKTS GEN    # OF PKTS ETE RX    ETE DELAY
     4              57                 32             161400

SOURCE                        2

DESTINATION    # OF PKTS GEN    # OF PKTS ETE RX    ETE DELAY

SOURCE                        3

DESTINATION    # OF PKTS GEN    # OF PKTS ETE RX    ETE DELAY
```

```
SOURCE                              4

DESTINATION   # OF PKTS.GEN.    # OF PKTS ETE RX.   ETE DELAY
     1             52                  27             145055

EVENTS
CNTEVENTS_SCH(10)=        4        CNTEVENTS_THC(10)=       4;
CNTEVENTS_SCH(11)=      133        CNTEVENTS_THC(11)=     131;
CNTEVENTS_SCH(12)=      237        CNTEVENTS_THC(12)=     237;
CNTEVENTS_SCH(13)=     1147        CNTEVENTS_THC(13)=    1146;
CNTEVENTS_SCH(14)=        0        CNTEVENTS_THC(14)=       0;
CNTEVENTS_SCH(15)=        0        CNTEVENTS_THC(15)=       0;
CNTEVENTS_SCH(16)=     1659        CNTEVENTS_THC(16)=    1516;
CNTEVENTS_SCH(17)=      725        CNTEVENTS_THC(17)=     725;
CNTEVENTS_SCH(18)=     1042        CNTEVENTS_THC(18)=    1042;
CNTEVENTS_SCH(19)=     1042        CNTEVENTS_THC(19)=    1042;
CNTEVENTS_SCH(20)=     1003        CNTEVENTS_THC(20)=    1003;
CNTEVENTS_SCH(21)=     1147        CNTEVENTS_THC(21)=    1147;
CNTEVENTS_SCH(22)=        1        CNTEVENTS_THC(22)=       1;

IN STMT 1566 PROGRAM RETURNS FROM MAIN PROCEDURE
```

The Building-Block Methodology Used for Studying PRNET

Zaw-Sing Su
UCLA
March 30, 1979

I. Introduction

Since the development of the ALOHANET in the late 60's, the concept of packet broadcasting network has received considerable attention as a feasible solution for rapid network deployment, for the use of mobile network elements, and as an economic alternative for point-to-point packet switching. A number of projects have been established to study such networks.

The Packet Radio Network (PRNET) developed under the support of DARPA is an experimental packet switching network facility using a common radio broadcast channel as its transmission medium. Among the objectives of its development are to demonstrate and to characterize the capabilities of packet radio networks. Much effort has been devoted in this regard. A number of studies have been carried out addressing issues concerning various facets of packet radio networks. Such efforts have been proven fruitful in studying specific issues. In this temporary note, we describe a methodology which is used for achieving a systematic characterization of throughput/delay and capacity behavior of the PRNET, as well as for supporting its continuing development.

The variable configuration of a PRNET leads naturally to a building-block approach. Each PRNET deployment may take a different configuration. The possible existence of mobile network elements also allows configuration changes throughout a single deployment.

For analysis using a building-block approach, the PRNET is decomposed into components (building blocks). Characteristics of a building block are studied as functions of its environment. Then, a network or a subnet with any configuration can be analyzed as a compound system consisting of building blocks. For a building-block approach to be effective, it is important that the network decomposition results in building blocks such that:

i) analysis of a building block is much easier than the analysis of an entire network,

ii) analysis of an integration of building blocks can be efficient, and

iii) interaction (coupling) among the building blocks can be sufficiently considered in their integration so that accurate analysis may be achieved.

Taking a building-block approach may lead to the understanding of not only the behavior of networks of various configurations, but also the impact of a subnet configuration on the network behavior. A building-block model which suits our studies is presented in the next section. In Section III, the building-block methodology used for PRNET studies is described. We discuss, in Section IV, a simple and efficient system-level simulation as an integral part of our methodology.

II. The Building-Block Model

The behavior of a PRNET can be viewed as the aggregative experience of packets traversing the network one hop at a time. Since the effect of packet broadcasting on network behavior is the major concern of our studies, for the model considered the Packet Radio Units (PRUs) and the relations among them form the basic structure of a PRNET.

The one-hop neighbors by radio connection of a PRU, together with the PRU itself, form a subnet. The configuration of this subnet together with the rate of packets for transmission from each of these PRUs form this PRU's one-hop neighborhood. The experience of a packet travelling one hop from a PRU to another can be characterized by the success probability of a transmission and, for CSMA, the time it takes to access the channel. This experience of a packet depends largely upon the one-hop neighborhoods of the receiving and the transmitting PRUs. The influence of the remaining network is reflected in traffic condition of these one-hop neighborhoods. We consider, as the building block of our PRNET model, a PRU and the one-directional hops to its one-hop neighbors. Formally, the building block of our model consists of network functions affecting the progression of a packet between entering the transmit queues of two consecutive PRUs.

The basic building block characterization consists of the success transmission probability and the distribution of channel access time (non-zero for CSMA). They are quantified as functions of the one-hop neighborhoods of the transmitting and receiving PRUs respectively. We categorize a one-hop neighborhood by the following:

1. the number of independent groups* in the one-hop neighborhood, and the number of PRUs in each independent group,

2. the rate of packets for transmission in each PRU.

For a realistic configuration, it is reasonable to assume that the number of one-hop neighbors of a PRU is not more than 8 (i.e., the maximum number of mutually hidden PRU is 4 -- see Figure 1).

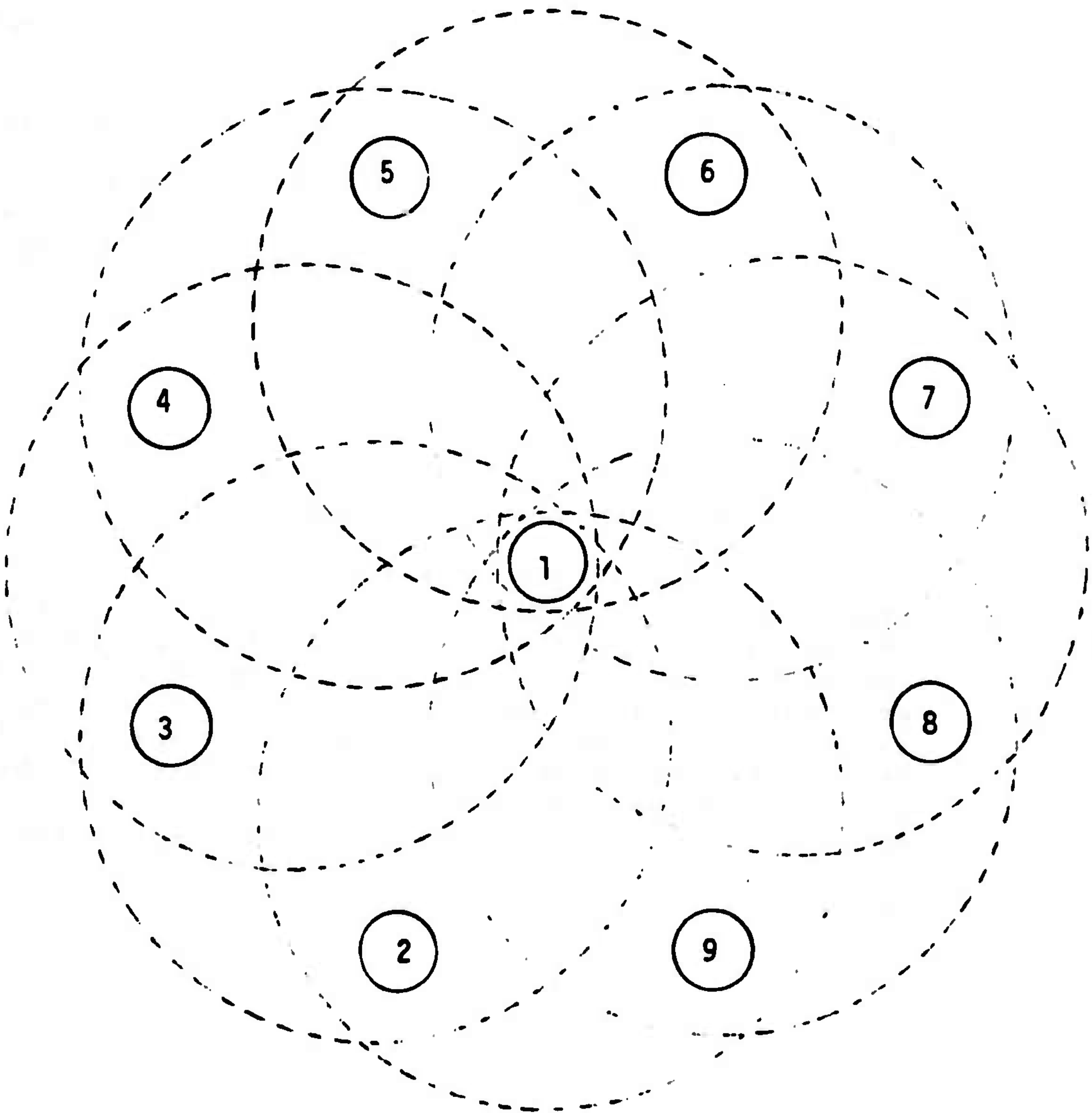
III. A Methodology for Studying PRNET

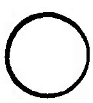
The tools available for studying the PRNET include the measurement facilities incorporated in the current PRNET implementation and a detailed simulation of this implementation designed and constructed at UCLA. The advantage of using these tools is their capability of directly studying or closely approximating realistic PRNET behavior. Use of the measurement facilities relies on the availability of the testbed network. It also consumes personnel resource for experiment setup. Both the measurement facilities and the detailed simulation suffer from being relatively inflexible for studying alternative design possibilities or hypothetical conditions, and difficult for studying large scale networks. To compensate, the use of a simple and efficient system-level simulation (Section IV) is incorporated as an integral part of the methodology.

This methodology is based on a building-block approach. Its basic scheme calls for the use of measurement experiments to calibrate the detailed simulation and to verify when necessary the study conclusions. Once calibrated, the detailed simulation can be used for quantifying the building block behavior under various one-hop neighborhoods, and used as a substitute for the testbed network to reduce our

---

*Two PRUs belong to the same independent group if they have exactly the same one-hop neighboring PRUs within the considered set.



 Packet Radio Unit (PRU)

-- the three PRUs within each region (dotted circle) can hear each
-- PRUs 2, 4, 6 and 8 ar mutually hidden from other

Figure 1

dependence on measurement activities. Knowledge of the building block behavior together with the use of system-level simulation form the basis for a flexbile and efficient methodology. This methodology may be employed in PRNET studies for:

i) systematic characterization of current PRNET implementation,

ii) evaluation of design alternatives,

iii) supporting future PRNET developments.

Appendix A details these aspects of PRNET studies and includes a description of the studies currently in progress. In Appendix B, we give an example describing the application of this methodology.

IV. The System-Level Simulation

The system-level simulation is a modulized program with a building-block simulator and a network simulator forming its basic structure.

The building-block simulator simulates the basic PRU functions, e.g., a Poisson process for packet reception, an exponential distribution for packet processing time, and a success probability for its transmission. For features necessary for a specific study, the building-block simuator calls subroutines. Whenever appropriate, this simulator approximates by a constant, a probability or a probability distribution the effect of a system function or the influence of a network condition.

The integration of building blocks is accomplished in the network simulator. The coupling of building blocks is taken into consideration by their mutual influence in terms of probabilistic effects. The network simulator also simulates end-to-end functions (e.g., end-to-end protocol) and network control functions (e.g., routing).

Advantages of such a system-level simulation include the following:

i) The system-level simulation is made efficient by closely simulating only aspects of the PRNET relevant to a specific study.

ii) Due to its simplicity and modulization it is easy to construct, simple to calibrate, and can be readily modified. Thus, for studying design alternatives, the modification of a network feature may be a matter of replacing a subroutine.

iii) Wherever appropriate we consider the influence of a network condition or the effect of a network function rather than simulate in detail the reality itself. Thus, the system-level simulation may simplify as well as generalize the model it uses. For example, we may substitute the study of a large scale network by the study of its 'critical paths' and the influence of the remaining network on these critical paths. We may simplify the model of a large network to a relatively simple model of its critical paths. As another example, the system-level simulation uses success probability of a transmission together with the time distribution of the interval between transmission attempts for simulating the effect of a channel access mode. It not only eliminates the complexity of simulating the channel access modes, but the resulting model is also generalized and can be used for studying the impact of different channel access modes by simply varying the probabilistic effects.

V. Conclusion

In this note, we have described a methodology employing a building-block approach for studying PRNET. This methodology combines the use of measurement experiments, detailed simulation, and system-level simulation. It takes advantages of each and mutually compensates their weaknesses. It reduces our dependence on the measurement activities. It is also readily adaptable for supporting future development of the PRNET such as the design of multistation and stationless PRNETs. Decomposition of the study of packet radio networks into relatively simple problems offers another advantage of the building-block approach. These simpler problems may yield more readily to mathematical modeling and analysis. Mathematical modeling and analysis may therefore be used in conjunction with this methodology.

Appendix A: PRNET Studies

The objective of the PRNET studies is perceived to be two-fold: to characterize its current implementation and to support its further development. For achieving this objective, we will systematically evaluate the basic capabilties (throughput/delay behavior and network capacity) of the current implementation with respect to various network configurations and varied input traffic. The established methodology and the basic network characterization provide us the capability for supporting the continuing development of PRNET. This end may be achieved by studying the impact of design alternatives. The studies currently in progress include the following:

I. PRNET Characterization

1. Systematic characterization of current PRNET implementation

The objective of this study is to characterize for the current PRNET implementation the impact of various network configurations and input traffic on end-to-end delay, throughput and network capacity. This study will give us not only an evaluation of its basic capabilities, but also a foundation for studies supporting continuing PRNET development. This study requires measurement experiments for calibrating the detailed simulation, the calibrated detailed simulation for quantifying building block behavior, and a system-level simulation program for network characterization. The measurement experiments accomplished to date can be used for calibrating the detailed simulation. Future experiments, expecially those with multi-hop configuration, will help to refine the calibration. Calibration of detailed simulation is progressing in parallel with its further development. The development of the system-level simulation is in progress in conjunction with other studies listed below. Concerted effort using mathematical modeling and analysis is also in progress.

2. Analysis of PRNET involving a Gateway

For interconnecting the PRNET with other networks, a gateway implementing internetworking functions is needed. The impact of the gateway on PRNET network behavior must therefore be assessed. Alternative means for connecting the gateway also need to be evaluated. In the current implementation, whether the gateway resides in the station or it becomes a physically separate device, it is connected to the PRNET through a PRU. From system behavior point of view the gateway can be viewed as an end device. In view of the lack of a physical gateway, its impact is studied by taking it as a terminal with different system parameters. Its study, involving a set of basic PRNET configurations, is currently in progress with priority.

II. Evaluation of Design Alternatives

3. Transmission Order Scheduling

This is a comparative study to evaluate the impact of different PRU transmission order scheduling algorithms on PRNET behavior. This study is in progress using a model consisting of a single PRU and its one-hop neighborhood. The building-block simulator of the system-level simulation is implemented for this study. We study the effect of different scheduling algorithms on the PRU's interaction (transmitting and receiving packets) with its neighbors. The effect is evaluated by the PRU throughput and the one-hop round-trip delay of the PRU traffic as well as the PRU buffer occupancy.

4. Congestion Control Studies

Due to the broadcast nature of the PRNET, congestion control is a significant issue having its impact on various design aspects. No concrete study on this issue is in progress. However, studying the impact of alternate routing is in its planning stage. The impact of LROPs and PDPs on congestion and their effectiveness for maintaining reliability is also in consideration as a potential area for investigation.

Studying design alternatives in other issues, such as flow control, protocol, and channel access mode, is also conceivable. They will be considered once the PRNET characterization is accomplished.

III. Supporting Future Development

5. Extension of the current model to study design issues for multistation PRNET is under investigation. The immediate objective is to devise a building-block model for studying its routing strategies. Our current building-block model with minor adjustments appears suitable for its throughput/delay studies.

Appendix B: An Example Illustrating the Methodology

In this example we illustrate the building-block methodology by its application to a study of the end-to-end delay and throughput characteristics betweeen terminals A and B of a network depicted in Figure B1. In this network, we assume PRUs 1, 2 and 3 are in each other's hearing range; so are PRUs 3, 4 and 5, as well as PRUs 5 and 6.

By assuming shortest path without alternate routing, we study in this example the experience of packets travelling the path linking terminals A and B, and PRUs 2, 3 and 4. For wired links we assume a constant delay and a throughput limit. Building blocks for this path consist of PRUs 2, 3 and 4 and the one-dimensional hops to their one-hop neighbors. The one-hop neighborhood of PRU 3, for example, consists of the set of PRUs 1, 2, 3, 4, and 5, and the rate of packets for transmission from these PRUs. Impact of the traffic between terminals B and C on the success probability of a transmission from PRU 2 to PRU 3 is reflected in the rate of packets for transmission from PRUs 4 and 5. Using the building block characterization available to us, a network simulator incorporating end-to-end functions (e.g., end-to-end protocol) would simulate packets travelling this path and yield an evaluation of the end-to-end behavior.

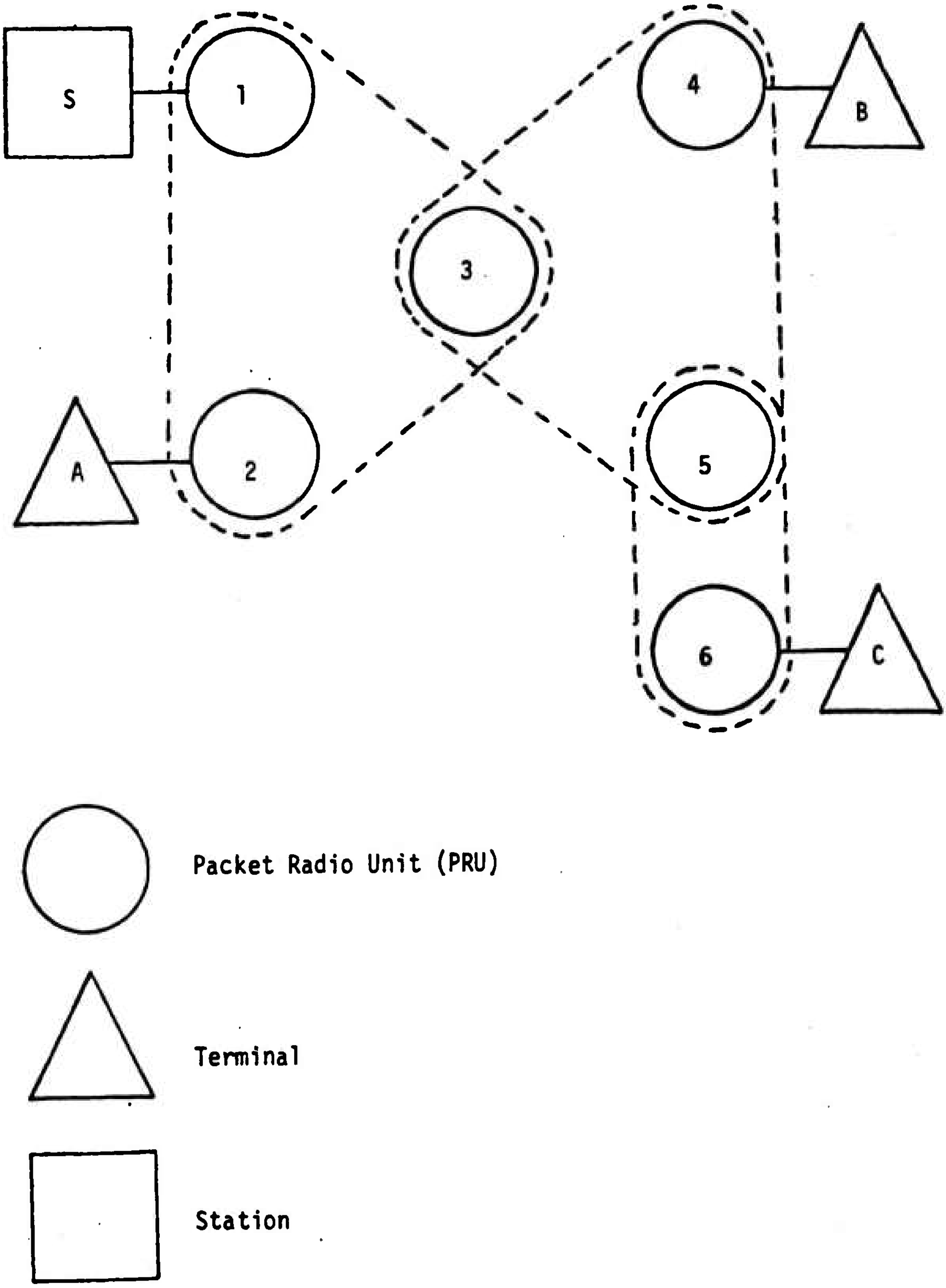


Figure B1

Documentation for PRNET System-Level Simulation

Zaw-Sing Su

UCLA

I. Introduction:

The Packet Radio Network (PRNET) is an operational packet switching network developed under the support of Defense Advanced Research Projects Agency. It uses a common radio channel as transmission medium. The packet radio technology offers the advantages of mobility, rapid deployment, potentially high bandwidth and reliable data transmission for military applications. A number of such network installations have been established. The objectives of their development is to install such networks, to demonstrate the feasibility, and to explore the potentials of packet radio technology.

A PRNET consists of three types of network elements: the packet radio units (PRUs), the terminal interface units (TIUs), and one or more stations. The PRUs function as store-and-forward packet switches. They are connected together via radio links to form the packet radio subnet. Each PRU may be used to connect an end-device, a TIU or the station, via a wire connection or stand a one as a repeater. The station embodies network control functions, while the TIUs, as the name implies, are used for interfacing used terminals (including hosts, and internetworking gateways). For the current implementation, there is one station to each PRNET. To extend the range of a PRNET, future development may incorporate more than one station, or eliminate the station by distributing the network control functions.

The establishment of operational PRNETs has demonstrated the feasibility and potentials of packet radio technology. Further advancement of such technology by continuing development can be effectively supported by coordinated efforts employing analysis, measurement, and simulation. Mathematical modelling and analysis are used to establish basic principles and to guide the investigations of such technology. As a research project, extensive measurement capabilties have been incorporated in the design and development of PRNET. Measurement experiments can be used for extracting basic system parameters, calibrating, and verifying investigation results. Simulation programs have been constructed to complement the capabilities of measurement tools and mathematical analysis techniques. Simulation may extend the capabilities to pursuing more extensive enumeration, to the studies of large scale networks, as well as to effectively studying design alternatives. For achieving these goals, a system-level simulation package has been developed under the following guidelines:

1. Accurately capturing essential functiors of the PRNET design. When calibrated, it should be able t assist in quantitative system evaluation;

2. Modulized so that it is flexible for studying design

alternatives in support of further advancement in packet radio technology;

3. Efficient and cost effective to be feasible for extensive investigations;

4. Well-instrumented to be capable of providing sufficient information for system analysis and calibration.

In view of these design objectives and the proposed building-block approach to PRNET studies (see PRTN#268), three simulators with different levels of details have been developed for the System-Level Simulation package. The building-block (BB) simulator simulates a PRU with the remaining network as its environment. While the general broadcasting network simulator (SLP) links a number of PRUs into a network without employing a specific protocol, the PRNET simulator (SL) simulates CAP5 specific implementations.

This document introduces the reader to the architecture of the System-Level Simulation package. It is also intended as a quide to the use of this package.

II. Basic Model

The model used in the System-Level Simulation package consists of a packet broadcasting subnet and a set of end-devices. Each end-device is connected to the subnet by a point-to-point (e.g., wire) connection to a broadcasting unit. At most one end-device can be attached to each broadcasting unit. In the case of PRNET, a terminal interface unit (TIU) or a station is an end-device attached to a PRU. A set of PRUs sharing a common radio channel form the packet broadcasting subnet.

The model for a PRU comprises two servers and a buffer pool. Of the two servers, one simulates the PRU processor and the other its transmitter. Upon its arrival at a PRU, a packet joins the processing queue awaiting to be processed. After being processed, it enters a transmission queue, and is then scheduled for transmission. Positive acknowledgement is assumed for packet hop transport. A packet occupying a PRU buffer may be an entry in its processing queue, in its transmission queue, or waiting to be acknowledged.

III. Description of the Simulators

A. Buildi -Block (BB) Simulator

Design of the PRNET System-Level Simulation follows a building-block approach to PRNET studies (see PRTN#268). The Building-Block Simulator simulates one PRU, and the remaining network as the environment of the PRU. The environment of a PRU consists of its neighbors and the operating condition of its neighborhood. Its neighbors include a number of 'next' PRUs to which packets are forwarded and from which acknowledgements are received; and a number of 'previous' PRUs from which packets are received and to which acknowledgements are sent. The operating condition of its neighborhood represents the impact of an instantaneous state of the remaining network. Specification for this operating condition defines the success transmission probabilities to and from its neighbors, the percentage of overhead traffic (e.g., unintended packets and control packets), and the distribution for required channel access time (which accounts for the number of channel sensings per transmission and transmission randomization delay). For a successfully acknowledged packet, an acknowledgement delay is accounted for. The hop acknowledgement delay is specified by its distribution.

The packet arrival process is assumed to be Poisson. A fixed portion of the input traffic can be specified as overhead for the simulated PRU. Therefore, two PRU processing time distributions are assumed: one for intended and the other for overhead packets, and both are exponentially distributed. For channel access time, a uniform distribution is used. Different channel access schemes are simulated by specifying appropriate channel access time and success transmission probabilities. The transmission scheduling algorithm can be parametrically selected as FIFO (CAP4), CAP4.9 cyclic, or CAP5. The linear-backoff retransmission algorithm specified in the current Channel Access Protocol (CAP) is also simulated.

The FIFO algorithm schedules the packets for transmission in their order of arrival. It allows at most one packet unacknowledged. Therefore, the transmission of a new packet would not take place until the currently transmitted packet has been either acknowledged or discarded after a prespecified maximum number of transmissions.

The transmission scheme for CAP4.9 implements multiple independent FIFO transmission que.ues, one for each 'next' neighbor. Only the packets on top of each FIFO queue may be outstanding unacknowledged. When the transmitter becomes available, the packets on top of the transmission queues are scanned in a cyclic order for a packet to be transmitted.

In CAP5, the transmission scheme allows any ready for transmission packet to be transmitted.

After its transmission, a packet waits to be acknowledged. If the acknowledgement does not arrive within a time-out period, the packet will be retransmitted. In BB, the acknowledgement delay (the time from a packet transmission to the return of its acknowledgement) is simulated by a prespecified exponential delay distribution. It is compared with the time-out period which is calculated using the following linear-backoff scheme:

RTXDLY + RETXDY * (times the packet has been transmitted).

The initial delay, RTXDLY, represents a minimum interval between any two consecutive transmissions. Each of the consecutive time-out periods for a packet is linearly incremented by RETXDY. The packet is discarded after a maximum number of transmissions.

B. Generalized Broadcasting Network (SLP) Simulator

The SLP incorporates a network simulator which drives a multiple incidents of the PRU simulation used in BB. A forward transmission from a PRU places the transmitted packet in the input (processing) queue of the receiving PRU. It thus realizes the acknowledgement delay by the time actually incurred incurred for processing and waiting in the receiving PRU, and that spent for two-way transmission. For packet hop transport, echo acknowledgement is assumed (with active acknowledgement for the last hop of a route). Having a common channel broadcasting transmission medium, a packet forwarding to the to the 'previous' PRU.

For network simulation, mechanisms are provided for route specification. The input traffic is distributed in prescribed proportions among the routes. Additional traffic must be specified for unintended and control packets entering individual PRUs. Beyond simulating the packet radio subnet, the SLP simulator models an end device by a fixed delay and the capability for generating end-to-end acknowledgements. End-to-end as well as round-trip statistics are collected for presenting network behavior. Only FIFO and CAP5 scheduling schemes are implemented in this simulator.

C. CAP5 PRNET (SL) Simulator

Tailoring SLP to the study of CAP5 PRNET, a number of additional details for essential functions are incorporated in SL to closer reflect the PRNET reality. The implemented features additional to SLP include:

1. connectivity specification -- a binary hearing matrix is used to define PRU connectivity. A '1' for the (i-j)th entry of the hearing matrix indicates that PRU i can hear PRU j;

2. channel access mode -- nonpersistent CSMA is simulated for channel access;

3. packet bumping -- according to CAP5 specification, the PRU buffer management scheme ensures radio reception. If all buffers are occupied, it frees a buffer for radio reception by discarding a packet in the transmit queue. If there is no packet available in the transmit queue, then reception is not guaranteed and the arriving packet is dropped;

4. when the PRU processor is busy, no radio transmission can be initialized. Upon completion of processing, the packet that have been scheduled for transmission, if any, since the beginning of this processing period is transmitted;

5. also simulated is the manual flow control mechanism enforcing a minimum interval of TRMIDY msec between two consecutive packet transmissions over the 1822 wire interface from a TIU to a PRU.

IV. Recommended Usage

Each of the three simulators of this package simulates PRNET at a different level of details. Owing to its important role as the switch of PR subnet, the characterization and optimization of PRU functions are of prime interest in the PRNET design process. BB simulator provides the capability for studying PRU functions such as scheduling, retransmission scheme, and the impact of PRU processing speed. Reducing complexity of the overall system by using probabilistic distribution functions to condense the environment of a PRU, we may expose the relevant issues concerning the reactions of a PRU to its environment.

SLP provides capabilities for preliminary studies at network level. In addition to studying the impact of PRU functions on network performance, we may also study the effect of different hop acknowledgment schemes. Without details for channel access and packet collision allows us to investigate network behavior without concerning ourselves with the constraints of a particular configuration. On the other hand, the impact of a specific configuration can be embedded in the specification of input parameters for success transmission probabilities, rate of overhead traffic, and channel access delays. Proper design of an experiment may also allow the use of SLP to simulate a portion of a large network by specifying the input parameters reflecting the presence of the remaining network.

For studying networks featuring a specific protocol, the details of such a protocol need to be incorporated. SL incorporates details of those aspects CAP5 protocol considered to have significant impact on the network throughput delay performance. These aspects are described in the last section. This simulator is intended for the use of quantitative studies of the current PRNET implementation.

Accuracy of a simulation increases with complexity, while it trades off with efficiency. The design of this package has restricted its usage to studying steady-state behaviors of packet transport functions. A simulation for studying functions, such as network congestion control, monitoring, mobile operations, require substantial dynamic interactions among different network entities. Following the building-block approach for this simulation package, it is conceivable to construct a network simulator for such purpose. It is, however, not the intention of the design of this simulation.

V. Program Structure

The PRNET System-Level Simulation is a discrete-time simulation with a millisecond time resolution. It has a modulized design to accommodate possible modifications and additions. The PRNET System-Level Simulation consists of a set of processes. Each simulates a distinct system operation.

The simulation programs are event driven. Each event corrensponds to the execution of a process. The future events are linked together according to their chronological order forming an event chain. At the completion of executing an event, the event chain is searched for the next event. Whenever the need of a future operation is recognized, during the processing of an event, a new event is created, scheduled, and linked to the event chain. The initialization event generates the first packet, schedules an event for it, and thus initiates the simulation.

In the following, we give a brief description for each process (flow diagrams for a number processes are included in Appendix I):

-- DRIVER: The driver takes off the event on top of the event list and sends it to the appropriate procedure for processing. 'ENDSIM' and the output interval of this simulator can be specified by the simulated time or the number of packets generated. The driver stops processing events when 'ENDSIM' is reached.

-- ADDEVENT: It adds a future event onto the event list in a chronological order.

-- ARRIVAL: It generates the arrival of the next packet and determines which route it will take according to the user specified probabilities, RTEPROBs.

-- ARRIVED (see flow diagram): Entering this process signifies that the packet has arrived at a PRU. A packet coming from a terminal must have waited at least TRMIDY msec since the last transmission from that TIU. If the packet is starting it's route, then call ARRIVAL to generate the next packet. If the buffer is full and there is a packet on the transmit queue not transmitting or waiting for a channel, and the arriving packet did not come from a TIU, then the packet on the transmit queue is bumped. If the buffer is full and no packet can be bumped, then the arriving packet is dropped; otherwise the arriving packet is placed on the receiving queue, and processed if it is the only packet on the receiving queue.

-- PROCESS: The packet is now being processed. The time it takes to process the packet depends on whether the packet is an intended user packet.

-- PROCESSED (see flow diagram): The packet has now been processed. If another packet is waiting for transmission, then transmit it first. If the packet is going to the terminal and this is it's first processing, then it must be reprocessed. If the packet is an acknowledgement packet then the procedure ACK is called. If the packet is not intended, then it is dropped. If the packet is going to a TIU, then end-to-end and/or one-way statistics are collected, if desired. The intended packet is then placed on the transmit queue and if it is the first packet on the transmit queue, it is transmitted. If there are more packets on the receiving queue, then the first packet on the queue is processed.

-- TRANSMIT (see flow diagram): The packet is now ready for transmission. If a packet is currently being processed, then the packet must wait until the packet has finished processing so it can use the CPU. If it is the first time the packet is to be s nt, then the packet must wait for a period of RTXDLY msec before being transmitted. If the packet is going to the TIU, then a packet is sent to the TIU where it will be processed and sent back to the next PRU on the route. The channel is then sensed, and if busy, the packet must wait until the end of the transmission before it can sense again. The packet is then transmitted to all members in it's hearing regions and will arrive safely if there are no collisions.

-- TRANSMITTED: The packet has now been transmitted. If the acknowledgement is returned successfully, then a packet is generated and arrives at the queue of the waiting PRU. If the packet successfully reaches it's destination, then hop delay statistics are collected if necessary, and either the packet or a duplicate arrives at the PRU. Unintended packets, if successful, arrive at the other PRUs in the hearing region. If the packet received it's acknowledgement during the transmission, then the packet is dropped off the queue, otherwise it waits for the acknowledgement.

-- TRANSWAIT: The channel was busy when the PRU tried to transmit earlier, and now the other transmission is complete. The PRU must now wait for a short period before sensing the channel again.

-- ACK (see flow diagram): The acknowledgement packet has been processed. If the packet waiting for the acknowledgement is not on the transmit queue, then nothing happens. Otherwise next hop statistics are collected, if desired. Th packet on the transmit queue is then dropped, unless it is in the process of being transmitted, upon where it must wait until the end of transmission to be dropped.

-- ONTXQ: It finds the packet on the transmit queue with the same ID as another designated packet, and returns a pointer to this packet if found, and a null pointer otherwise.

-- BUFFER: It decrements the buffersize and collects statistics for average queue size if necessary.

-- DELETE: It deletes the packet from the queue and calls BUFFER to decrement the buffersize.

-- SCHEDULE (see flow diagram): The next transmission may be scheduled depending on the conditions. If an ack occurred, or the wait period is up, and a FIFO scheme is used, then we transmit the top packet. If a packet has just been transmitted, or all the packets were sent the previous time and a cyclic scheme is used, then we search through the queue for the next packet to be sent. If a packet had just been transmitted, then we wait for a period before transmitting the next otherwise we transmit now. If none of the above conditions are satisified, then nothing happens.

For SL, execution efficiency is in the range of 10 - 100 simulated time to simulation execution time ratio on UCLA-IBM3033 depending upon network topology and traffic distribution. The core memory it requires for execution is about 150K also depending on configuration of the simulated network.

The only system-specific program module is the random number generator used. It needs to be replaced when these programs are use on a system other than that of UCLA-IBM3033.

VI. Statistics Collection

The statistics collection interval can be specified either in terms of the number of generated packets or of simulated time (msec). Statistics can be optionally discarded for an initial simulation period by specifying a non-zero value for BEGINSIM. The accumulated statistics is reported at the specified interval in addition to a cumulative final report. Most statistics are repo ted in terms of average, standard deviation, and maximum. Hist grams is provided as an option. The statistics reported include:

1. Pairwise Hop Statistics (in matrix form)

-- packets transmitted
-- dropped packets
-- discarded packets
-- packets bumped
-- rate of collision
-- unintended packets
-- duplicated packets
-- transmissions/packet
-- beyond success transmissions/packet
-- # of channel attempts/transmission
-- success transmission probability (link Q)
-- hop delay
-- throughput

2. PRU statistics

-- buffer occupancy (over time)

3. Route Statistics

-- one-way delay
-- one-way throughput
-- round-trip delay
-- round-trip throughput
-- ETE packet loss

VII. Input/Output Specification

1. Image of Input Files

Each input file consists of two sections. The first section has fixed entries. In the second section, input is specified in groups. Those parameters need to be specified for each PRU (route, or other entities) are grouped as such. Each input data format can be free or fixed. For those with free format, identifiable by the delimiters '=' preceeding it and a ';' following it, the data must be entered between the delimiters. For numerical data, every blank between the delimiters is filled with a zero. The input images for each of the three simulators are as follows:

-- BB Simulator

```
Line#
  1 | 111111 121111 131111 141111 151111 161111 171111 181111 191111 1011:
  2 |SCHEDULING=CYCLIC;    HGRAM   =0;     PKTS=0;PKTFLOW=0;       DB=0;
  3 |BUFFERMAX=          5;  TRANSMAX=          6;  NOOFPRUS=         4;
  4 |PRINT_INT=      60000;  BEGINSIM=          0;  ENDSIM    =   60000;
  5 |AMEAN      =      55.;  PFMEAN    =       10;  PNFMEAN  =         3;
  6 |PKTLEN     =      272;  TRSPD     =      100;  ACKPROB  =       .95;
  7 |RTXDLY     =      8.2;  RETXDY    =    10.24;  NRT_MIN  =        11;
  8 |NRT_MEAN  =       155;  SCHWAIT  =       8.2;  MINTIME  =       15.;
  9 |MAXTIME    =       15;  BKTSIZE=       15.;
 10 |ORIGPROB=0.25;  DESTPROB=0.15;  SUCCPROB=0.95;
 11 |ORIGPROB=0 25;  DESTPROB=0.15;  SUCCPROB=0.95;
 12 |ORIGPROB=0.25;  DESTPROB=0.15;  SUCCPROB=0.95;
 13 |ORIGPROB=0.25,  DESTPROB=0.15;  SUCCPROB=0.95;
```

-- SLP Simulator

```
Line#
|SCHEDULING=CYCLIC;    HGRAM   =1;    PKTS=1;PKTFLOW=0;      DB=0;
|NOOF_PRUS=          4;  PKTLEN   =         1;  TRSPD    =     100;
|NOOF_RTES=          2;  LGST_RTE=          7;  TRMIDY   =      10;
|TRANSMAX =          6;  RTXDLY   =       8.2;  RETXDY   =   10.24;
|AMEAN    =        400;  PFMEAN   =        10;  PNFMEAN  =       3;
|PRINT_INT=        200;  BEGINSIM=        100;  ENDSIM   =     500;
|MINTIME  =        100;  MAXTIME =       1500;  BKTSIZE =      100;
|TERMMAX  =         10;  SARR     =    111111;  SRTE     =  999999;
|BUFFERMAX=    6;  LAMBDA =     10;  DMEAN   =    100;  EMEAN   =    7.2;
|BUFFERMAX=    6;  LAMBDA =     10;  DMEAN   =    100;  EMEAN   =    7.2;
|BUFFERMAX=    6;  LAMBDA =      0;  DMEAN   =400000;  EMEAN   =    7.2;
|BUFFERMAX=    6;  LAMBDA =      5;  DMEAN   =    200;  EMEAN   =    7.2;
| 111111 121111 131111 141111 151111 161111 171111 181111
| 211111 221111 231111 241111 251111 261111 271111 281111
| 311111 321111 331111 341111 351111 361111 371111 381111
| 411111 421111 431111 441111 451111 461111 471111 481111
|     1.00     1.00     .9724    1.00
|     1.00     1.00     .9724    1.00
|     1.00     1.00   1.00       1.00
|     1.00     1.00     .9724    1.00
|0 1 3 4 0 4 3 1 0
|0 2 3 4 0 4 3 2 0
| .5  .5
```

-- SL Simulator

```
Line#
 |SCHEDULING=CAP5.0;  HGRAM    =0;  PKTS=1;PKTFLOW=0;  DB=0;  DEBUG=1;
 |NOOF_PRUS=           4;  SARR      =  11111111;  TRSPD     =        100;
 |NOOF_RTES=           6;  LGST_RTE=          5;  TRMIDY    =         10;
 |NOOF_REGS=           1;  RTXDLY    =        8.2;  RETXDY    =      10.24;
 |AMEAN     =         15.;  PFMEAN    =         10;  PNFMEAN =          3;
 |PRINT_INT=     111350;  BEGINSIM=          0;  ENDSIM    =        300;
 |MINTIME   =          50;  MAXTIME =        750;  BKTSIZE =         50;
 |TRANSMAX =           6;  CMEAN     =          2;  SRTE      =      55555;
 |TTRSPD    =         50.;  TERMMAX =         10;
 |BUFFERMAX=    6;  EMEAN    =    7.2;
 |BUFFERMAX=    6;  EMEAN    =    7.2;
 |BUFFERMAX=    6;  EMEAN    =    7.2;
 |BUFFERMAX=    6;  EMEAN    =    7.2;
 |  111111  121111  131111  141111
 |  211111  221111  231111  241111
 |  311111  321111  331111  341111
 |  411111  421111  431111  441111
 |0 1 4 0 4 1 0
 |0 2 4 0 4 2 0
 |0 3 4 0 4 3 0
 |0 4 1 0 1 4 0
 |0 4 2 0 2 4 0
 |0 4 3 0 3 4 0
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |RTEPROB   =       .167;  PKTLEN    =          1;
 |    '1'B   '1'B   '1'B   '1'B
 |    1   1   1   1
 |    1   1   1   1
 |    1   1   1   1
 |    1   1   1   1
```

2. Annotation

-- Fixed Entries

```
Line 1 input of BB Simulator contains seeds for 10 random
              number generators used in that simulator
SCHEDULING - '  FIFO', or 'CYCLIC' (CAP5.0)
HGRAM      - 1 if histograms desired
             0 if histograms not desired
PKTS       - 1 if simulation run by number of packets generated
             0 if simulation run by time
PKTFLOW    - 1 for event tracing
             0 for no event tracing
DB         - 1 for debugging output
             0 for no debugging output
DEBUG      - 1 for detailed debugging output
             0 for no detailed debugging output
NOOF_PRUS  - number of PRUs to be simulated
NOOFFRUS   - number of neighboring PRUs (BB)
PKTLEN     - packet length (user words)
TRSPD      - radio transmission speed  (kbits/sec)
ACKPROB    - probability for a successful ack transmission
NOOF_RTES  - number of routes
LGST_RTE   - longest route (number of PRUs)
TRMIDY     - 1822 flow control parameter (msec)
TRANSMAX   - maximum number of radio transmissions per packet
NOOF_REGS  - number of regions within each of which all PRUs
             can hear each other
RTXDLY     - minimum interval between radio transmissions
RETXDY     - retransmission delay (time-out period) increment
AMEAN      - average packet interarrival time (one traffic
             source generating packets for all routes)
PFMEAN     - average PRU processing time for intended packets
PNFMEAN    - average PRU processing time for unintended packets
PRINT_INT  - output interval (msec or pkts).
             From BEGINSIM on, statistics are summarized every
             PRINT_INT msec or pkts.
BEGINSIM   - beginning of statistics collection (msec or pkts)
ENDSIM     - end of simulation (msec or pkts)
NRT_MIN    - minimum time from transmission of a packet to
             return of its acknowledgement
NRT_MEAN   - mean time from transmission of a packet to return
             of its acknowledgement
SCHWAIT    - minimum interval between radio transmissions (same
             as RTXDLY, a redundant specification)
MINTIME    - lower bound for histogram
MAXTIME    - upper bound for histogram
BKTSIZE    - histogram bucket size
TERMMAX    - maximum number of packets can be buffered in a TIU
TTRSPD     - transmission speed over 1822 wire interface
SARR       - seed for packet generations (large, odd integer)
SRTE       - seed to determine route taken by a packet (large,
```

odd integer)

```
CMEAN       - mean time for channel access
BUFFERMAX   - buffer pool size
LAMBDA      - offerred channel traffic rate (packets/sec)
DMEAN       - average interarrival rate for unintended packets
EMEAN       - average time spent processing in TIU
ORIGPROB    - probability of a PRU from which a packet is
              received
DESTPROB    - probability of a PRU to which a packet is forwarded
              the remaining packets are assumed unintended
SUCCPROB    - success transmission probability from the simulated
              PRU to a neighboring PRU
RTEPROB     - probability for a newly generated packet to take
              a specific route.
```

-- Group Entries:

Group I (PRU specifications)
Particulars of each PRU

Group II (seeds for random number generators)
The seeds must be large, odd integers, preferably different.

Group III of of SLP (success transmission probabilities)
A set of inputs in an (n x n) matrix, where n is the number of PRUs, and the (i,j)th entry is the success transmission probability from the ith to the jth PRU.

Group IV of SLP and Group III of SL (route specification)
Each route must begin and end with a '0'.
Each route must be of length LGST_RTE+2 (one '0' on each end)
(pad shorter routes with '0's at the end)
The specification of each route must begin on a new line.
A round-trip route (for end-to-end acknowledged packets) is
specified with a '0' in between the two way specification.

Group V of SLP (route probabilities)
For each route is specified a probability that a packet will take this route.
The total sum of the probabilities must be 1.

Group IV of SL (route specification)
packets are allowed to be of different lengths for each route

Group V of SL (power level)

OUTPUT STATISTICS:

SIMULATION TIME - current simulation time, the time the last event occured.

PACKETS GENERATED - the number of packets that have been generated by the traffic generator.

PRU Statistics (one column per PRU):

AVE QUEUE SIZE - a time average over the interval between the last change in queue size before the previous output and the last change before the current output.

MAX QUEUE SIZE - maximum queue sizes of the PRUs for the same inteval as that of the AVERAGE QUEUE SIZE.

ARRIVING PKTS STILL IN TIU - generated packets waiting, in the TIU attached to this PRU, for entering the PR subnet.

PACKETS DROPPED FROM TIU - packets dropped when sent from the attached TIU due to buffer overflow in this PRU.

END TO END ACKS DROPPED FROM TIU - end-to-end acknowledgements dropped when sent from the attached TIU due to buffer overflow in this PRU.

Hop Transport Statistics (presented in matrix form):

PACKETS TRANSMITTED - # of packets transmitted from one PRU to the other.

END TO END ACKS TRANSMITTED - # of end-to-end acknowledgements (as oppose to forwarding packets) transmitted.

DROPPED PKTS FROM TOO MANY TX - packets dropped because no acknowledgement was received after the maximum allowed # of transmissions.

DROPPED PACKETS FROM BUFFER OVF - packet tx dropped due to receiving PRU buffer overflow.

PKTS BUMPED FROM TX Q OF RX-PRU - packets discarded in the transmit queue of the receiving PRU to assure radio reception.

NUMBER OF COLLISIONS - number of collisions involving a packet transmitted on this link.

UNINTENDED PACKETS RECEIVED - unintended packets transmitted on this link.

DUPLICATE PACKETS RECEIVED - duplicate transmissions for intended packets.

AVE NUMBER OF TX BEYOND SUCCESS - per packet aveage of # of retransmissions beyond that tx which successfully reaches the next PRU.

AVE NUMBER OF TX PER PKT - ave # of total txs per packet.

AVERAGE # OF CHANNEL ATTEMPTS - average number of channel attempts per packet.

SUCCESS TX PROBABILITY - success transmission probability (or link quality) for a link from a PRU to another.

Statistics gathered for each of the following times include its AVERAGE, STANDARD DEVIATION, and MAXIMUM.

HOP DELAY - delay from the time a packet arrives at a PRU until it successfully reaches the next PRU.

ACKNOWLEDGEMENT TIME - the time from the first transmission of a packet until the reception of its acknowledgement. The (i, j) entry is the acknowledgement time for packets from PRU i to PRU j.

ONE-WAY END-TO-END DELAYS

FORWARD - interval from the time the packet's generation until the end of its arrival at the destination TIU.

RETURN - interval from the end-to-end acknowledgement leaving destination TIU until its arrival at the originating TIU.

ROUND-TRIP DELAY - interval from the generation of a packet until the completion of the processing of its end-to-end ack at the PRU connecting the originating TIU. This is the sum of forward and return one-way delays, time spent waiting for transmission from both the PRU and the TIU, and the processing time in TIU.

HISTOGRAMS: Histograms are optional. It is produced when the input value for parameter 'HGRAM' is '1'. The required statistics are collected throughout the collection period. When the option for histogram is specifed, they are printed once at the end together with the overall statistics. Histograms provided are for round-trip and end-to-end delays.

VIII. An Example (using SL Simulator)

A 4-PRU network is simulated using SL. Its configuration is depicted in Figure I. PRU#4 is assumed to be connected to a gateway. Traffic flows through three routes: (1) from TIU#1 to the gateway, (2) response traffic from the gateway to TIU#1, and (3) from TIU#2 to the gateway. The traffic on route (3) is end-to-end acknowledged. The total network traffic rate is of 10 packets/second, equivalent to an average interarrival interval of 100 msec. It is then portioned into 30%, 30%, and 40% among the three routes.

A copy of the simulation output is attached in Appendix II. It first summarizes the input parameters for the run. Follwing that is the statistics for the three 5 sec intervals. The final portion of the printout is a summary of the overall statistics together with histograms. Statistics for an initial period of 5 seconds (specified by 'BEGINSIM') is discarded.

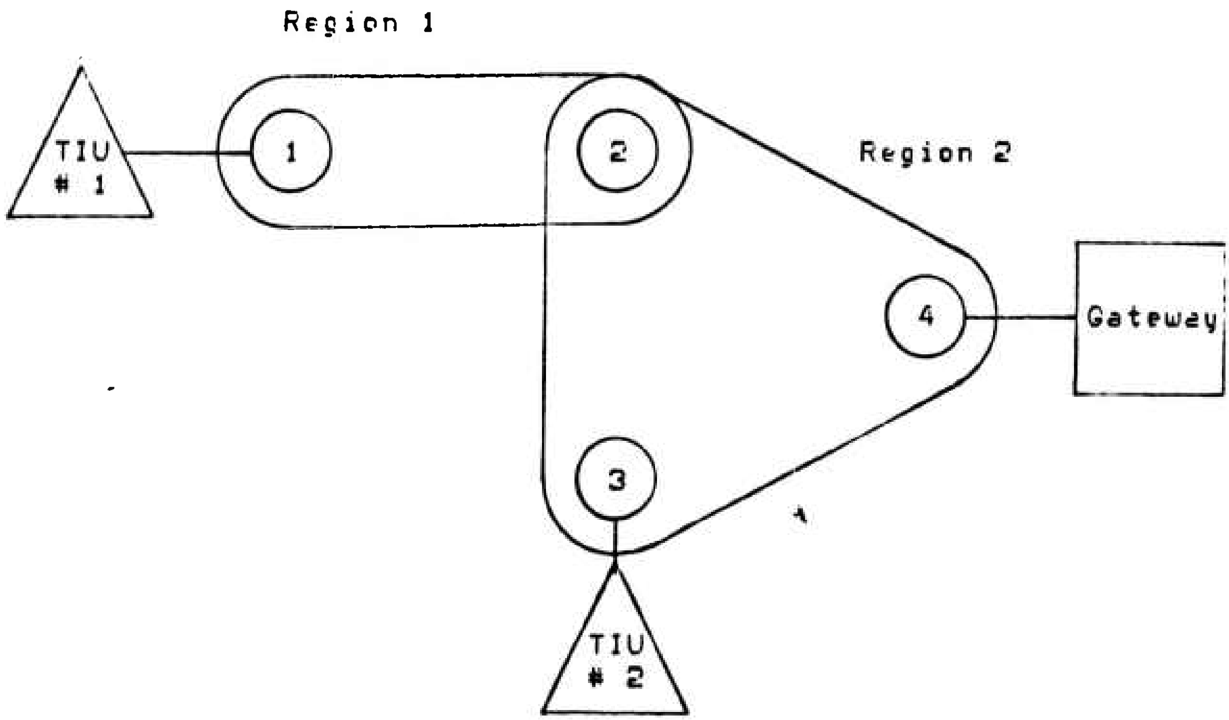


Figure I

Acknowledgement

The programming efforts for this simulation package by Nancy Ishii and her support in preparing this document are greatly appreciated.

Appendix I Flow Diagrams

Abbreviations used in the following flow diagrams:

-- TX = transmission or transmit

-- RX = receiving

-- Q = queue

-- TOP = time out period

-- pkt = packet

-- S = start of the process

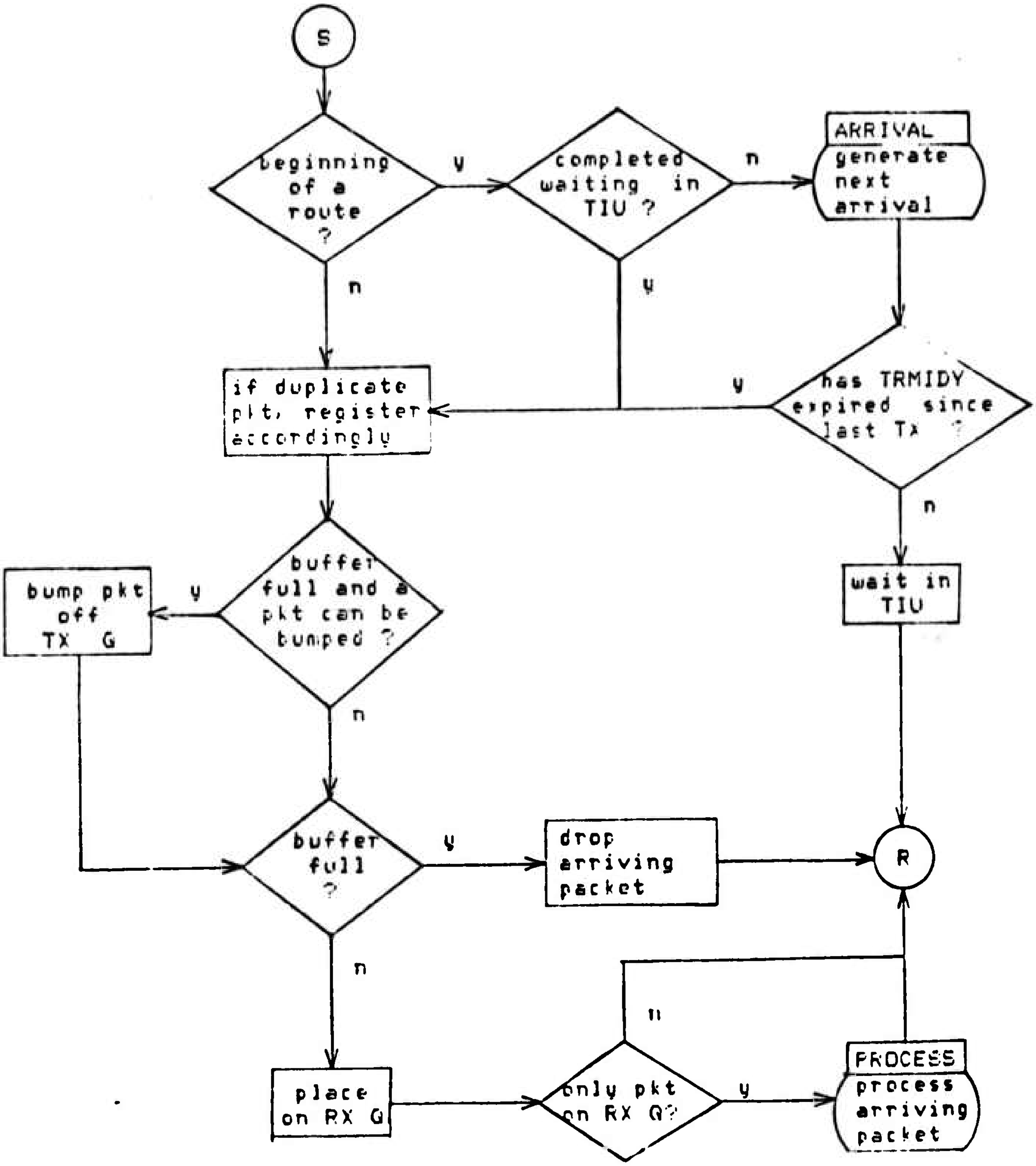
-- R = end of the process, returning to the calling process

-- ack = acknowledgement

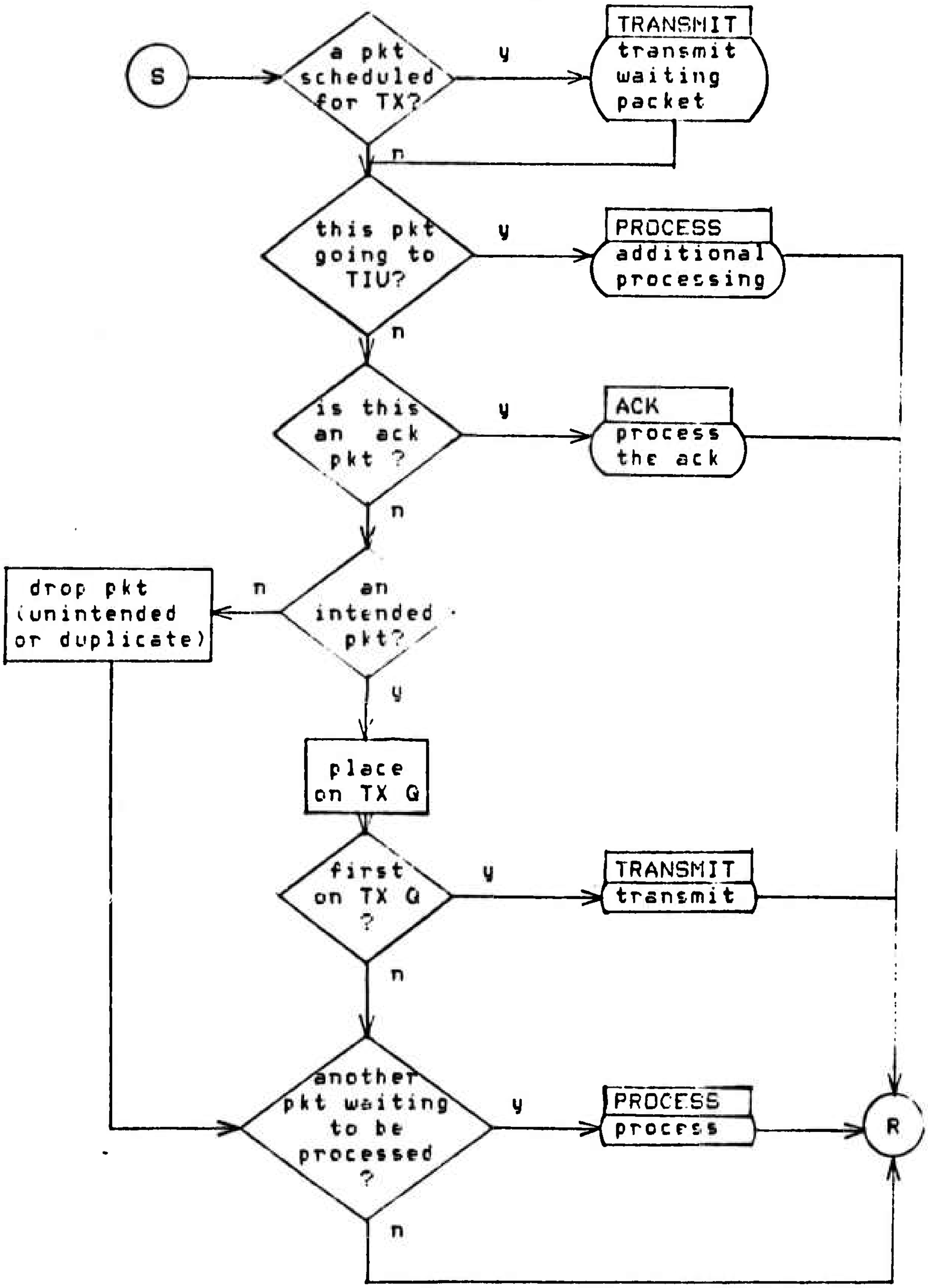
-- ack'd = acknowledged

-- stats = statistics

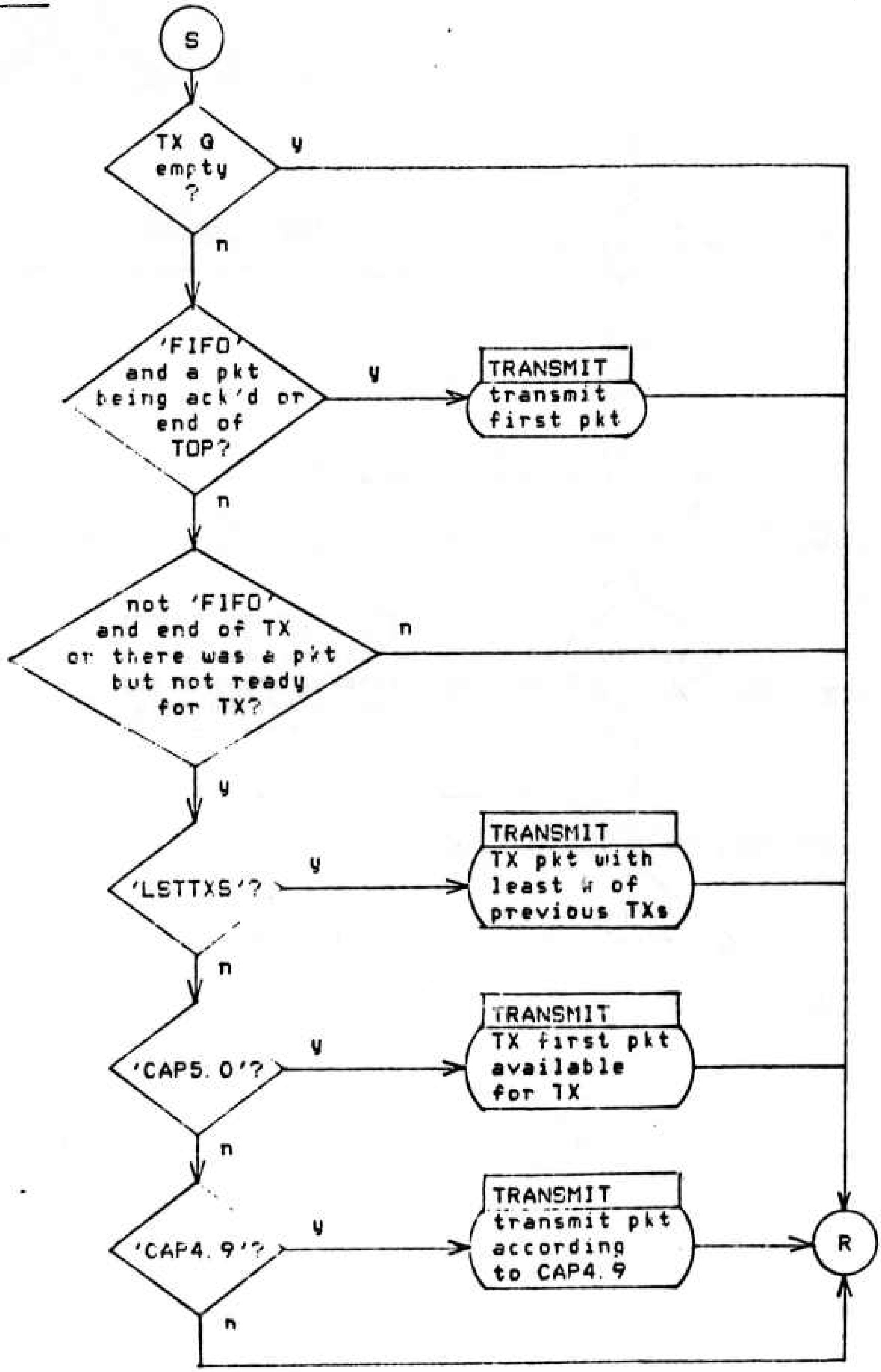
ARRIVED


S
beginning of a route ?
y
completed waiting in TIU ?
n
ARRIVAL
generate next arrival
n
y
if duplicate pkt, register accordingly
y
has TRMIDY expired since last Tx ?
n
wait in TIU
buffer full and a pkt can be bumped ?
y
bump pkt off TX Q
n
buffer full ?
y
drop arriving packet
R
n
place on RX Q
only pkt on RX Q?
y
PROCESS
process arriving packet

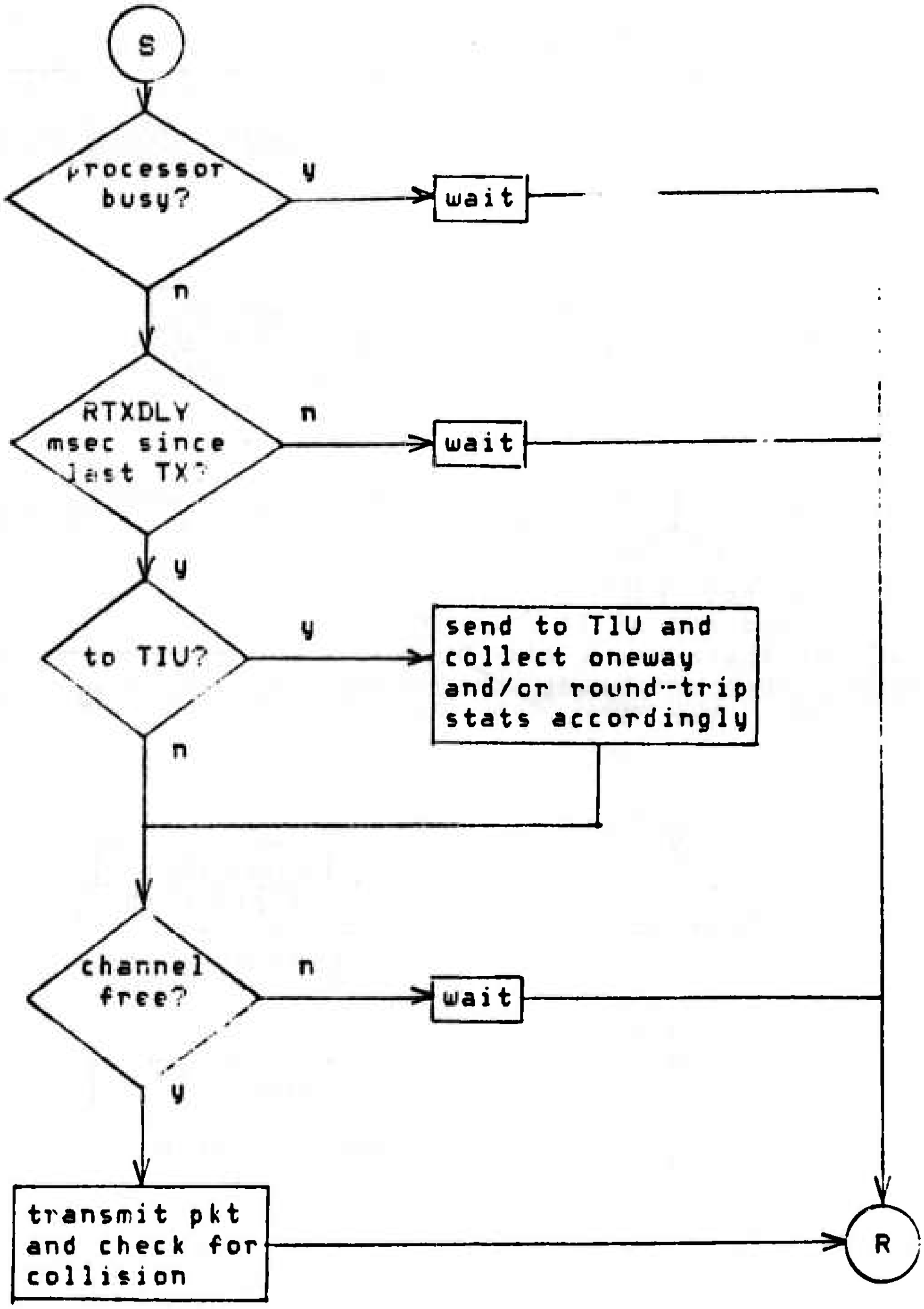
PROCESSED


S
a pkt scheduled for TX?
y
TRANSMIT
transmit waiting packet
n
this pkt going to TIU?
y
PROCESS
additional processing
n
is this an ack pkt ?
y
ACK
process the ack
n
an intended pkt?
n
drop pkt (unintended or duplicate)
y
place on TX Q
first on TX Q ?
y
TRANSMIT
transmit
n
another pkt waiting to be processed ?
y
PROCESS
process
R
n

SCHEDULE

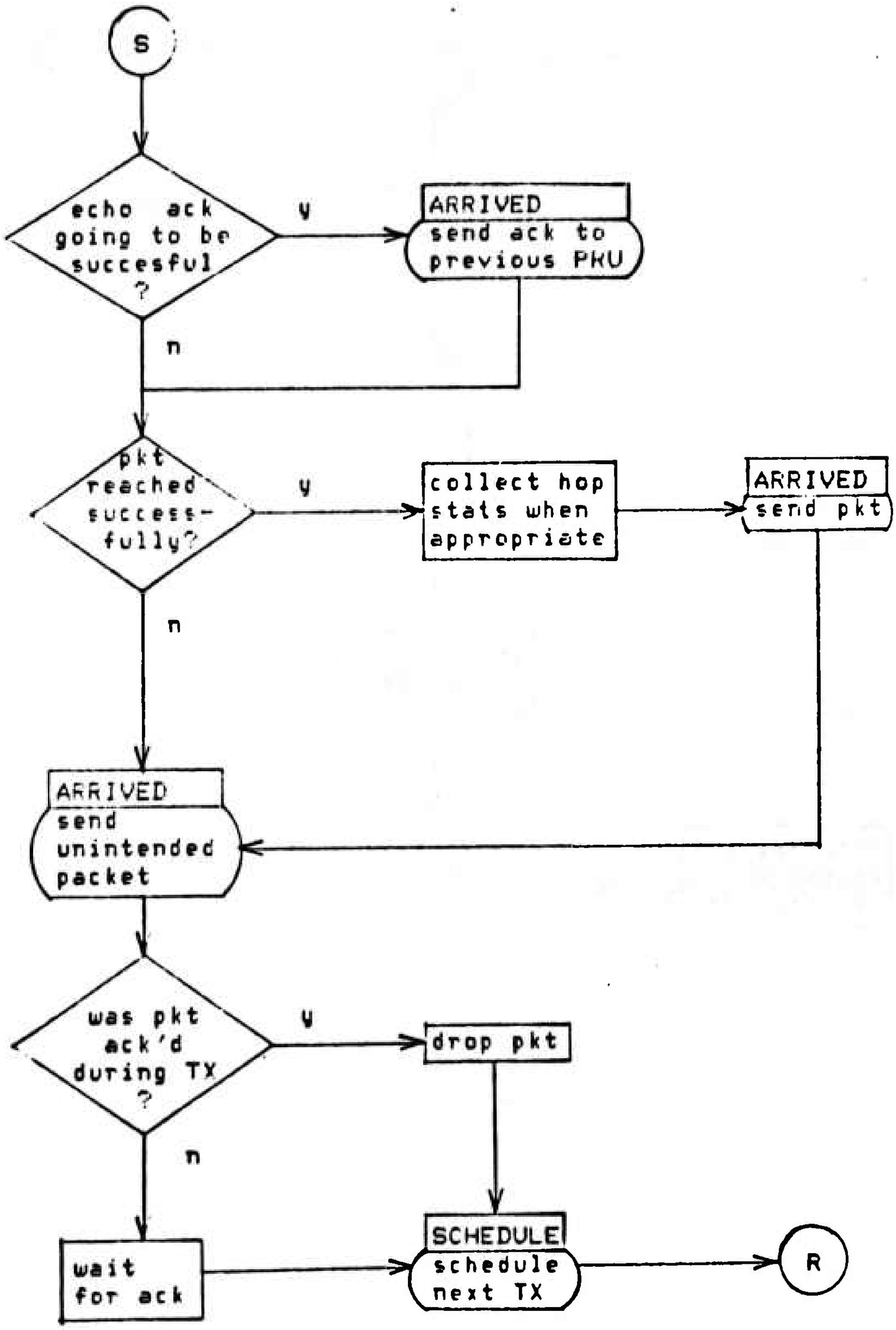

S
TX Q empty ?
y
n
'FIFO' and a pkt being ack'd or end of TOP?
y
TRANSMIT
transmit first pkt
n
not 'FIFO' and end of TX or there was a pkt but not ready for TX?
n
y
'LSTTXS'?
y
TRANSMIT
TX pkt with least # of previous TXs
n
'CAP5.0'?
y
TRANSMIT
TX first pkt available for TX
n
'CAP4.9'?
y
TRANSMIT
transmit pkt according to CAP4.9
n
R

TRANSMIT

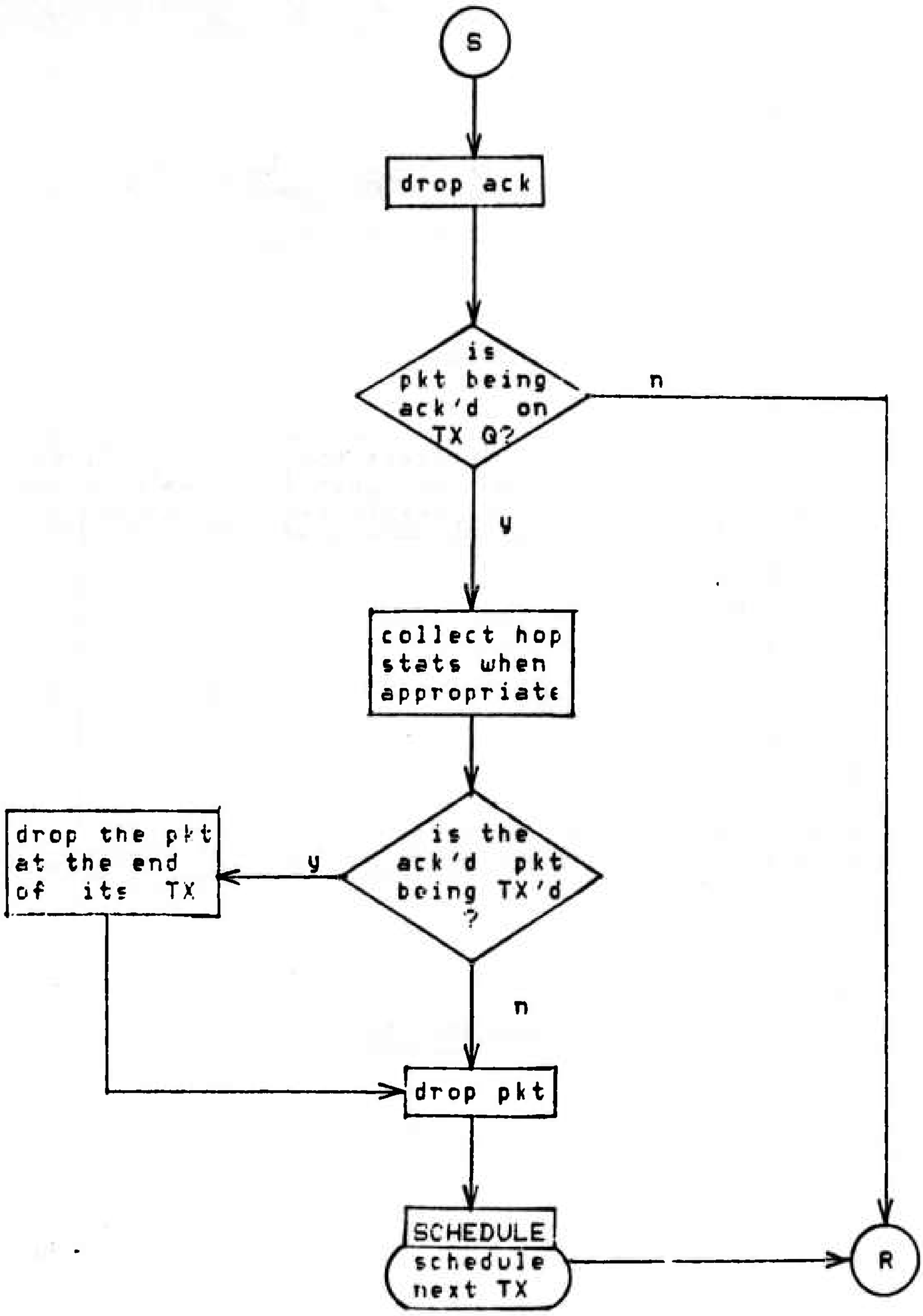


S
processor busy?
y
wait
n
RTXDLY msec since last TX?
n
wait
y
to TIU?
y
send to TIU and collect oneway and/or round-trip stats accordingly
n
channel free?
n
wait
y
transmit pkt and check for collision
R

TRANSMITTED



ACK


S
drop ack
is pkt being ack'd on TX Q?
n
y
collect hop stats when appropriate
is the ack'd pkt being TX'd ?
y
drop the pkt at the end of its TX
n
drop pkt
SCHEDULE
schedule next TX
R

APPENDIX II

```
//FYI648  .      GH,5,000,ID=NETEXEC.R','NANCY'
// EXEC           GI      OK
//STEPLIB DD DI        ),DSN=  .I648.DGH.LIBRARY
//SYSPRINT DD SY      =A,SPACE=(TRK,(20,20),RLSE)
//INFILE DD DISP=OLD,DSN=FYI648.DGH.DATA42
//FT06F001 DD SYSOUT=A,
//  DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//
IEF236I ALLOC. FOR FYI648EA
IEF237I 363    ALLOCATED TO STEPLIB
IEF237I 716    ALLOCATED TO SYSPRINT
IEF237I 363    ALLOCATED TO INFILE
IEF237I 173    ALLOCATED TO FT06F001
```

```
SIMULATION INPU   PAMETER,  (TIME SPECIFIED IN MSEC. UNLESS O/W SPECIFIED)

PRU PARAMETERS
TYPE OF SCHEDULING                                           CAP5.0
# BUFFERS IN                       PRU # 1                        6
                                   PRU # 2                        6
                                   PRU # 3                        6
                                   PRU # 4                        6
MAXIMUM NUMBER OF TRANSMISSIONS PER PACKET                        6
AVE PROCESSING TIME FOR INTENDED PACKETS                      10.00
AVE PROCESSING TIME FOR UNINTENDED PACKETS                     3.00
TX SPEED - PRU/PRU (KBITS/SEC)                               100.00
TX SPEED - TO/FROM TERMINAL (KBITS/SEC)                       50.00
1822 FLOW CONTROL PARAMETER--TRMIDY (MSEC)                    10.00
MIN INTERVAL BETWEEN RADIO TXS--RTXDLY                         8.20
RETRANSMISSION DELAY INCREMENT--RETXDY                        10.24
(HOP ACK TIME OUT PERIOD=RTXDLY+RETXDY*THE NUMBER OF TRANSMISSIONS)

CONFIGURATIONS AND ROUTE SPECIFICATIONS
NUMBER OF PRUS                                                    4
REGION MEMBERS (1 IF MEMBER, 0 IF NOT)
REGION #/PRU #         1     2     3     4
       1               1     1     0     0
       2               0     1     1     1

NUMBER OF ROUTES                                                  3
LONGEST ROUTE (# OF PRUS ON ROUTE)                                5

ROUTE #                          1       2       3
TRAFFIC DISTRIBUTION         0.300   0.300   0.400
ROUTES:
                                 4       1       3
                                 2       2       4
                                 1       4       0
                                 0       0       4
                                 0       0       3

INPUT TRAFFIC SPECIFICATIONS
AVERAGE INTERARRIVAL TIME                                    100.00
CHANNEL ACCESS RANDOMIZATION MEAN                              2.00
AVERAGE PACKET LENGTH (16 BIT WORDS)
    ROUTE # 1                                                  1.00
    ROUTE # 2                                                  1.00
    ROUTE # 3                                                100.00
MEAN TIME SPENT IN THE TIU CONNECTING TO:
    PRU   # 1                                                  7.20
    PRU   # 2                                                  7.20
    PRU   # 3                                                  7.20
    PRU   # 4                                                  7.20

EXPERIMENT DURATION AND OUTPUT FREQUENCY SPECIFICATIONS
SIMULATION RUN BY TIME
BEGINNING OF STATISTICS COLLECTION                             5000
END OF SIMULATION                                             20000
```

PRINT INTERVAL	5000

JGRAMS

HISTOGRAMS PRINTED	
LOWER BOUND FOR HISTOGRAM	25
UPPER BOUND FOR HISTOGRAM	500
BUCKET SIZE	25

ATION TIME	10003.17			
ACKET RADIO UNIT #	1	2	3	4
AVE QUEUE SIZE	0.55	0.75	0.87	1.32
MAX QUEUE SIZE	6	6	6	6
ARRIVING PKTS STILL IN TIU	0	0	0	0
PACKETS DROPPED FROM TIU	0	0	0	0
END TO END ACKS DROPPED FROM TIU	0	0	0	2

PACKET RADIO UNIT # 1				
PACKETS TRANSMITTED	0	12	0	0
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	10	0	0
UNINTENDED PACKETS RECEIVED	0	0	0	0
DUPLICATE PACKETS RECEIVED	0	1	0	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.50	0.00	0.00
AVE NUMBER OF TX PER PKT	0.00	1.83	0.00	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.05	0.00	0.00
SUCCESS TX PROBABILITY	0.00	0.83	0.00	0.00
*** HOP DELAY ***				
AVERAGE	0.00	20.85	0.00	0.00
STANDARD DEVIATION	0.00	19.20	0.00	0.00
MAXIMUM	0.00	74.92	0.00	0.00
THROUGHPUT	0	12	0	0
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	40.80	0.00	0.00
STANDARD DEVIATION	0.00	30.06	0.00	0.00
MAXIMUM	0.00	104.25	0.00	0.00
THROUGHPUT	0	12	0	0

PACKET RADIO UNIT # 2				
PACKETS TRANSMITTED	22	0	0	14
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	2	6
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	6
NUMBER OF COLLISIONS	0	0	0	0
UNINTENDED PACKETS RECEIVED	0	0	84	0
DUPLICATE PACKETS RECEIVED	6	0	0	6
AVE NUMBER OF TX BEYOND SUCCESS	1.00	0.00	0.00	1.86
AVE NUMBER OF TX PER PKT	2.00	0.00	0.00	2.86
AVERAGE # OF CHANNEL ATTEMPTS	1.09	0.00	0.00	1.07
SUCCESS TX PROBABILITY	1.00	0.00	0.00	1.00
*** HOP DELAY ***				
AVERAGE	20.73	0.00	0.00	20.66
STANDARD DEVIATION	17.03	0.00	0.00	18.38
MAXIMUM	55.84	0.00	0.00	61.11
THROUGHPUT	22	0	0	14

*** ACKNOWLEDGEMENT TIME ***

AVERAGE	47.41	0.00	0.00	86.84
STANDARD DEVIATION	29.98	0.00	0.00	58.92
MAXIMUM	104.83	0.00	0.00	219.34
THROUGHPUT	21	0	0	14

PACKET RADIO UNIT # 3

PACKETS TRANSMITTED	0	0	0	12
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	6	0	0
UNINTENDED PACKETS RECEIVED	0	43	0	0
DUPLICATE PACKETS RECEIVED	0	0	0	14
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.00	0.00	1.50
AVE NUMBER OF TX PER PKT	0.00	0.00	0.00	2.50
AVERAGE # OF CHANNEL ATTEMPTS	0.00	0.00	0.00	1.09
SUCCESS TX PROBABILITY	0.00	0.00	0.00	1.00

*** HOP DELAY ***

AVERAGE	0.00	0.00	0.00	50.57
STANDARD DEVIATION	0.00	0.00	0.00	23.87
MAXIMUM	0.00	0.00	0.00	90.20
THROUGHPUT	0	0	0	13

*** ACKNOWLEDGEMENT TIME ***

AVERAGE	0.00	0.00	0.00	120.14
STANDARD DEVIATION	0.00	0.00	0.00	60.16
MAXIMUM	0.00	0.00	0.00	224.01
THROUGHPUT	0	0	0	12

PACKET RADIO UNIT # 4

PACKETS TRANSMITTED	0	18	0	0
END TO END ACKS TRANSMITTED	0	0	11	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	2	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	2	3	0
NUMBER OF COLLISIONS	0	6	0	0
UNINTENDED PACKETS RECEIVED	0	39	52	0
DUPLICATE PACKETS RECEIVED	0	3	4	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.38	1.33	0.00
AVE NUMBER OF TX PER PKT	0.00	1.38	2.33	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.48	1.19	0.00
SUCCESS TX PROBABILITY	0.00	0.92	1.00	0.00

*** HOP DELAY ***

AVERAGE	0.00	39.25	41.47	0.00
STANDARD DEVIATION	0.00	53.75	32.76	0.00
MAXIMUM	0.00	209.65	117.21	0.00
THROUGHPUT	0	18	11	0

*** ACKNOWLEDGEMENT TIME ***

AVERAGE	0.00	30.94	66.98	0.00
STANDARD DEVIATION	0.00	23.73	53.25	0.00
MAXIMUM	0.00	88.69	194.37	0.00
THROUGHPUT	0	16	9	0

ROUTE #	1	2	3
*** ONE WAY . IO END JELAY ***			
FORWARD			
AVERAGE	96.75	103.06	142.19
STANDARD DEVIATION	60.75	45.65	64.24
MAXIMUM	275.46	185.09	257.66
THROUGHPUT	18	12	13
RETURN			
AVERAGE	0.00	0.00	97.58
STANDARD DEVIATION	0.00	0.00	49.82
MAXIMUM	0.00	0.00	218.82
THROUGHPUT	0	0	10
*** ROUND TRIP DELAY ***			
AVERAGE	0.00	0.00	242.42
STANDARD DEVIATION	0.00	0.00	88.60
MAXIMUM	0.00	0.00	343.0[illegible]
THROUGHPUT	0	0	10

SIMULATION TIME 15003.59

PACKET RADIU UNIT #	1	2	3	4
AVE QUEUE SIZE	0.26	0.39	0.65	0.80
MAX QUEUE SIZE	4	4	6	6
ARRIVING PKTS STILL IN TIU	0	0	0	0
PACKETS DROPPED FROM TIU	0	0	0	0
END TO END ACKS DROPPED FROM TIU	0	0	0	1

PACKET RADIC UNIT # 1

PACKETS TRANSMITTED	0	17	0	0
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	5	0	0
UNINTENDED PACKETS RECEIVED	0	0	0	0
DUPLICATE PACKETS RECEIVED	0	4	0	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.29	0.00	0.00
AVE NUMBER OF TX PER PKT	0.00	1.47	0.00	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.00	0.00	0.00
SUCCESS TX PROBABILITY	0.00	0.89	0.00	0.00
*** HOP DELAY ***				
AVERAGE	0.00	16.57	0.00	0.00
STANDARD DEVIATION	0.00	23.14	0.00	0.00
MAXIMUM	0.00	101.23	0.00	0.00
THROUGHPUT	0	17	0	0
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	25.49	0.00	0.00
STANDARD DEVIATION	0.00	22.80	0.00	0.00
MAXIMUM	0.00	111.40	0.00	0.00
THROUGHPUT	0	17	0	0

PACKET RADIO UNIT # 2

PACKETS TRANSMITTED	17	0	0	17
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	1	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	0	0	0
UNINTENDED PACKETS RECEIVED	0	0	60	0
DUPLICATE PACKETS RECEIVED	3	0	0	9
AVE NUMBER OF TX BEYOND SUCCESS	0.56	0.00	0.00	0.94
AVE NUMBER OF TX PER PKT	1.56	0.00	0.00	1.94
AVERAGE # OF CHANNEL ATTEMPTS	1.22	0.00	0.00	1.12
SUCCESS TX PROBABILITY	1.00	0.00	0.00	1.00

*** HOP DELAY ***

AVERAGE	14.78	0.00	0.00	13.15
STANDARD DEVIATION	9.21	0.00	0.00	6.76
MAXIMUM	36.60	0.00	0.00	28.71
THROUGHPUT	17	0	0	17

*** ACKNOWLEDGEMENT TIME ***

AVERAGE	32.76	0.00	0.00	34.86
STANDARD DEVIATION	19.30	0.00	0.00	17.33
MAXIMUM	86.61	0.00	0.00	71.83
THROUGHPUT	16	0	0	17

PACKET RADIO UNIT # 3

PACKETS TRANSMITTED	0	0	0	13
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	1
NUMBER OF COLLISIONS	0	5	0	0
UNINTENDED PACKETS RECEIVED	0	44	0	0
DUPLICATE PACKETS RECEIVED	0	0	0	16
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.00	0.00	1.38
AVE NUMBER OF TX PER PKT	0.00	0.00	0.00	2.38
AVERAGE # OF CHANNEL ATTEMPTS	0.00	0.00	0.00	1.14
SUCCESS TX PROBABILITY	0.00	0.00	0.00	1.00

*** HOP DELAY ***

AVERAGE	0.00	0.00	0.00	44.73
STANDARD DEVIATION	0.00	0.00	0.00	23.01
MAXIMUM	0.00	0.00	0.00	88.98
THROUGHPUT	0	0	0	13

*** ACKNOWLEDGEMENT TIME ***

AVERAGE	0.00	0.00	0.00	100.34
STANDARD DEVIATION	0.00	0.00	0.00	66.25
MAXIMUM	0.00	0.00	0.00	241.51
THROUGHPUT	0	0	0	13

PACKET RADIO UNIT # 4

PACKETS TRANSMITTED	0	14	0	0
END TO END ACKS TRANSMITTED	0	0	13	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	1	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	0	0	0

UNINTE... PACKETS RECEIVED	0	41	43	0
DUPLI... PACKETS RECEIVED	0	2	4	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.36	0.92	0.00
AVE NUMBER OF TX PER PKT	0.00	1.36	1.92	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.32	1.20	0.00
SUCCESS TX PROBABILITY	0.00	1.00	1.00	0.00
*** HOP_DELAY ***				
AVERAGE	0.00	40.37	36.02	0.00
STANDARD DEVIATION	0.00	43.41	20.16	0.00
MAXIMUM	0.00	156.33	87.08	0.00
THROUGHPUT	0	14	13	0
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	21.69	50.96	0.00
STANDARD DEVIATION	0.00	10.77	31.14	0.00
MAXIMUM	0.00	41.28	101.25	0.00
THROUGHPUT	0	14	12	0

ROUTE #	1	2	3
*** ONE WAY END TO END DELAY ***			
FORWARD			
AVERAGE	79.78	63.87	94.75
STANDARD DEVIATION	45.87	33.82	55.00
MAXIMUM	190.93	169.46	241.48
THROUGHPUT	13	17	12
RETURN			
AVERAGE	0.00	0.00	88.00
STANDARD DEVIATION	0.00	0.00	39.91
MAXIMUM	0.00	0.00	169.99
THROUGHPUT	0	0	12
*** ROUND TRIP DELAY ***			
AVERAGE	0.00	0.00	182.95
STANDARD DEVIATION	0.00	0.00	71.74
MAXIMUM	0.00	0.00	334.30
THROUGHPUT	0	0	12

SIMULATION TIME 20012.40

PACKET RADIO UNIT #	1	2	3	4
AVE QUEUE SIZE	0.85	0.76	0.76	1.15
MAX QUEUE SIZE	6	6	6	6
ARRIVING PKTS STILL IN TIU	0	0	0	0
PACKETS DROPPED FROM TIU	0	0	0	1
END TO END ACKS DROPPED FROM TIU	0	0	0	1

<u>PACKET RADIO UNIT # 1</u>

PACKETS TRANSMITTED	0	21	0	0
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	2	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q CF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	35	0	0

UNINTENDED PACKETS RECEIVED	0	0	0	0
DUPLICATE PACKETS RECEIVED	0	3	0	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.37	0.00	0.00
AVE NUMBER OF TX PER PKT	0.00	1.79	0.00	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.09	0.00	0.00
SUCCESS TX PROBABILITY	0.00	0.56	0.00	0.00
*** HOP DELAY ***				
AVERAGE	0.00	54.57	0.00	0.00
STANDARD DEVIATION	0.00	72.38	0.00	0.00
MAXIMUM	0.00	292.51	0.00	0.00
THROUGHPUT	0	20	0	0
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	46.92	0.00	0.00
STANDARD DEVIATION	0.00	44.87	0.00	0.00
MAXIMUM	0.00	171.40	0.00	0.00
THROUGHPUT	0	19	0	0

PACKET RADIO UNIT # 2

PACKETS TRANSMITTED	12	0	0	22
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	2	0
PKTS BUMPED FROM TX Q OF RX-PRU	1	0	3	4
NUMBER OF COLLISIONS	0	0	0	0
UNINTENDED PACKETS RECEIVED	0	0	82	0
DUPLICATE PACKETS RECEIVED	8	0	0	9
AVE NUMBER OF TX BEYOND SUCCESS	2.31	0.00	0.00	0.86
AVE NUMBER OF TX PER PKT	3.31	0.00	0.00	1.86
AVERAGE # OF CHANNEL ATTEMPTS	1.39	0.00	0.00	1.27
SUCCESS TX PROBABILITY	1.00	0.00	0.00	1.00
*** HOP DELAY ***				
AVERAGE	26.67	0.00	0.00	23.11
STANDARD DEVIATION	22.83	0.00	0.00	18.25
MAXIMUM	83.15	0.00	0.00	62.88
THROUGHPUT	12	0	0	22
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	108.69	0.00	0.00	44.43
STANDARD DEVIATION	95.71	0.00	0.00	31.06
MAXIMUM	394.82	0.00	0.00	123.09
THROUGHPUT	13	0	0	22

PACKET RADIO UNIT # 3

PACKETS TRANSMITTED	0	0	0	16
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	2
NUMBER OF COLLISIONS	0	8	0	0
UNINTENDED PACKETS RECEIVED	0	9	0	0
DUPLICATE PACKETS RECEIVED	0	0	0	18
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.00	0.00	1.21
AVE NUMBER OF TX PER PKT	0.00	0.00	0.00	2.21
AVERAGE # OF CHANNEL ATTEMPTS	0.00	0.00	0.00	1.17
SUCCESS TX PROBABILITY	0.00	0.00	0.00	1.00

*** HOP DELAY ***				
AVERAGE	0.00	0.00	0.00	36.82
STANDARD DEVIATION	0.00	0.00	0.00	13.75
MAXIMUM	0.00	0.00	0.00	62.20
THROUGHPUT	0	0	0	15

*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	0.00	0.00	81.74
STANDARD DEVIATION	0.00	0.00	0.00	37.82
MAXIMUM	0.00	0.00	0.00	153.37
THROUGHPUT	0	0	0	14

PACKET RADIO UNIT # 4				
PACKETS TRANSMITTED	0	13	0	0
END TO END ACKS TRANSMITTED	0	0	14	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	2	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	1	0
NUMBER OF COLLISIONS	0	9	0	0
UNINTENDED PACKETS RECEIVED	0	35	50	0
DUPLICATE PACKETS RECEIVED	0	4	1	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.55	0.79	0.00
AVE NUMBER OF TX PER PKT	0.00	1.55	1.79	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.84	1.25	0.00
SUCCESS TX PROBABILITY	0.00	0.94	1.00	0.00

*** HOP DELAY ***				
AVERAGE	0.00	43.14	30.00	0.00
STANDARD DEVIATION	0.00	37.96	16.97	0.00
MAXIMUM	0.00	130.14	64.54	0.00
THROUGHPUT	0	12	14	0

*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	30.69	49.14	0.00
STANDARD DEVIATION	0.00	17.56	28.71	0.00
MAXIMUM	0.00	65.74	110.02	0.00
THROUGHPUT	0	11	14	0

ROUTE #	1	2	3
*** ONE_WAY_END_TO_END DELAY ***			
FORWARD			
AVERAGE	131.13	120.64	91.88
STANDARD DEVIATION	55.36	103.19	39.96
MAXIMUM	251.61	466.53	161.79
THROUGHPUT	13	20	16
RETURN			
AVERAGE	0.00	0.00	81.31
STANDARD DEVIATION	0.00	0.00	32.85
MAXIMUM	0.00	0.00	159.73
THROUGHPUT	0	0	14
*** ROUND TRIP DELAY ***			
AVERAGE	0.00	0.00	171.50
STANDARD DEVIATION	0.00	0.00	80.73
MAXIMUM	0.00	0.00	401.21

THROUGHPUT	0	0	14	

OVERALL STATISTICS

SIMULATION TIME	20012.40			
PACKET RADIO UNIT #	1	2	3	4
AVE QUEUE SIZE	0.54	0.63	0.76	1.09
MAX QUEUE SIZE	6	6	6	6
ARRIVING PKTS STILL IN TIU	0	0	0	0
PACKETS DROPPED FROM TIU	0	0	0	1
END TO END ACKS DROPPED FROM TIU	0	0	0	4

PACKET RADIO UNIT # 1

PACKETS TRANSMITTED	0	50	0	0
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	2	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q OF RX-PRU	0	0	0	0
NUMBER OF COLLISIONS	0	50	0	0
UNINTENDED PACKETS RECEIVED	0	0	0	0
DUPLICATE PACKETS RECEIVED	0	8	0	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.38	0.00	0.00
AVE NUMBER OF TX PER PKT	0.00	1.69	0.00	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	2.05	0.00	0.00
SUCCESS TX PROBABILITY	0.00	0.73	0.00	0.00
*** HOP DELAY ***				
AVERAGE	0.00	33.13	0.00	0.00
STANDARD DEVIATION	0.00	52.28	0.00	0.00
MAXIMUM	0.00	292.51	0.00	0.00
THROUGHPUT	0	49	0	0
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	37.80	0.00	0.00
STANDARD DEVIATION	0.00	36.00	0.00	0.00
MAXIMUM	0.00	171.40	0.00	0.00
THROUGHPUT	0	48	0	0

PACKET RADIO UNIT # 2

PACKETS TRANSMITTED	51	0	0	53
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	5	6
PKTS BUMPED FROM TX Q OF RX-PRU	1	0	3	10
NUMBER OF COLLISIONS	0	0	0	0
UNINTENDED PACKETS RECEIVED	0	0	226	0
DUPLICATE PACKETS RECEIVED	17	0	0	24
AVE NUMBER OF TX BEYOND SUCCESS	1.20	0.00	0.00	1.15
AVE NUMBER OF TX PER PKT	2.20	0.00	0.00	2.15
AVERAGE # OF CHANNEL ATTEMPTS	1.23	0.00	0.00	1.16
SUCCESS TX PROBABILITY	1.00	0.00	0.00	1.00

*** HOP DELAY ***

AVERAGE	20.15	0.00	0.00	19.27
STANDARD DEVIATION	17.20	0.00	0.00	16.15
MAXIMUM	83.15	0.00	0.00	62.88
THROUGHPUT	51	0	0	53
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	58.65	0.00	0.00	52.56
STANDARD DEVIATION	61.62	0.00	0.00	43.03
MAXIMUM	394.82	0.00	0.00	219.34
THROUGHPUT	50	0	0	53
PACKET RADIO UNIT # 3				
PACKETS TRANSMITTED	0	0	0	41
END TO END ACKS TRANSMITTED	0	0	0	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	0	0
PKTS BUMPED FROM TX Q CF RX-PRU	0	0	0	3
NUMBER OF COLLISIONS	0	29	0	0
UNINTENDED PACKETS RECEIVED	0	126	0	0
DUPLICATE PACKETS RECEIVED	0	0	0	48
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.00	0.00	1.36
AVE NUMBER OF TX PER PKT	0.00	0.00	0.00	2.36
AVERAGE # OF CHANNEL ATTEMPTS	0.00	0.00	0.00	1.13
SUCCESS TX PROBABILITY	0.00	0.00	0.00	1.00
*** HOP DELAY ***				
AVERAGE	0.00	0.00	0.00	43.69
STANDARD DEVIATION	0.00	0.00	0.00	21.22
MAXIMUM	0.00	0.00	0.00	90.20
THROUGHPUT	0	0	0	41
*** ACKNOWLEDGEMENT TIME ***				
AVERAGE	0.00	0.00	0.00	99.76
STANDARD DEVIATION	0.00	0.00	0.00	57.74
MAXIMUM	0.00	0.00	0.00	241.51
THROUGHPUT	0	0	0	39
PACKET RADIO UNIT # 4				
PACKETS TRANSMITTED	0	45	0	0
END TO END ACKS TRANSMITTED	0	0	38	0
DROPPED PKTS FROM TOO MANY TX	0	0	0	0
DROPPED PACKETS FROM BUFFER OVF	0	0	5	0
PKTS BUMPED FROM TX Q CF RX-PRU	0	2	4	0
NUMBER OF COLLISIONS	0	15	0	0
UNINTENDED PACKETS RECEIVED	0	115	145	0
DUPLICATE PACKETS RECEIVED	0	9	9	0
AVE NUMBER OF TX BEYOND SUCCESS	0.00	0.41	0.97	0.00
AVE NUMBER OF TX PER PKT	0.00	1.41	1.97	0.00
AVERAGE # OF CHANNEL ATTEMPTS	0.00	1.54	1.21	0.00
SUCCESS TX PROBABILITY	0.00	0.95	1.00	0.00
*** HOP DELAY ***				
AVERAGE	0.00	40.67	35.38	0.00
STANDARD DEVIATION	0.00	46.66	24.03	0.00
MAXIMUM	0.00	209.65	117.21	0.00
THROUGHPUT	0	44	38	0
*** ACKNOWLEDGEMENT TIME ***				

```
AVERAGE                            0.00     27.71     54.35     0.00
STANDARD D                         0.00     19.00     38.04     0.00
MAXIMUM                            0.00     88.69    194.37     0.00
THROUGHPUT                            0        41        35        0


ROUTE #                               1         2         3

  *** ONE WAY END TO END DELAY ***
   FORWARD
AVERAGE                          101.90     96.64    108.67
STANDARD DEVIATION                58.69     76.62     57.79
MAXIMUM                          275.46    466.53    257.66
THROUGHPUT                           44        49        41

   RETURN
AVERAGE                            0.00      0.00     88.06
STANDARD DEVIATION                 0.00      0.00     41.03
MAXIMUM                            0.00      0.00    218.82
THROUGHPUT                            0         0        36

  *** ROUND TRIP DELAY ***
AVERAGE                            0.00      0.00    195.02
STANDARD DEVIATION                 0.00      0.00     85.55
MAXIMUM                            0.00      0.00    401.21
THROUGHPUT                            0         0        36

THROUGHPUT (PKTS/MSEC)             0.001799
AVE DELAY TIME (MSEC)                195.02

ROUTE #    1

ROUND TRIP DELAY TIME

    0    0         0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  500

ONE WAY END TO END DELAY TIME

 FORWARD

    0   11    7    6    4    8    3    2    1    0    1    1    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  500

 RETURN

    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  500

 ROUTE #    2
```

ROUND TRIP DELAY TIME

```
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

ONE WAY END TO END DELAY TIME

FORWARD

```
     0   12   13   11    1    3    4    1    1    1    1    0    0    0    0    0    0    0    1    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

RETURN

```
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

ROUTE #   3

ROUND TRIP DELAY TIME

```
     0    0    0    4    5    3    4    6    5    0    1    1    2    4    0    0    1    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

ONE WAY END TO END DELAY TIME

FORWARD

```
     0    3   12    8    5    5    3    1    0    3    1    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

RETURN

```
     0    8    6   10    7    1    3    0    1    0    0    0    0    0    0    0    0    0    0    0
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
0   25   50   75  100  125  150  175  200  225  250  275  300  325  350  375  400  425  450  475  50
```

```
IEF14?I  STEP EXECUTED - COND CODE 1000
 285I    FYI648.DGH.LIBRARY                                    KEPT
  85I    VOL SER NOS= DATI31.
  85I    SYS80036.T114236.SV000.FYI648EA.R0000001              SYSOUT
IEF285I  VOL SER NOS= SYSDA2.
IEF285I  FYI648.DGH.DATA42                                     KEPT
IEF285I  VOL SER NOS= DATI31.
IEF285I  SYS80036.T114236.SV000.FYI648EA.R0000002              DELETED
IEF285I  VOL SER NOS= SYSDA3.

CCN301I  SYSPRINT PRINTED    13 TOTAL PAGES, REQUIRED     9 TRACKS.

   REGION ALLOCATION      CORE USED
         200K               164K

     STEP CPU TIME        JOB CPU TIME       STEP I/O COUNT       JOB I/O COUNT
         1.41S               1.41S                713                  713

********** END OF JOB **********
```

10-Sep-79 19:15:15-PDT,6898;000000000001
Date: 10 Sep 1979 1915-PDT
From: ZSU
Subject: PTP Route Assignment & PRU Power Control
To: Westcott at BBN-TENEXA
cc: Cerf at ISI, Beeler at BBN-TENEXA, ZSu

Hi, Jil

I have looked at the two problems you suggested to us. I am writing to keep you informed on the preliminary conclusions drawn from our study. Your feedback and/or further input will be very appreciated.

PTP Route Assignment

The two issues that need be considered here are the rate of packet collisions, and the amount of PRU processing for intended and unintended packets.

First, we observe that routing traffic through a single path should generate no fewer collisions than would splitting the traffic through two parallel adjacent paths. Along two adjacent paths the PRUs are assumed pairwise in hearing range with each other. Let us consider a configuration of three consecutive pairs of PRUs, (i-1)th, ith, (i+1)th, a typical configuration along two parallel adjacent paths. Ignoring the remaining network for the time being, and assuming perfect CSMA operations, we may say that a collision occurs over the ith PRU pair if and only if a packet transmission from the (i-1)th pair overlaps with one from the (i+1)th pair. Since two PRUs in a pair can hear each other, presumably they won't transmit at the same time. Then, the probability of such a collision won't change noticably when the PRU pairs are collapsed into single PRUs. In fact, packet destruction due to collision might be reduced by routing traffic through parallel paths if the PRU hearing ranges are restricted so that the PRUs on different paths cannot hear each other unless they belong to the same pair. In such a case, a collision over the ith PRU pair occurs only if the overlapped packet transmissions were from the PRUs on the same path (one PRU of the (i-1)th pair and one of the (i+1)th pair).

To verify this observation, I have simulated three configurations studying the cases of a single-source, single-sink, three-hop network. We vary the input traffic rate for each configuration and observe the round-trip network behavior (delay vs throughput). As a typical PRNET behavior, the throughput is observed to retreat at high input traffic rate. Thus a network capacity, defined as the maximum throughput over the input rate, is obtained for each configuration. Parallel-path routing over two adjacent routes (50%-50% traffic splitting) has a 35% advantage over single-path routing. This advantage increases to 45% when we restrict the hearing ranges of the PRUs as we described above. These advantages may be exaggerated due to the lack of other traffic on the network.

When the network is processing bound, distributing the processing load is desirable. Parallel-path routing appears to do exactly that. The even distribution of intended packets achieved by parallel-path routing is obvious. The number of unintended packets generated by a transmission depends on the number of neighbors of the transmitting PRU. The number of packets to be transmitted over the network is independent of routing. Assuming the same number of hops for alternative routes and uniform geographic density of PRUs, the total number of unintended packets and therefore their processing in a network should remain relatively constant. Using a similar argument, we may conclude that parallel-path routing may also help to distribute the channel load taking advantage of spatial reuse of the channel in multi-hop environment.

The above discussion has been restricted to the consideration of steady-state network behavior. Ideally, dynamic behavior should also be studied.

PRU Power Control

For this question, let us consider a 3-PRU subnet consisting of two high-powered repeater PRUs, R1 and R2, and a terminal PRU, T, with T and R1 hidden from each other. Collisions may thus occur between packets from R1 and T. Let Pr and Pt be two collided packets from R1 and T respectively. If Pr is a little ahead of Pt, Pr may survive the collision, saved by its stronger signal. If Pt is ahead of Pr, the receiver would have been synchronized with Pt when the preemble of Pr arrives. The receiver would thus lose both packets. According to this scenario, we see that while the success transmission probability (STP) from terminal-PRU T to repeater R2 remains unchanged from that of equal-powered transmission, the STP from repeater R1 to repeater R2 is increased from exp(-2*RT*t) to exp(-RT*t), where RT is the offered traffic rate from T, and t the packet length in time. We are in the process of modifying the system-level simulation to accommodate the requirements for studying the subject of PRU power control. I will get back to you on this.

From the above discussion, it appears PRU power control would increase STP from repeater to repeater and hence reduce collisions. It would be a 'pure' gain in channel utilization without affecting processing load in any obvious way. For a processing bound system, however, it may not have much effect in network performance.

Another issue which may deserve attention is that the differential in power level may result in one-way communication, namely, a terminal-PRU would hear a repeater but not the other way. Referring to the 3-PRU example cited earlier, with CSMA the terminal-PRU, T, may not waste its transmission if it can hear repeater R1 transmitting. In this case, the one-way communication adds to the advantages of PRU power control. Likewise, it may inhibit a terminal-PRU unneccessarily. I cannot think of any intrinsic difficulty which may arise in dealing with this issue. But it may worth your attention for implementation.

In related works, Kleinrock and Silvester have obtained the 'magic number six', and in a subsequent paper* they considered PRUs operating at different power levels. Their studies pointed out that a happy medium for PRU transmission power exists, and reducing interference may increase network throughput. Unfortunately, their quantitative results are based upon slotted ALOHA and other assumptions such as pairwise 'partnership' among PRUs. Their study may point out meaningful directions for PRNET studies, while direct application is not recommended here.

*'On the Capacity of ALOHA Packet Radio Networks for Local Traffic Matrices', presented at 12th Hawaii International Conference on System Science, Honolulu, January 1979, pp. 231-237.

Regards,

Zaw-Sing

18-Sep-79 10:13:26-PDT,5348;000000000001
Date: 18 Sep 1979 1013-PDT
From: ZSU
Subject: Simulation Study of Ft. Bragg Configuration
To: Frankel at SRI-KL
cc: Beeler at BBN-TENEXA, Cerf at ISI, Chu,
cc: Kunzelman at SRI-KL, Quilici, ZSu

Dear Mike,

Thank you for the detailed information on the initial Ft. Bragg deployment. We have derived from it a PRNET model. Using this model, we ran our simulation to estimate the performance of the initial Ft. Bragg PRNET deployment. I would like to brief you on the initial results we have obtained, and to continue a dialogue with you in estimating and predicting the performance behavior of the PRNET at Ft. Bragg as well as an integral part of our effort in studying PRNET characteristics in general. I will start with a description of the model we are using, and state the assumptions we have made. Then, I will present our simulation results. We are continuing our study to estimate the impact of a number of variations in the traffic distribution, system parameters, and network configuration. These planned studies are outlined below.

The model we use is a 4-PRU network, 3 for connecting the Corp HQ TIU, the Division HQ TIU, and the hot-standby/training TIU, and one for the integrated station/gateway. Given that all PRUs are in line-of-sight of each other (fully connected), we, for this initial study, did not take into account the existing repeater. We have also assumed negligible traffic from NCC during normal operation. For input traffic distribution, we assume that each terminal generates the same amount of traffic for a host via the gateway. The host generates response traffic at the same rate it receives from each TIU. Letting one unit of input traffic be generated by each terminal, we have an input traffic distribution of 2 units from the training TIU, 4 units each from the Corp HQ and the Division HQ TIUs, and 10 units of response traffic from the station/gateway. The packet length for both user and response traffic is assumed to be one text word. Other assumptions we have made include:

- CAP5 cyclic scheduling algorithm;

- CSMA channel access mode;

- A PRU buffer pool of 6 buffers shared between the processing & transmit queues (including packets waiting for acknowledge);

- a maximum of 6 transmissions per packet with no alternate routing;

- The average PRU processing times for intended and unintended packets are 10 msec and 3 msec respectively;

- No processing is required for either active acknowledgement generation or acknowledgement processing;

- The channel access randomization has a mean of 2 msec;

- Both wire and radio transmissions are assumed to be at 100 kbps;

- The TIU processing time is assumed to have a mean of 7.2 msec;

- No ETE acknowledgement or ETE time-out;

- The flow control parameter (TRMIDY) is reduced from the default value of 82 msec to 10 msec so it would not interfere with the behavior of the PRNET itself;

- The initial transmission delay (RTXDLY) and the retransmission delay increment (RETXDY) are set at their default values of 8.2 msec and 10.24 msec respectively.

Using this model and parameters described above, we simulated the network with varying input rates. The network peformance is characterized in terms of end-to-end (ETE) round-trip (RT) delay vs throughput (see table below). The throughput is observed to retreat at high input rates as expected. The network capacity, defined as the maximum throughput over input rate, is thus obtained at about 15 packets/sec (i.e., 0.75 packets/sec of user traffic per terminal). The ETE RT delays appear to be excessively high. Looking at statistics collected for other parameters, we have noticed that the congestion occured at the station/gateway PRU (the obvious bottleneck of this network configuration) and was causing buffer overflow and excessive retransmissions (from other PRUs). It indicates the usefulness of packet bumping in the current PRU implementation. We are taking steps to include bumping in our simulation.

Beyond our current study, we are planning to use the same configuration and traffic distribution for studying the following:

- the impact of varying RETXDY and RTXDLY,

- the impact of longer response packets,

- the impact of lower PRU processing times (for the IPRs),

Other studies we are planning that vary from this configuration and traffic distribution include:

- the separation of the gateway from the station (exported gateway),

- varying the number of TIUs (2, 3, 5, 10),

- varying the response traffic rate from that of input traffic,

- a multi-hop configuration (in anticipation of using the existing repeater to connect more distant TIUs).

We will keep you posted of our progress. In the mean time, please keep us abreast of your observations.

Network Input (pkts/sec)	12.46	15.07	17.22	20.09	30.14
Throughput (pkts/sec)	11.57	13.81	14.98	14.87	13.66

ETE RT Delay (msec)	164.6	200.2	225.7	283.1	379.2

Regards,

Zaw-Sing

Some Results on a Simulation Study of CAP4.9 Transmission Scheme

Zaw-Sing Su
UCLA
July 2, 1979

I. Introduction

In PRTN #265, Jil Westcott of BBN pointed out that the FIFO transmission scheme used for the PRU may be inefficient. According to that transmission scheme, a PRU would allow at most one packet unacknowledged at any one time regardless for which neighboring PRU the packet is intended. A second packet would not be transmitted until the first packet is either properly acknowledged, or dropped after a maximum number of retransmissions. Westcott suggested to allow more than one unacknowledged packets simultaneously outstanding. In CAP 4.9, a transmission scheme, often referred to as the cyclic transmission scheme, is constructed. This transmission scheme allows more than one packet to be simultaneously unacknowledged, but at any one time there may be at most one outstanding packet intended for each neighboring PRU.

In this PRTN, we report some simulation results on a study of the CAP 4.9 transmission scheme, using the building-block simulator [PRTN#268]. The results presented include the throughput improvement of the CAP4.9 transmission scheme over the FIFO transmission scheme, the effects of success transmission probabilities, and of the initial transmission delay, RTXDLY. First, we describe the simulation model and the parameters used. We then present and discuss the simulation results.

II. The Building-Block Simulation Model

The building-block simulator simulates the functions of one PRU. A PRU is modeled as consisting of two servers in series: a processor and a transmitter (Figure 1). The input, or 'received', traffic is of Poisson distribution with a constant packet length of 17 words (i.e., 272 bits). Upon arrival they form a queue for processing, and a queue for transmission after the completion of processing. The total number of buffers for these two queues is limited by the size of the common buffer pool, which is set at 5. Packets received are placed in the processing queue. According to preassigned probabilities, each packet is assigned a 'previous PRU' ID and a 'next PRU' ID. The previous PRU is the PRU from which the packet is to have been received; and the next PRU is the PRU to which this packet is to be forwarded. A zero next PRU ID indicates that this received packet is not intended for the receiving PRU. Throughout this study, we evenly distribute the intended packets among the next PRUs. Depending on whether it is intended, a packet after being received incurs a processing delay from one of two distributions. In this study, the processing times for intended and unintended packets are exponentially distributed with means equal to 10 msec and 3 msec respectively.

After being processed, an intended packet is ready for transmission. After a packet has been transmitted, the PRU waits a specified time out period for an acknowledgement. The time out periods, i.e., the transmission and retransmission delays, are structured the same as in CAP 4, based on the current default values for RTXDLY (8.2 msec) and RETXDY (10.24 msec). The CSMA channel access mode is simulated by a random number of channel sensings, with a range of [1, 5], and a random delay between two channel sensings, with a range of (0.5, 2] msec. After the transmission of a packet, an acknowledgement may be received in an exponentially distributed next-hop turn-around time (NRT). A probability (FTSP), as a function of the neighboring PRU, is assigned to the successful transmission of a forwarded packet; also, a probability is assigned to the successful transmission of the returning acknowledgement. If an acknowledgement is not received before the expiration of the time out period, a retransmission would take place then, unless the processor is busy. A maximum number of transmissions for each packet is set at 6. (Since only one PRU is simulated, actual alternate routing is not an issue.)

III. Results and Discussions

The simulation program used in this study was designed as the building-block of the system-level simulation described in PRTN #268. This building-block simulator has demonstrated behaviour in close correlation with that observed in the measurement experimental results, when used stand-alone it cannot fully reflect dynamic interaction among the PRUs. The reader is cautioned to take the results presented here as relative comparisons between different schemes and parameter values.

The statistics collected are cumulative over all neighbors. They include: the one-hop delay (from the arrival of a packet at the simulated PRU to its successful arrival at the next PRU), the throughput (the number of packets successfully transmitted to the next PRU), the buffer occupancy, the number of dropped (after a maximum number of retransmissions) and discarded packets (due to buffer overflow), the previous-hop acknowledgement time (time elapsed from the arrival of a packet until the successful transmission of its first acknowledgement to the previous PRU), and the next-hop acknowledgement time.

In this report, the behaviour of a PRU is presented based upon the one hop throughput-delay relations for the traffic passing through that PRU. Due to the operational nature of the PRU, the traffic of two neighboring PRUs in a PRNET are highly correlated. How do they actually correlate depends upon the network configuration, the traffic pattern, as well as other factors. Due to this correlation, the behaviour of a PRU in a network depends a great deal on the operating points of its neighboring PRUs. In particular, an increase in the throughput of a PRU usually implies higher traffic to its neighbors. It may therefore take longer for a neighbor to acknowledge packets intended for him. In turn, such higher acknowledgement time may cause higher delay and lower throughput for this PRU. While simulating a single PRU, only limited above mentioned dynamic interaction between neighboring PRUs can be taken into account.

For comparison purpose, we define an equilibrium trace along which the average next-hop acknowledgement time (NAT), and the average previous-hop acknowledgement time (PAT) are equal. Such a trace defines a 'maximum traffic', or 'capacity', of a PRU. we will use this 'capacity' for comparison. Such a trace may be interpreted as describing the steady state throughput-delay relation of a PRU with all its neighbors operating at the same operating point as its own.

For graph presentation, we show the comparison of the traces as well as the contruction of each trace from a set of throughput-delay curves, each with a different next hop turn-around time.

1. Comparison of CAP 4.9 ~~Cyclic~~ vs FIFO Transmission Schemes

Conceptually, the CAP 4.9 transmission scheme implements multiple independent FIFO transmit queues. Only the packets on the top of a FIFO queue may be transmitted. In this view of the CAP 4.9 transmission scheme, the FIFO transmission becomes its special case of one FIFO queue.

The advantage of CAP 4.9 transmission scheme over the FIFO transmission scheme increases with the number of neighboring PRUs. This advantage reaches a limit when the number of neighbors becomes much greater than the number of buffers available. In which case, the likelyhood of two packets for the same neighbor waiting in the queue for transmission vanishes. The simulation results presented in Figure 6 indicate that significant improvement can be achieved by CAP 4.9 cyclic transmission. Using the above defined measure, we have observed about 72% improvement over FIFO with 2 neighbors, 132% with 3 neighbor, and 184% with 4 neighbors.

2. The Effect of Success Transmission Probability

The 'Hidden terminal' effect has been a concern when CSMA is used for multi-hop packet radio networks. The effect of 'hidden terminals' is that the advantage of CSMA over ALOHA channel access mode may be significantly reduced due to higher collision rate among packets from mutually hidden PRUs. In our study, the collision rate is translated into the success transmission probability. To investigate the extend of the adversity due to the 'hidden terminal' effect, we ran the building-block simulator with varying success transmission probability. The conclusion one may draw from the results presented in Figure 9 is that a lower success transmission probability would not significantly affect the PRU performance. This is because the failure probability diminishes exponentially with the number of retransmissions. The retransmissions may increase the traffic delay. But the system appears to become saturated due to other factors before the effect of 'hidden terminals' becomes significant. This observation coincides with our previous measurement experience that channel does not seem to be the bottle-neck of the system.

3. The Impact of Reducing Initial Transmission Delay (RTXDLY)

In the Hop-by-Hop measurement experiment, we observed that the default values for RTXDLY and RETXDY may not be set at their optimum. It indicated that RTXDLY, the initial transmission delay, should be reduced. Recently, we also heard the argument that giving priority to the first transmission of a packet may improve PRNET performance. This argument is well shared. The CAP4.9 transmission scheme and the subsequent CAP 5 transmission scheme also reflects such view. Reducing the value of RTXDLY, the initial transmission delay, may be another step along the same direction. Although some improvement by reducing RTXDLY is expected, it is surprising to observe its extent of improvement indicated by our simulation results shown in Figure 11. In the case of 3 neighbors, by reducing RTXDLY from 8.2 msec to 3.5 msec an improvement of 43% is observed. Although this result is not definitive, it may indicate that a varification by measurement is warranted.

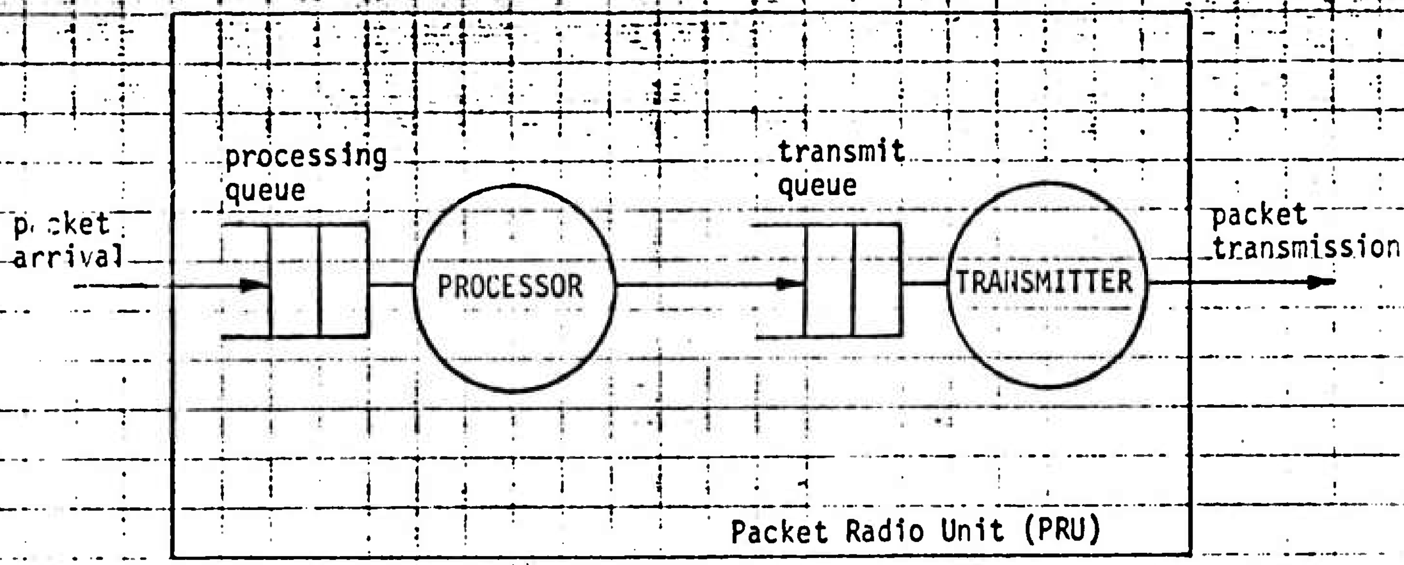

processing queue
transmit queue
packet arrival
PROCESSOR
TRANSMITTER
packet transmission
Packet Radio Unit (PRU)

Figure 1

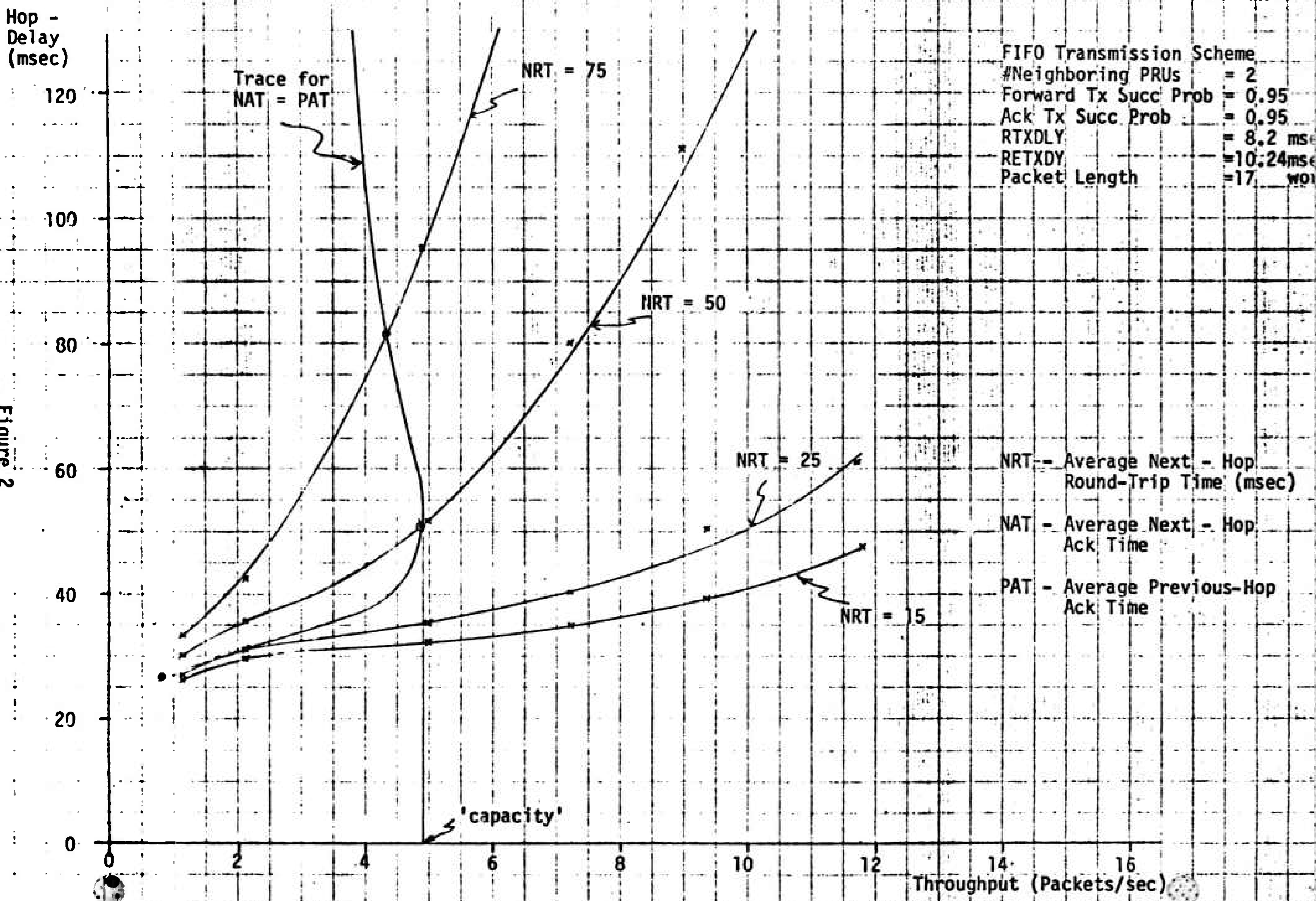
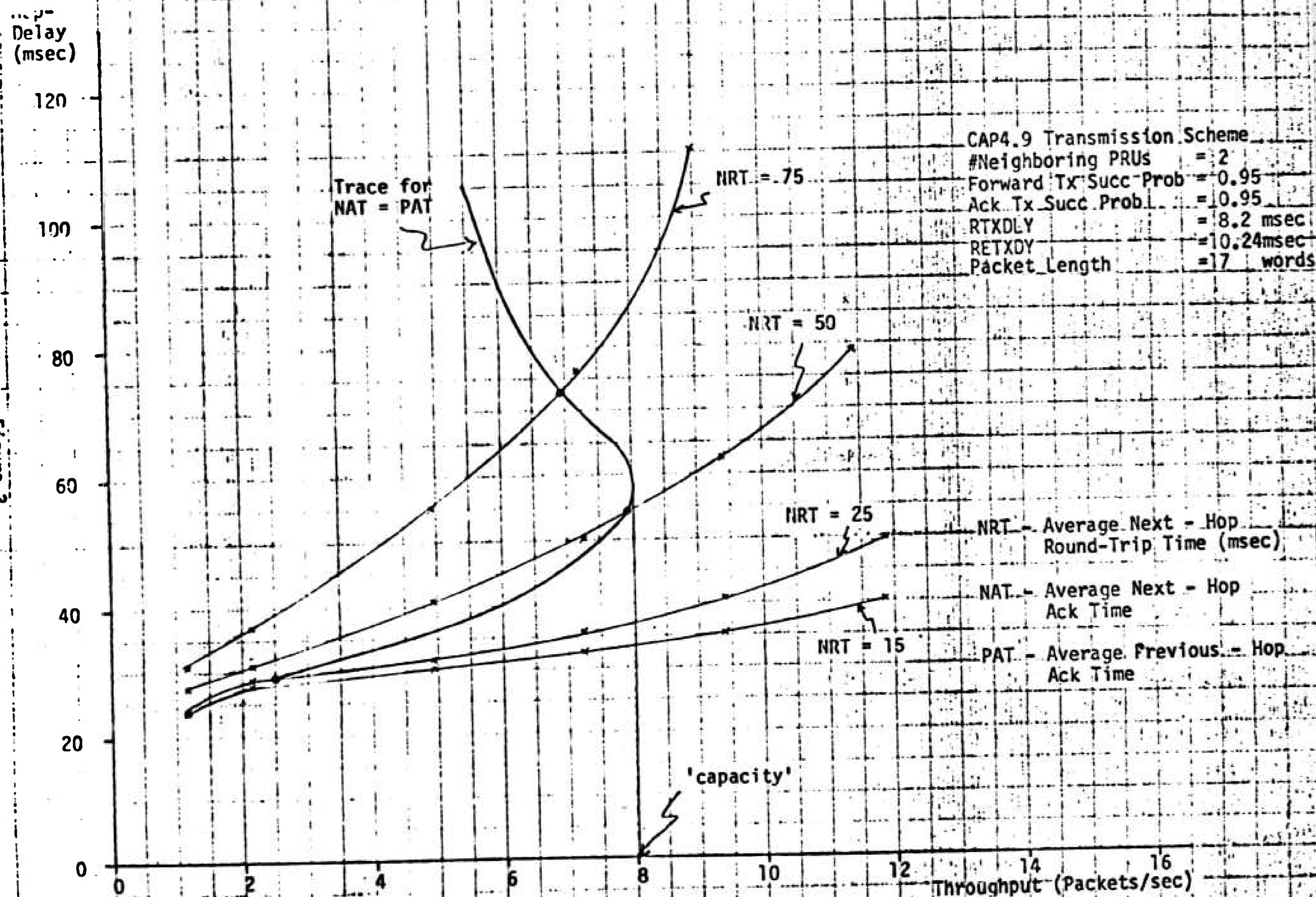

Hop -
Delay
(msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (Packets/sec)
Trace for
NAT = PAT
NRT = 75
NRT = 50
NRT = 25
NRT = 15
'capacity'
FIFO Transmission Scheme
#Neighboring PRUs = 2
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2 ms
RETXDY =10.24mse
Packet Length =17 wo
NRT - Average Next - Hop Round-Trip Time (msec)
NAT - Average Next - Hop Ack Time
PAT - Average Previous-Hop Ack Time

Figure 2



Delay
(msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (Packets/sec)
Trace for
NAT = PAT
NRT = 75
NRT = 50
NRT = 25
NRT = 15
'capacity'
CAP4.9 Transmission Scheme
#Neighboring PRUs = 2
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2 msec
RETXDY = 10.24msec
Packet Length = 17 words
NRT - Average Next - Hop Round-Trip Time (msec)
NAT - Average Next - Hop Ack Time
PAT - Average Previous - Hop Ack Time

Figure 3

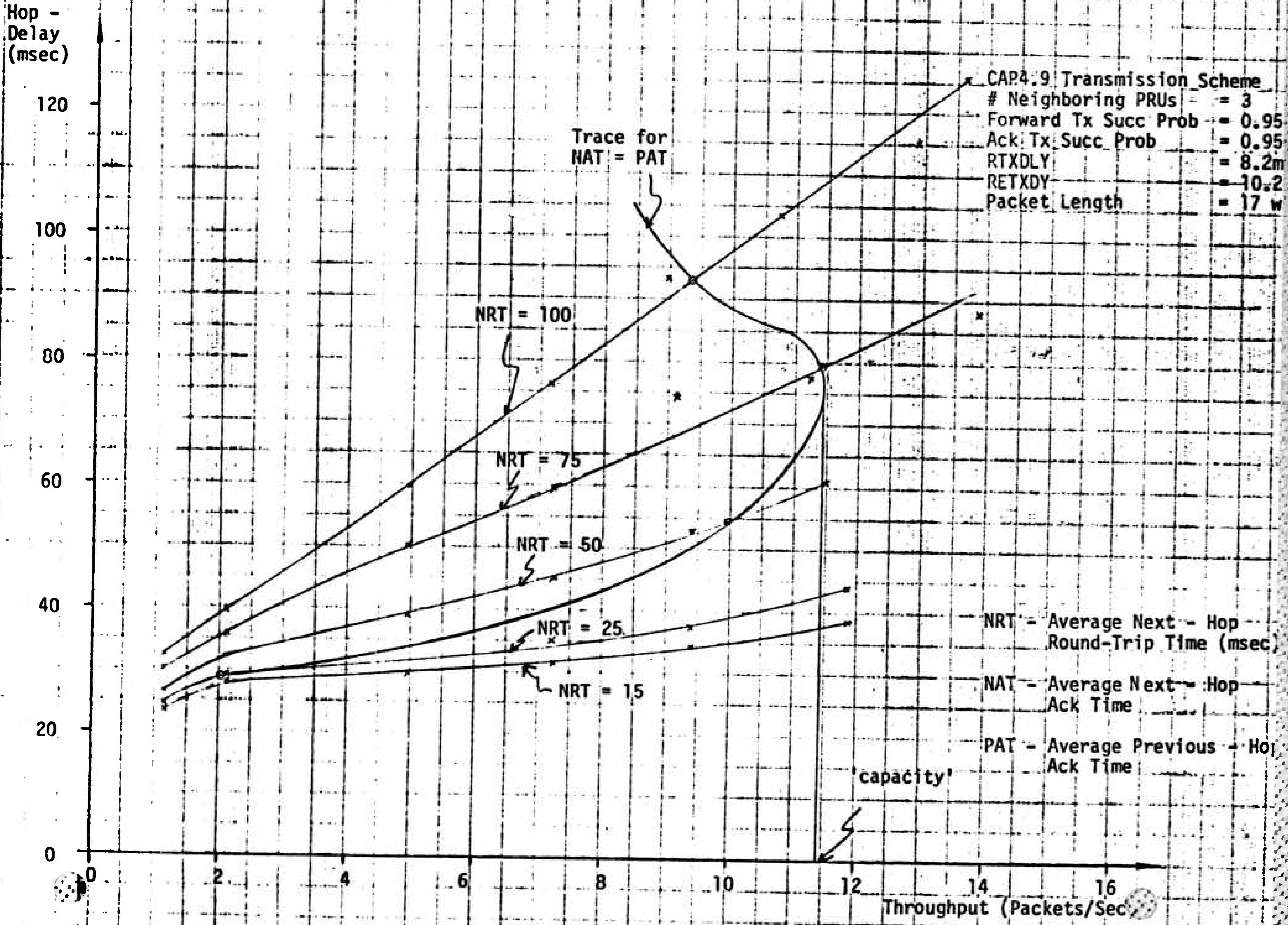
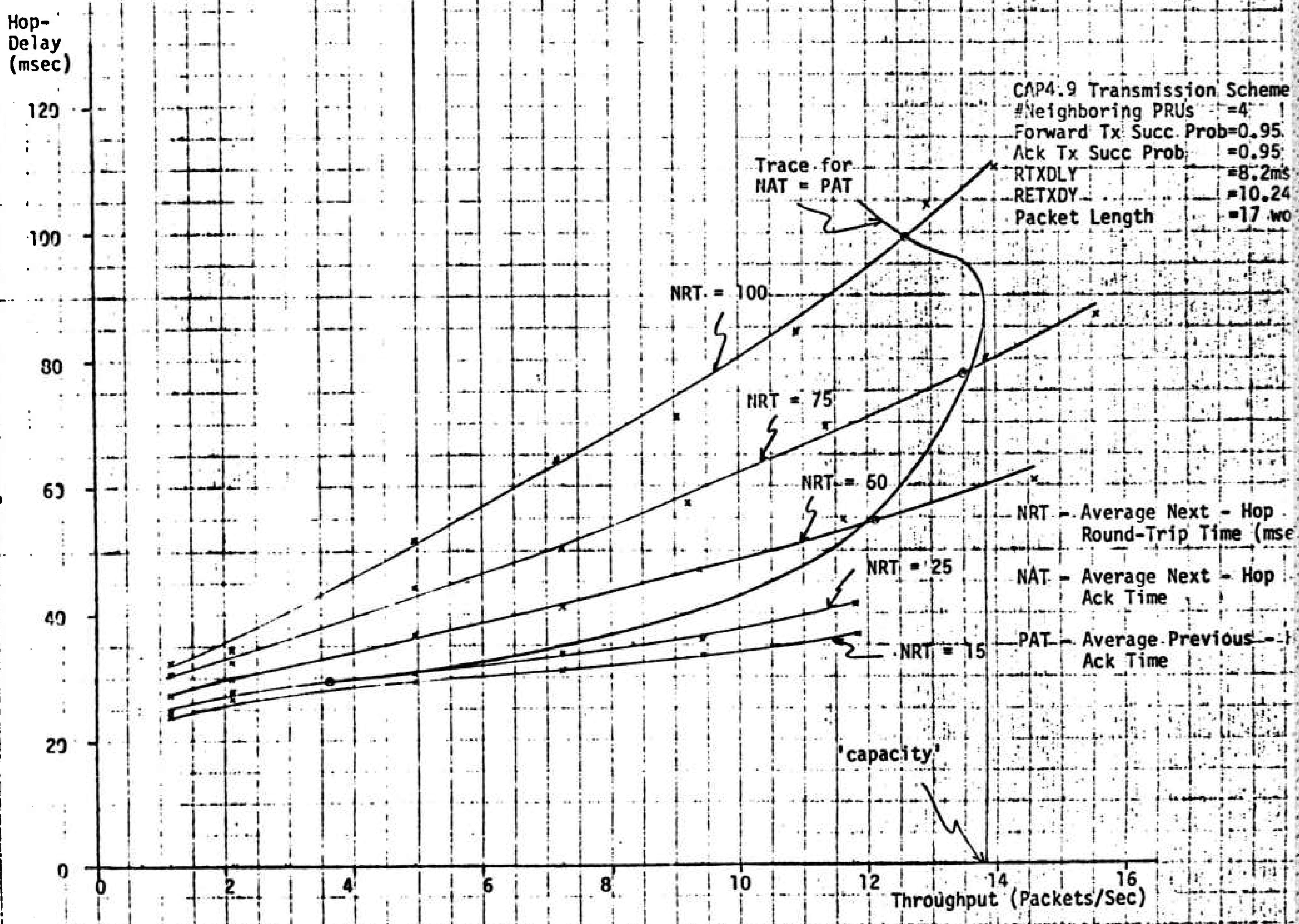

Hop - Delay (msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (Packets/Sec)
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2m
RETXDY = 10.2
Packet Length = 17 w
Trace for NAT = PAT
NRT = 100
NRT = 75
NRT = 50
NRT = 25
NRT = 15
NRT - Average Next - Hop Round-Trip Time (msec)
NAT - Average Next - Hop Ack Time
PAT - Average Previous - Ho
Ack Time
'capacity'

Figure 4



Hop-Delay (msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (Packets/Sec)
CAP4.9 Transmission Scheme
#Neighboring PRUs =4
Forward Tx Succ Prob=0.95
Ack Tx Succ Prob =0.95
RTXDLY =8.2ms
RETXDY =10.24
Packet Length =17 wo
Trace for NAT = PAT
NRT = 100
NRT = 75
NRT = 50
NRT = 25
NRT = 15
NRT - Average Next - Hop Round-Trip Time (mse
NAT - Average Next - Hop Ack Time
PAT - Average Previous - Ack Time
'capacity'

Figure 5

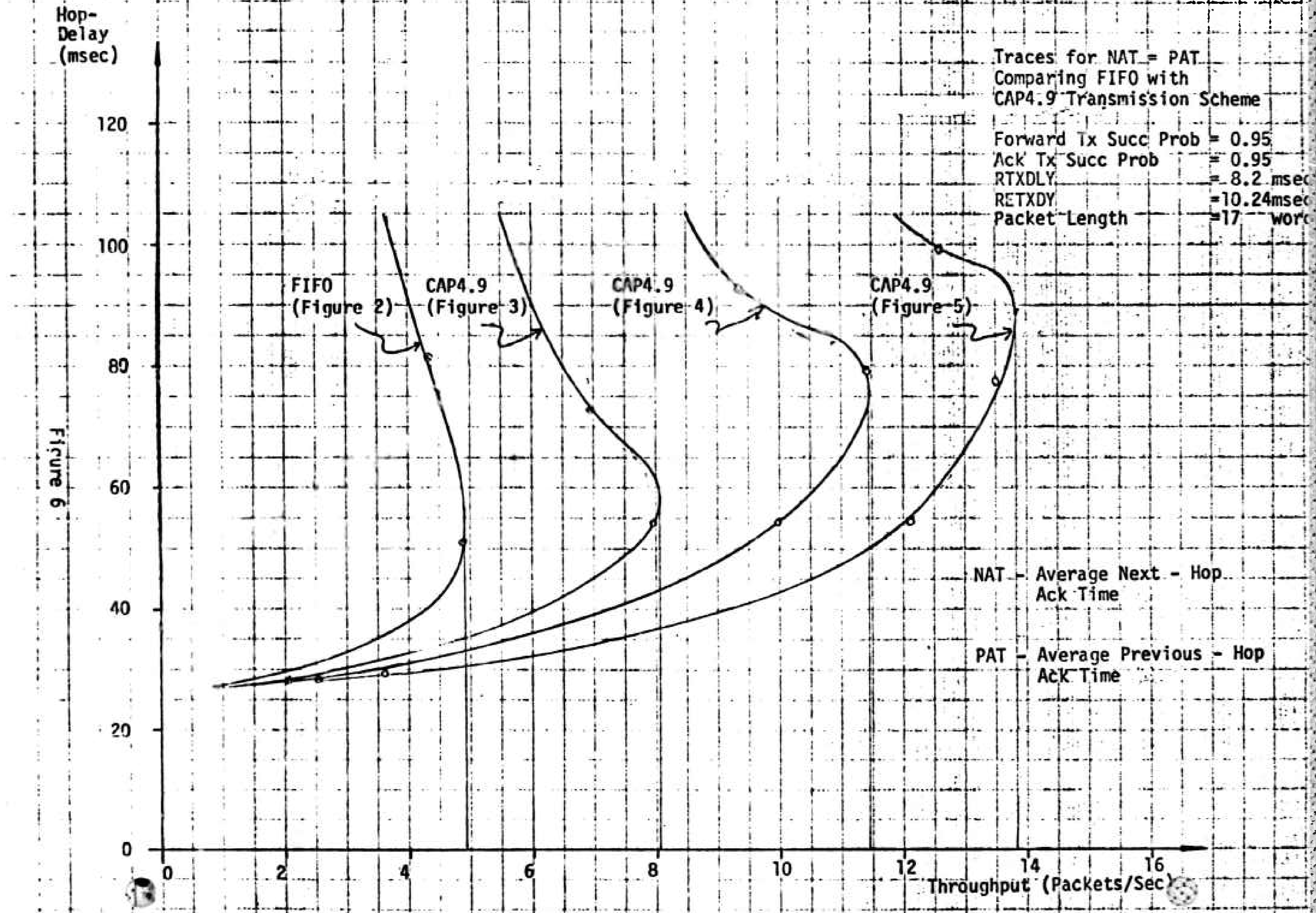

Hop-Delay (msec)
Traces for NAT = PAT
Comparing FIFO with
CAP4.9 Transmission Scheme
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2 msec
RETXDY =10.24msec
Packet Length =17 word
FIFO (Figure 2)
CAP4.9 (Figure 3)
CAP4.9 (Figure 4)
CAP4.9 (Figure 5)
NAT - Average Next - Hop Ack Time
PAT - Average Previous - Hop Ack Time
0
20
40
60
80
100
120
0
2
4
6
8
10
12
14
16
Throughput (Packets/Sec)

Figure 6

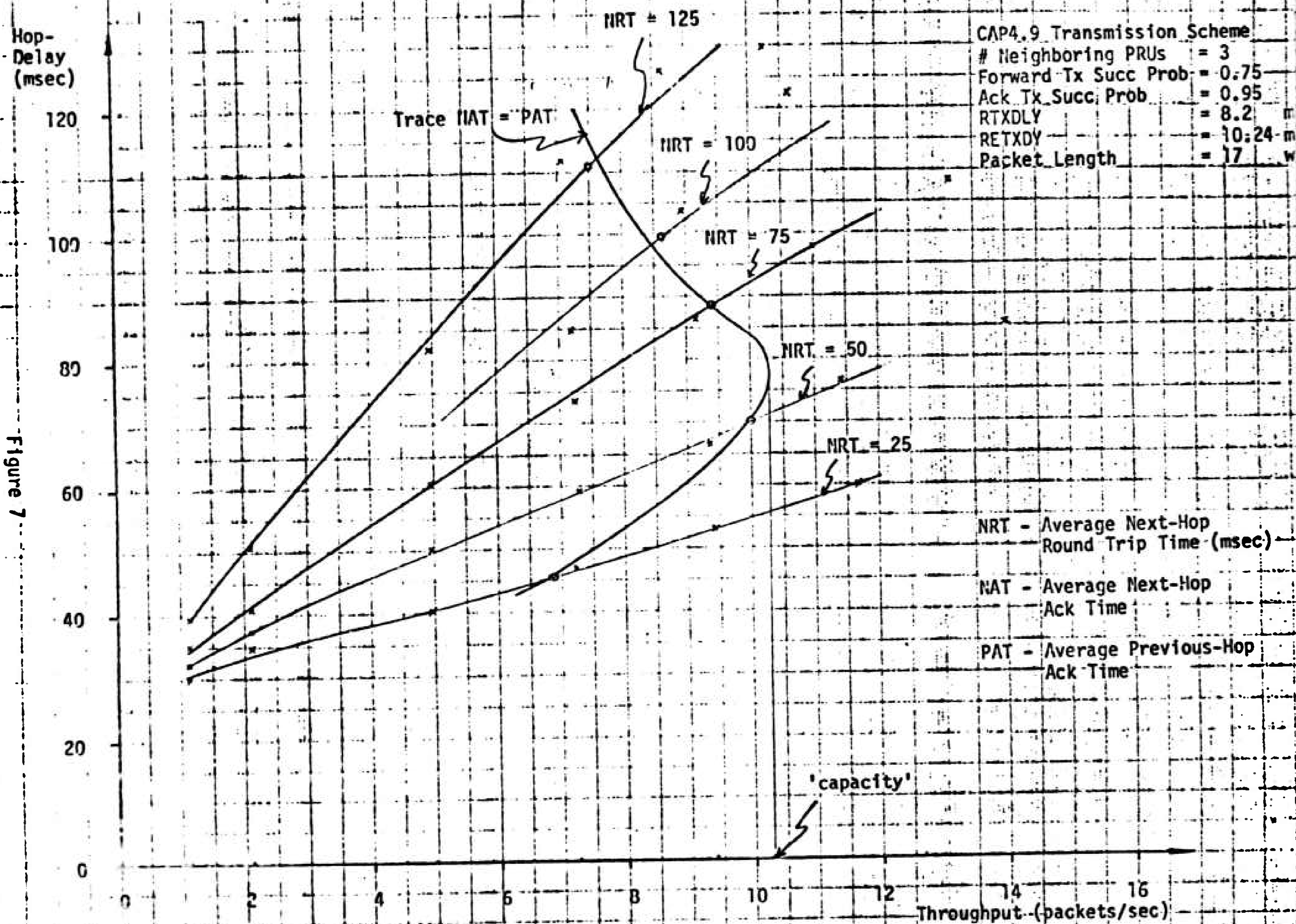
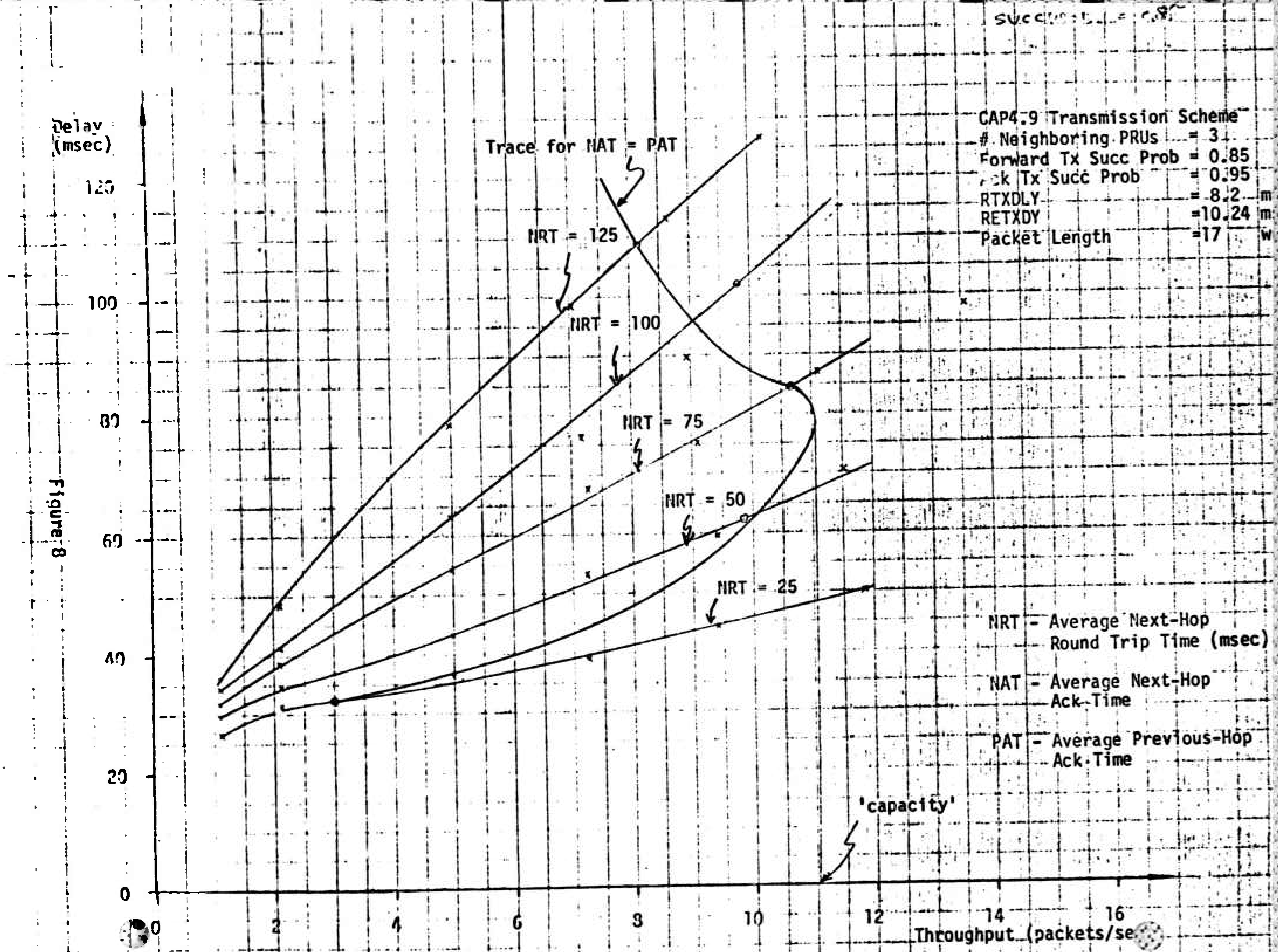
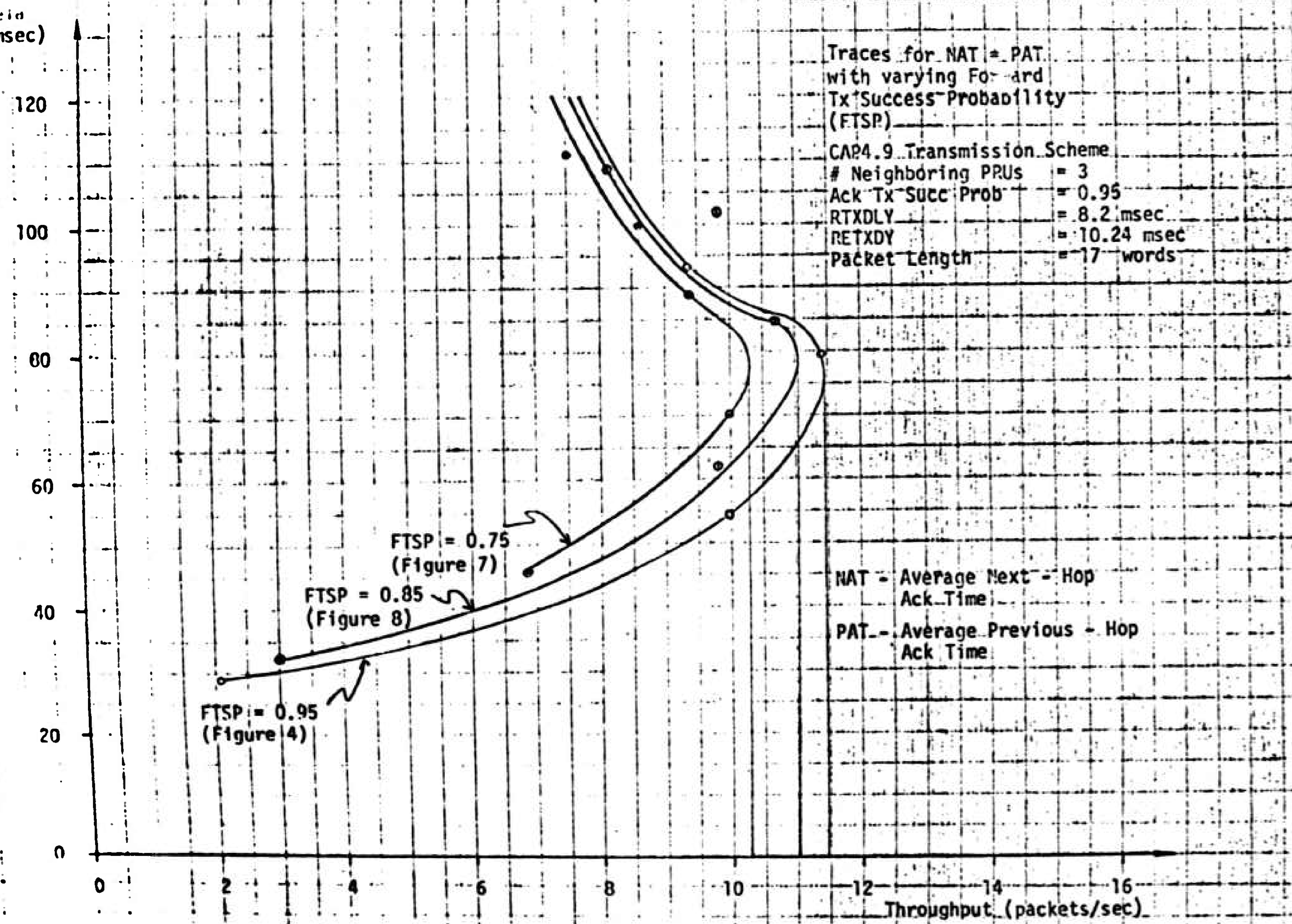

Hop-Delay (msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (packets/sec)
NRT = 125
NRT = 100
NRT = 75
NRT = 50
NRT = 25
Trace NAT = PAT
'capacity'
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Forward Tx Succ Prob = 0.75
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2
RETXDY = 10.24
Packet Length = 17
NRT - Average Next-Hop Round Trip Time (msec)
NAT - Average Next-Hop Ack Time
PAT - Average Previous-Hop Ack Time

Figure 7



Delay (msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (packets/se
Trace for NAT = PAT
NRT = 125
NRT = 100
NRT = 75
NRT = 50
NRT = 25
'capacity'
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Forward Tx Succ Prob = 0.85
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2 m
RETXDY =10.24 m
Packet Length =17 w
NRT - Average Next-Hop Round Trip Time (msec)
NAT - Average Next-Hop Ack Time
PAT - Average Previous-Hop Ack Time

Figure 8



Delay (msec)
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
Throughput (packets/sec)
Traces for NAT = PAT with varying Forward Tx Success Probability (FTSP)
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Ack Tx Succ Prob = 0.95
RTXDLY = 8.2 msec
RETXDY = 10.24 msec
Packet Length = 17 words
NAT - Average Next - Hop Ack Time
PAT - Average Previous - Hop Ack Time
FTSP = 0.75 (Figure 7)
FTSP = 0.85 (Figure 8)
FTSP = 0.95 (Figure 4)

Figure 9

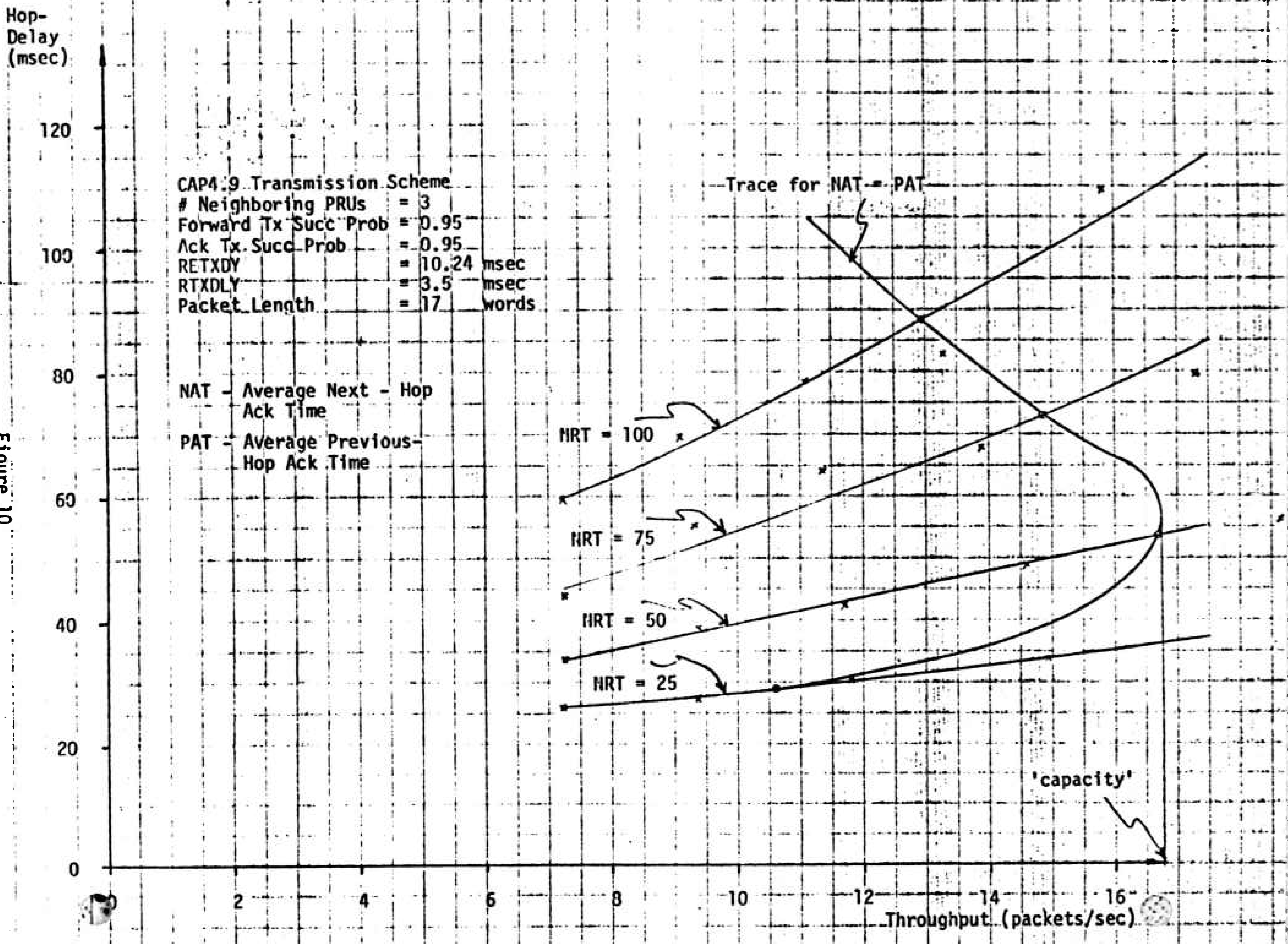

Hop-Delay (msec)
120
100
80
60
40
20
0
2
4
6
8
10
12
14
16
Throughput (packets/sec)
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RETXDY = 10.24 msec
RTXDLY = 3.5 msec
Packet Length = 17 words
NAT - Average Next - Hop Ack Time
PAT - Average Previous-Hop Ack Time
Trace for NAT = PAT
NRT = 100
NRT = 75
NRT = 50
NRT = 25
'capacity'

Figure 10

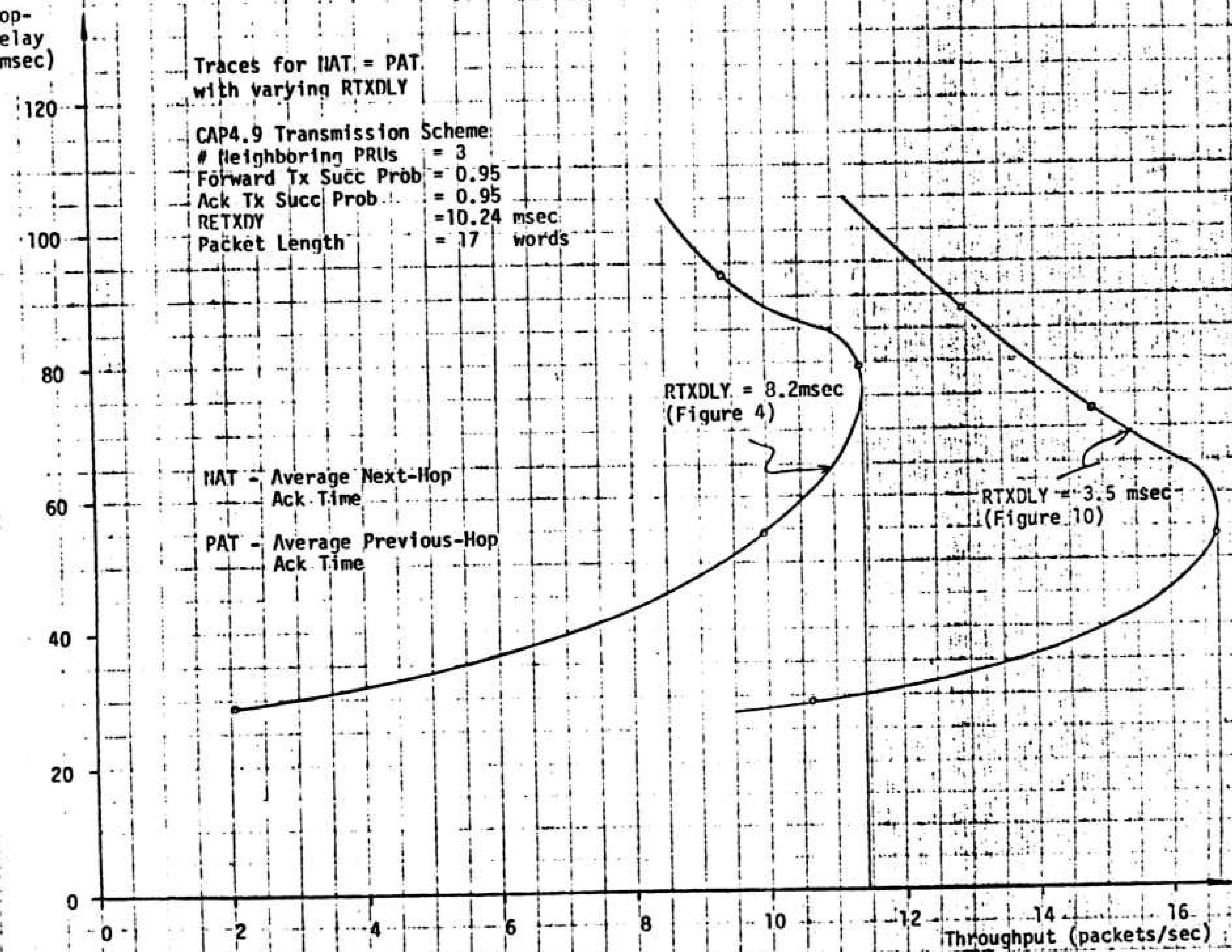

Hop-Delay (msec)
120
100
80
60
40
20
0
Traces for NAT = PAT with varying RTXDLY
CAP4.9 Transmission Scheme
Neighboring PRUs = 3
Forward Tx Succ Prob = 0.95
Ack Tx Succ Prob = 0.95
RETXDY =10.24 msec
Packet Length = 17 words
NAT - Average Next-Hop Ack Time
PAT - Average Previous-Hop Ack Time
RTXDLY = 8.2msec (Figure 4)
RTXDLY = 3.5 msec (Figure 10)
0
2
4
6
8
10
12
14
16
Throughput (packets/sec)

Figure 11

PRTN # 253
Stan Lieberson

Fouad Tobagi

UCLA

CURRENT PR MEASUREMENT PLAN, MARCH, 1978

This plan constitutes the description of a number of measurement activities to be conducted on the experimental PRNET. It is expected and intended that this document will be modified and expanded to reflect increasing knowledge of the capabilities, limitations and parameters of the experimental PRNET, as well as changes in PRNET protocol and hardware.

The plan consists of the following:

Part I: Objectives of the measurement program including a description of the PRNET measurement facilities.

Part II: A list of specific network experiments, along with the required configurations, tools, parameters and data reduction.

Part III: Actual test bed configurations to carry out the network experiments listed in Part II. (Drawings obtained from SRI.)

Part IV: A tentative time table for the execution, data reduction and analysis of the experiments in Part II.

PART I

MOTIVATION AND OBJECTIVES OF THE PACKET RADIO MEASUREMENT PROGRAM

D.O.D PACKET RADIO COMMUNICATIONS REQUIREMENTS

The advantages in using broadcast radio communications are many: easy access to central computer installations and computer networks; collection and dissemination of data over local distributed geographical areas independent of the availability of preexisting (telephone) wire networks; the suitability of wireless connections for comunications with and among mobile users.

The military applications are further supported by the potential rapid deployment of packet radio systems, the mixing of voice and data transmission over the same communications media and the achievement of distributed control of shared channels rendering the system both cost-effective and reliable.

THE NEED FOR THE PACKET RADIO MEASUREMENT EFFORT

The concern for measurement is due to several factors. Firstly, these measurements provide a meana to evaluate the performance of the protocols employed and the identification of their key parameters. Moreover, this realistic observation of the system behavior will assist in the validation and improvement of existing analytical models devised to study aome of these schemes, such as the access modes and routing strategies. Secondly, these measurements will allow for the detection of aystem inefficiencies and the identification of design flaws such as the inadvertent creation of a deadlock condition. Thirdly, measurement facilities and data, when used to improve network design, are a valuable feedback process in which design deficiencies are detected and subsequently corrected.

MEASUREMENT FACILITIES

Immediately below we describe the various types of statistics to be collected in the Packet Radio Net, the traffic sources required in measurement experiments and the techniques available for measurement data collection.

CUMULATIVE STATISTICS (CumStats)

As their name suggests these consist of data regarding a variety of events, accumulated over a given period of time, and proveded in the form of sums, frequencies and histograms. We shall distinguish between those data collected at the PRUs (PRU based CumStats) and those collected at the end devices (the end-to-end CumStats). The PRU based CumStats provide information about the local environment and behavior such as traffic load, channel access, routing performance, and repeater activity. Conversely, end-to-end statistics collected at network sources and sinks, that is stations and terminal devices, will reflect more global network behavior such as user delays and network throughput.

TRACE STATISTICS

The trace capability allows one to literally follow a packet through the network, and to trace the route which it takes and the delays which it encounters at each hop. In the packet radio network, the collection of trace data to the repeaters is prohibited by the limited size of storage in the PRU. To overcome this problem, we have introduced a new type of packet called the Pickup Packet. These packets are generated with an empty text field by traffic generators at end devices. As these packets flow normally in the network according to the transport protocols, selected repeaters will gather the trace statistics and will store them within the text field of the pickup packets themselves.

SNAPSHOT STATISTICS

Snapshots give an instantaneous peek at a PRU, showing its state at that moment with regard to buffer assignment and queue lengths. (In the ARPANET, which is a decentralized network in which each node contains routing algorithms and data, snapshots also include routing related information; in the Packet Radio Network, such information is available at the station). Changes to appropriate station tables will be time stamped and collected as the station's snapshot function.

ARTIFICIAL TRAFFIC GENERATORS

The creation of streams of packets between given points in the net, with given durations, intervals, packet lengths, and packet types (Information and Pickup Packets) is clearly a requirement of any experimental system. While it might be desirable to provide each PRU with the capability of creating such traffic, this additional burden on the PRU software can be avoided if there exists a reasonable number of terminals with processors attached which, along with the station, will be programmed to provide the traffic-source functions indicated above.

Specifically, traffic-source features which the terminals (and Station) should provide are:

(1) Information Packets - the user specifies the packet length, frequency, destination and duration of one or more streams of Information Packets. (The text content may be arbitrary.)

(2) Pickup Packets - the user specifies the packet length, frequency, destination and duration of one or more streams of Pickup Packets.

STATION MEASUREMENT PROCESS

Since the station is the central node and provides central control for the operation of the entire network it therefore plays a central role in the execution of measurement functions. It is through the station that the initiation and termination of measurement experiments are controlled. In particular, the station enables and disables the CumStat and Pickup packet functions at the PRU's, and assigns to the various elements the intervals for CumStat collections, and to the artificial traffic generator, their corresponding parameters. Moreover, it is to the station that all measurement data is ultimately destined; upon arrival at the station, the data is time-stamped and stored in a single measurement file for off-line reduction and analysis. In addition, all changes to the station's internal tables (routing, connectivity, PRU operational parameters, etc.) will be reflected by an entry into the measurement file, thus allowing the correlation of measurement results to the actual network configuration. A (Measurement) process at the station will perform all of the above functions.

SUMMARY OF MEASUREMENT ITEMS

Pickup Packets (at each PRU, the following data items are collected in the Pickup packet):

Time of arrival of the packet at the PRU
Time the Pickup packet was first placed on the transmit queue
Time of each transmission

Time HBH ack arrived (stored in next Pickup packet)
The current PRU ID

PRU based Cumulative Statistics

Number of packets received in error
Number of packets received but not intended for this PRU
Histogram of number of transmissions per successful packet
Number of unsuccessful packets (dropped because of lack of ack)
Number of packets discarded because of lack of buffer space
Number of alternately routed ("ALL") packets received
Table counting number of correctly received packets from immediate neighbors
Number of transmissions beyond success
Table sampling frequency of buffer states (and transceiver states)

End-Device Cumulative Statistics

Histogram of round-trip times
Number of packets transmitted
Number of duplicate packets detected
Number of packets discarded by the sender because of lack of ETE ack
Histogram of number of transmissions per successful (ETE) packet
Histogram of packet intergeneration time

Note: certain CumStat items will distinguish between inbound (to the station) and outbound (from the station) traffic, as well as 100 and 400 KBPS Rates.

PART II

PROPOSED PACKET RADIO MEASUREMENTS

Table of Symbols:

○ PRU

□ Station

△ Terminal Device

---- Routing Assigned

◯ Area of Bi-Directional Connectivity

EVALUATE CHANNEL ACCESS PROTOCOLS IN ONE-HOP ENVIRONMENTS

In the initial experimental system pure ALOHA, "disciplined ALOHA" and non-persistent CSMA will be available. Our measurement aims are to validate the analytical models of the three access modes and to evaluate their performance in realistic environments.

In evaluating terminal access in a single hop system (a model commonly used in analysis), we consider an environment consisting of a single station and a population of terminals within range and in line-of-sight of the station. In order to determine the relationships between the network throughput (rate of successfully received packets at the station) and channel traffic (rate of packet transmissions over the channel), as well as the relationships between the network throughput and packet delays, the following quantities will be measured:

(a) the number of transmissions a packet incurs before success

(b) the one-hop packet delay: time elapsed since the packet is ready for transmission until it is acknowledged, i.e., until its acknowledgment packet is received from the station

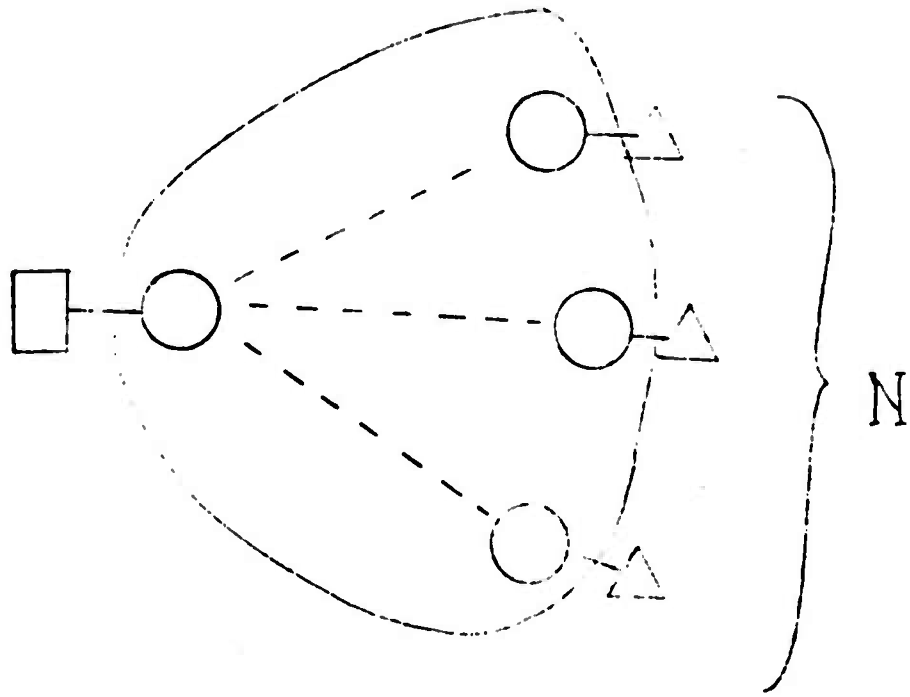
(c) network throughput: average number of packets received at the station per unit time

Items (a) and (b) are obtained in the form of histograms by the CumStat tools at the PRU and the end device respectively. Item (b) may also be obtained individually for each Pickup packet by having the originating device store the time its acknowledgment arrived. Item (c) is obtained at the station from end-to-end cumulative statistics.

Note: When comparing measurement results with analysis, it is important to note the following important discrepency: the analysis assumed that acknowledgement packets are instantaneous and for free; the measurement results will bear the effects of acknowledgement traffic, of its interference with information packets and of the loss of acks.

Area of Interest: 1-hop Channel Access

See SRI testbed figure(s): 2 (modified by moving a terminal)



Area of Interest: 1-hop Channel Access

Configuration: A 1-hop network with all involved PRUs in line-of-sight and within range

Tools: Cumstats (End-device and PRU), Pickup Packets

Data Items:
1) # of tx's to successful acknowledgement [CS-PRU]
2) # of successful tx's [CS-PRU] (number of tx's HBH ack'ed)
3) 1-hop (one-way) delay [Pickup Packets]
4) # of packets successfully rx'd at the station [CS-station]
5) # of packets successfully rx'd at the receiving PRU (station's PRU) [CS-PRU]

Parameters: Traffic sources sending information packets (one run with minimum size packets, one run with maximum size packets) sent at a range of intergeneration times, the range to Initial experiments will determine the limits of this range.

Retransmission delays (mean) will vary within a range of values over several runs of this experiment. (Range will be determined experimentally.)

Processing: mean, SD, pdf of: # of tx's, plot of:

a) delay vs throughput

b) # of tx's to success vs throughput

c) maximum throughput vs N

d) maximum throughput vs retransmission delay

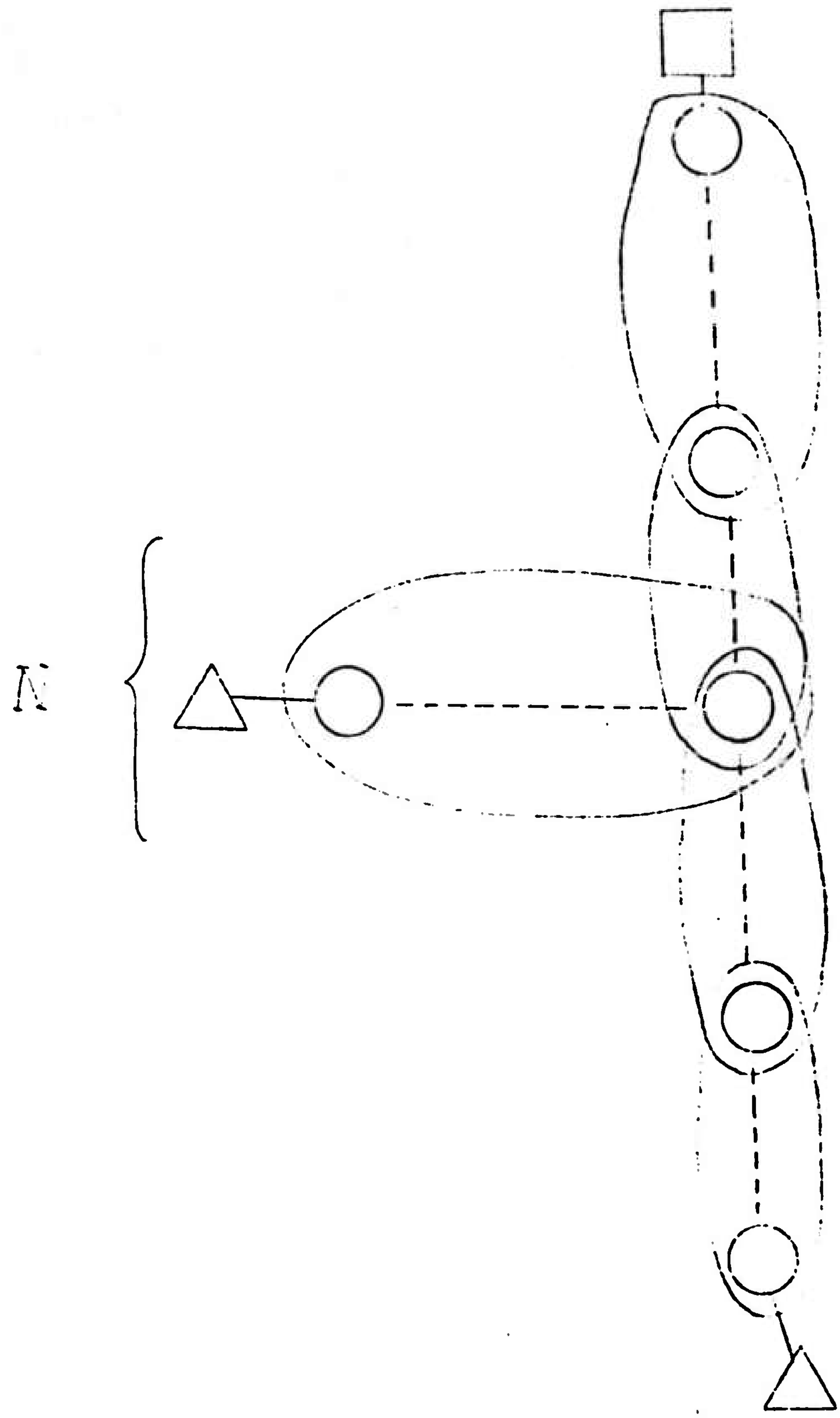
Comments: Each of the N configurations to be run for the two packet sizes, a range of traffic rates and mean retransmission times, and compared across the three access methods (ALOHA, "disciplined ALOHA", non-persistent CSMA). Other parameters to note: i) effect of signal power (capture), ii) all 400KB or 100KB rates.

EVALUATE CHANNEL ACCESS PROTOCOLS IN MULTI-REPEATER ENVIRONNMENTS

The tasks of measuring performance of terminal access techniques in multi-repeater environments differs from the previous one in that repeater-to-repeater traffic is present contending on the same channel. The environment consists of a number of repeaters and stations and a population of terminals, not necessarily all within range and in line-of-sight. The same quantities as listed above, measured over the terminal-to-repeater hop, will be collected using the same tools.

Area of Interest: Multi-Repeater 1-hop Channel Access

See SRI testbed figure(s): 1



Area of Interest: Multi-Repeater 1-hop Channel Access

Configuration: A pool of terminals accessing a repeater, with contending network traffic routed through that repeater.

Tools: Cumstats (end-device and PRU), Pickup packets

Parameters: The N terminals send information packets. The comments as to packet sizes and intergeneration times for 1-hop channel access apply here. The end terminal will provide a stream of information packets to the station at a similiar range of rates.

Data Items:
1) # of tx's to success [CS-PRU]
2) 1-hop (first hop) delay [PP]
3) # of packets successfully received at the station [CS-station]
4) # of successful tx's [CS-PRU]

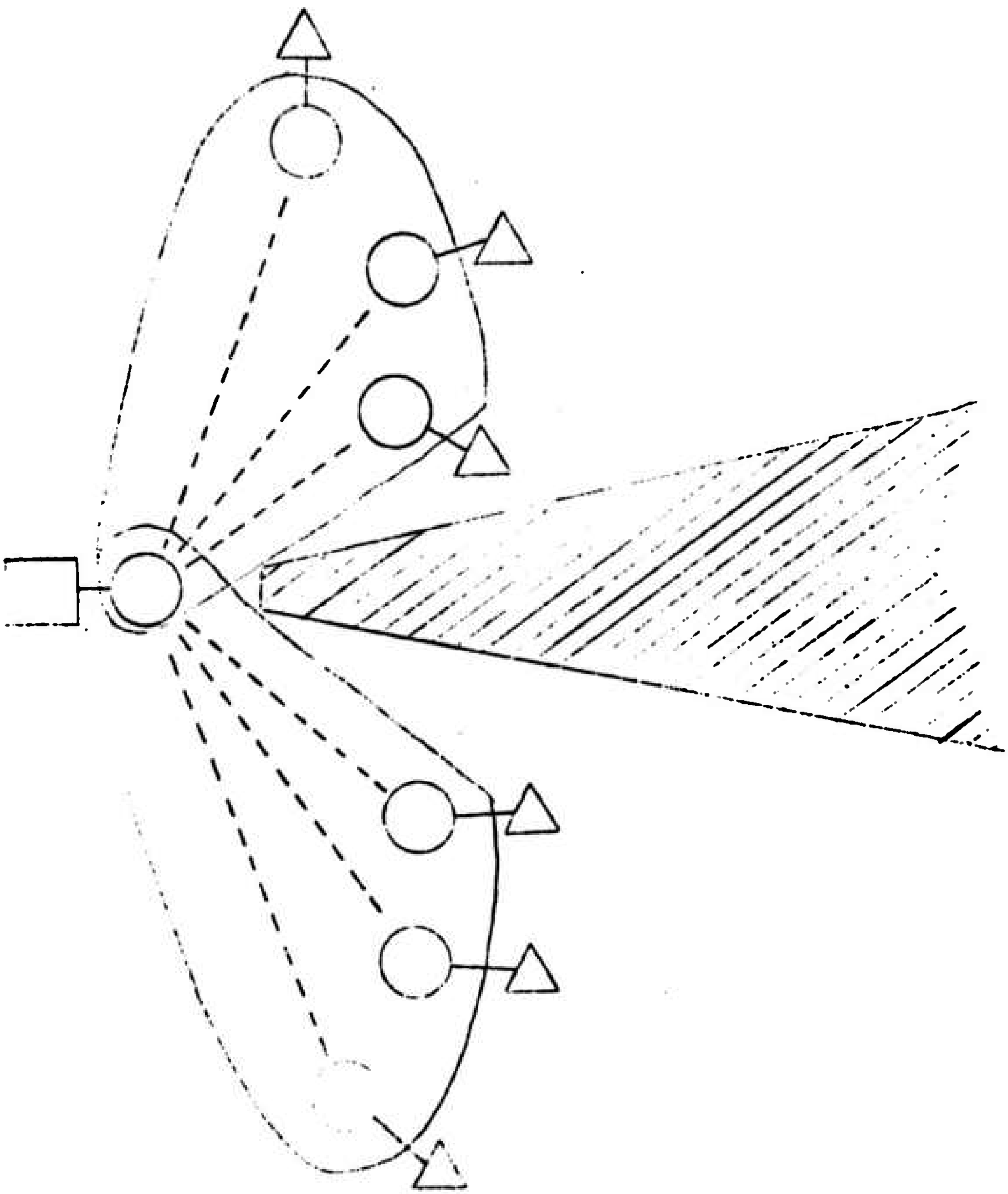
Processing: Same as for 1-hop channel access, but with all results plotted against level of interference caused by transit traffic (end terminal data rate), and N.

HIDDEN TERMINAL

In these one-hop channel access experiments, the performance of CSMA is dependent of the connectivity pattern among the contending terminals. The "hidden terminal" problem can, as shown by analysis, significantly degrade such a permormance. This experiment is aimed at a characterization of the effect of the hidden terminal problem on the performance measures discussed in the preceeding experiments (One-Hop Channel Access; Multi-Repeater One-Hop Channel Access).

Area of Interest: Hidden Terminal Problem in Non-Persistent CSMA

See SRI testbed figure(s): 2



Area of Interest: Hidden Terminal Problem in Non-Persistent CSMA

Configuration: Two groups of terminals, with no connectivity between the groups, accessing the same PRU.

Tools: Cumstats (end-device and PRU)

Parameters: Each group sending α and $1-\alpha$ of the traffic, respectively.

Data Items:
1) # of tx's to success [CS-PRU]
2) 1-hop delay [CS-terminal]
3) # of packets successfully rx'd (ETE) [CS-station]
4) # of successful transmissions

Processing: [same as in previous two channel-access experiments] Compare with analytical results of hidden terminal problem.

HOP-BY-HOP ACKNOWLEDGMENT PROTOCOL

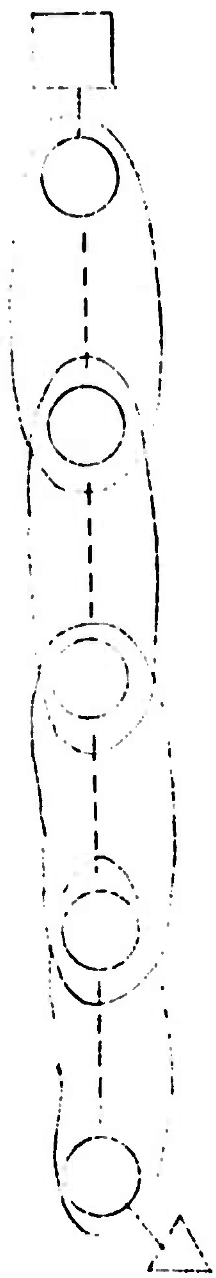
Hop-by-hop acknowledgment suffers from packet interference. The delay until the echo acknowledgment is received at the transmitter (provided the ack is not lost or conflicted with) is random. Thus, the packet may incur some additional transmissions beyond the first successful one creating additional overhead on the channel and the PRU's. This number of additional transmissions is a measure of the inefficiency of echo acknowledgments; so too will be the number of packets discarded at the transmitter because of lack of reception of the echo acknowledgment. That is, the transmitter reached the maximum number of retransmissions of a packet before the ecno acknowledgment was received; although the packet may have been successful, the transmitter declares itself unsuccessful in establishing communication!

Thus, we shall measure the efficiency of the Echo Acknowledgment protocol by measuring the number of additional transmissions beyond success incurred by a packet. To compute this number, a PRU must have two pieces of information; it must know how many times the packet has been transmitted, and it must also know which of those retransmissions was the one that reached the next repeater successfully. This information will be contained in two fields in each packet header, which we refer to here as fields A and B. Field B is used by the PRU to store the current transmission number of the packet. When the packet is successfully heard by the intended receiver, the contents of field B are saved by being stored into field A; when the Echo acknowledgment is successfully heard by the sending PRU, field A of the echo acknowledgment is compared with the current number of transmissions of the packet, i.e., the contents of field B in the sender's copy of the packet. If these two numbers differ, then the magnitude of that difference represents the number of times that the packet was retransmitted after it had already been successfully received at the next hop. This data is collected as part of the cumulative statistics of the sending PRU.

The independent parameters for these experiments are the input traffic rates, the retx delay, and the time-out periods at each device. The data items to be collected are the number of tx's beyond success as well as packet delyas (either end-to-end or single hop delays). An increase in the "time-out" period may decrease the number of tx's beyond success, but they may increase the delay. [The time-out period is that minimum delay each packet must experience before it can be transmitted, i.e. the fixed time to which the random delay is added.]

Area of Interest: Hop-by-hop acknowledgement protocols

See SRI testbed figure(s): 1



Area of Interest: Hop-by-hop acknowledgement protocols

Configurations: two basic configurations will be considerd: a string of repeaters of some length; and a ring of terminals around a station, forming a single hop system.

Tools: Cumstats (PRU)

Parameters: For both configurations, we vary the input rates at the traffic sources, the retx delays and time-out periods for rescheduling.

Data Items:
1) # of tx's beyond success.
2) packet delays, either end-to-end (round-trip) or single hop delays.

Processing: mean, SD, pdf of: # of tx's beyond succcess (vs N, and vs traffic load), plot: # of tx's beyond success vs traffic load (for N)

Evaluation of Repeater's Performance:

The evaluation of the performance of a repeater is most important in the anaylsis of network behavior; it allows us to break down key network measures (such as packet delay and throughput) into their elementary components and to examine the effects on these measures of the repeater activity and design (including buffer management, queueing discipline, and packet processing priorities).

PACKET PROCESSING DELAYS

The quantities relevant to packet delay at a PRU are:

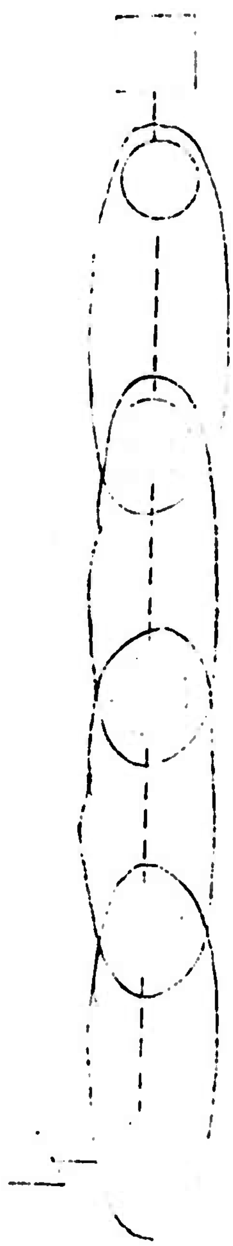
(a) The processing time of a packet flowing through a repeater; this is counted in Pickup Packets as the time lapse between the packet's arrival and the time it is placed on the transmission queue. This processing includes various checks such as checksum, packet type, routing tables. etc.

(b) the packet queueing delay at a repeater; this is also counted in Pickup Packets as the time elapsed from when the packet is placed on the transmission queue until it is considered for transmission (i.e., until it is at the head of the line, in a first-come-first-served discipline).

(c) the packet's service time; this is also counted in Pickup packets as the time elapsed from when the packet is at the head of the queue until its HBH acknowledgment is correctly received. Note that the actual service time (time until the packet is correctly received at the next repeater) is smaller than the one measured here due to the echo acknowledgment protocol used in this system. Note also that the service times of consecutive packets may be correlated.

Area of Interest: Packet processing time

See SRI testbed figure(s): 1



Area of Interest: Packet processing time

Configurations: A string of PRUs appears to be the most obvious configuration to consider. A minimum of 5 PRUs guarantees the existence of all combinations of data rates (100/400 KBPS) and HBH ack types (echo/active).

Tools: Pickup Packets

Parameters: Varying input rates at traffic source, varying retx delays at PRUs.

Data Items: (as defined in text above)
Processing time (PP)
Queueing delay (PP)
Service time (PP)

Processing: For each of the PRU's, plot - processing time, queueing delay and service time vs traffic load and retx delay.

TRANSCEIVER ACTIVITY

The quantities related to a repeater's communications activity are:

(a) percent of time the PRU transceiver is busy transmitting and receiving; this can be obtained in the PRU CumStat by regular sampling of the transceiver's state.

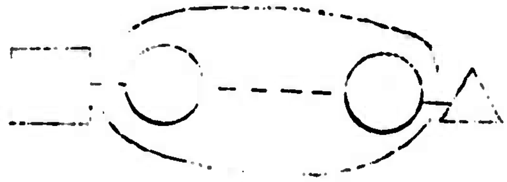
(b) the percent of traffic received with checksum error (obtained in the PRU CumStats).

(c) the percent of traffic received correctly but not intended for this repeater (obtained in the PRU CumStats).

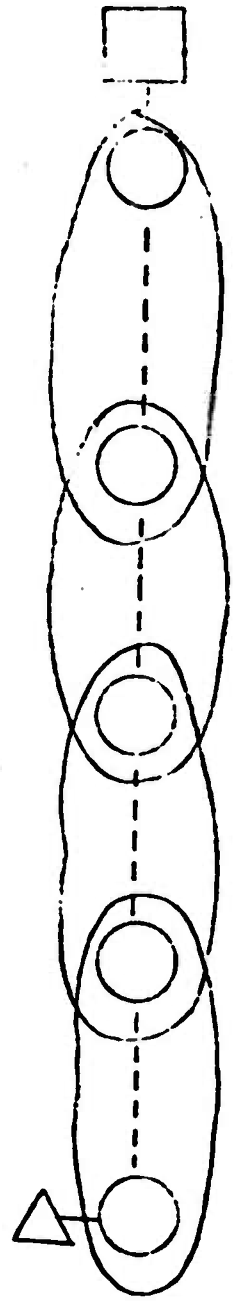
Area of Interest: Transceiver activity

See SRI testbed figure(s): b) 1; c) 5,6 [modified]

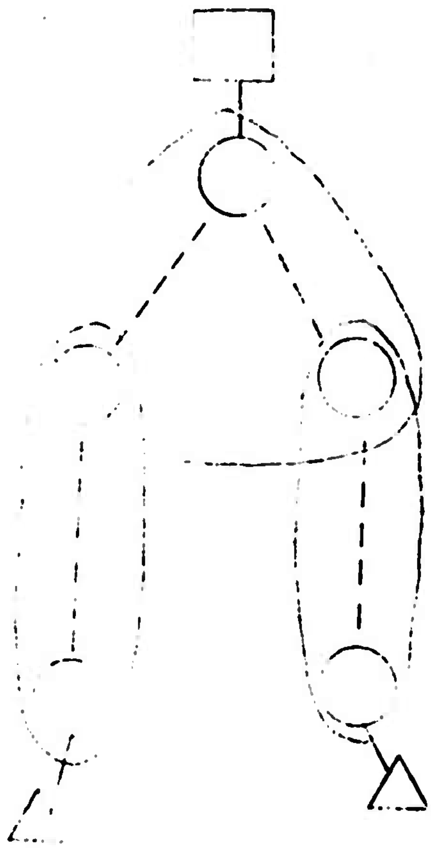
a)



b)



c)



Area of Interest: Transceiver activity

Configurations: (a) for the simplest configuration; (b) for a string containing all combinations of data rates and HBH acks (100/400 and active/echo); (c) for more generality in contention among PRUs

Tools: Cumstats (PRU), Snapshots (PRU)

Parameters: various input rates, retx delays.

Data Items:
Transceiver States:
fraction of time transceiver is:
a) disabled
b) tx'ing
c) rx'ing
d) rx enabled (listening)

Other Items:
- # of words tx'd
- # of packets correctly tx'd
- # of packets correctly rx'd
- # of packets incorrectly rx'd

Processing: plot of: frequency of above transceiver states vs traffic load, fraction of correct tx's, fraction of correct rx's.

Comments: The size and timing of the transceiver disabled state ("black hole") is of special concern.

BUFFER MANAGEMENT

The quantities relevant to buffer management and occupancy are:

(a) the percent of time packet buffers are in a given state (free, queued for packet transmission, reserved for packet receive). This can be obtained in the PRU CumStat by a regular sampling of the buffer states.

(b) the frequency of buffer overflow as a function of the load; this is obtained also in the CumStats by counting the number of packets discarded due to lack of buffer space.

Area of Interest: PRU buffer Management

Configurations: (a) for the simplest configuration; (b) for a string containing all combinations of data rates and HBH acks (100/400 and active/echo); (c) for more generality in contention among PRUs [See Transceiver Activity]

Tools: Cumstats (PRU), Snapshots (PRU)

Parameters: various input rates, retx delays.

Data Items:

Buffer States:

fraction of time buffers are:

a) free
b) queued for packet tx
c) reserved for packet rx (radio 100 KB, 400 KB; sta/ter)

Other Items:

- number of packets correctly tx'd
- number of packets correctly rx'd
- number of packets incorrectly rx'd

Processing: plot of: frequency of above buffer states vs traffic load

Routing Protocols:

In this experimental network, the hierarchical routing scheme(*) in use is based on a tree structure with the station as its root. The initial tree structure is created dynamically by the Initialization Procedure in which the station used PRU connectivity information to create a tree that attempts to "minimize" the number of hops between each repeater and the station. (However, when the first choice shortest path cannot be used, the packet departs from this path and uses an alternate route.)

The analysis of a routing algorithm, particularly in a broadcast, and thus mobile, network, is a complex task, in that routing is topology - and load-dependent, and involves, with varying degrees of subtlety, all of the system's protocols. Thus, routing considerations are really a synthesis of most elements of the system design, and as such, the measurement of the algorithm involves at times the study of the interaction of the many system protocols.

TRAFFIC DISTRIBUTION AND EFFICIENCY

Given the patterns of input load on the network, the distribution of traffic flow in the net is an indication of the behavior and efficiency of the routing and initialization algorithms. One may detect the concentration of traffic on specific routes creating congestion while alternate routes are not assigned; thus smaller delay routes may have been ignored in favor of the shorter routes provided by the initialization procedure. To obtain the distribution of traffic flow, the following quantities are to be measured.

(a) the total number of packets received and transmitted at each repeater (obtained in the PRU CumStats)

(b) the fraction of time the transceiver is busy (obtained by snapshot statistics, or in the PRU CumStat by regular sampling of the transceiver's state).

Also, the "point-to-point" nature of this routing algorithm, restricts a packet at a given hop to a single repeater as its immediate destination, does not take advantage of the broadcast nature of the channel, in which several neighbors may actually hear the transmission and be capable of relaying the packet. Thus the following quantity is relevant:

(*) In forthcoming CAP versions, point-to-point routing algorithms will be implemented. As this is done, new experiments will be designed to examine these protocols. For the time being, we assume hierarchical routing.

(c) the number of packets correctly received and discarded because they are destined to other components in the net (obtained in the PRU CumStat).

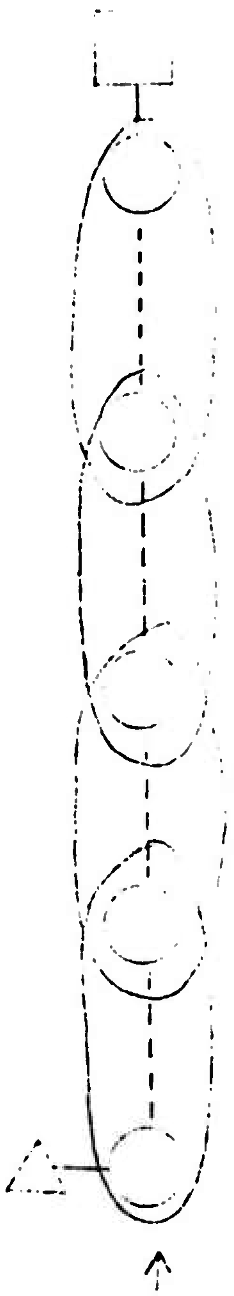
Moreover, to measure the potential of each neighboring repeater (say, repeater "n") as an immediate destination, it is essential to know the probability of success P(n) repeater n has to correctly receive a broadcast packet. This we do by maintaining in each PRU a table counting the number of successfully received packets from each immediate neighbor. The ratio of the number of packets correctly received from a given neighbor, to the number of packets transmitted by that neighbor, is a measure of P(n).

Most of the above concerns examining the efficiency of operational user networks by observing the actual traffic distribution determined by the routing algorithm. In this experimental net we intend to examine the relationship between traffic distribution and network efficiency. This experiment is aimed at learning whether it is more efficient to route a given throughput on a single branch or to equally split the traffic along the two branchs.

Area of Interest: Traffic distribution and efficiency

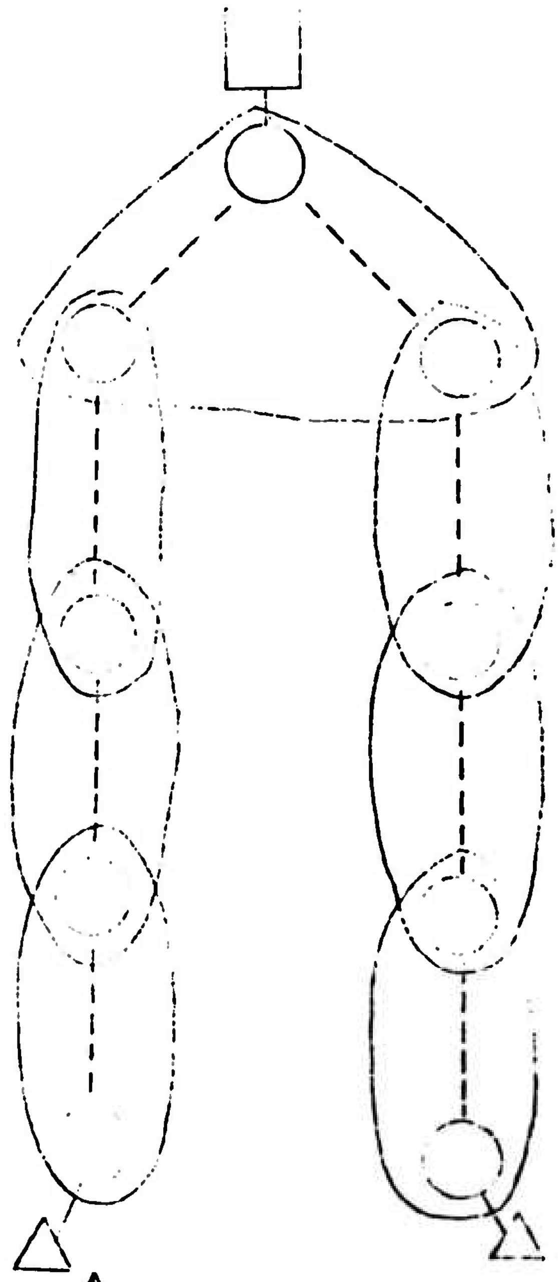
See SRI testbed figure(s): a) 1; b) 3,5,n [modified]

a)



2S

b)



1S 1S

Area of Interest: Traffic distribution and efficiency

Configurations: Two independent paths of repeaters, meeting at the station.

Tools: Cumstats (end-device and PRU)

Parameters: For a range of input rates (minimal channel usage, through saturation), S, apply the traffic as indicated in the accompanying configuration figure.

Data Items: 1) Maximum achievable thourghput (CS-PRU)
2) ETE delay [CS-end-device]

Processing: Compute delay vs throughput for the two methods under the various traffic loads.

ROUTING SENSITIVITY TO TRAFFIC FLOW

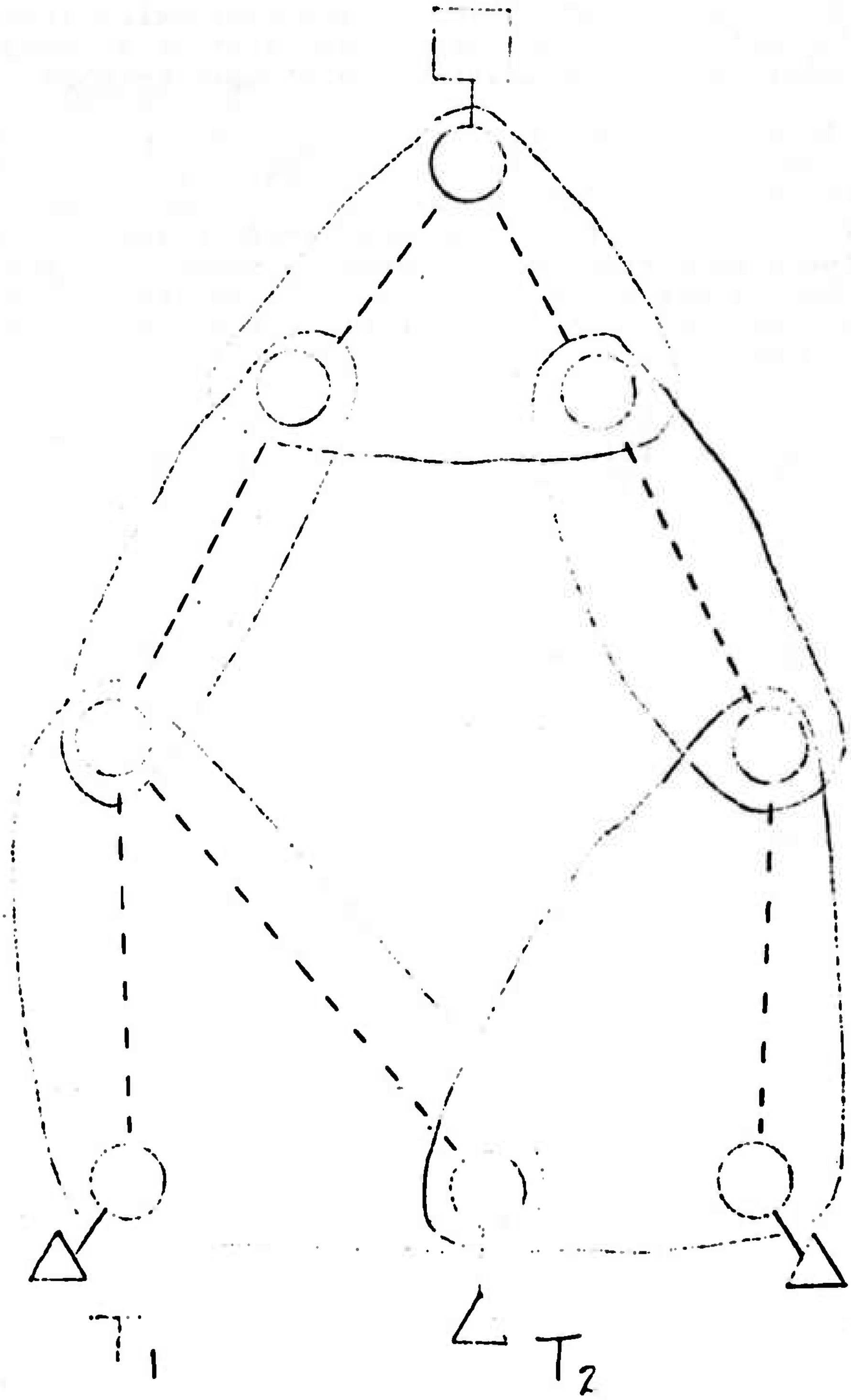
The station routing algorithm is intended to speed the flow of traffic by detecting and responding to non-optimal traffic distribution patterns. In this experiment we will use a configuration in which we systematically create unbalanced traffic patterns to observe the station's response and to determine the threshold at which it will respond.

This configuration also allows for a pattern in which the station's response can only result in the shifting of some traffic from one route to another, thereby recreating a mirror image of the original unbalanced traffic pattern, for which the station's only response could be to return the routes to their original pattern again, etc. An example would be where the respective input rates of the three terminals (see figure with configuration) are:

T1: 1s
T2: 5s
T3: 1s

Area of Interest: Routing sensitivity to traffic flow

See SRI testbed figure(s): 6



Area of Interest: Routing sensitivity to traffic flow

Configurations: Terminals T1 and T3 have separate paths to the station, and terminal T2 can route traffic along either of those paths. The relative input traffic of the three terminals will determine the congestion on each of the two routes.

Tools: Cumstats (end-device and PRU), Snapshots (station)

Data Items:
1) # of packets rx'd and tx'd at each node [CS-PRU]
2) # of packets rx'd at station from each terminal [CS-station] (throughput)
3) route assignments [snapshots-station]
4) ETE delay [CS-end-device]

Parameters: Vary the traffic load on Terminals T1, T2 and T3.

Processing:
1) Find threshold values for routing changes
2) Find effectiveness of routing changes (thoughput, delay)

Comment: In case efficiency is enhanced by uniform distribution of traffic over available branches, we must check for "flipping", i.e. a condition in which the station continually alters T2's route from one branch to the other in an attempt to equalize the traffic flow.

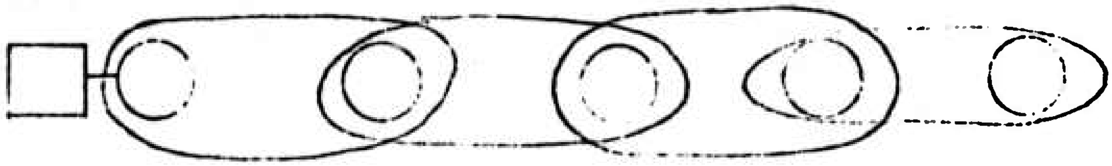
NETWORK INITIALIZATION TIME

This experiment is aimed at the determination of the relationship between network size and initialization time. Using time-stamps of updates in the connectivity and routing tables (initially empty) at the station, the level-by-level progress of initialization can be measured, and the relationship between initialization time and the number of levels and elements per level can be described. The extrapolation of this function to large-scale networks will be of particular interest.

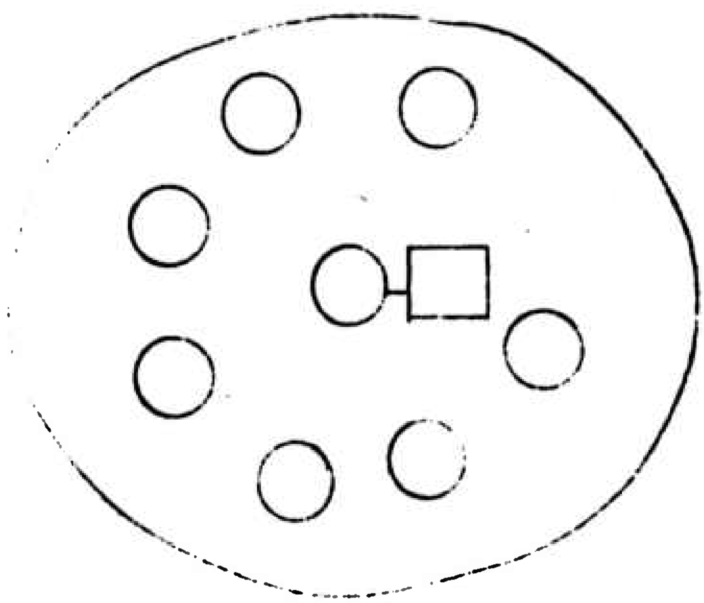
Area of Interest: Network Initialization Time

See SRI testbed figure(s): 1 (STRING)

STRING:



RING:



Area of Interest: Network Initialization Time

Configurations: We will first examine simple configurations such as a string (which corresponds to a tree of K levels, with 1 repeater at each level) and a ring (which corresponds to a tree of 1 level and N elements in that level). It will also be interesting to consider the "largest" network that the test-bed can support, in terms of the number of levels and repeaters at each level.

Tools: Station connectivity table
Station routing table

Parameters: K (number of levels), N (number of repeaters/level); ROP frequencies.

Data Items: Station time-stamp of additions to connectivity and routing tables.

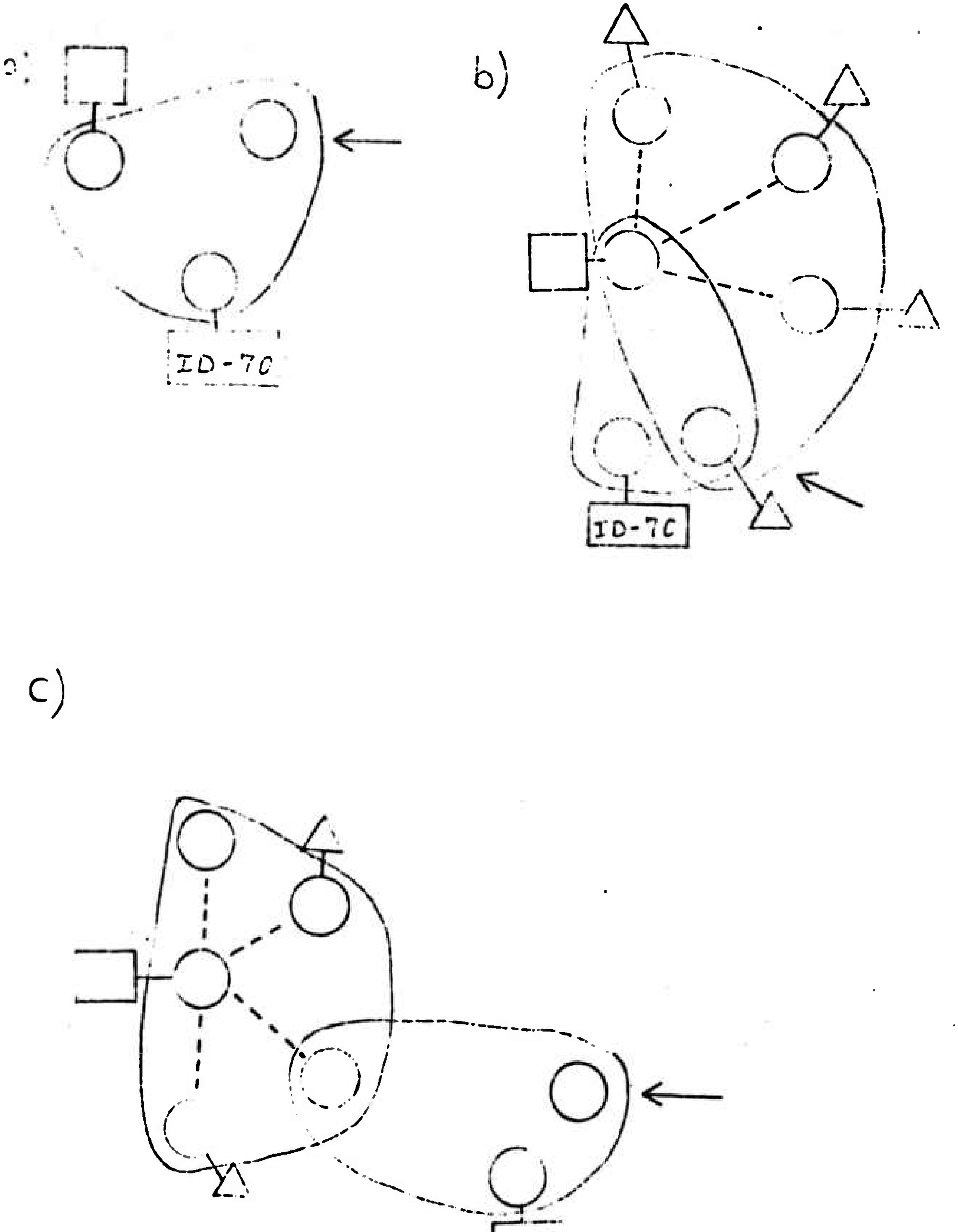
Data Reduction: Progress of initialization: component initialization time in relation to K and N.

COMPONENT INITIALIZATION TIME

This experiment is a companion experiment to the Network Initialization Time experiment, with the exception that we consider here the labeling of a component thrown into an already initialized network. The focus here is on the time of occurrence of various events (first tx of unlabeled ROP; reception of label packet) from the point of view of the added component itself.

Area of Interest: Component Initialization Time

See SRI testbed figure(s): To Be Arranged



Area of Interest: Component Initialization Time

Configurations: Three examples are selected. The first is the simplest: the network consists of only a station; the second is a one-level network, to which is added the unlabeled PRU at the same level (here, the station has to process a larger number of ROP's relayed by the other components); the third involves a multihop network. In the figure, the arrow points to the component to be initialized.

Tools: Interdata-70 with superhose monitor

Parameters: varying traffic rates from pool of terminals, varying ROP frequencies.

Data Items: ID-70 trace of ROPs, label packet(s), labeled ROPs.

Data Reduction: Delay from 1st ROP to rx of label packet.

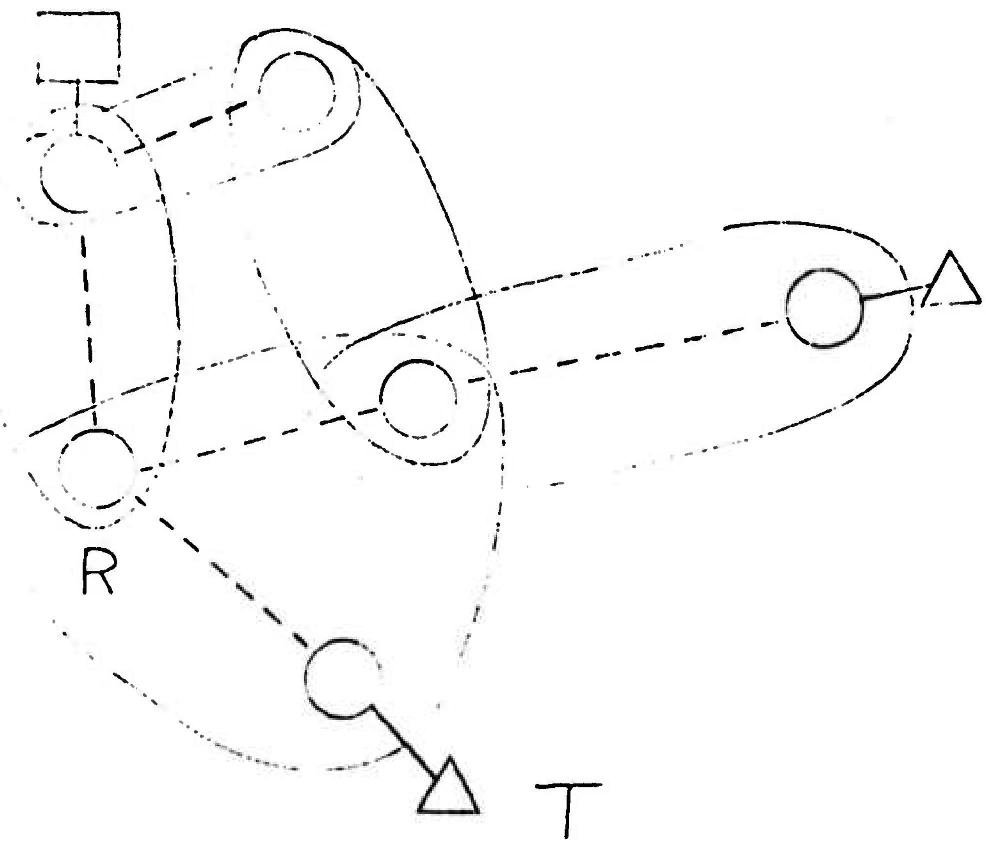
RESPONSE TO COMPONENT FAILURE

A key consideration in network design is the ability to react dynamically, swiftly and efficiently to the sudden loss of a component. In the PRNET we are particluarly concerned with the response of the network to the unexpected failure of a repeater. We want to know how quickly the controlling station can detect and respond to the failure, and also the effects on local traffic of the failure and of the station's response.

The measures of interest will be the lapse of time between failure and the station's response (measured as the relabeling of components affected by the failure); packet delays at affected terminals before failure, after failure and prior to the station's response, and following the station's response. These local effects (around the failed component) will be measured in terms of local single hop delays, and one-way delays (time from source to destination) obtained via the Pickup Packets.

Area of Interest: Response to Component Failure

See SRI testbed figure(s): Possibility: 5 [illegible]



Area of Interest: Response to Component Failure

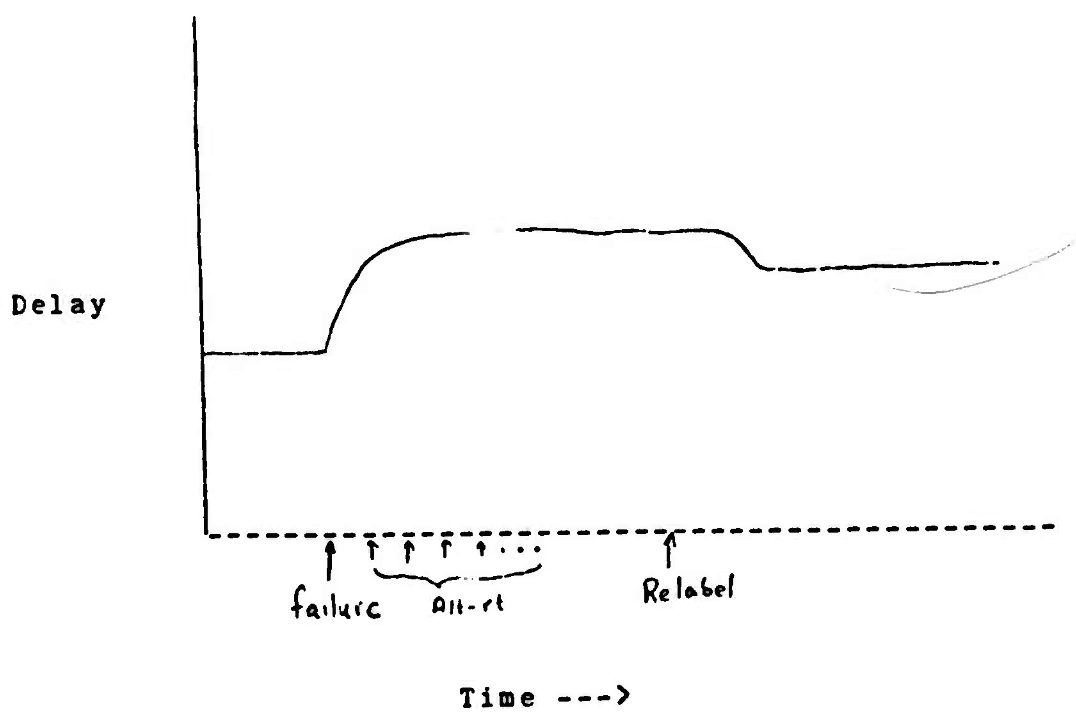
Configurations: We consider the configuration depicted, in which repeater R is the failing repeater. Terminal T is the directly affected component.

Tools: Pickup Packets; Station labels

Parameters: Pickup packets generated at given rates from both terminals.

Data Items: Time-stamps, routes taken [PP]; time new label generated [station], local single-hop delay; number of alt-routed packets between failure and station response.

Data Reduction: On a time-line, show delay between component failure and terminal relabeling, with intermediate events (# of alt-rt pkts rx'd), and plot delay (single-hop and one-way) [PP] as shown below:

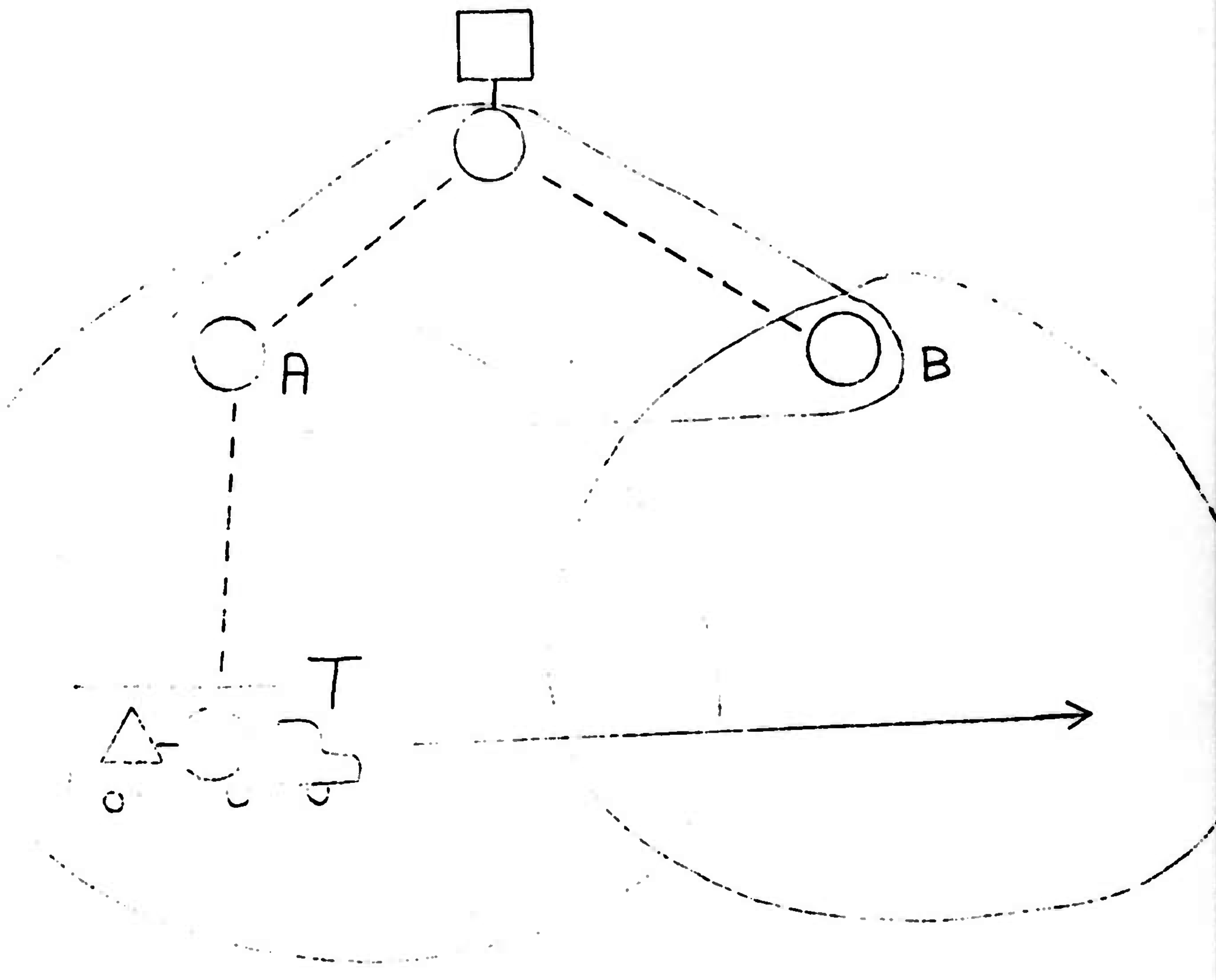


MOBILE TERMINAL

With mobile terminals, the issue is the efficient relaying of traffic originating at the terminals as they move from one repeater's range to another's. The measures of efficiency are the percent of time the terminal PRU is using the wrong label (and thus is alternately routing), and the single-hop delay from the terminal to its relaying repeater, and ETE delay.

Area of Interest: Mobile Terminals

See SRI testbed figure(s): Possibility: 5 (modified)



Area of Interest: Mobile Terminals

Configuration: A simple configuration for this experiment has been selected. As terminal T moves, it leaves repeater A's range to enter repeater B's range.

Tools: CumStat (PRU and End-Device); Pickup Packets

Parameters: RCP rates, # of tx's before alt-routing.

Data Items: Histogrms of tx's to success [CS-PRU]
ETE delay [CS-End Device]
1-hop delay [PP]

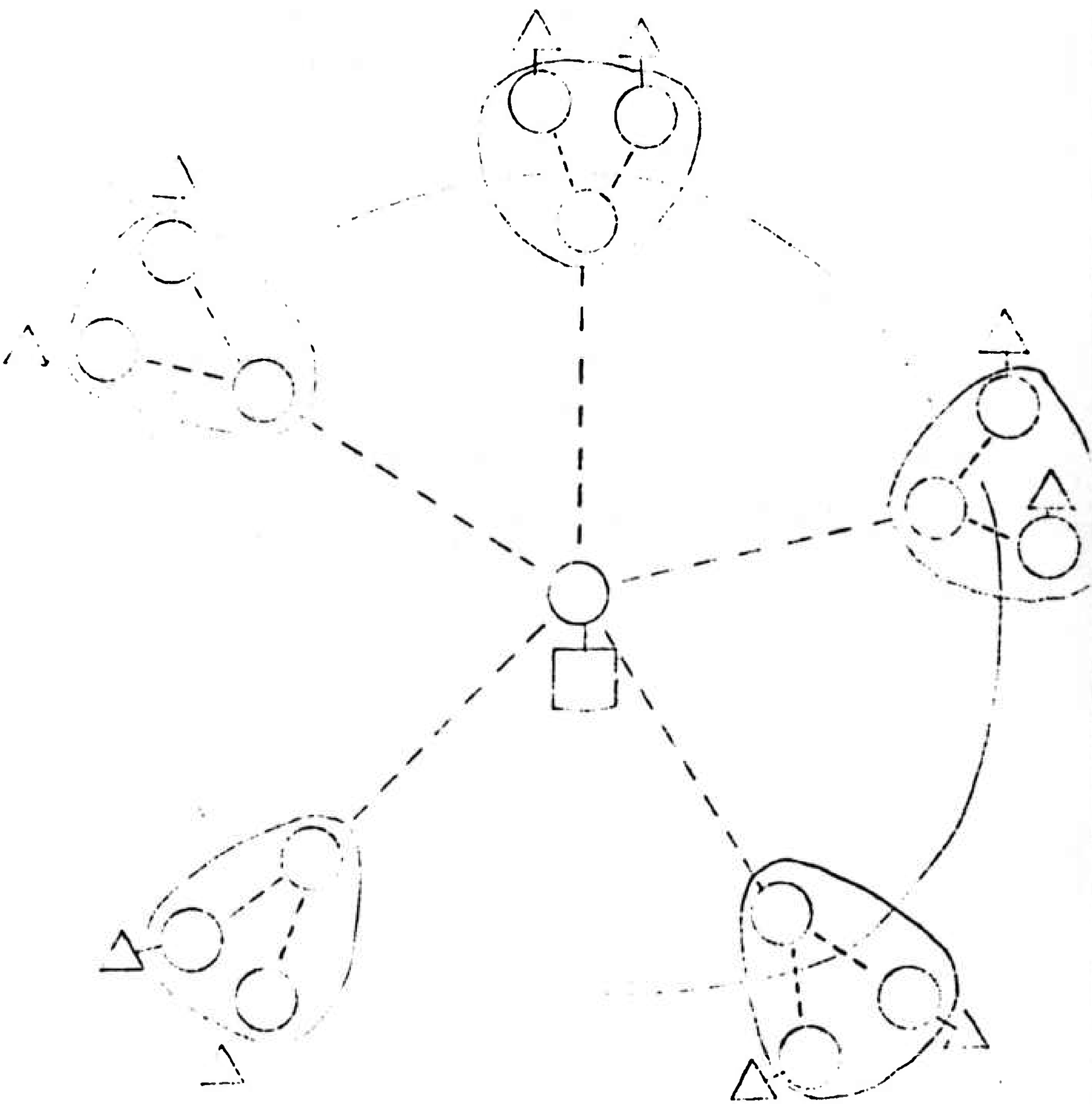
Processing: Use alt-routing data from CumStat histograms to compute the percent of time the terminal PRU was using the wrong label. Correlate with the ETE and 1-hop delays.

EFFECTS OF 'N' ON CSMA

The analysis of a simple two-hop network configuration using slotted ALOHA and CSMA was published in PRTNs 247, 248, and 249. In this experiment we intend to use the same configuration and attempt to verify some of the analytic results, mostly those relating to CSMA (since it is implemented in the PRNET). Basically the analysis has considered the fully-connected two-hop network configuration in which all repeaters are within range and in line-of-sight of each other and of the station. With each repeater is associated a population of terminals generating traffic which is destined to the station. Terminals and repeaters follow the nonpersistent CSMA protocol.

Area of Interest: Effect of # of Repeaters on Access Mode Performance

See SFI testbed figure(s): To Be Arranged



Area of Interest: Effect of # of Repeaters on Access Mode Performanc.

Configuration: A ring of repeaters around the station, each repeater with it's population of terminals that have connectivity only with their repeater.

Tools: CumStats (end-device and PRU), Pickup packets

Parameters: Traffic sources sending information and PPs (of various sizes) at various rates s such that
0 LE Ns LE f(N) (as per analysis)
e.g. if N = 10 then analysis indicates f(10) = 0.43
Other parameters: (i) retransmission delay;
(ii) Initial transmission delay (0, i.e. immediate first transmission IFT; and non-zero, i.e. delayed first trnasmission DFT).

Data Items: # of tx's to success [CS-PRU]
# of successful tx's [CS-PRU]
Round trip delay [CS-end device]
Access delay (terminal to repeater) [PP]
Delay at repeater [PP]

Processing: Mean and standard deviation of items above

Delay vs throughput for various N (0 < N <= 10)
Maximum throughput vs N

Comments: Do same with ALOHA schemes

PART III

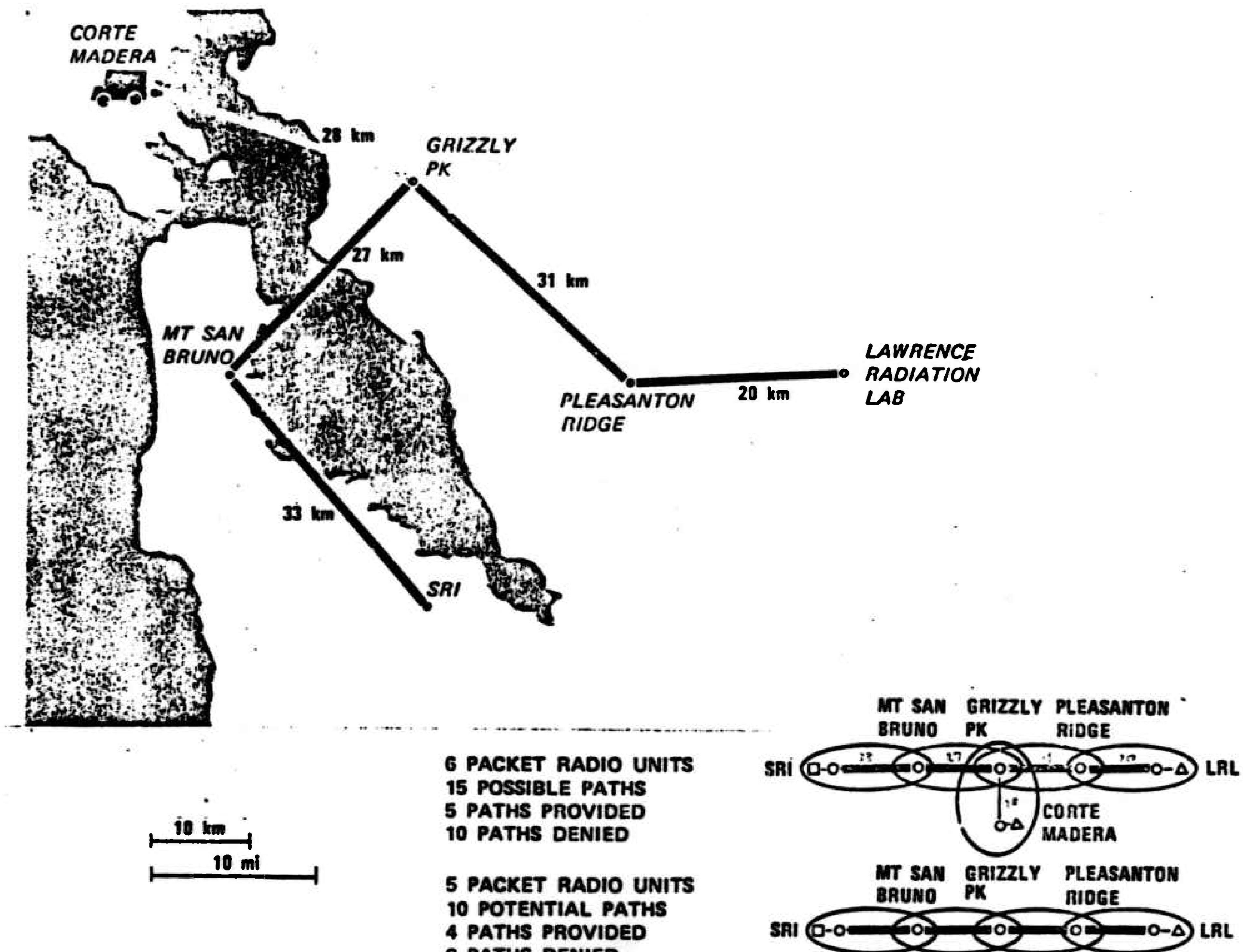
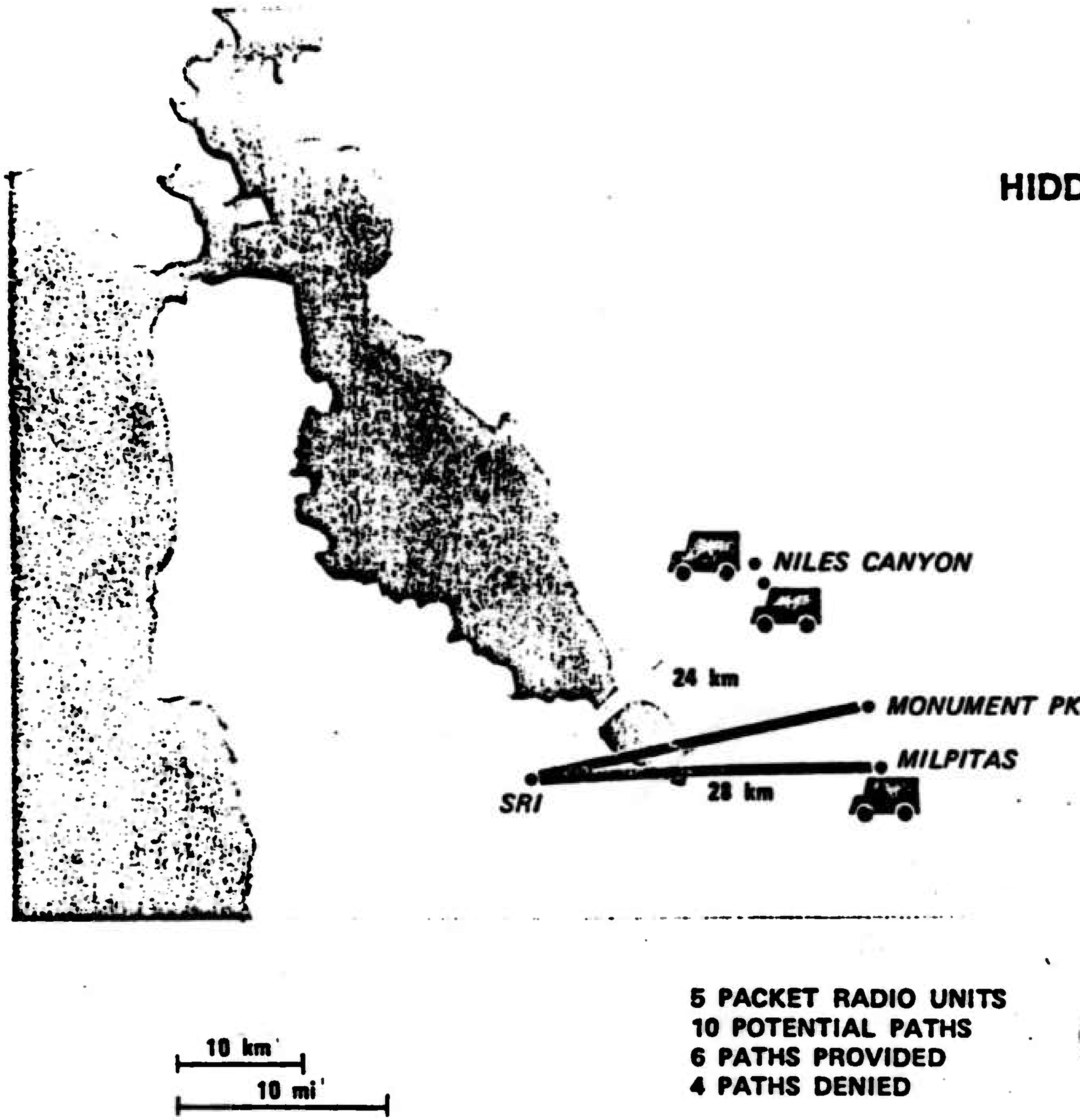

CORTE MADERA
28 km
GRIZZLY PK
27 km
31 km
MT SAN BRUNO
LAWRENCE RADIATION LAB
PLEASANTON RIDGE
20 km
33 km
SRI
10 km
10 mi
6 PACKET RADIO UNITS
15 POSSIBLE PATHS
5 PATHS PROVIDED
10 PATHS DENIED
MT SAN BRUNO
GRIZZLY PK
PLEASANTON RIDGE
SRI
LRL
CORTE MADERA
5 PACKET RADIO UNITS
10 POTENTIAL PATHS
4 PATHS PROVIDED
6 PATHS DENIED
MT SAN BRUNO
GRIZZLY PK
PLEASANTON RIDGE
SRI
LRL

Figure 1

HIDDEN TERMINALS



5 PACKET RADIO UNITS
10 POTENTIAL PATHS
6 PATHS PROVIDED
4 PATHS DENIED

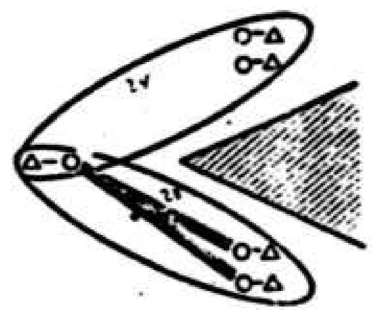


Figure 2

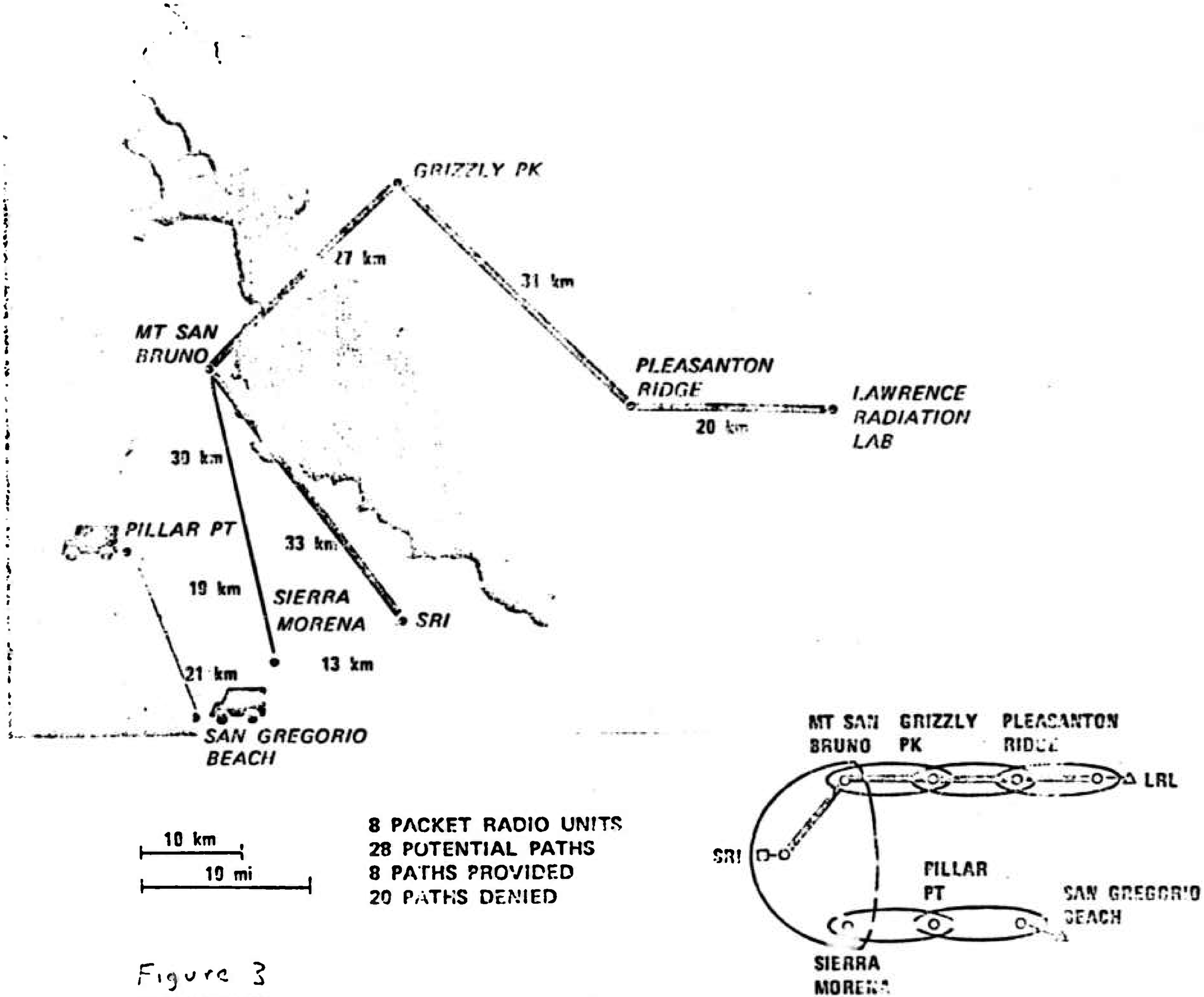

GRIZZLY PK
27 km
31 km
MT SAN BRUNO
PLEASANTON RIDGE
20 km
LAWRENCE RADIATION LAB
30 km
PILLAR PT
33 km
19 km
SIERRA MORENA
SRI
13 km
21 km
SAN GREGORIO BEACH
10 km
10 mi
8 PACKET RADIO UNITS
28 POTENTIAL PATHS
8 PATHS PROVIDED
20 PATHS DENIED
MT SAN BRUNO
GRIZZLY PK
LRL
SRI
PILLAR PT
SIERRA

Figure 3

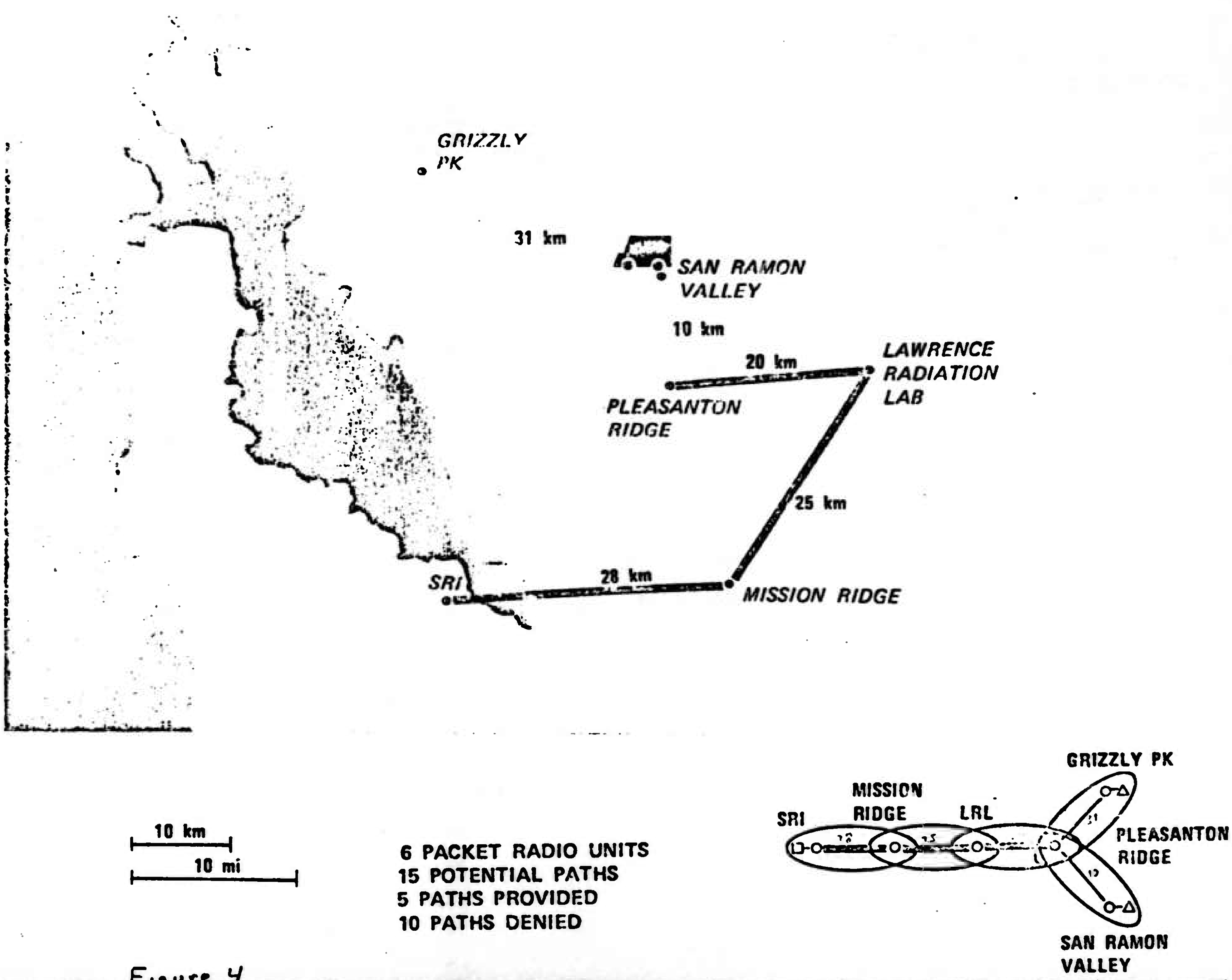
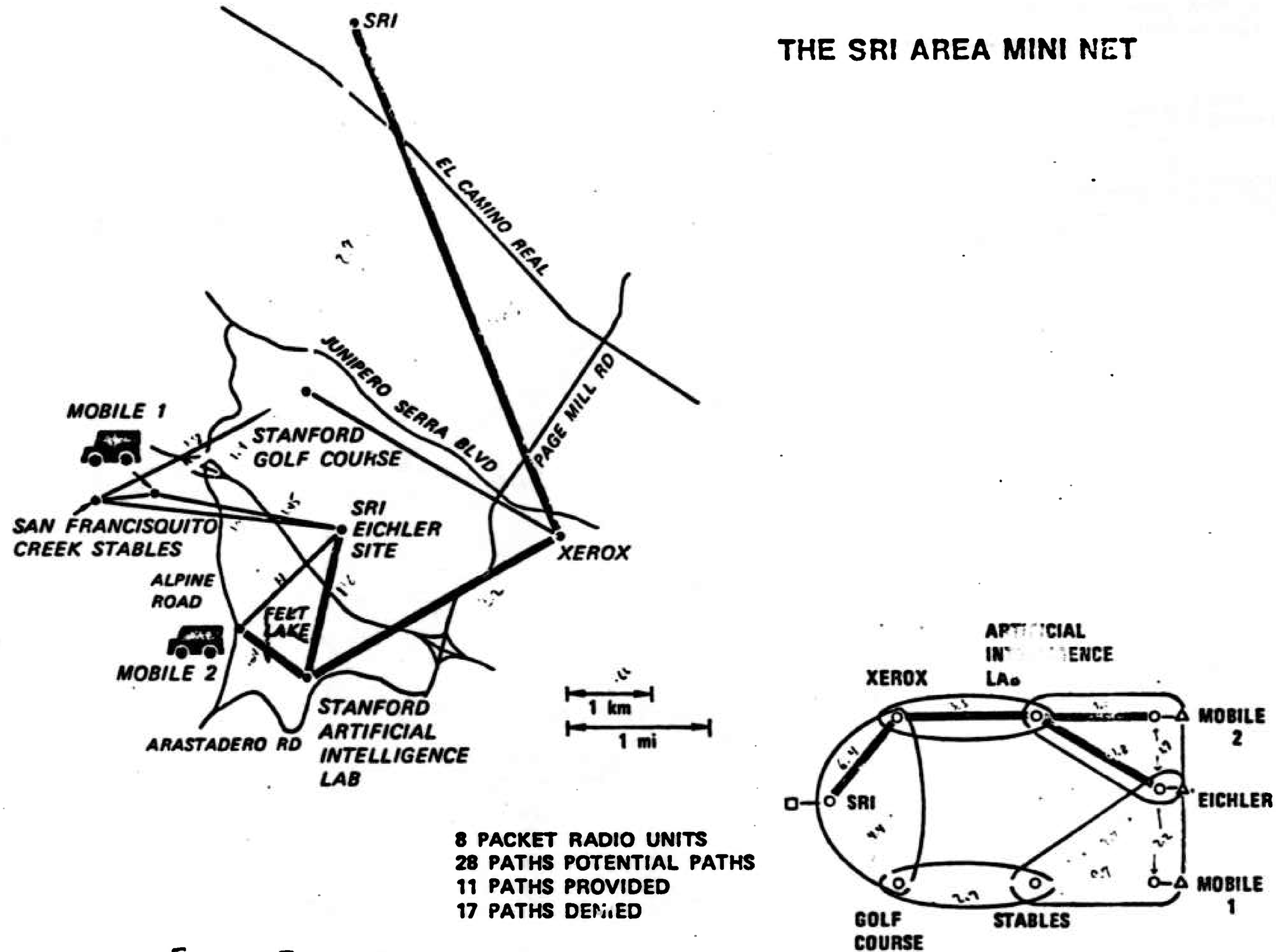

GRIZZLY
PK
31 km
SAN RAMON
VALLEY
10 km
20 km
LAWRENCE
RADIATION
LAB
PLEASANTON
RIDGE
25 km
28 km
SRI
MISSION RIDGE
10 km
10 mi
6 PACKET RADIO UNITS
15 POTENTIAL PATHS
5 PATHS PROVIDED
10 PATHS DENIED
SRI
MISSION
RIDGE
LRL
GRIZZLY PK
PLEASANTON
RIDGE
SAN RAMON
VALLEY

Figure 4

THE SRI AREA MINI NET



8 PACKET RADIO UNITS
28 PATHS POTENTIAL PATHS
11 PATHS PROVIDED
17 PATHS DENIED

Figure 5

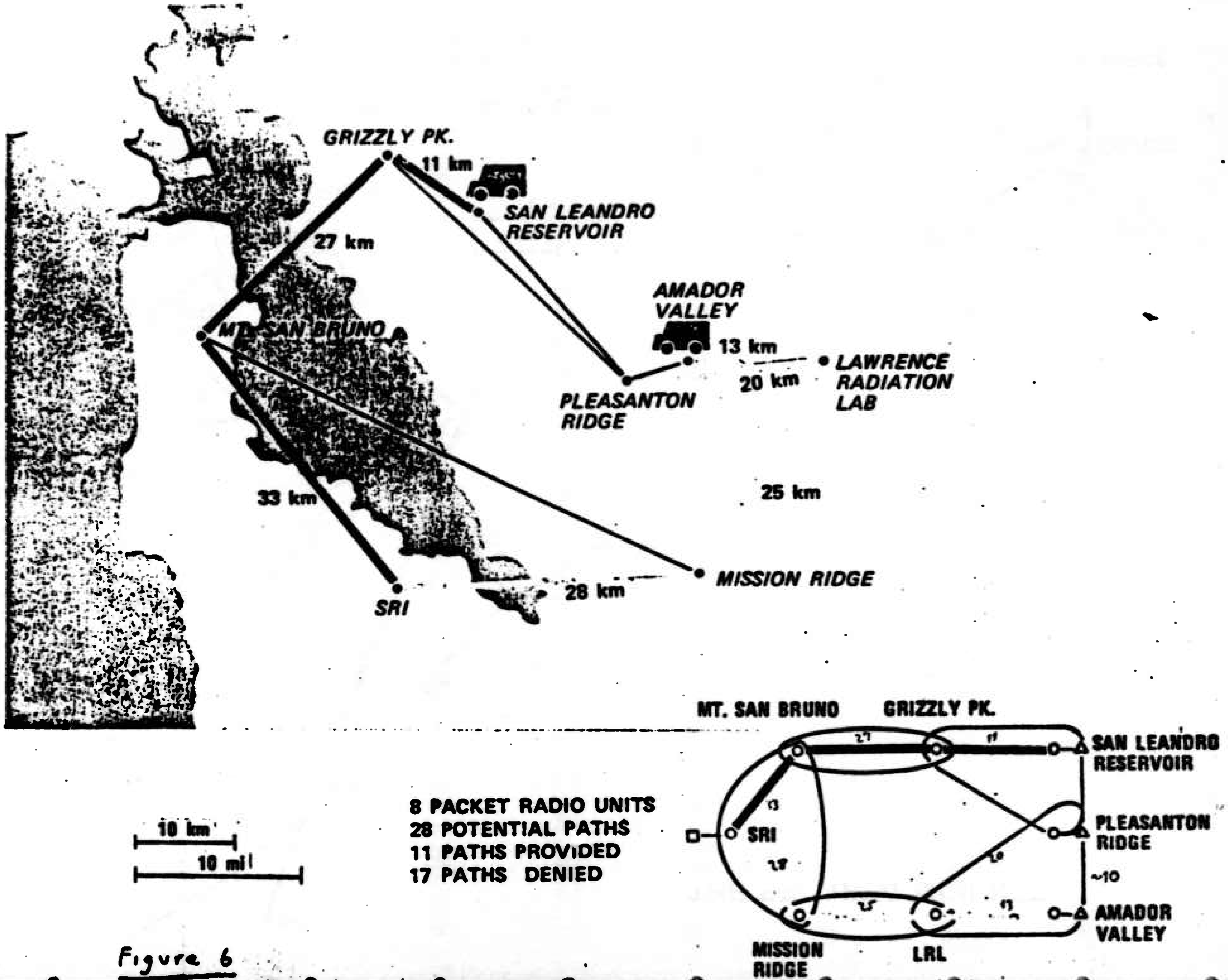
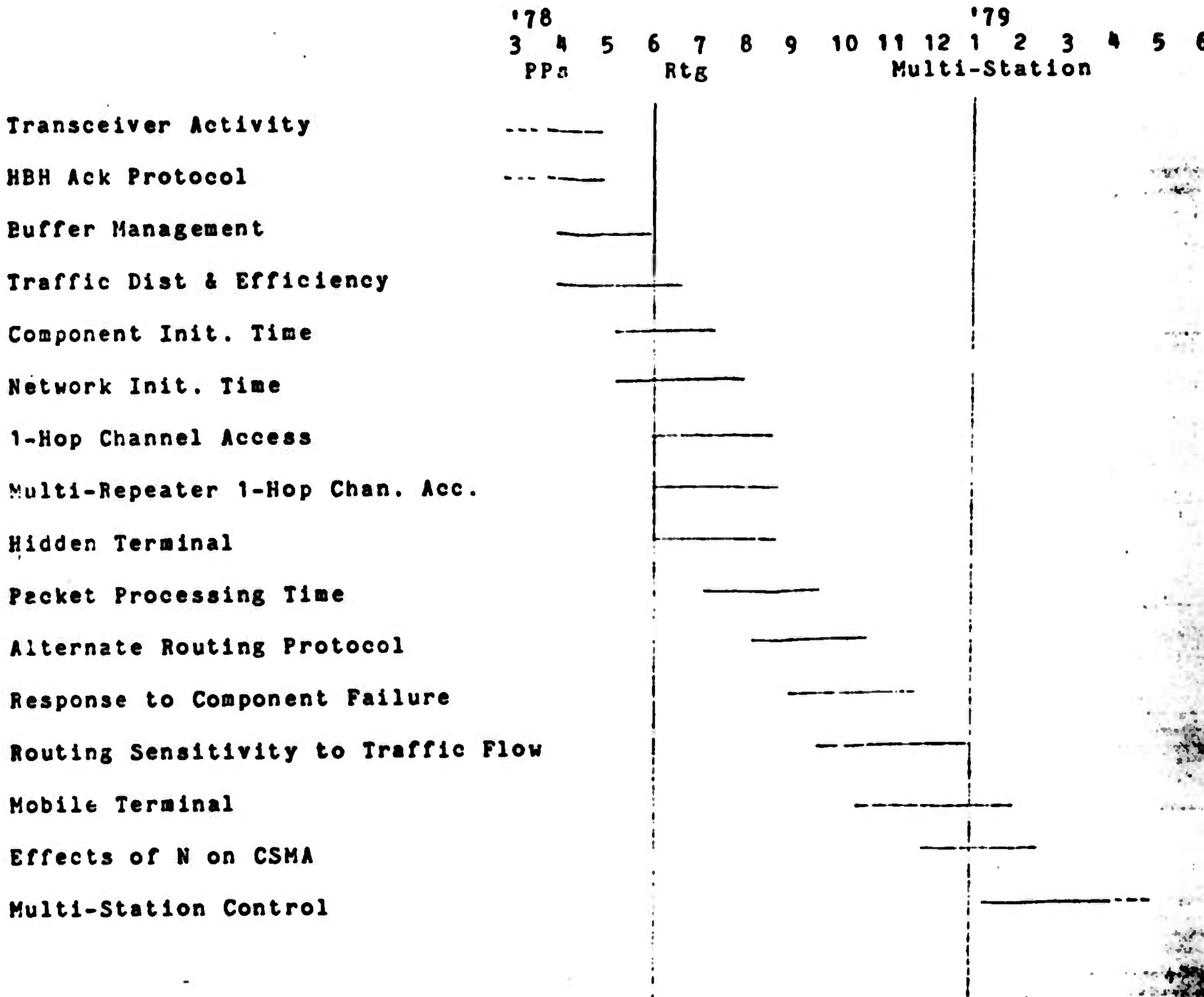

GRIZZLY PK.
11 km
SAN LEANDRO RESERVOIR
27 km
AMADOR VALLEY
MT. SAN BRUNO
13 km
LAWRENCE RADIATION LAB
20 km
PLEASANTON RIDGE
33 km
25 km
MISSION RIDGE
28 km
SRI
10 km
10 mi
8 PACKET RADIO UNITS
28 POTENTIAL PATHS
11 PATHS PROVIDED
17 PATHS DENIED
MT. SAN BRUNO
GRIZZLY PK.
SAN LEANDRO RESERVOIR
PLEASANTON RIDGE
SRI
~10
AMADOR VALLEY
MISSION RIDGE
LRL

Figure 6

PART IV

Tentative Working Schedule



S

Current Measurement Results

for

Hop-by-Hop Acknowledgement Scheme Experiment

Zaw-Sing Su

UCLA

October 1978

I. PURPOSE

Purpose of the hop-by-hop acknowledgement scheme experiment is to characterize the echo acknowledgement scheme which is used in the current PRNET implementation. A comparison with other schemes (e.g., active acknowledgement scheme) may be done when the PRNET simulation package becomes available.

II. DESCRIPTION OF THE EXPERIMENT

1. Testbed Layout

Configuration of the experimental network was as depicted in Appendix A. The network consisted of 3 repeaters and 3 terminals with T1, T2 as source terminal PRU's, and T3 a sink terminal PRU. The two source terminals received input traffic at the same rate throughout this experiment. The sink terminal end-to-end acknowledged each packet it received. Due to the constraints on the possible number of connections to the station for delivering measurement statistics, this experiment was run with connections only to the 3 terminals and the middle repeater, R.

2. Traffic Generator

The traffic generator used for this experimnet generated packets of constant length. The input traffic was generated such that in the network there could be at most one outstanding (unacknowledged) packet for each input terminal at one time. A new packet could only be generated after the source terminal received an end-to-end acknowledgement for the previous packet. The interval between the receipt of an end-to-end acknowledgement and the generation of the next packet is termed the post-ACK delay. The post-ACK delay is the parameter for setting the input traffic rate. In reality, the inter-generation interval of two consecutive input packets was the sum of the round-trip delay for the first packet and the preset post-ACK delay. The actual input rate was therefore dynamically dependent upon the round-trip delay and therefore the network behaviour. Consequently, when studying the results below, one should bear in mind the effects of this implicit flow control mechanism.

3. Channel Access Mode

The non-persistent CSMA (Carrier Sense Multiple Access) mode was used throughout the experiemnt.

4. Hop-by-Hop Acknowledgement Scheme

The echo acknowledgement Scheme was adopted for hop-by-hop acknowledgement throughout this experiment, i.e., active acknowledgements were used only where necessary.

5. Scheme for Varying Transmission Delay

When a new packet was to be transmitted over a hop, a predetermined initial transmission delay (RTXDLY) was applied. A retransmission would occur if the packet is not acknowledged after a retransmission delay, where

$$\text{retransmission delay} = RTXDLY + n * RETXDY,$$

where RETXDY is a predetermined 'unit retransmission delay' and n the number of the retransmission. For example, a retranmission delay of (RTXDLY + 2 * RETXDY) would apply to the second retransmission (or the third transmission attempt) of a packet.

6. Parametric Variables

The experiment was run with the following parameters varied:

i) input rate (set via post-ACK delay),
ii) delay for the first transmission (RTXDLY),
iii) unit retransmission delay (RETXDY), and
iv) packet length.

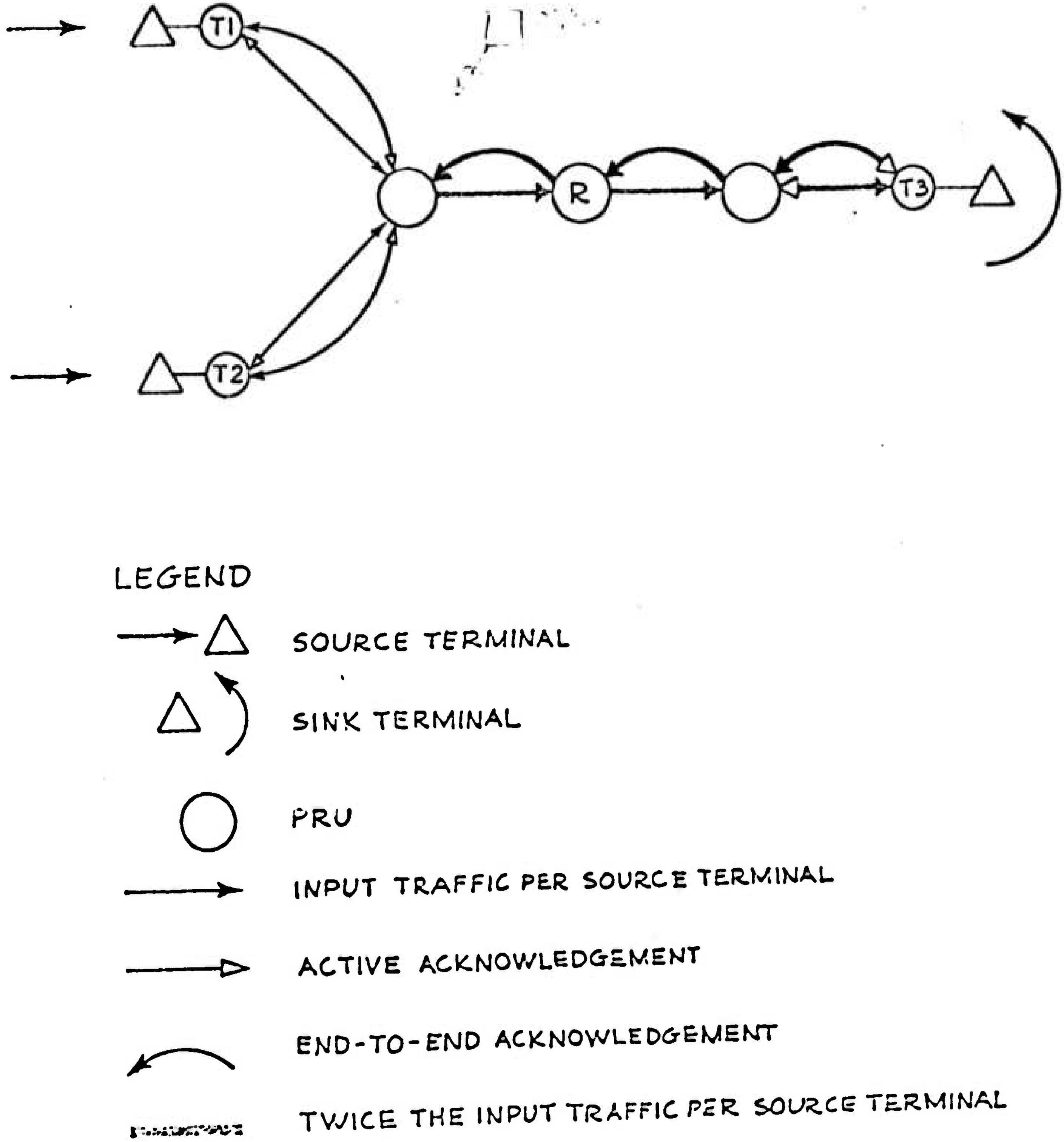
III. MEASUREMENT RESULTS (Appendix B)

The results are presented in two groups:

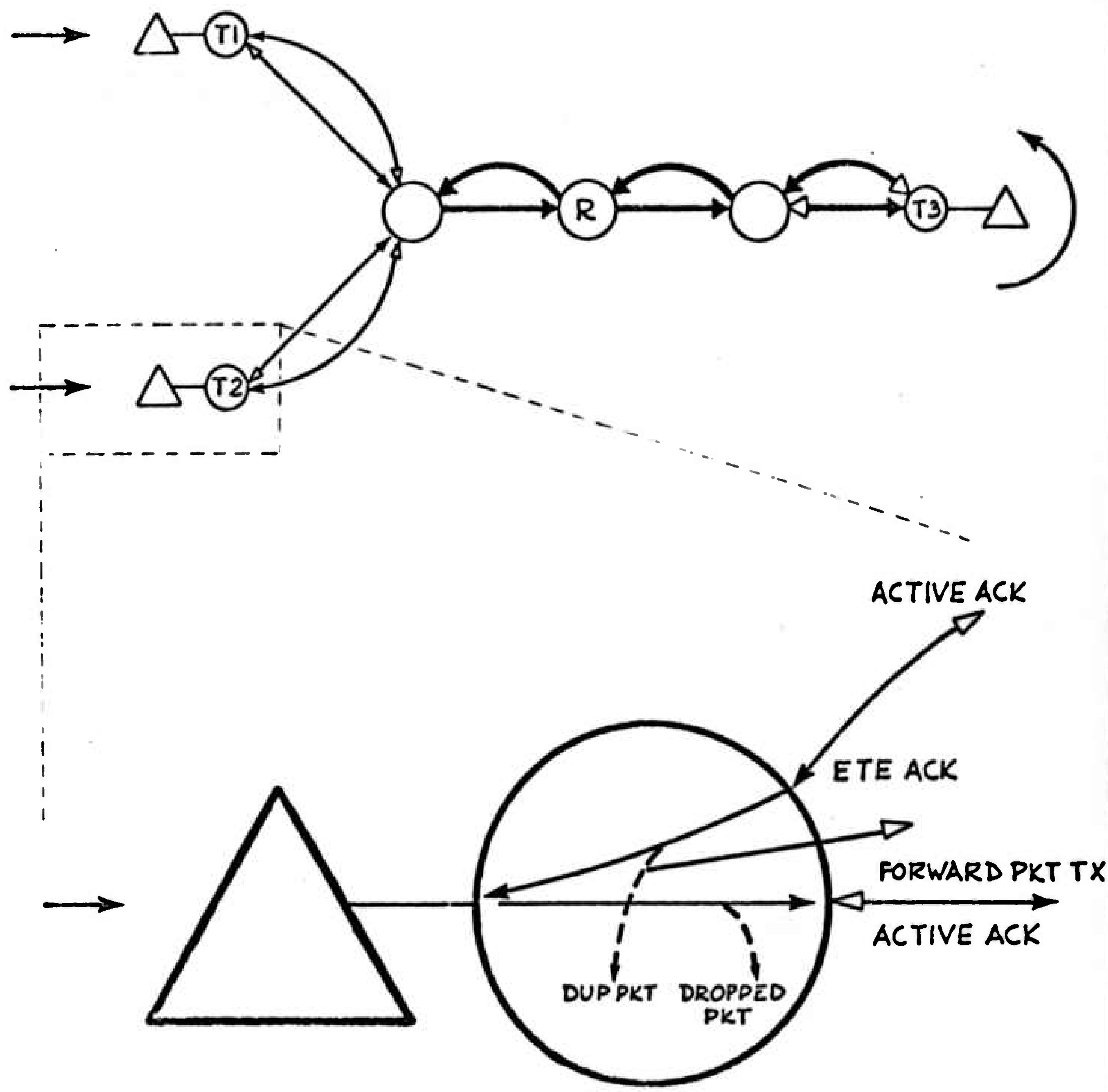
i) with varying input rate, and
ii) with varying transmission delays.

The results obtained for varying packet length are too limited for any meaningful interpretation and therefore are not included.

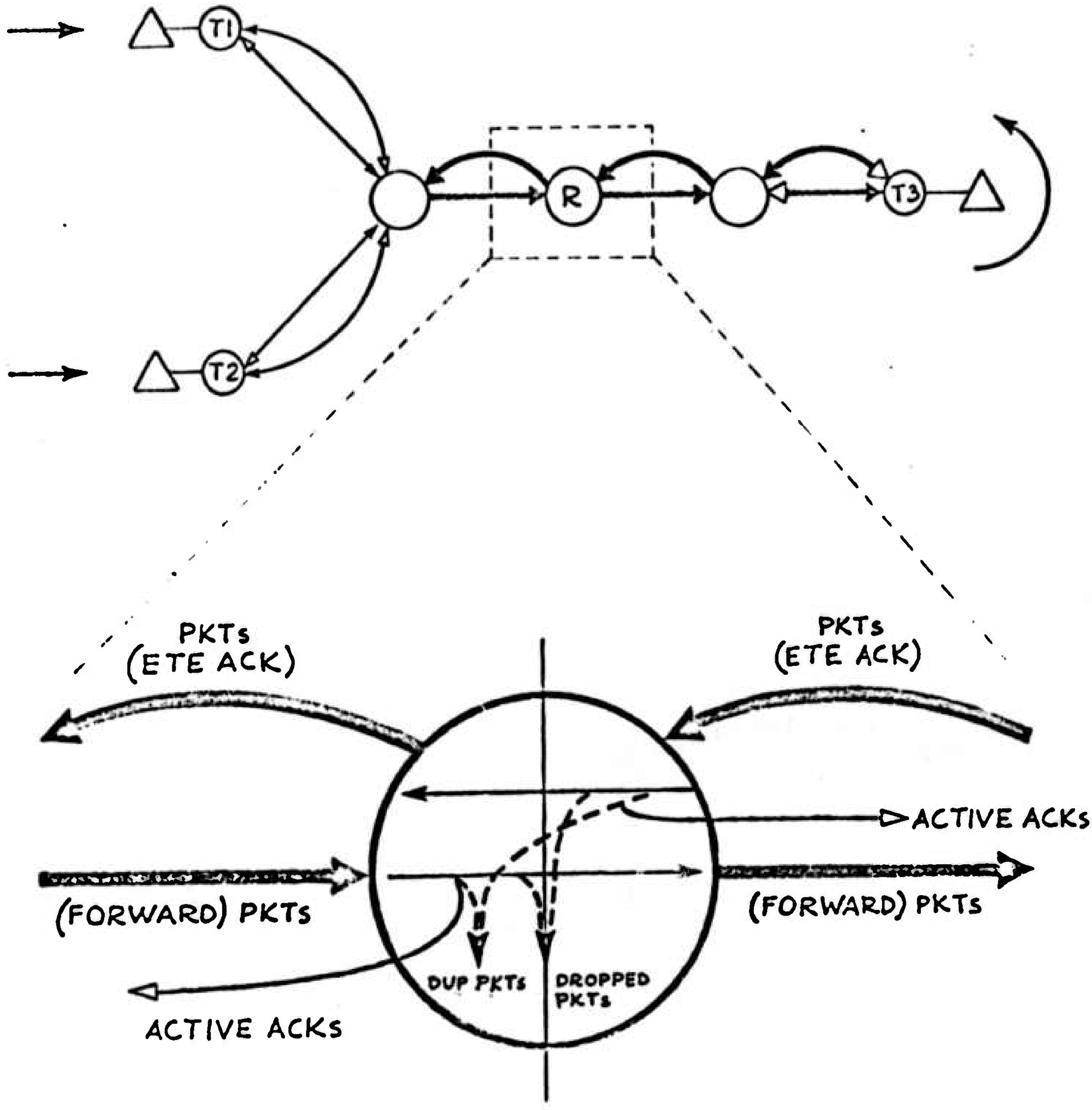
APPENDIX A: CONFIGURATION



Testbed Layout

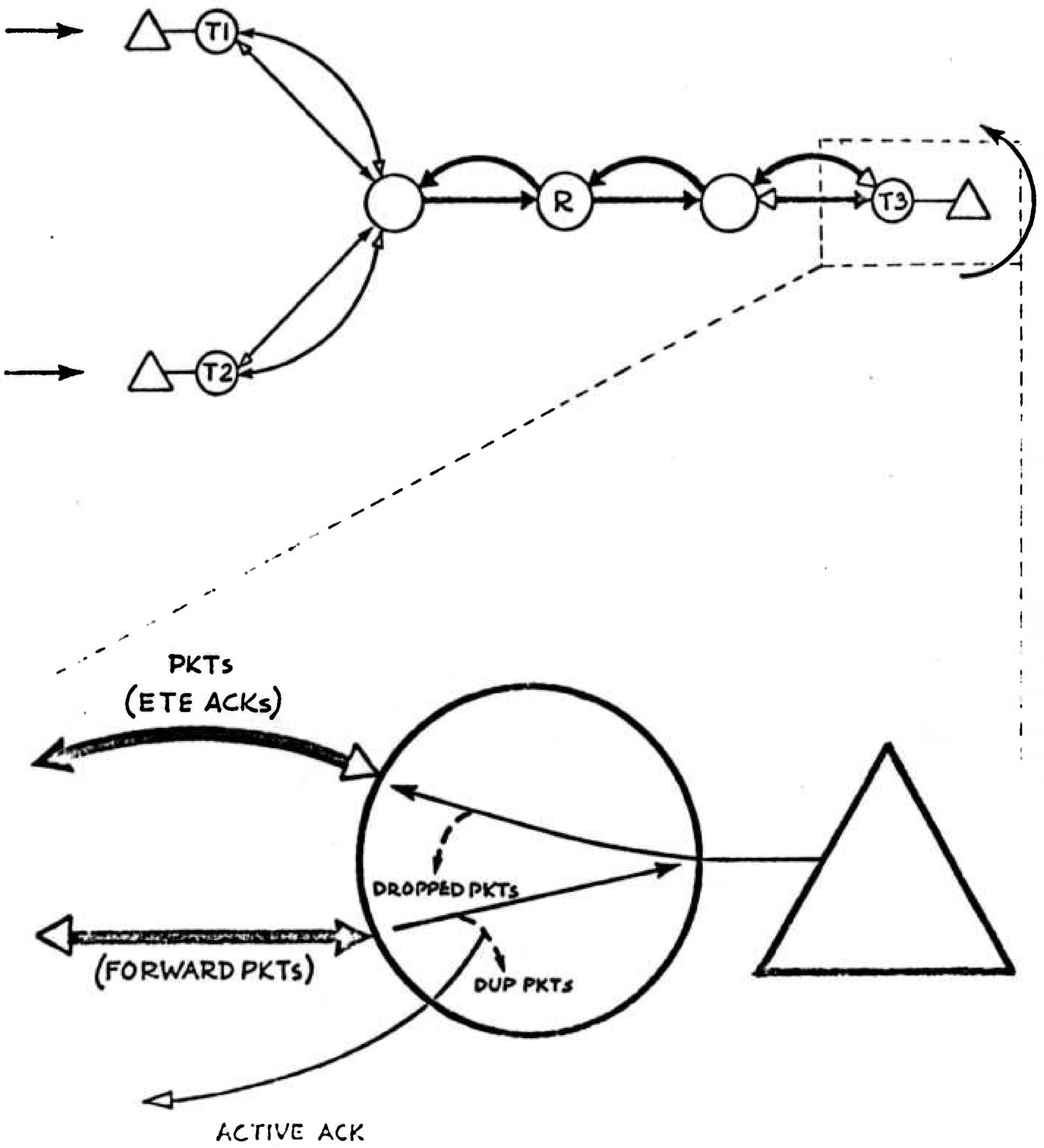


SOURCE TERMINAL



T1
T2
R
T3
PKTs
(ETE ACK)
PKTs
(ETE ACK)
ACTIVE ACKs
(FORWARD) PKTs
(FORWARD) PKTs
DUP PKTs
DROPPED
PKTs
ACTIVE ACKs

REPEATER R



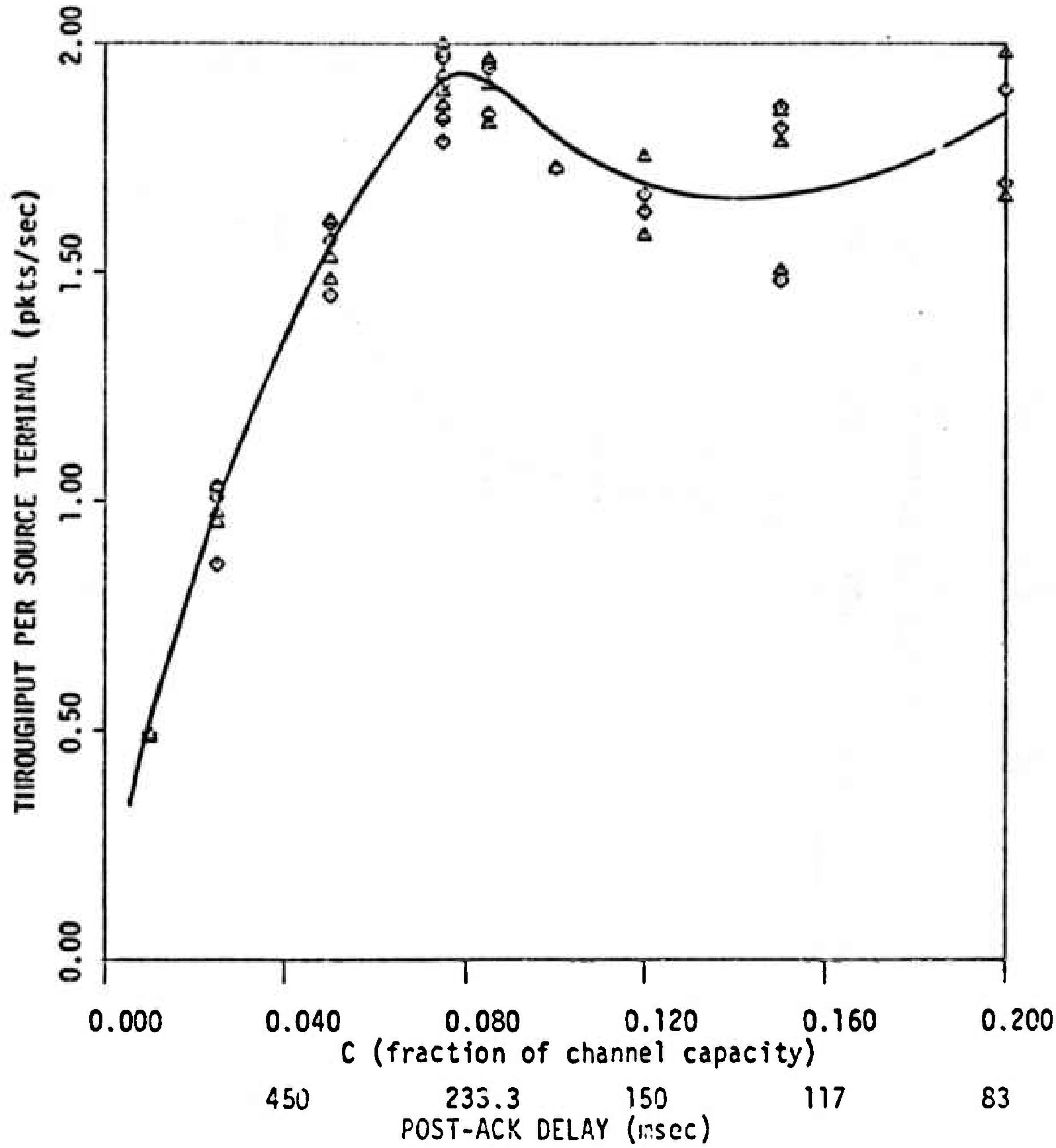
SINK TERMINAL

APPENDIX B: MEASUREMENT RESULTS

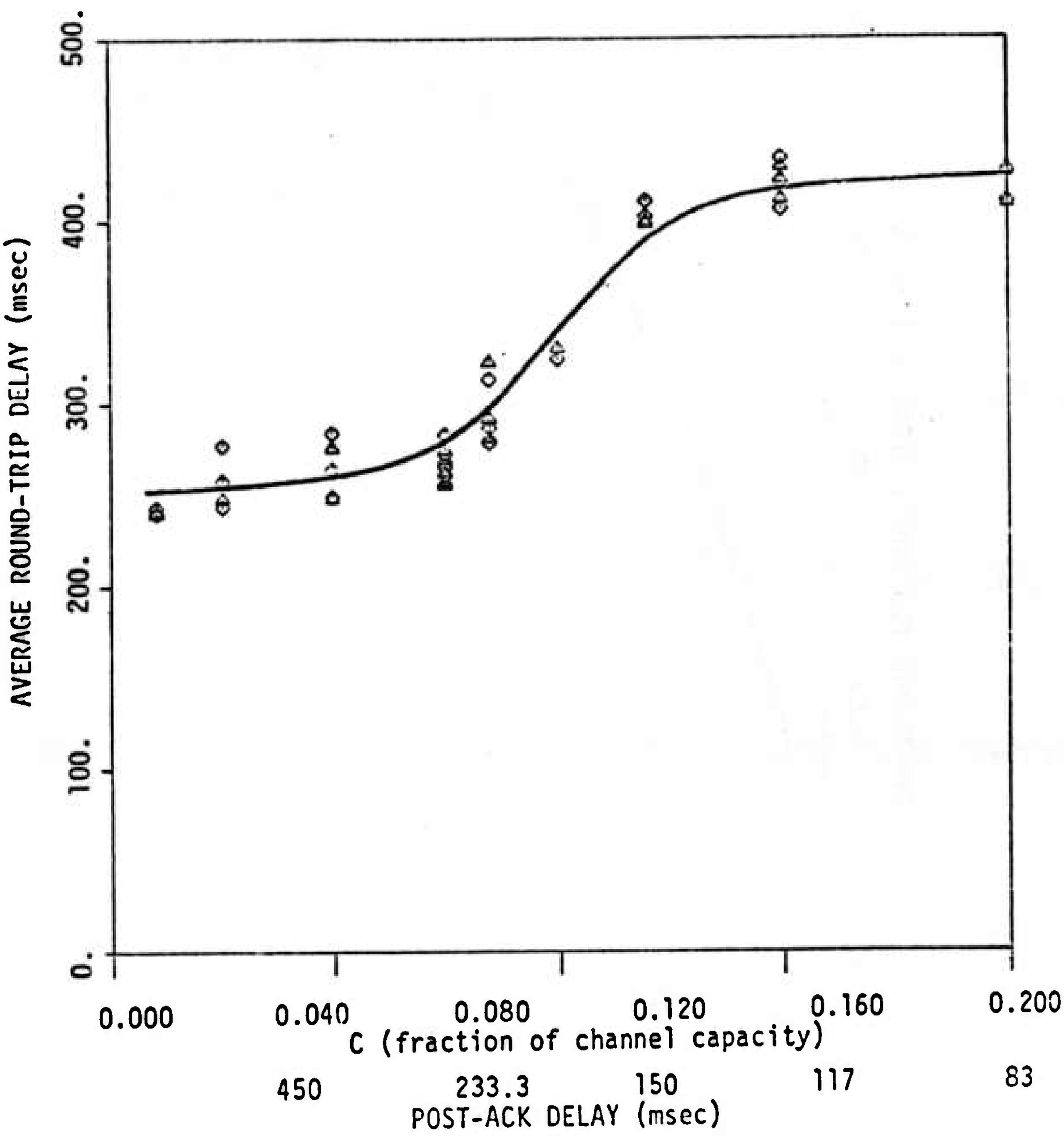
I. Network and Component Behaviour under Varying Input Rate with

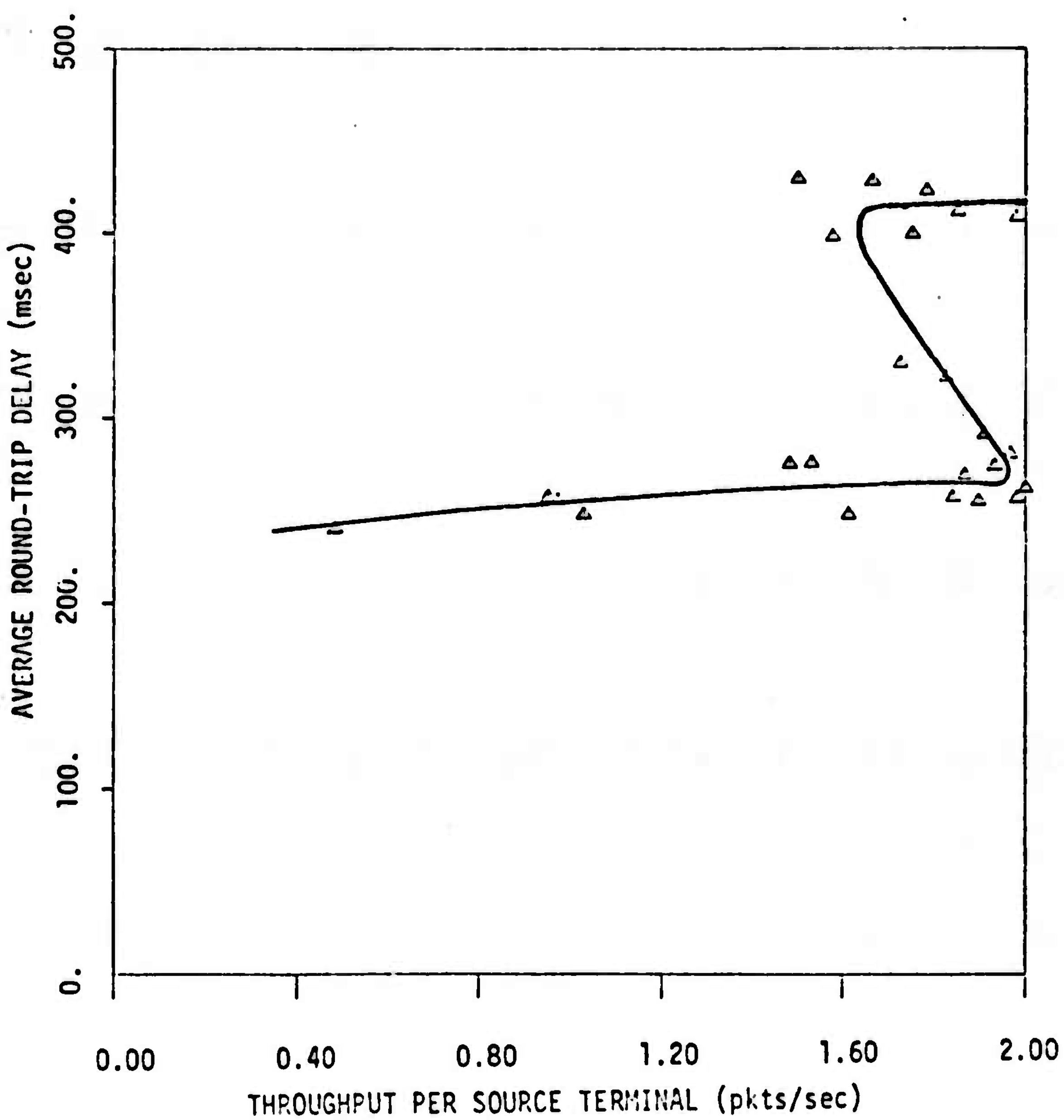
- Transmission Delay (RTXDLY) = 8.2 msec
- Unit Retransmission Delay (RETXDY) = 10.24 msec
- Packet Length = 17 words (total for transmission)

1. Network Round-Trip Behaviour

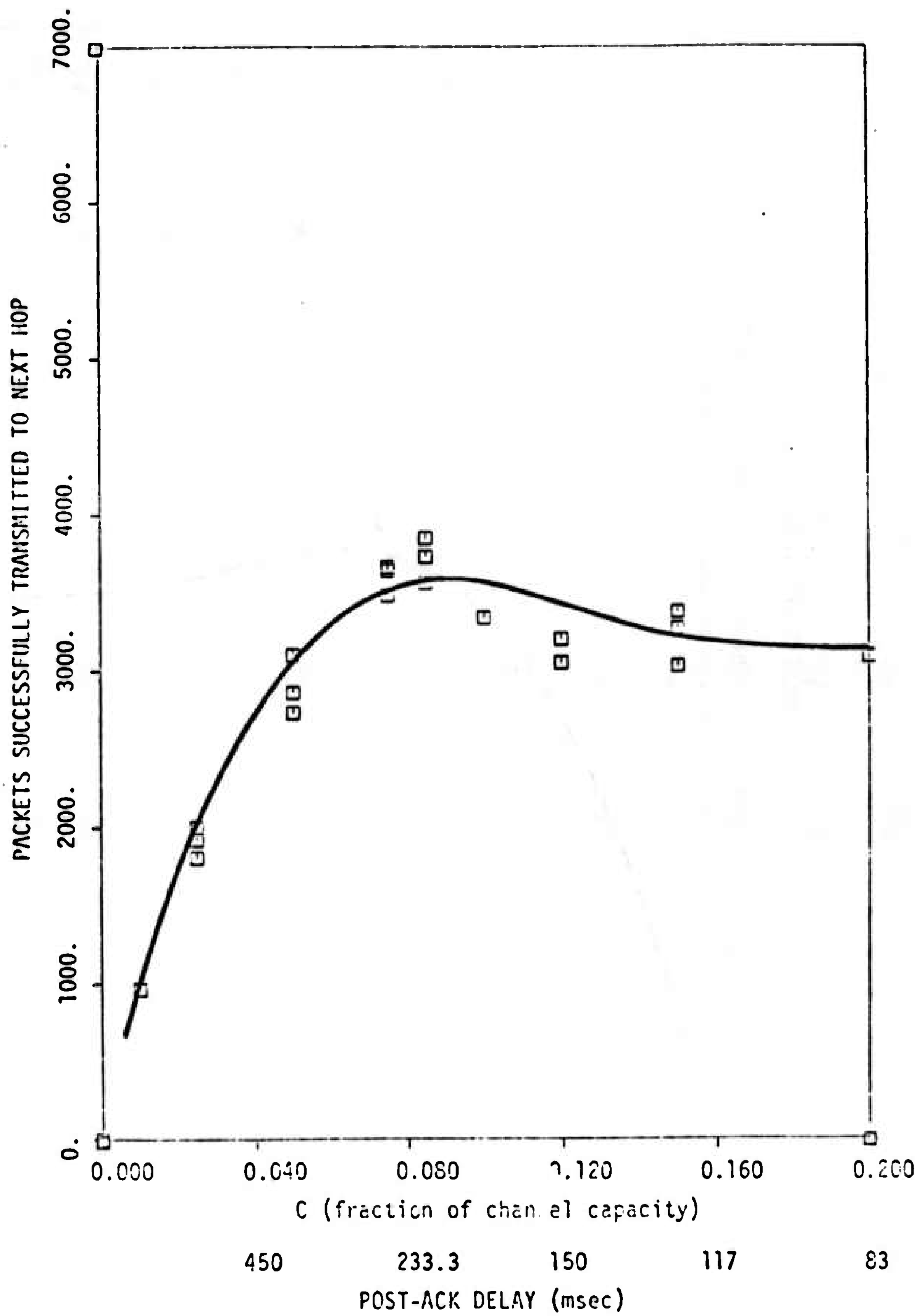


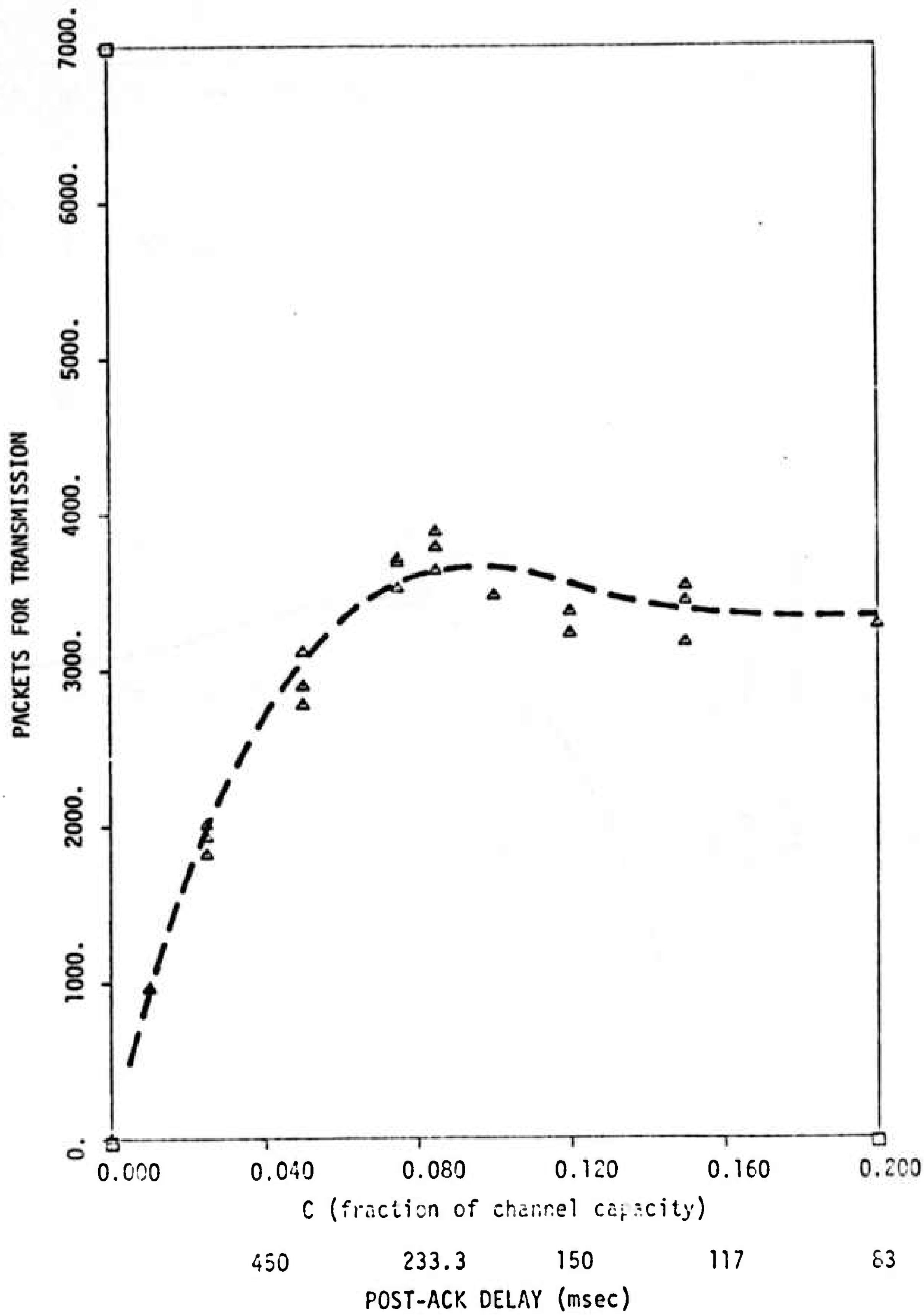
THROUGHPUT PER SOURCE TERMINAL (pkts/sec)
0.00
0.50
1.00
1.50
2.00
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)


AVERAGE ROUND-TRIP DELAY (msec)
500.
400.
300.
200.
100.
0.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

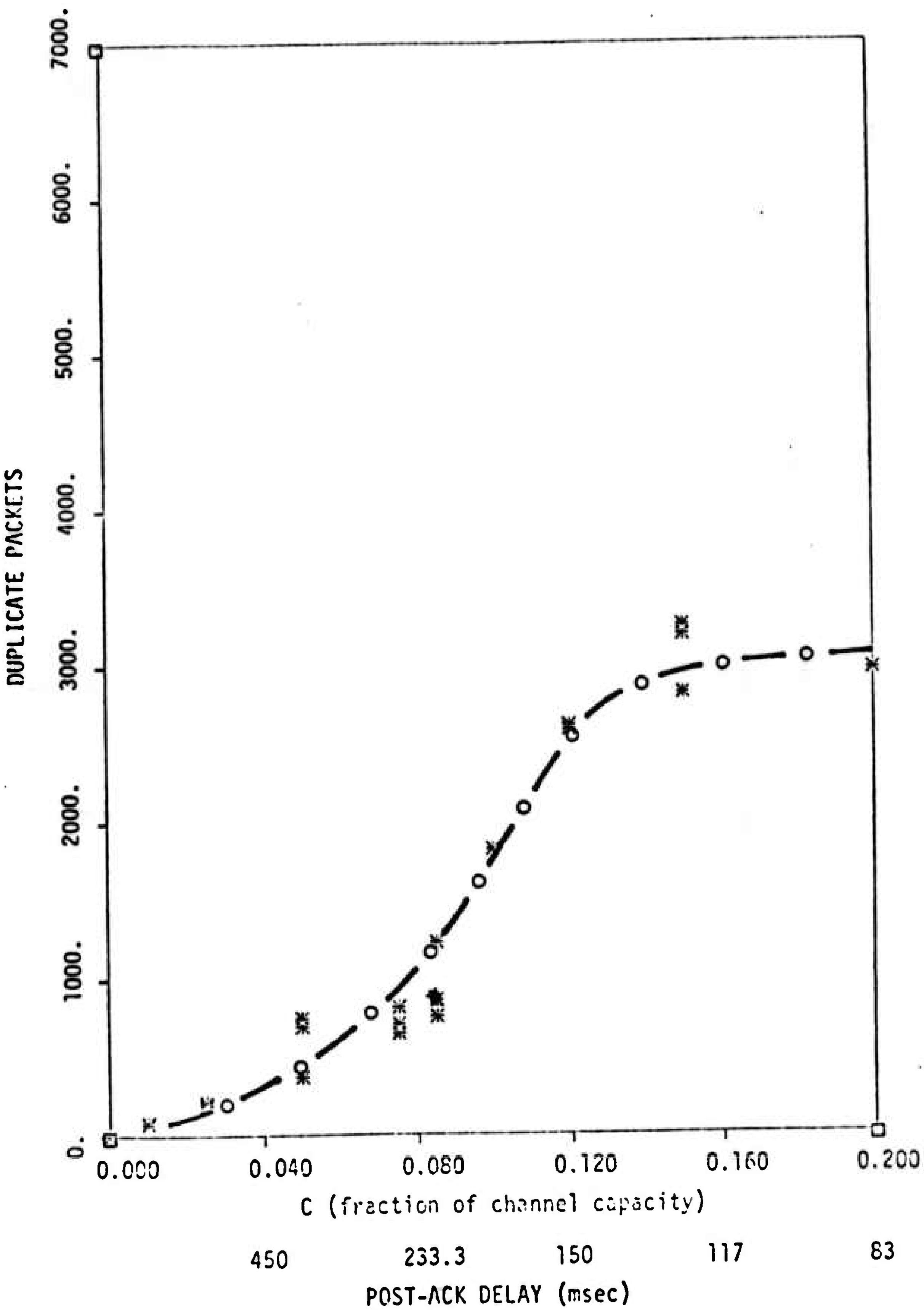

AVERAGE ROUND-TRIP DELAY (msec)
0.
100.
200.
300.
400.
500.
0.00
0.40
0.80
1.20
1.60
2.00
THROUGHPUT PER SOURCE TERMINAL (pkts/sec)

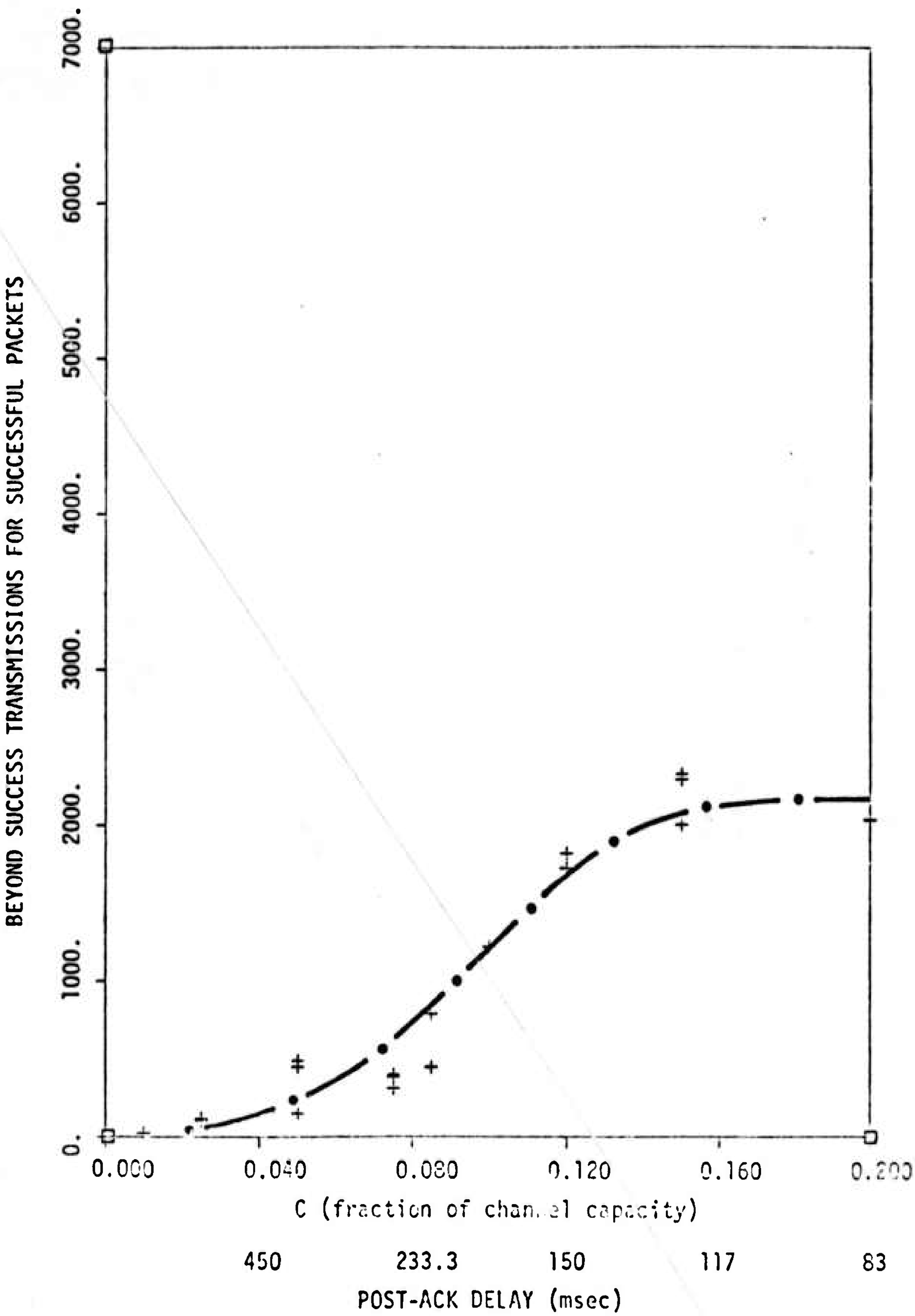
2. Behaviour of Repeater R

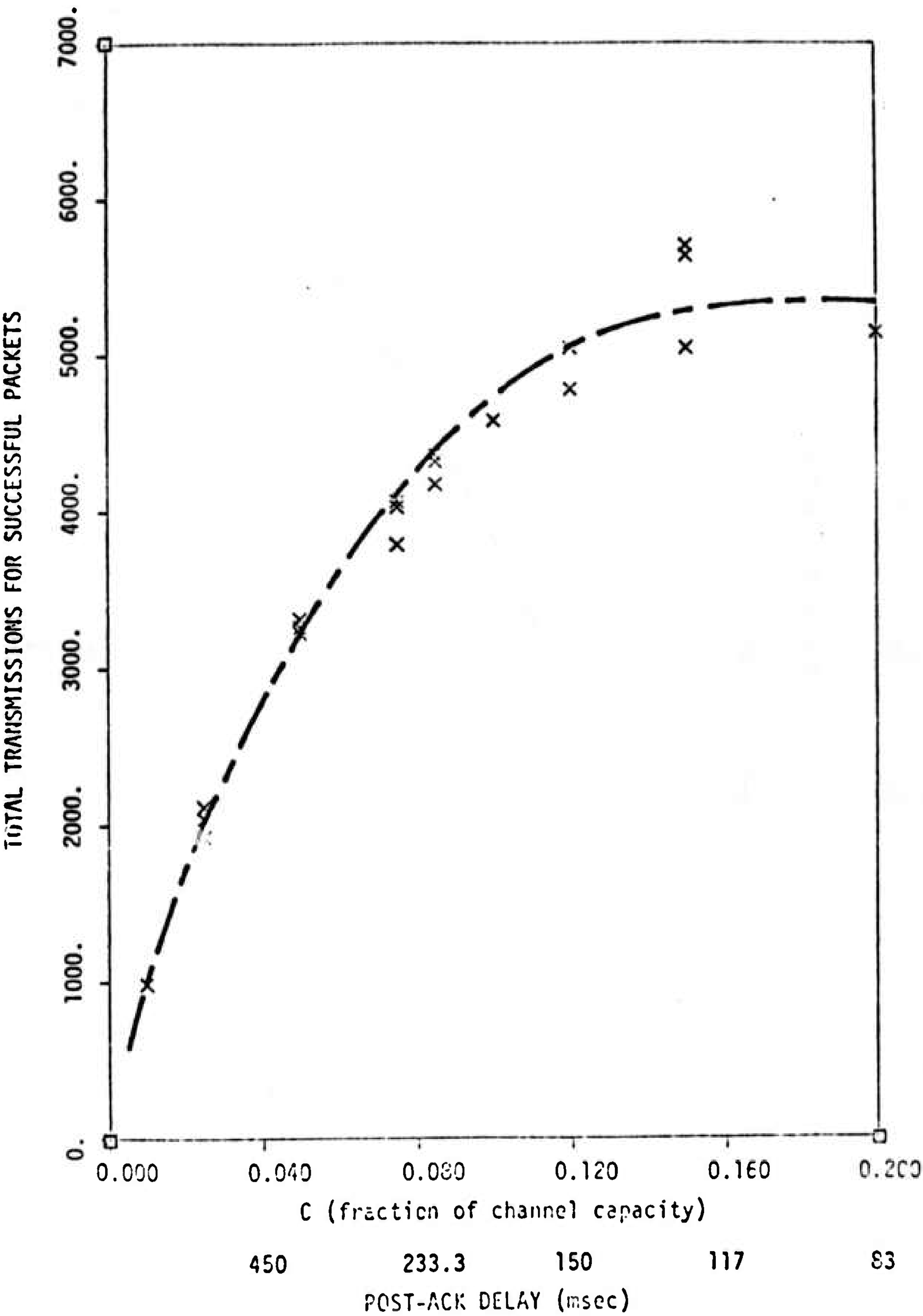

PACKETS SUCCESSFULLY TRANSMITTED TO NEXT HOP
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

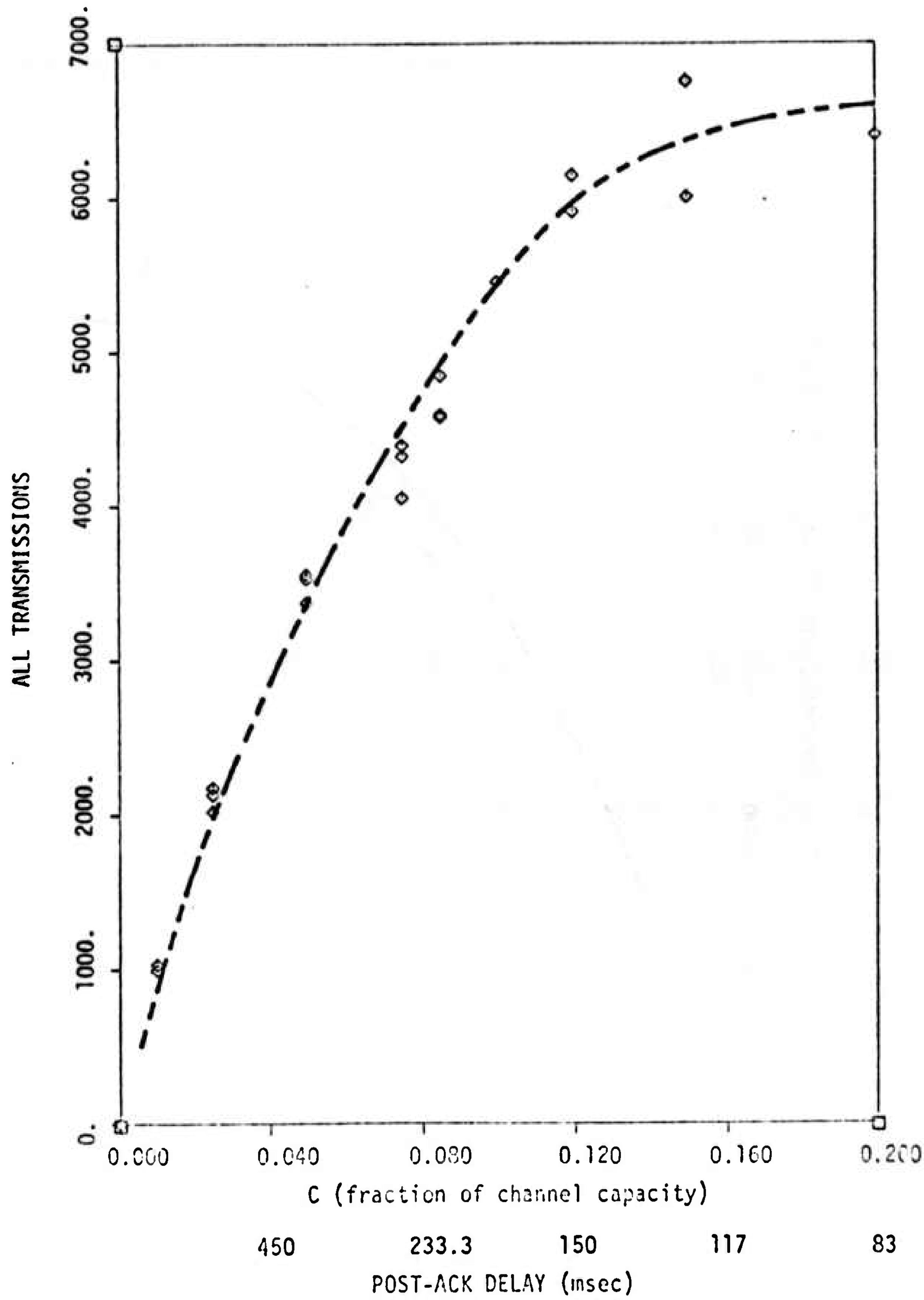


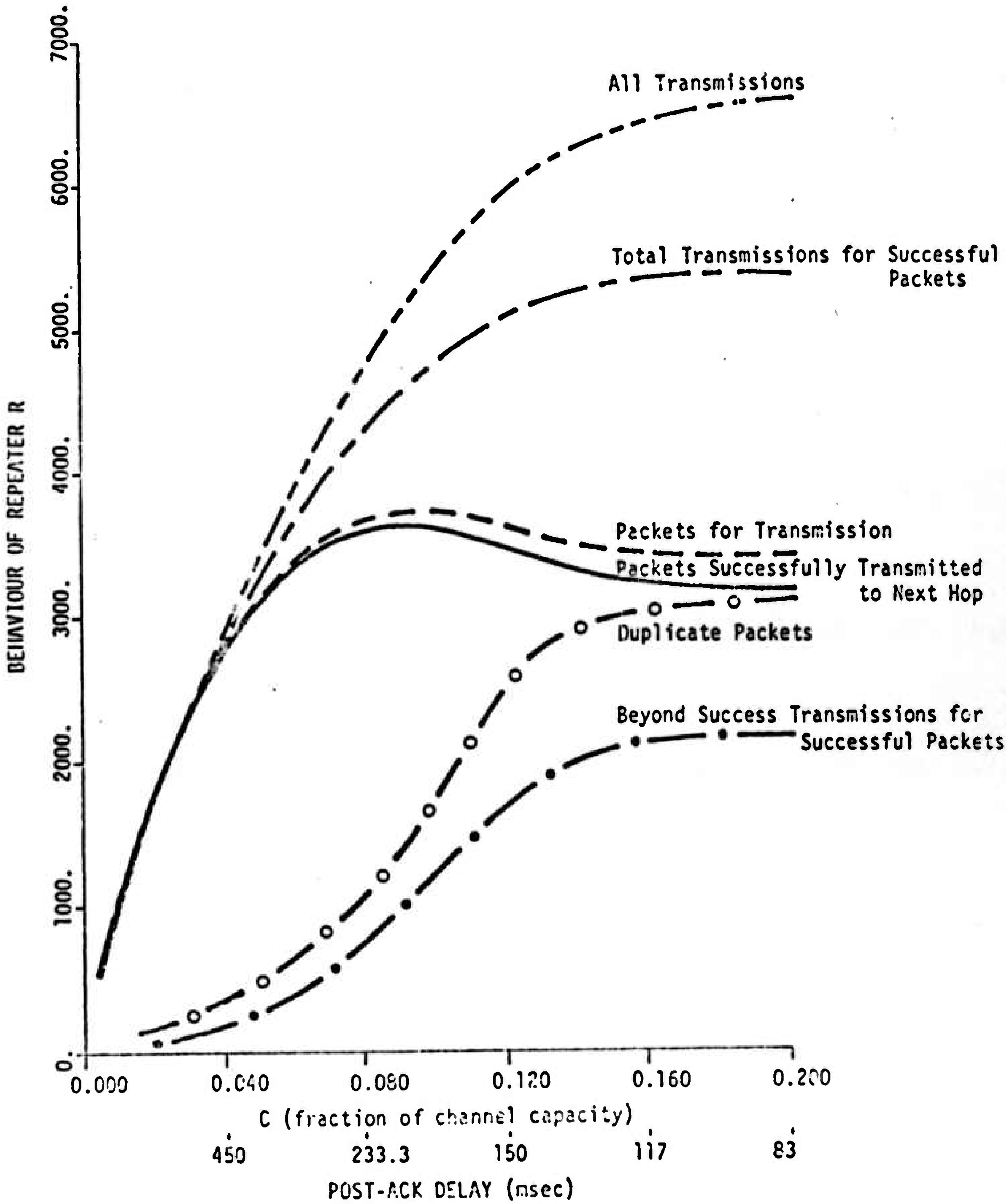
PACKETS FOR TRANSMISSION
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)


DUPLICATE PACKETS
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)


BEYOND SUCCESS TRANSMISSIONS FOR SUCCESSFUL PACKETS
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

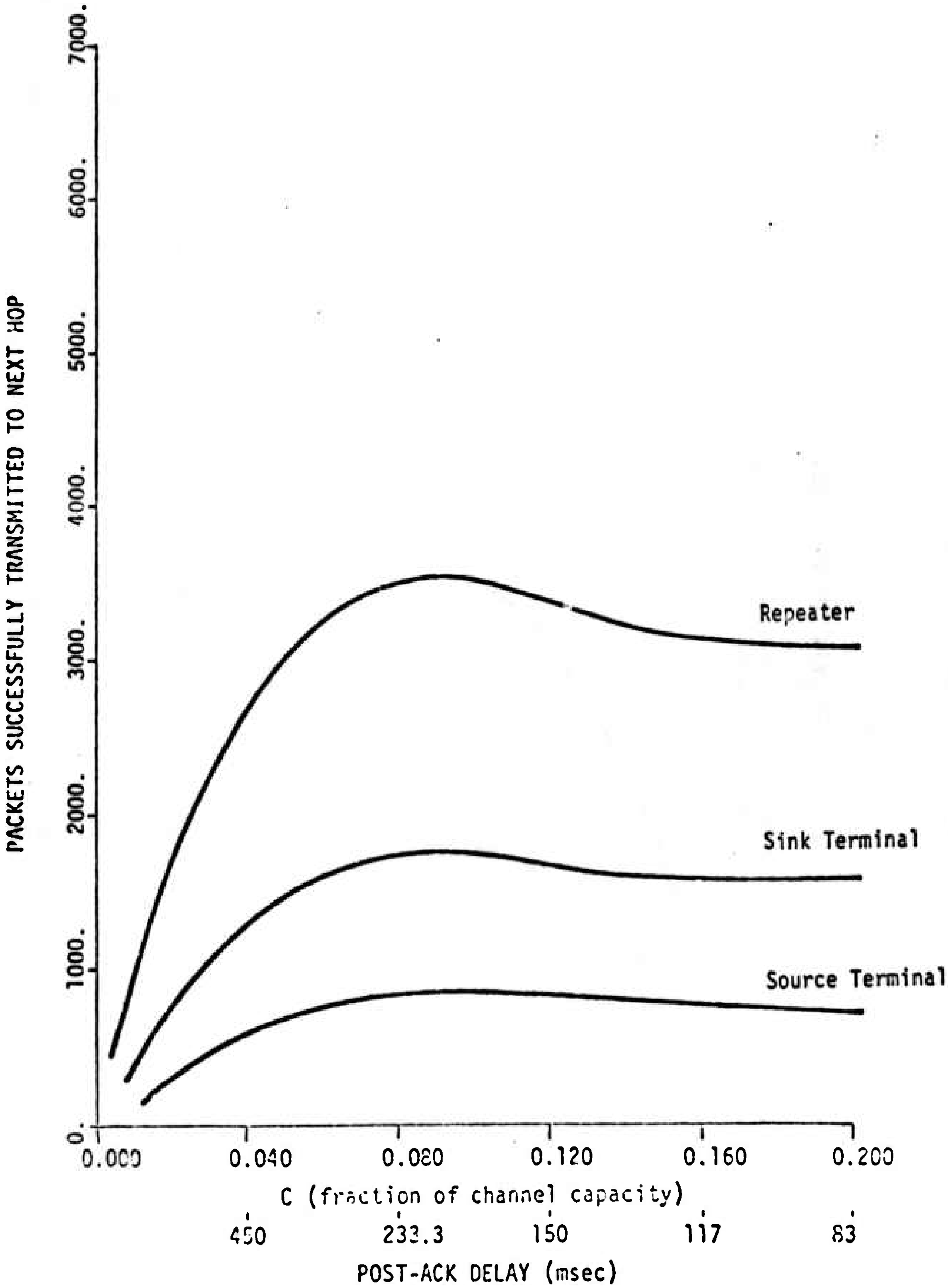

TOTAL TRANSMISSIONS FOR SUCCESSFUL PACKETS
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

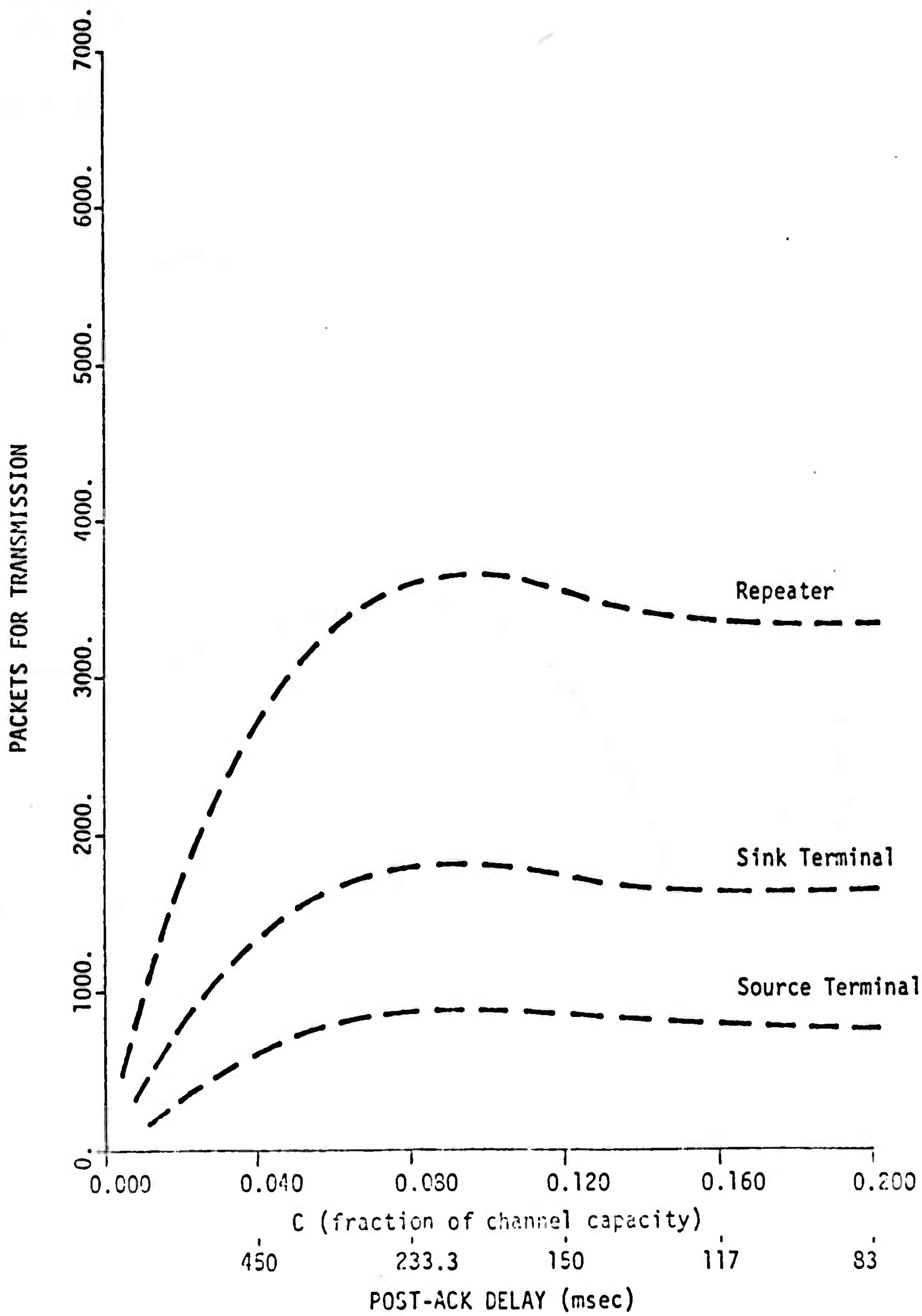

ALL TRANSMISSIONS
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

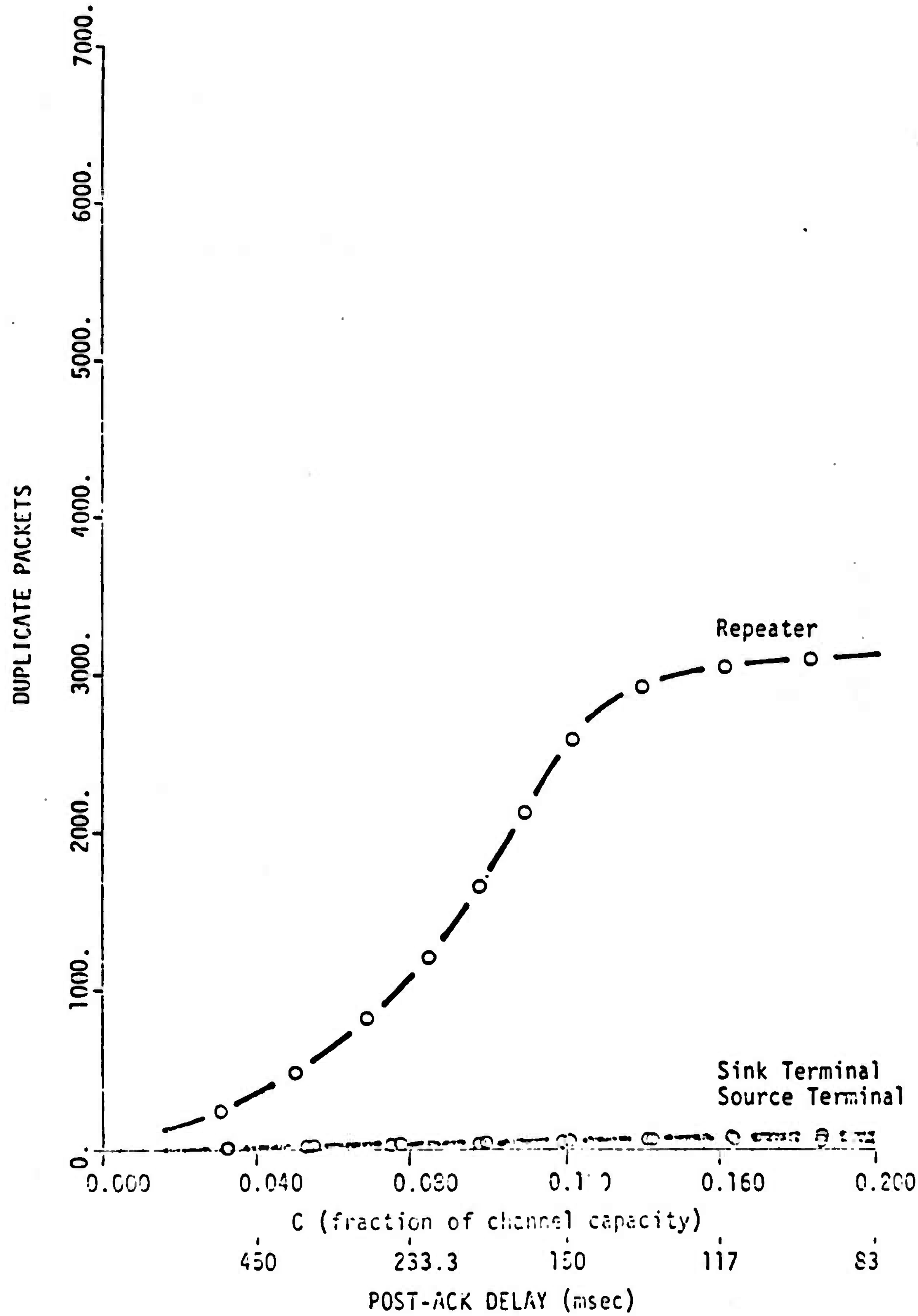


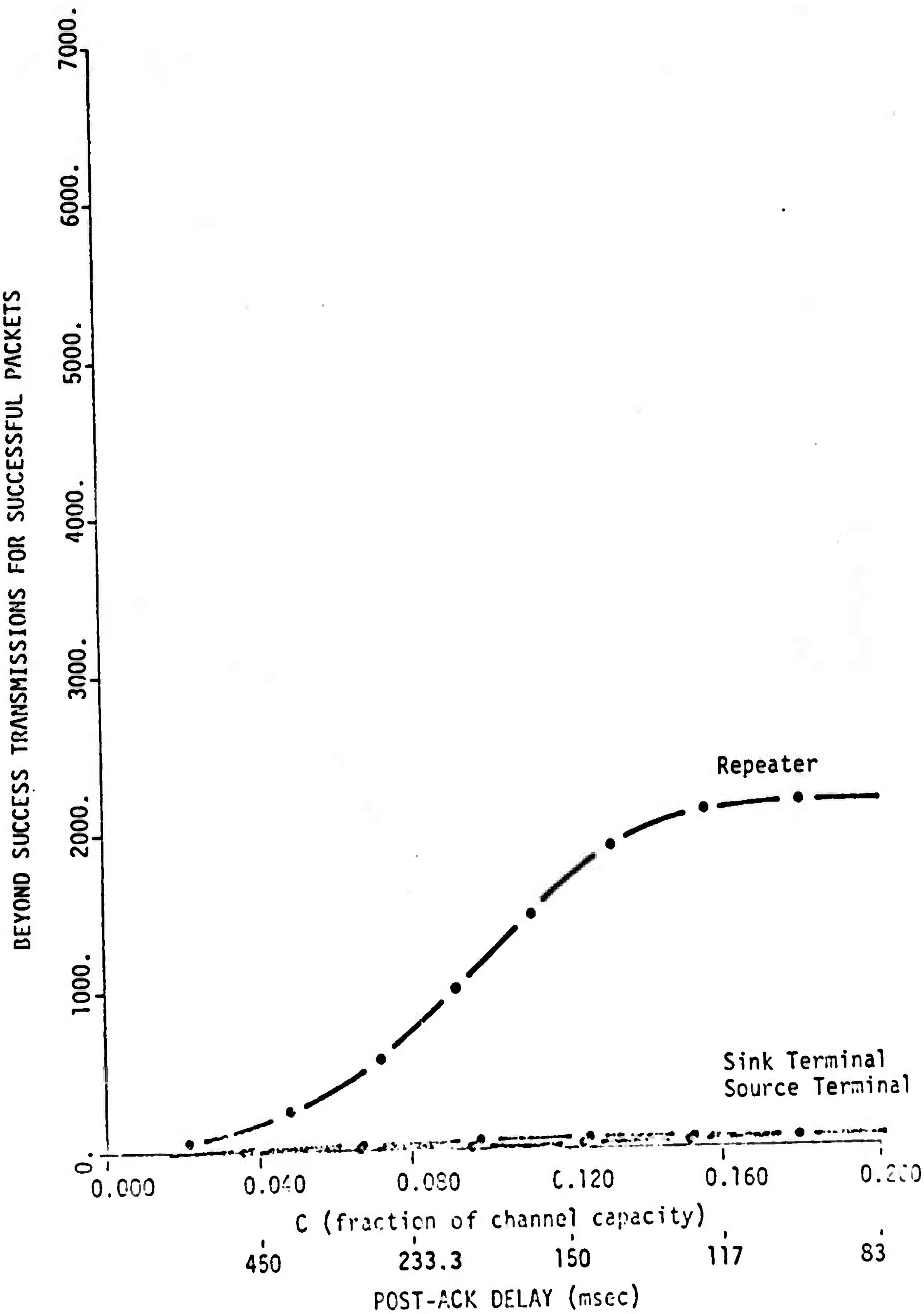
All Transmissions
Total Transmissions for Successful Packets
Packets for Transmission
Packets Successfully Transmitted to Next Hop
Duplicate Packets
Beyond Success Transmissions for Successful Packets
BEHAVIOUR OF REPEATER R
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

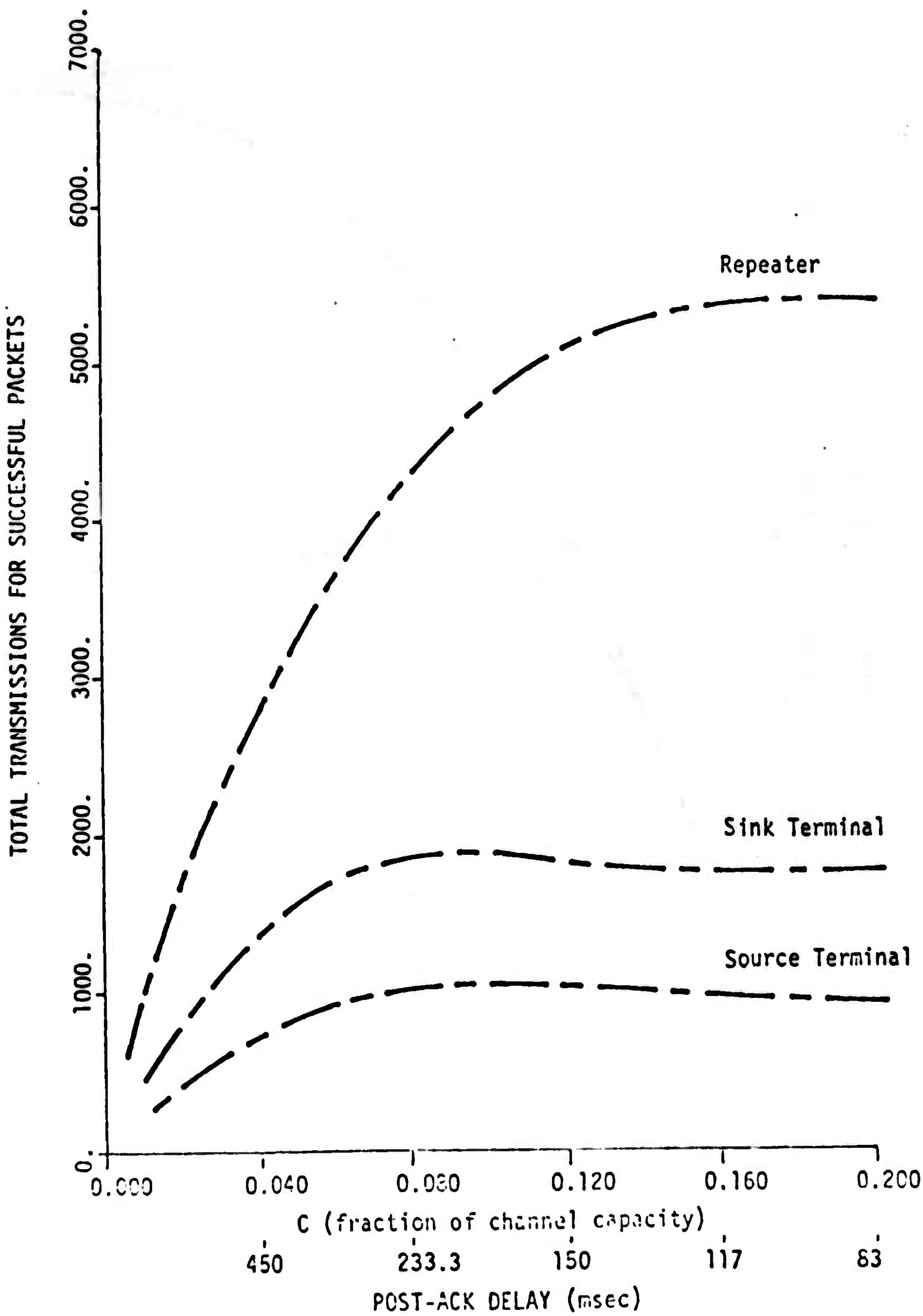
3. Comparison of the Source Terminal, The Sink Terminal, and Repeater R


PACKETS SUCCESSFULLY TRANSMITTED TO NEXT HOP
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

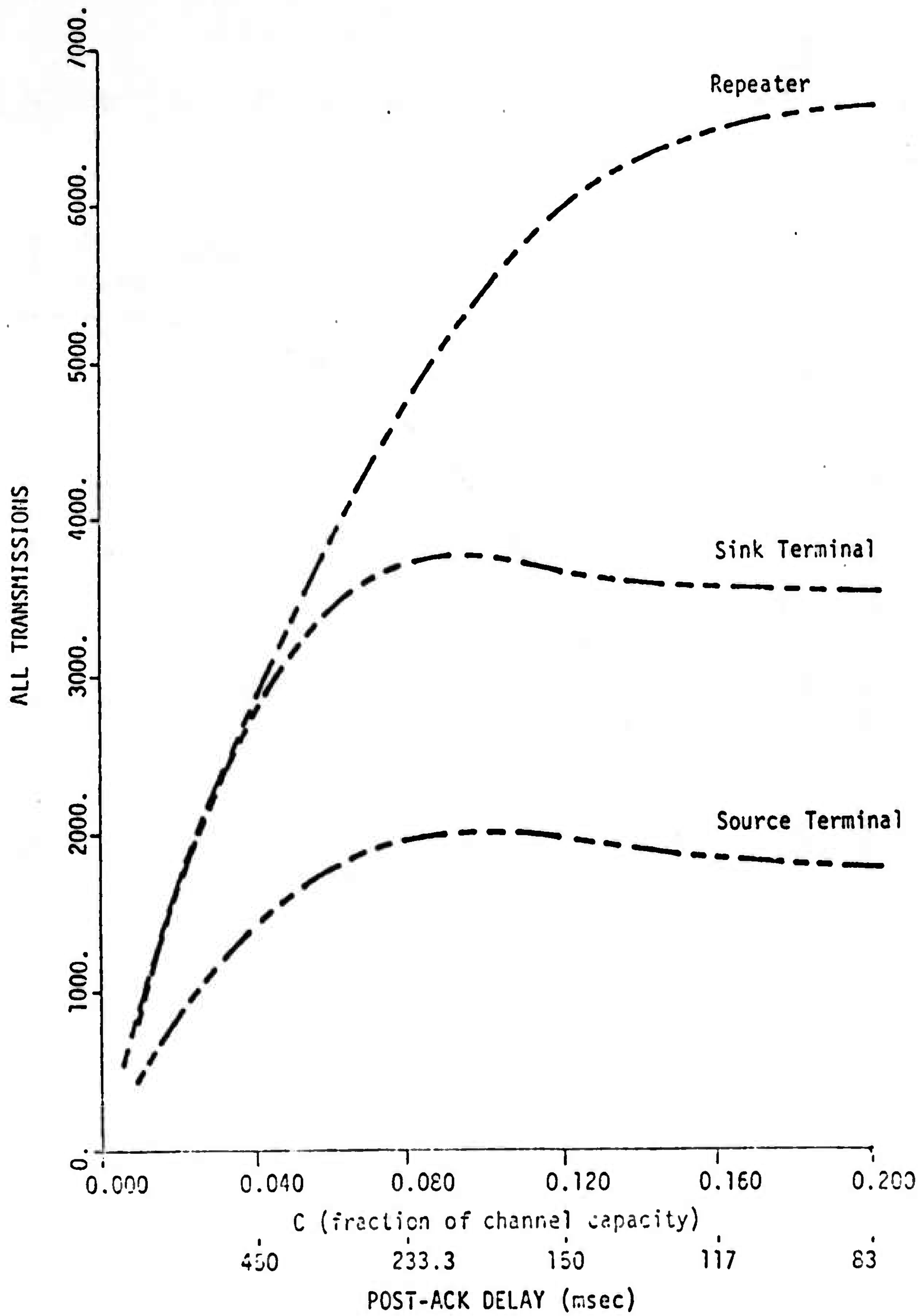

PACKETS FOR TRANSMISSION
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)


DUPLICATE PACKETS
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)


BEYOND SUCCESS TRANSMISSIONS FOR SUCCESSFUL PACKETS
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)



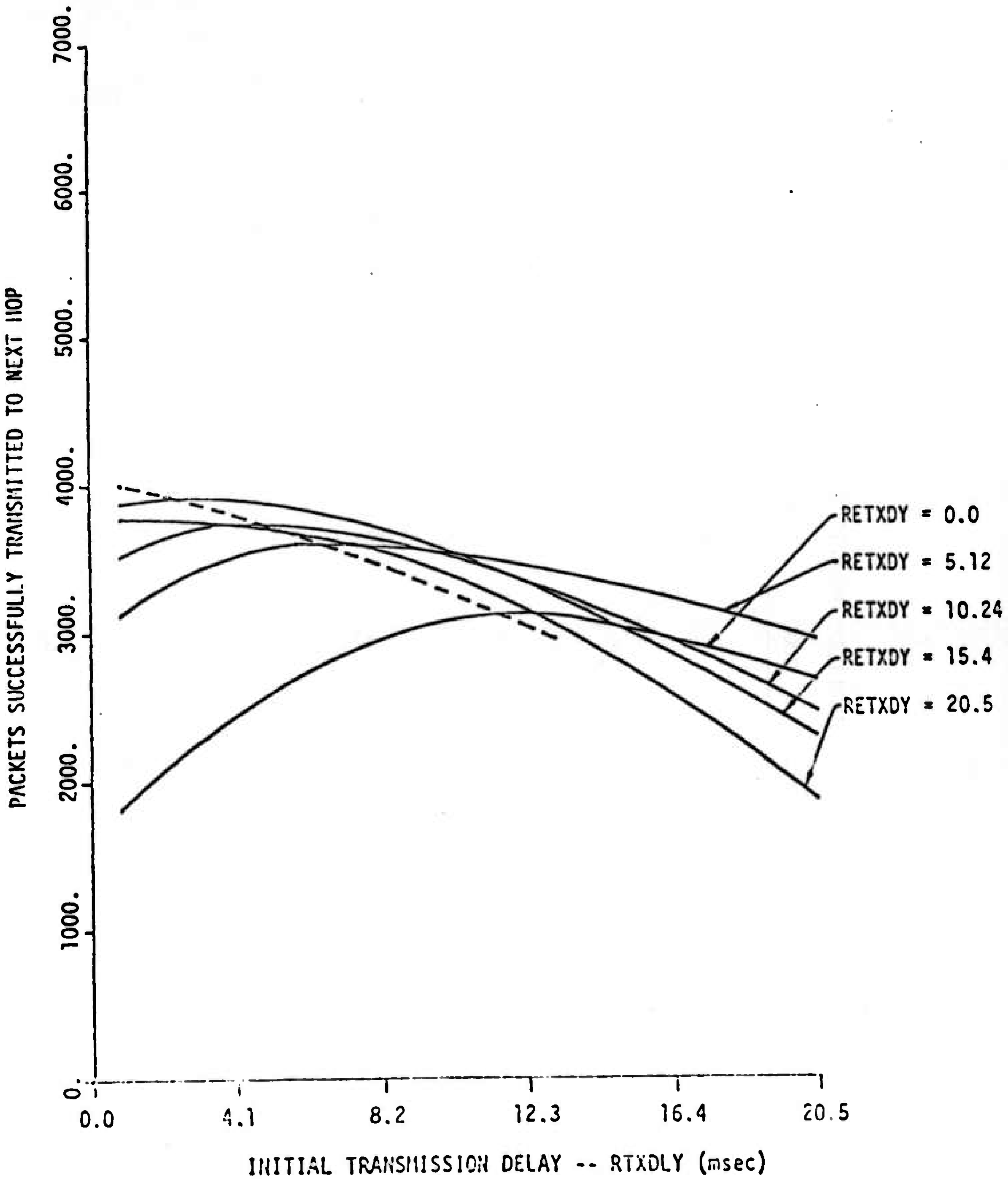
TOTAL TRANSMISSIONS FOR SUCCESSFUL PACKETS
0.
1000.
2000.
3000.
4000.
5000.
6000.
7000.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

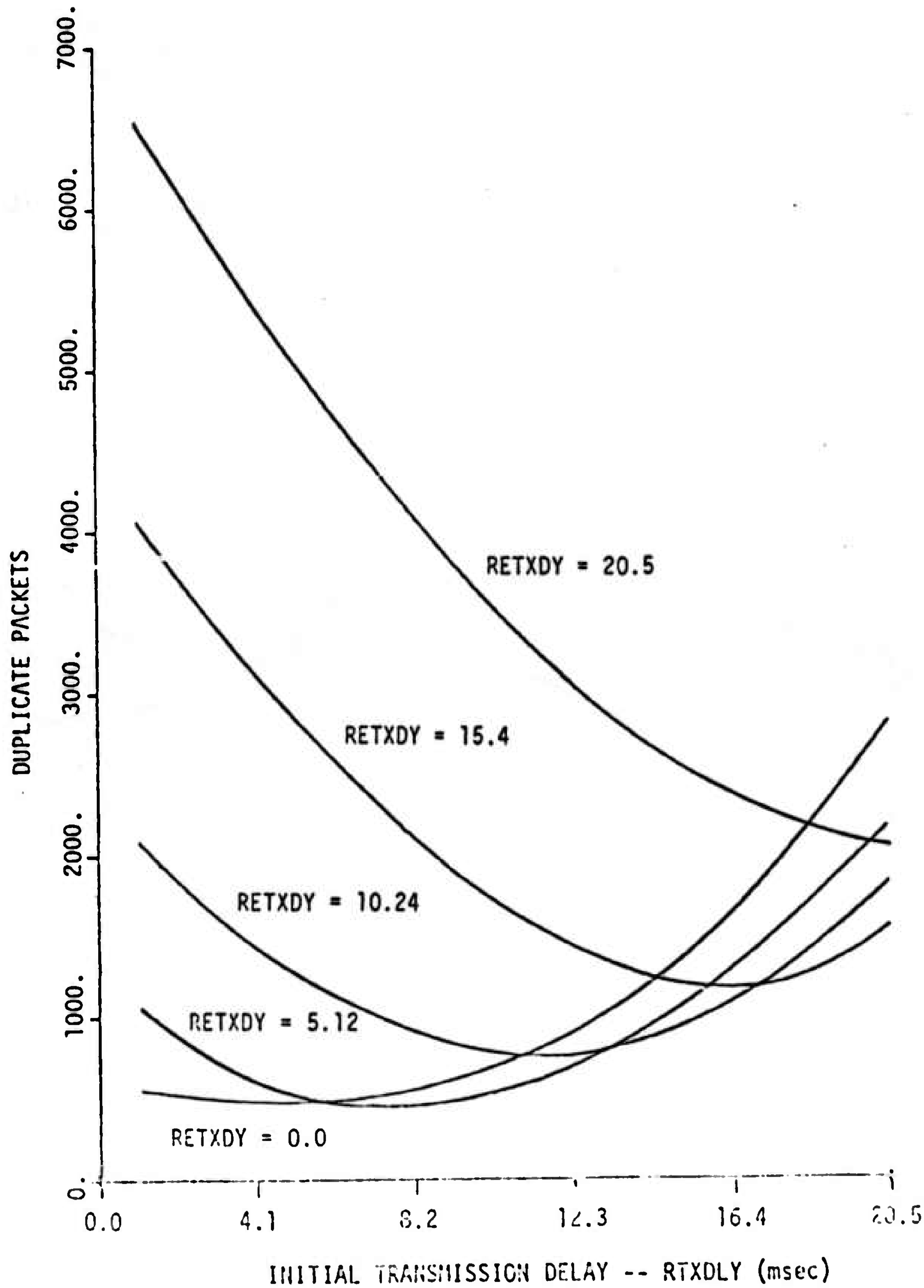


ALL TRANSMISSIONS
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
Repeater
Sink Terminal
Source Terminal
0.000
0.040
0.080
0.120
0.160
0.200
C (fraction of channel capacity)
450
233.3
150
117
83
POST-ACK DELAY (msec)

II. Behaviour of Repeater R under
Varying Transmission Delays with

- Input Rate C = 0.75 or Post-ACK Delay = 233.3 msec
- Packet Length = 17 words (total for transmission)


PACKETS SUCCESSFULLY TRANSMITTED TO NEXT HOP
7000.
6000.
5000.
4000.
3000.
2000.
1000.
0
RETXDY = 0.0
RETXDY = 5.12
RETXDY = 10.24
RETXDY = 15.4
RETXDY = 20.5
0.0
4.1
8.2
12.3
16.4
20.5
INITIAL TRANSMISSION DELAY -- RTXDLY (msec)



7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
DUPLICATE PACKETS
RETXDY = 20.5
RETXDY = 15.4
RETXDY = 10.24
RETXDY = 5.12
RETXDY = 0.0
0.0
4.1
8.2
12.3
16.4
20.5
INITIAL TRANSMISSION DELAY -- RTXDLY (msec)

Measurement Results from the Exporting Gateway Experiment

PRTN # 275

Stanley E. Lieberson
Zaw-Sing Su

UCLA

September, 1979

I. Introduction

A measurement experiment was conducted on the SRI PRNET testbed to study the possible advantage of physically separating the gateway from the station. The experiment was not completed due to component failures and testbed unavailablity. In this PRTN we provide the collected results to make them available to the interested parties

The model we used consists of a set of source PRUs and one or two destination PRUs, each PRU with an attached TIU. The experiment was run with a varying number of source PRUs and varying traffic load. The network is fully connected.

In the next section, we detail the experiment configurations. Section III contains a chart listing the collected results. A few plots for round-trip delay vs throughput and dropped packets vs throughput are included in the Appendix

II. Experiment Configurations

A number of experiments are detailed below. (The configurations and motivations for those experiemnts that could not be run will not be discussed here.) According to their configuration, we group the runs into four series. During these measurement runs, PRUs send their cumulative statistics (CumStats) to the measurement file every 2 minutes; TIUs sent their CumStats to the measurement file every three minutes (different rates were chosen to avoid swamping the net with simultaneous measurement packets). Each of the experiment runs was of seven minutes duration; thus allowing for four PRU CumStats and two TIU CumStats to be received from each such device during the course of each experiment run, and resulting in three PRU CumStat intervals of two minutes each, and one TIU CumStat interval of three minutes, for each such device in each run.

Thoughout this study, all runs use SPF protocol, CAP 4.7 FIFO transmission scheme, and all runs have zero-text-word packets.

Data collected from each PRU included:

- # of rx's and # of tx's over 1822 interface;
- # of radio tx's of forwarded packets;
- # of radio tx's of active ack's;
- # of correct radio rx's for intended packets;
- # of duplicate pkts rx'd;
- # of received pkts dropped for lack of hop ack;
- # of received packets discarded for lack of buffer space;
- # of alternately routed packets received.

Data collected from each TIU included:

throughput: # of packets ack'ed ETE;

mean round-trip delay for ETE ack'ed packets.

A schematic below lists the runs and the traffic input rate for each run. A symbolic configuration is shown after each series number. (The series numbers indicate the number of source and destination PRUs, e.g. "1-1".) Within each series, we label each run with a letter. A traffic source (or destination) is specified in conjunction with the TIU to which it is connected. The second traffic source of TIU#1, for example, is specified as "T1-2". The packet generation facility on the testbed generates feedback-dependent traffic. For each traffic-source / traffic-sink pair we indicate the packet generation rate by the "post-ack" delay in TIU clock ticks (60ths of a second). The "post-ack" delay specifies the interval between reception of the ETE ack of the previous packet, and the generation of a new packet. (Thus, only one packet from a given traffic source may be simultaneous outstanding in the network.)

Series 1-1 (18 runs): One source PRU and one destination PRU.

[TIU#1]-[PRU] [PRU]-[TIU#4]

	A	B	C	D	E	F
T1-1 -> T4-1	0	1	2	4	8	16

	G	H	I	J	K
T1-1 -> T4-1	0	1	2	4	8
T1-2 -> T4-2	0	1	2	4	8

	L	P	Q
T1-1 -> T4-1	0	4	8
T1-2 -> T4-2	0	4	8
T1-3 -> T4-3	0	4	8

	R	S	T	U
T1-1 -> T4-1	0	2	4	8
T1-2 -> T4-2	0	2	4	8
T1-3 -> T4-3	0	2	4	8
T1-4 -> T4-4	0	2	4	8

Series 2-1 (4 runs): Two source PRUs and one destination PRU.

```
[TIU#1]-[PRU]  [PRU]-[TIU#4]
[TIU#2]-[PRU]
```

	A	B	C	D
T1-1 -> T4-1	1	2	4	16
T2-1 -> T4-2	1	2	4	16

==

Series 3-1 (6 runs): Three source PRUs and one destination PRU.

```
[TIU#1]-[PRU]
[TIU#2]-[PRU]  [PRU]-[TIU#4]
[TIU#3]-[PRU]
```

	A	B	D	F
T1-1 -> T4-1	1	2	8	32
T2-1 -> T4-2	1	2	8	32
T3-1 -> T4-2	1	2	8	32

	G	H
T1-1 -> T4-1	4	8
T1-2 -> T4-2	4	8
T2-1 -> T4-3	4	8
T3-1 -> T4-4	4	8

Series 2-2 (9 runs): Two source PRUs and two destination PRUs.

[TIU#1]-[PRU] [PRU]-[TIU#4]

[TIU#2]-[PRU] [PRU]-[TIU#5]

	A	B	C	D
T1-1 -> T4-1	1	2	4	8
T2-1 -> T5-1	1	2	4	8

	E	F	G	I	J
T1-1 -> T4-1	1	8	16	8	4
T1-2 -> T5-1	1	8	16	32	64
T2-1 -> T4-2	1	8	16	8	4
T2-2 -> T5-2	1	8	16	32	64

III. Collected Results

In the Appendix we have compiled much of the data that resulted from the above runs. For each run we list the run number, and for each PRU in that run we list the following, in terms of packets/second:

(1) receptions over the wire from the attached TIU;

(2) transmissions over the wire to the attached TIU;

(3) radio receptions that were correctly heard, and intended;

(4) transmissions of active acknowledgements;

(5) transmissions of all other packets (i.e. forwarded packets);

(6) reception of duplicate packets (packets already successfully received);

(7) a listing of the raw count of the number of:

(i) alternately-routed packets received;

(ii) packets discarded for lack of buffer space;

(iii) packets dropped for lack of an acknowledgement.

For each PRU that had an attached TIU acting as a traffic source, we give:

(1) the TIU traffic source number (corresponds with Section II);

(2) the mean round-trip delay, in milliseconds;

(3) the throughput, in packets per second.

We have also made a number of plots. Due to the feedback-dependent nature of the input traffic, the intended input rate need not correspond to the achieved throughput. Since no ETE retransmissions were observed, we use the achieved throughput (number of ETE acknowledged packets) as the standard reference (the horizontal axis) for our plots.

For series 1-1 we have plotted round-trip (RT) delay vs throughput (Figure 1), and the total number of duplicate packets received by all PRUs vs throughput (Figure 2).

Figures 3 and 4 show similar plots for series 2-1 and series 3-1.

In Figure 5 we plot the RT delay vs throughput for the single-traffic-source-per-TIU portion of series 2-2 (i.e. runs A, B, C, and D), and also the two-traffic-sources-per-TIU portion, in which traffic is equally split between the two destinations. Figure 6 shows the associated curve for duplicate packets. Figure 7 shows curves, to be discussed in detail below, that are transformations of the curves in Figure 1.

IV. Discussion

Looking at Figure 1, we see that increasing the traffic has no effect upon the RT delay in runs A-F, indicating the absence of congestion in the network. Indeed, this is to be expected, since these runs consist of but a single traffic source, and thus only one "real" packet would be outstanding in the network at any time. It may be instructive to apportion the approximately 103 ms of RT delay among the various subsystems. Briefly, a packet in this one-hop network undergoes the following: it is

1) transmitted across the wire channel (1822 interface) at 50 KBPS from the TIU to the PRU (11+1 words for header and text);
2) processed by the PRU and placed on the (empty) radio tx queue;
3) transmitted at 100 KBPS (3+11+1+2 words for preamble, header, text, and checksum);
4) received and processed by the destination PRU; this processing results in the packet being placed on the (empty) "active ack" radio tx queue;
5) header-modified packet (3+11+0+2 words) tx'd as the active ack;
6) packet removed from active ack tx queue and processed again, and placed on the 1822 tx queue (to the destination TIU);
7) tx'd over the 1822 to the destination TIU (11+1 words);
8) processed by the destination TIU, who generates an ETE ack;
9) ETE ack tx'd over the 1822 to the PRU (11+0 words);
10) packet processed by the PRU and placed on the radio tx

queue;

11) packet tx'd at 100 KBPS (3+11+0+2 words);

12) packet rx'd at next PRU, processed, and placed on the (empty) active ack queue;

13) active ack tx'd at 100 KBPS (3+11+0+2 words);

14) packet removed from active ack queue, processed, and placed on the 1822 tx queue;

15) packet tx'd over the 1822 channel. Finished.

Based upon our oscilliscope measurements for PRU timings, and information received from SRI for TIU timings, we find that the above steps can be expected to require the following times (in msec's):

1) 1822 tx	3.84
2) proc.	10.
3) radio tx	2.72
4) proc.	10.
5) ack tx	2.56
6) proc.	10.
7) 1822 tx	3.84
8) TIU proc.	7.2
9) 1822 tx	3.52
10) proc.	10.
11) radio tx	2.56
12) proc.	10.
13) ack tx	2.56
14) proc.	10.
15) 1822 tx	3.52
Total:	~ 92 msec

This is about 10 ms shy of the measured average RT time, which is calculated using TIU clock tics (16.7 msec/tic). (Actually, it is about 18 msec shy, since the real RT time would be, on the average, half a clock tic longer than that recorded by the TIU.) One additional source of delay not mentioned above is the PRU radio hardware channel access delay, which delays each transmission of a packet from 0 ms to approximately 6 ms (zero, one, two, or three steps of up to about 2 ms each). The four transmissions a packet undergoes would thus account for perhaps 12 of the 18 outstanding msec's.

This accounting may illustrate two points:

(1) even in a simple, one-hop, conflict-free system, a lot of work is performed to deliver and acknowledge one packet; and

(2) two-thirds of the delay results from PRU packet processing. This speaks to the importance of hardware speed, software efficiency, and protocol/header simplicity.

At a first look, the throughput (S) vs delay (D) relationship for Series 1-1 beyond runs A-F appears unfamiliar. The throughput continues to increase instead of retreating at higher input rates. The increase in RT delay, rather than accelerating, levels off. Most puzzling of all is the decrease in duplicate packets. An explanation which may be offered for this result has much to do with the effect of input feedback-dependency.

Due to the input feedback-dependency, the throughput becomes dependent upon the delay. The greater the RT delay, the longer the interval between the generation of packets, and thus the smaller the throughput. More precisely, since no ETE retransmissions is observed, the throughput achieved should be inversely proportional to the RT delay: $S \sim 1/D$. The curve in Figure 1 deviates from $S = 1/D$ for two reasons: (i) the delay plotted is RT delay, and does not include the experimenter-imposed inter-generation delay between receipt of the ETE acknowledgement and the generation of the next packet (the post-ack-delay, PAD); and (ii) the throughput, S, is the total network throughput, rather than the per-traffic-source throughput (the feedback-dependency exists on a per-source bas' :).

Figure 7 shows the adjustments for these two factors. Shown are the original D vs S curves for Series 1-1. The curves with their PADs included are superimposed. And finally, these curves, with the throughput divided by the number of traffic sources, are plotted. The resulting curves conform almost perfectly to the curve S = 1/D.

This explanation for the effects of throughput dependency upon RT delay is not sufficient to answer the questions we have raised. In particular, what caused the number of duplicate packets to drop, and the throughput-delay curve to flatten, at the high throughput end?

It is known that, at the same traffic level, a stream of packets may move swiftly through a network, or may move slowly and suffer from repeated retransmissions and the generation of many duplicates. While the appropriate conditions for unencumbered traffic are not known for arbitrary configurations, the effects of traffic that is in some sense synchronous, and traffic in the same configuration that is asynchronous, have been noted. The conditions for synchronization are more likely to be sustained when the external environments remain relatively constant. This leads to our explanation of a synchronization effect imposed by the PRU flow control parameter TRMIDY. It is noticed that a possible result of synchronization, a flattened throughput-delay curve and a drop in duplicate packets, occurred beginning around 12 packets/second of throughput. An apparent limit on throughput in these configurations results from the def. ılt setting of TRMIDY at 160 PRU clock ticks (about 82 msec). This parameter enforces a minimum time between the enabling of reception on the PRU's 1822 wire channel, the "station/terminal" channel. For the configurations studied (i.e. a "Ft Bragg" configuration wherein all user traffic is destined to a station/gateway PRU), all traffic must pass through that PRU and across its station/terminal 1822 channel. Return traffic must enter into the PRU over the same channel. It is at this point that the TRMIDY parameter can create a bottleneck: with its (default) value of 81.92 msec required between arriving packets on the wire channel, a maximum flow of 12.2 pps into the PRU can be accommodated, which is the same as 12.2 ETE acknowledged packets per second, acknowledged packets being the definition of throughput used herein. Because the traffic generation process is feedback-dependent, the maximum possible actual input rate (summed over all traffic generators) is thereby limited to 12.2 pps. At that operating point the interarrival time at the gateway becomes constantly equivalent to the TRMIDY setting, and the input across both wire channels (at the originating and destination TIUs' becomes very regular. Thus, we speculate that as the traffic input rate increased in Series 1-1, and as the duplicates and retransmissions increased, a point of synchrony was reached in which transmissions and/or PRU processings were no longer random or "out of synch," resulting in a smoother flow of traffic and subsequently a reduction in the generation of duplicates.

In the absence of more information on the timing of the detailed functions of the PRU, we are not able to explain how the interval synchronization occurs. While knowledge of the factors that contribute to an in-synch flow of traffic would be useful, it is likely that the varying nature of input traffic would limit the application of such knowledge.

One major unexplained point regards the observation of duplicate packets created in runs for single traffic source configurations (runs A-F of Series 1-1). The circumstances that lead to these duplicate packets is unclear; this is a one-hop configuration with only one traffic source and thus only one packet outstanding at a time. The timing sequence presented earlier does not appear to warrant retransmissions due to time-outs. Yet in run B, for example, during the six minute CumStat collection interval, almost 500 duplicate packets were received by the two PRUs in the course of some 3000 round-trip packets (i.e. the sending of a packet and the receipt of its ETE acknowledgement), during which there were some 3200 packet transmissions (not including transmissions of active ack's, which should not create duplicates) at the source PRU, and 3350 at the destination PRU. This means that about ten percent of packet transmissions were of duplicates (as distinct from necessary retransmissions). More precisely, it means that almost every retransmission of a packet was unnecessary and resulted in a duplicate. The question that arises is: how could an unnecessary (or even a necessary) retransmission occur in such an empty network?

Because the traffic is feedback-dependent, a new packet is generated only after its predecessor has been ETE acknowledged (in the case of run 1-1B, 17 ms after that previous packet's ETE ack has arrived at the source TIU). Thus, one would expect that a generated packet would enter an empty network, be successfully transmitted the one hop to the destination PRU on its first attempt, be successfully active-acknowledged about ten ms later (well before the 8.2 + 10.24 ms retransmission time-out, and well after however long may be required for the source PRU to reenable radio rx after having transmitted the packet). Obviously, radio channel contention should not be a factor. Nor should channel link quality be so poor as to require retransmissions, since the two radios were connected via coaxial cable. In any case, lost or unheard active acknowledgements could not have been the source of retransmissions, since the number of active acknowledgements is almost equal to the number of packets received over the 1822 (wire) channel, indicating that additional or repeated tx's of active ack's were not necessary.

In fact, the 500 duplicates are almost completely accounted for by the number of retransmissions (excess of the number of non-ack tx's over the number of packets rx'd from the attached TIU). This implies the late reception of an active ack as the event triggering the sending of retransmissions that would result in duplicates. But, as we have seen above, in an unloaded network it does not seem possible that the tx of the active ack would be delayed beyond the 8.2 + 10.24 ms retransmission time-out period. Thus, we seem to be left with no explanation for the duplicate packets.

(One source of packets, and thus of channel contention, that has not been mentioned is the station. Although the station does initiate the sending of control packets, its traffic is at such a low level that it does not likely account for the duplicates noted. Likewise, the measurement packets themselves, occurring at two or three minute intervals, also should not account for the duplicates.)

From the above discussion, we may note the following points:

(1) As noted on page 11, about 2/3 of the delay in one-hop configurations under unloaded conditions resulted from PRU processing. The factors that make up this processing delay (protocol complexity, header structure, software design) are thus of some significance to overall network performance.

(2) We have shown how the TRMIDY (input-control) parameter can create a bottleneck in a gateway-oriented network. In the network used for these measurements, the TRMIDY parameter combined with feedback-dependency to limit the movement of packets into the network (thereby sometimes speeding the flow of those that were allowed to enter). In a network without the one-packet-outstanding limitation, e.g. TCP, the bottleneck may result in ETE retransmissions due to timing-out at the source. This extra load on the network would tend to result in further deterioration of the network performance beyond that caused by flow-control (TRMIDY) at the gateway PRU. Thus, the value of any PRU's TRMIDY parameter needs to be selected carefully so as to avoid the creation of unintended bottlenecks.

(3) The creation of duplicate packets, apparently resulting from transmissions beyond success, seems incompatible with a one-hop, single-source, one-packet-outstanding configuration. More details about the timings of activities within a PRU and the timings of interactions between PRUs are needed to explicate these results. The Pickup Packet measurement tool would provide such data.

(4) Feedback independent traffic sources are more desirable for future experiments.

Appendix:

Collected Results

		--------- r a t e s / s e c o n d -----------						---raw---	=== traffic source =		
Run	PRU	wire rx	wire tx	radio corr rx	act ack tx	non-ack tx	dups	alt-rt[A] disc[C] drop[D]	atta-ched TIU	mean delay	through put
1-1A	1	8.86	8.86	18.22	8.90	9.33	.46	A-1	1-1	107.7	8.67
A	4	8.74	8.74	18.04	8.78	9.27	.53				
1-1B	1	8.36	8.36	17.33	8.41	8.92	.55	A-1	1-1	103.3	8.19
B	4	8.23	8.23	17.59	8.27	9.30	1.07				
1-1C	1	7.44	7.45	15.39	7.46	7.92	.48	A-2	1-1	102.1	7.33
C	4	7.33	7.33	15.28	7.37	7.91	.58				
1-1D	1	5.99	5.99	13.41	6.05	6.27	.40		1-1	102.0	5.90
D	4	5.93	5.93	12.24	5.98	6.39	.33				
1-1E	1	4.30	4.30	8.94	4.36	4.59	.30		1-1	102.7	4.22
E	4	4.25	4.25	8.95	4.30	4.66	.39				
1-1F	1	2.76	2.76	5.69	2.80	2.89	.13		1-1	102.4	2.70
F	4	2.74	2.74	5.68	2.78	2.90	.15				
1-1G	1	10.00	10.00	24.18	11.87	14.13	4.33	A-10	1-1	198.9	4.84
									1-2	198.5	4.84
G	4	9.86	9.86	23.76	11.73	13.86	4.16	A-29			
1-1H	1	10.14	10.14	24.18	11.74	14.01	4.29	A-3	1-1	180.0	4.93
									1-2	180.1	4.93
H	4	9.91	9.91	23.77	11.70	13.82	4.04	A-8			
1-1I	1	10.11	10.11	23.78	11.27	13.62	4.03	A-2	1-1	164.7	4.91
									1-2	165.0	4.91
I	4	9.87	9.87	23.61	11.34	13.71	4.01	A-6			
1-1J	1	9.57	9.57	22.03	9.59	12.42	3.49	A-2	1-1	140.8	4.68
									1-2	141.3	4.68
J	4	9.43	9.43	21.64	10.23	12.17	2.77				
1-1K	1	8.37	8.37	17.68	8.46	9.26	.94		1-1	109.3	4.09
									1-2	108.4	4.09
K	4	8.23	8.23	17.43	8.33	9.15	.93	A-1			
1-1L	1	11.45	11.45	25.23	15.82	13.74	3.65	A-3	1-1	262.5	3.69
									1-2	261.7	3.70
									1-3	262.4	3.70
L	4	11.15	11.15	23.56	13.21	12.37	4.77	A-11			
1-1P	1	10.71	10.71	24.72	13.10	13.96	4.08	A-2	1-1	214.3	3.49
									1-2	213.8	3.49
									1-3	213.5	3.49
P	4	10.54	10.54	23.71	12.73	13.13	3.84	A-1			
1-1Q	1	9.98	9.97	23.26	11.21	13.24	3.83	A-2	1-1	169.9	3.23
									1-2	171.0	3.23
									1-3	170.4	3.23
Q	4	9.76	9.77	23.30	11.15	13.50	3.94	A-4			
1-1R	1	12.89	12.87	26.61	14.25	13.68	2.57	A-1	1-1	310.7	3.12
									1-2	311.9	3.12
									1-3	312.5	3.11
									1-4	310.9	3.11
R	4	12.59	12.59	25.80	14.64	13.17	1.62	A-2			

Run	PRU	wire rx	wire tx	radio corr rx	act ack tx	non-ack tx	dups	---raw--- alt-rt[A] disc[C] drop[D]	=== traffic source === atta-ched TIU	mean delay	through-put
		--------- r a t e s / s e c o n d					----------				
1-1S	1	11.88	11.86	25.09	13.14	13.17	4.28	A-6	1-1	303.0	2.91
									1-2	302.9	2.91
									1-3	302.8	2.91
									1-4	305.0	2.91
S	4	11.66	11.67	24.81	15.29	13.10	2.14	A-4			
1-1T	1	11.45	11.43	25.61	13.59	14.18	4.29	A-3	1-1	287.3	2.78
									1-2	287.1	2.78
									1-3	286.8	2.78
									1-4	286.2	2.78
T	4	11.21	11.19	23.88	13.43	12.63	2.91	A-2			
1-1U	1	11.17	11.15	25.49	13.48	14.28	4.19	A-4	1-1	230.6	2.70
									1-2	232.7	2.70
									1-3	231.3	2.69
									1-4	230.9	2.69
U	4	10.91	10.91	23.67	13.12	12.77	3.28	A-2			
2-1A	1	5.11	5.10	11.56	6.31	6.44	1.34	D-1 A-2	1-1	172.7	4.97
A	2	5.01	5.00	11.38	6.12	6.36	1.37	D-4 A-2	2-1	173.2	4.98
A	4	9.94	9.94	23.66	11.04	13.70	4.96	D-5 A-23			
2-1B	1	5.17	5.15	11.41	6.28	6.23	1.09		1-1	154.4	5.11
B	2	5.08	5.07	11.37	6.22	6.28	1.22	D-4 A-2	2-1	154.1	5.11
B	4	10.06	10.06	22.91	11.13	12.83	4.08	D-4 A-11			
2-1C	1	5.26	5.25	10.96	5.48	5.69	.43	D-2	1-1	122.1	5.10
C	2	5.17	5.16	11.09	5.40	5.82	.65	D-1 A-2 C-2	1-1	122.2	5.11
C	4	10.24	10.24	21.40	10.46	11.14	.99	D-2			
2-1D	1	2.73	2.73	5.72	2.75	2.98	.24	A-2	1-1	105.7	2.67
D	2	2.71	2.71	5.97	2.82	3.11	.38	A-3	2-1	106.1	2.67
D	4	5.36	5.36	11.52	5.39	6.15	.79				

		rates / second						raw	traffic source		
Run	PRU	wire rx	wire tx	radio corr rx	act ack tx	non-ack tx	dups	alt-rt[A] disc[C] drop[D]	attached TIU	mean delay	through-put
2-2A	1	6.49	6.49	13.29	6.64	6.79	.35		1-1	138.9	6.34
A	2	6.37	6.37	13.47	6.65	6.84	.60	A-15 D-1	2-1	138.7	6.36
A	4	6.33	6.33	13.51	6.72	6.41	.56	A-15 D-1			
A	5	6.37	6.36	13.09	6.53	3.56	.40	A-6			
2-2B	1	5.78	5.77	12.43	6.14	6.64	.96	A-2	1-1	140.3	5.60
B	2	5.69	5.69	12.54	6.11	6.61	1.08	A-20 D-2	2-1	138.8	5.63
B	4	5.59	5.59	12.67	6.28	6.71	1.16	A-15 D-2 C-1			
B	5	5.69	5.68	12.24	6.01	6.54	.96	A-6 D-1			
2-2C	1	3.53	3.53	7.70	3.75	3.97	.45	A-2	1-1	123.0	5.14
C	2	5.21	5.21	11.48	5.51	6.00	.92	A-7	2-1	124.0	5.14
C	4	5.18	5.18	10.74	5.32	5.56	.40	A-3			
C	5	5.20	5.19	10.83	5.36	5.62	.46	A-5			
2-2D	1	4.21	4.20	9.54	4.64	4.96	.75	A-24 D-2 C-5	1-1	107.4	4.11
D	2	4.15	4.15	9.15	4.39	4.76	.69	A-14 D-2	2-1	109.4	4.09
D	4	4.11	4.11	8.45	4.21	4.34	.24	A-3 D-1			
D	5	4.14	4.13	8.51	4.29	4.37	.25	A-6 D-1			
2-2E	1	5.99	5.99	14.49	8.56	8.29	2.99	A-53 D-2 C-1	1-1	313.2	2.97
									1-2	321.6	2.89
E	2	5.92	5.92	12.18	8.19	8.25	3.02	A-62	2-1	313.5	2.97
									2-2	320.4	2.92
E	4	5.94	5.94	14.36	8.14	8.41	3.23	A-39 D-1			
E	5	5.88	5.88	14.22	8.00	8.33	3.26	A-48			
2-2F	1	5.93	5.91	14.13	7.24	7.94	2.44	A-25 D-1 C-2	1-1	210.1	2.87
									1-2	208.5	2.83
F	2	5.85	5.84	13.74	6.89	7.87	2.46	A-21	2-1	206.1	2.88
									2-2	207.8	2.68
F	4	5.81	5.81	13.63	6.98	7.82	2.34	A-22			
F	5	5.84	5.82	13.60	6.89	7.77	2.26	A-8 D-1			
2-2G	1	4.96	4.96	11.32	5.47	6.17	1.38	A-15 C-2	1-1	142.3	2.42
									1-2	143.6	2.41
G	2	4.89	4.88	11.06	5.24	6.16	1.54	A-3	2-1	143.8	2.41
									2-2	142.2	2.41
G	4	4.86	4.86	11.01	5.20	6.15	1.34	A-6			
G	5	4.87	4.87	10.89	5.37	6.00	1.19	A-7			
2-2I	1	5.02	5.03	11.64	5.66	6.39	1.53	A-9 D-2 C-2	1-1	149.1	3.49
									1-2	170.6	1.42
I	2	4.98	4.98	11.00	5.45	6.01	1.16	A-3	2-1	147.8	3.50
									2-2	168.7	1.41
I	4	7.04	7.04	15.68	7.52	8.64	1.80	A-9 D-1			
I	5	2.87	2.87	6.54	3.24	3.66	.87	A-6 D-1			
2-2J	1	5.47	5.46	13.31	6.23	6.51	1.14	A-12 C-3	1-1	149.0	4.54
									1-2	191.7	0.79
J	2	5.41	5.41	11.73	5.96	6.31	1.03	A-7 D-1	2-1	149.0	4.54
									2-2	185.8	0.79
J	4	9.13	9.12	19.96	9.69	10.82	2.09	A-13			
J	5	1.64	1.63	3.75	1.97	2.10	.51				

		--------- r a t e s / s e c o n d -----------						---raw---	=== traffic source ===		
Run	PRU	wire rx	wire tx	radio corr rx	act ack tx	non-ack tx	dups	alt-rt[A] disc[C] drop[D]	atta-ched TIU	mean delay	through-put
3-1A	1	3.60	3.59	8.24	5.77	4.63	1.07	D-1 A-12	1-1	267.1	3.46
A	2	3.55	3.54	8.27	5.90	4.46	.99	D-1 A-18	2-1	266.6	3.46
A	3	3.55	3.55	8.04	5.55	4.48	.95	D-2 A-1 C-1	3-1	263.6	3.50
A	4	10.49	10.49	25.41	12.04	14.55	7.58	D-1 A-121 C-5			
3-1B	1	3.63	3.63	8.30	5.59	4.66	1.06	D-1 A-1	1-1	245.8	3.52
B	2	3.55	3.55	8.15	5.41	4.59	1.08	D-1 A-3	2-1	247.0	3.50
B	3	3.55	3.55	8.20	5.47	4.64	1.12		3-1	247.4	3.51
B	4	10.57	10.57	25.18	11.69	14.60	7.63	D-1 A-67			
3-1D	1	3.49	3.49	7.69	4.19	4.19	.71	A-4 D-1	1-1	157.9	3.38
D	2	3.43	3.43	7.54	4.09	4.10	.67	A-3	2-1	156.6	3.39
D	3	3.42	3.41	7.79	4.30	4.19	.78	A-4	3-1	156.8	3.38
D	4	10.13	10.13	23.57	11.18	13.24	4.20	A-27 D-1 C-2			
3-1F	1	1.60	1.60	3.44	1.66	1.84	.24		1-1	109.9	1.54
F	2	1.59	1.58	3.31	1.64	1.71	.12	A-2	2-1	110.3	1.55
F	3	1.59	1.59	3.60	1.72	1.83	.25	A-4	3-1	110.4	1.54
F	4	4.67	4.67	10.44	4.85	5.58	.92	A-4 C-1			
3-1G	1	5.52	5.51	12.36	8.41	6.74	1.29	A-10 D-2	1-1	304.8	2.64
									1-2	305.5	2.65
G	2	2.75	2.74	6.15	4.53	3.39	.66	A-13	2-1	297.5	2.71
G	3	2.75	2.75	6.37	4.80	3.42	.63	A-5 D-3 C-2	3-1	297.5	2.71
G	4	10.63	10.64	25.26	11.98	14.44	7.67	A-129 D-1			
3-1H	1	5.37	5.37	12.27	7.60	6.90	1.66	A-9 D-1	1-1	245.6	2.60
									1-2	245.8	2.60
H	2	2.65	2.66	6.11	4.09	3.45	.81	A-7	2-1	245.6	2.61
H	3	2.64	2.63	6.36	4.43	3.52	.86	A-6 D-2	3-1	242.0	2.62
H	4	10.38	10.38	24.79	11.73	14.14	6.97	A-96 D-1			

Figures

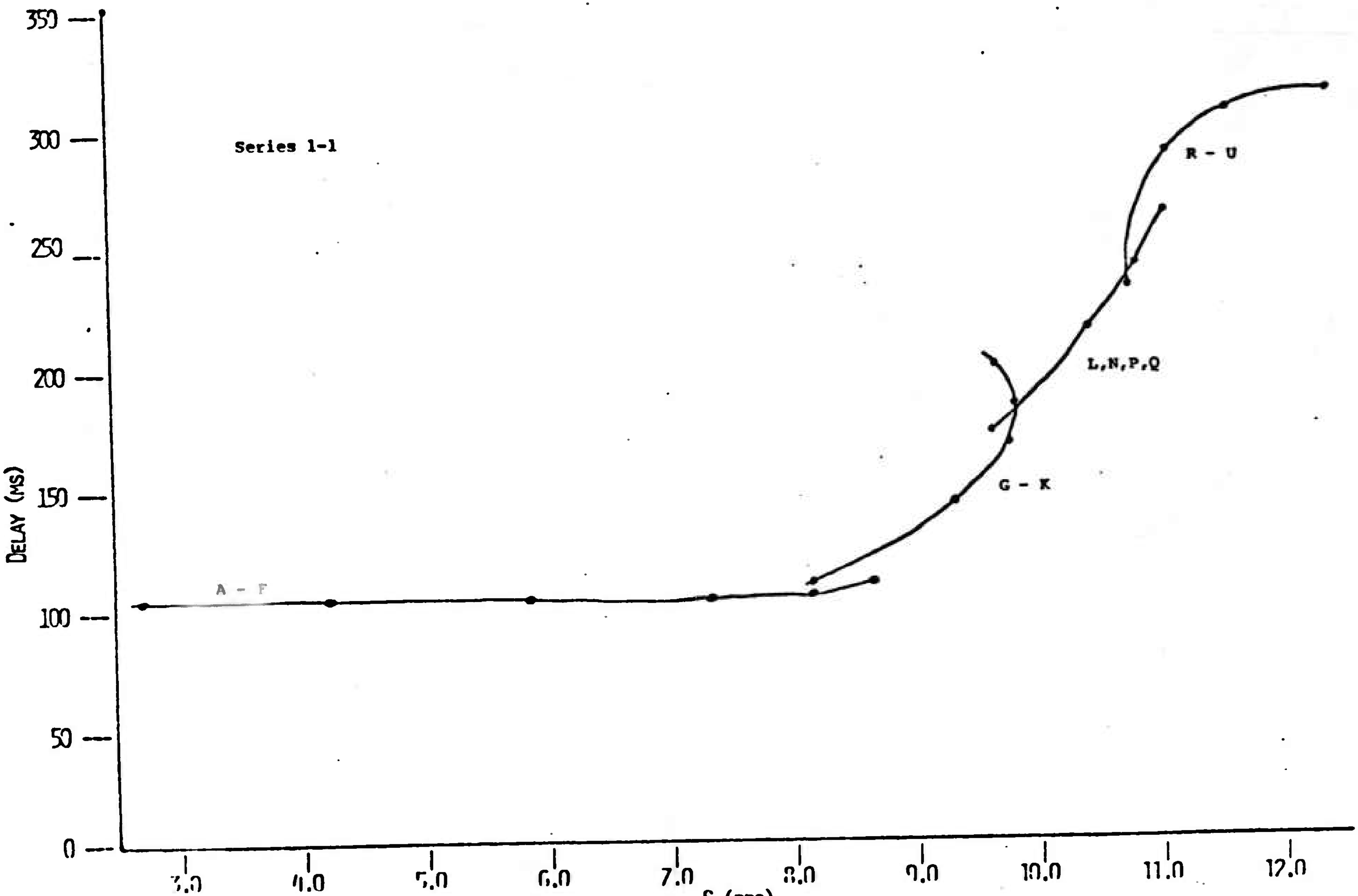
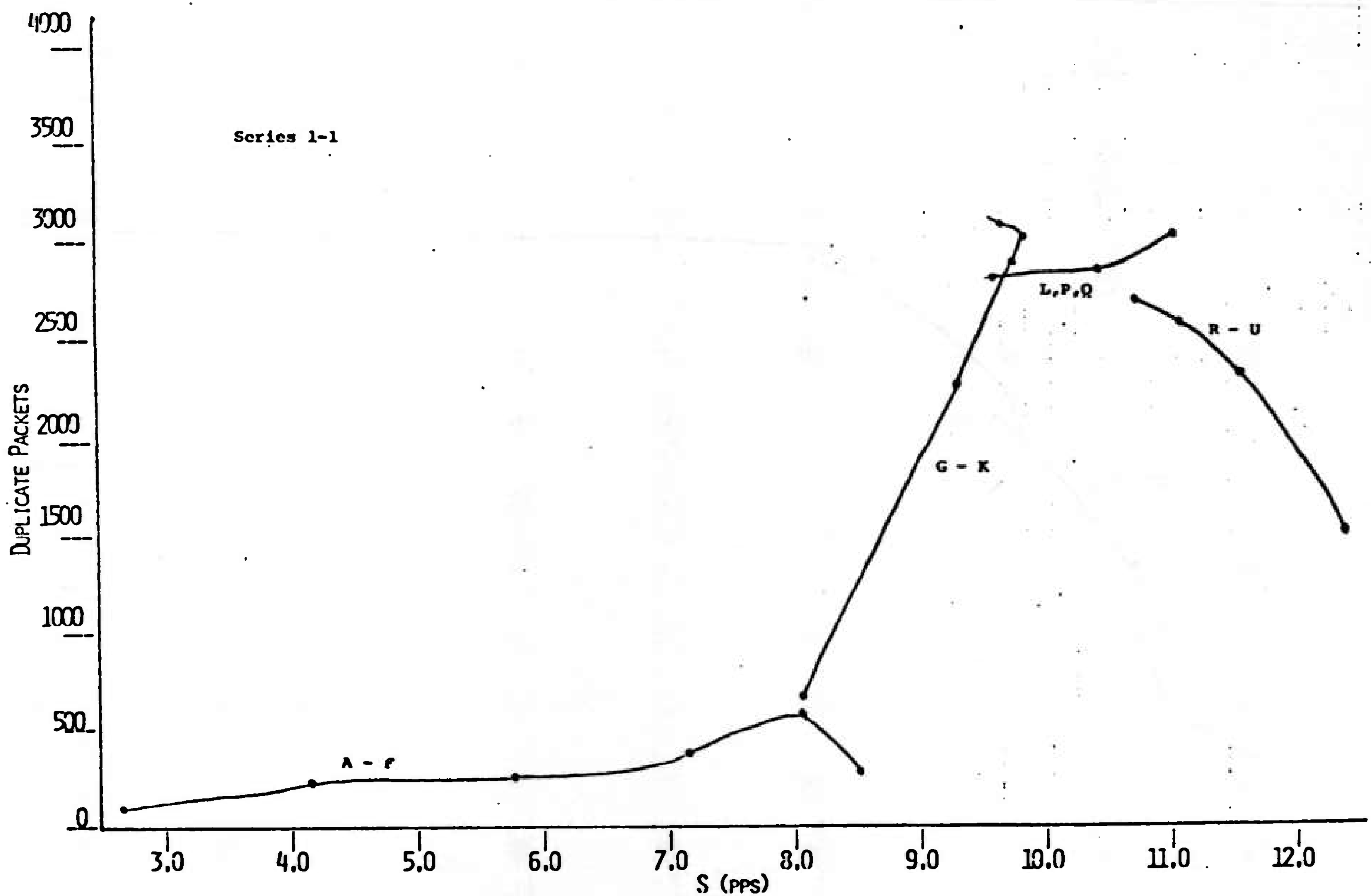

Series 1-1
DELAY (MS)
350
300
250
200
150
100
50
0
A - F
G - K
L,N,P,Q
R - U
4.0
7.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 1



Series 1-1
DUPLICATE PACKETS
4000
3500
3000
2500
2000
1500
1000
500
0
A - F
G - K
L,P,Q
R - U
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 2

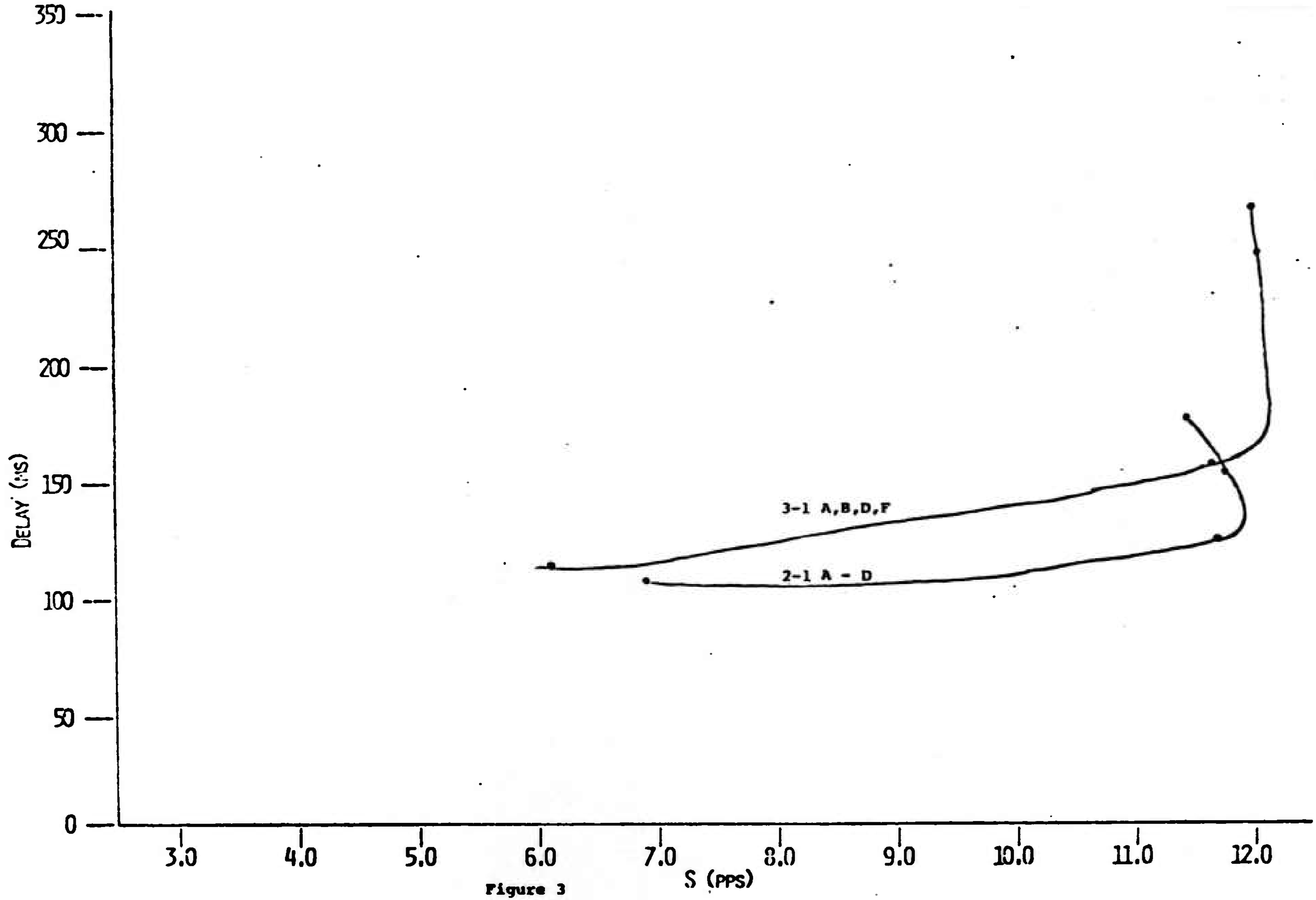

350
300
250
200
150
100
50
0
DELAY (MS)
3-1 A,B,D,F
2-1 A - D
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 3

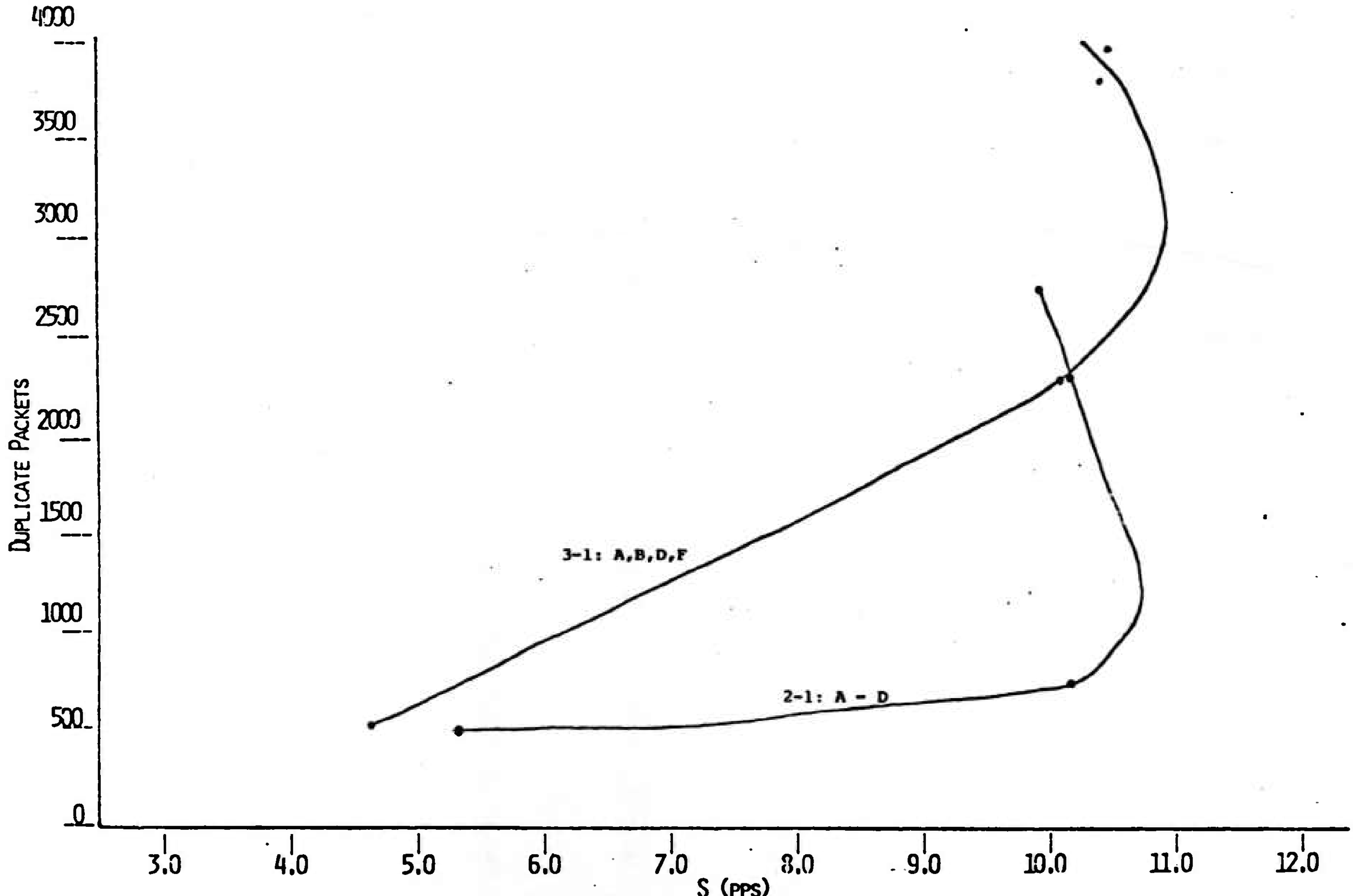
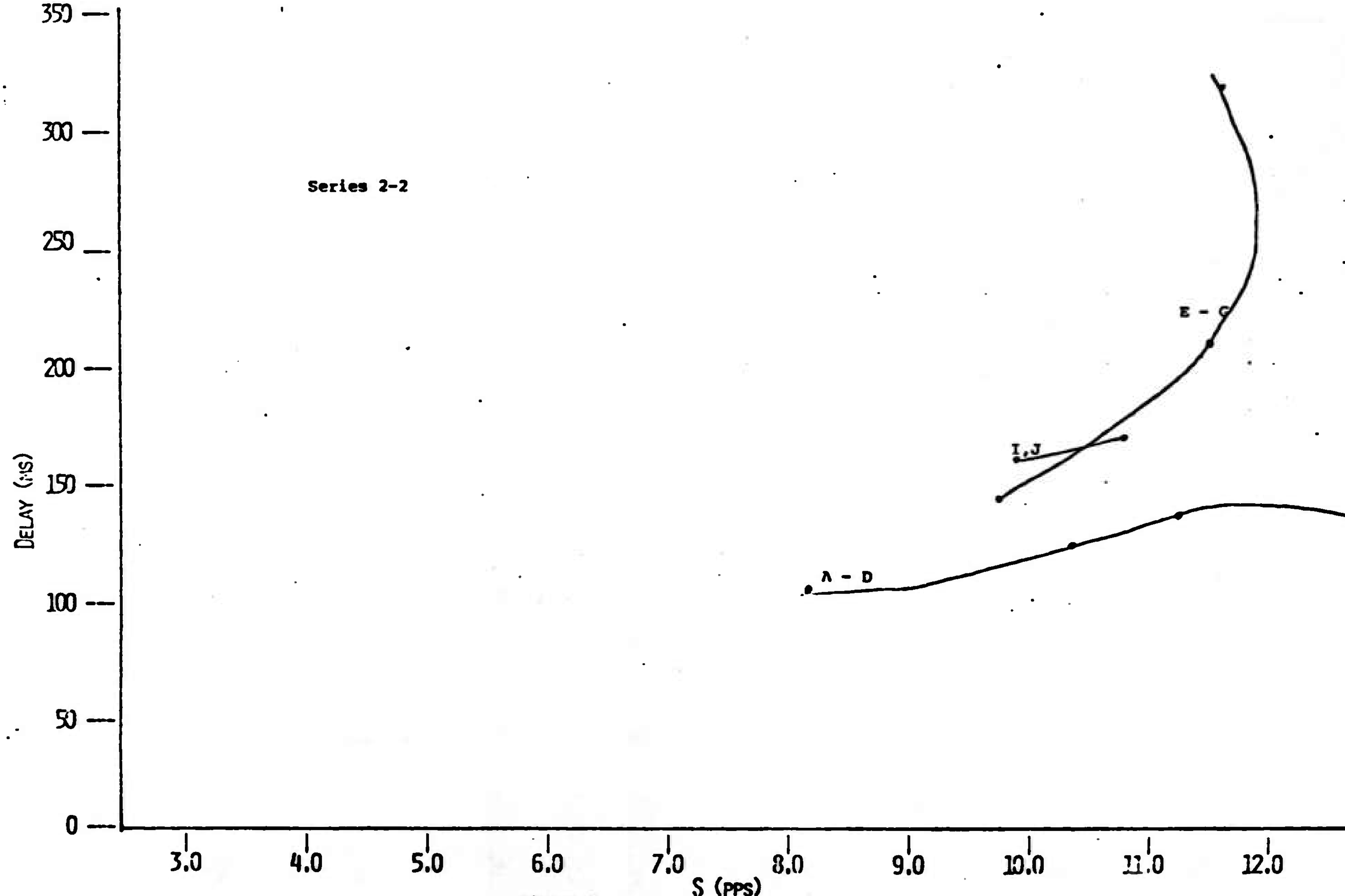

DUPLICATE PACKETS
4000
3500
3000
2500
2000
1500
1000
500
0
3-1: A,B,D,F
2-1: A - D
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 4



Series 2-2
DELAY (MS)
350
300
250
200
150
100
50
0
A - D
I,J
E - G
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 5

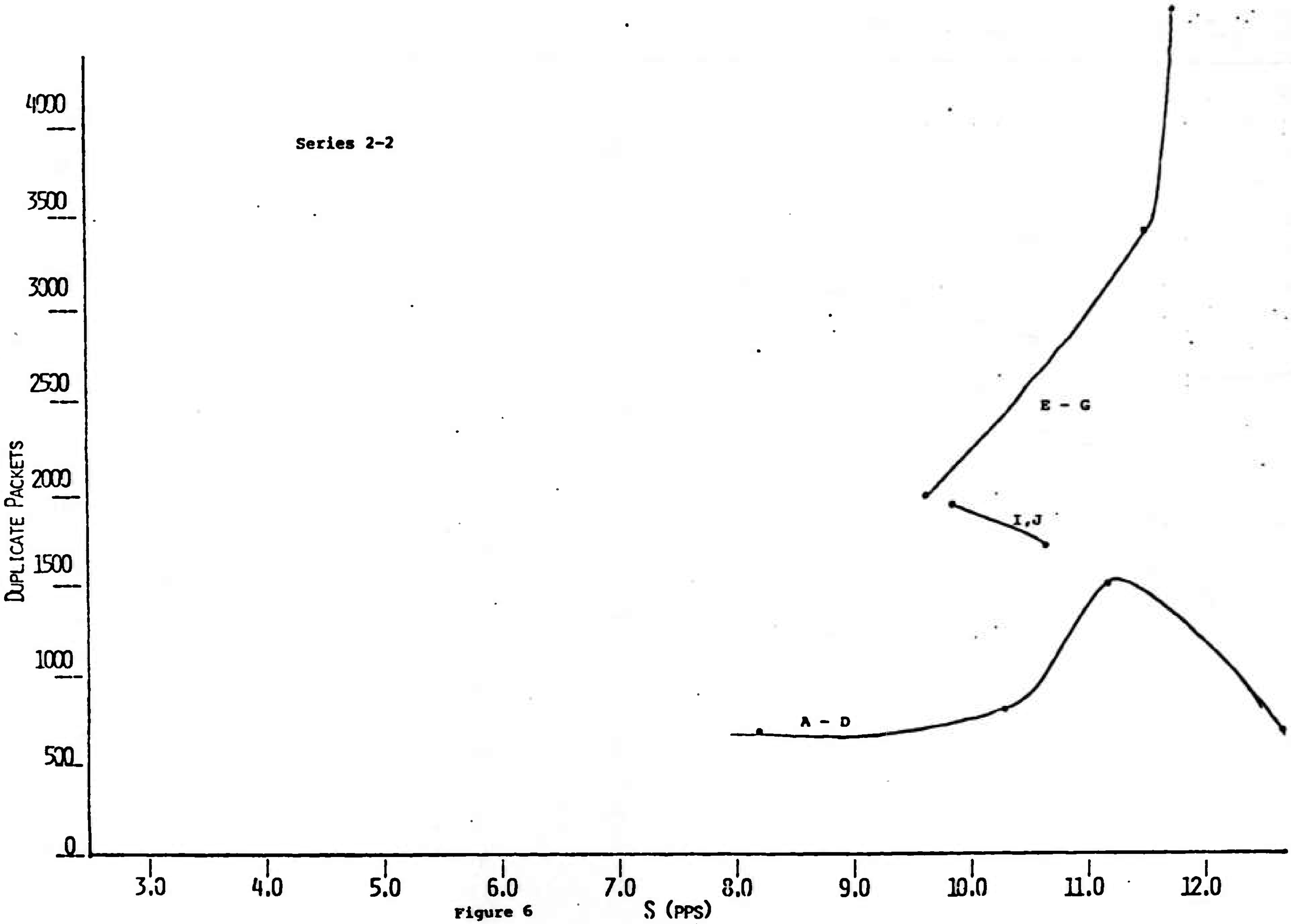
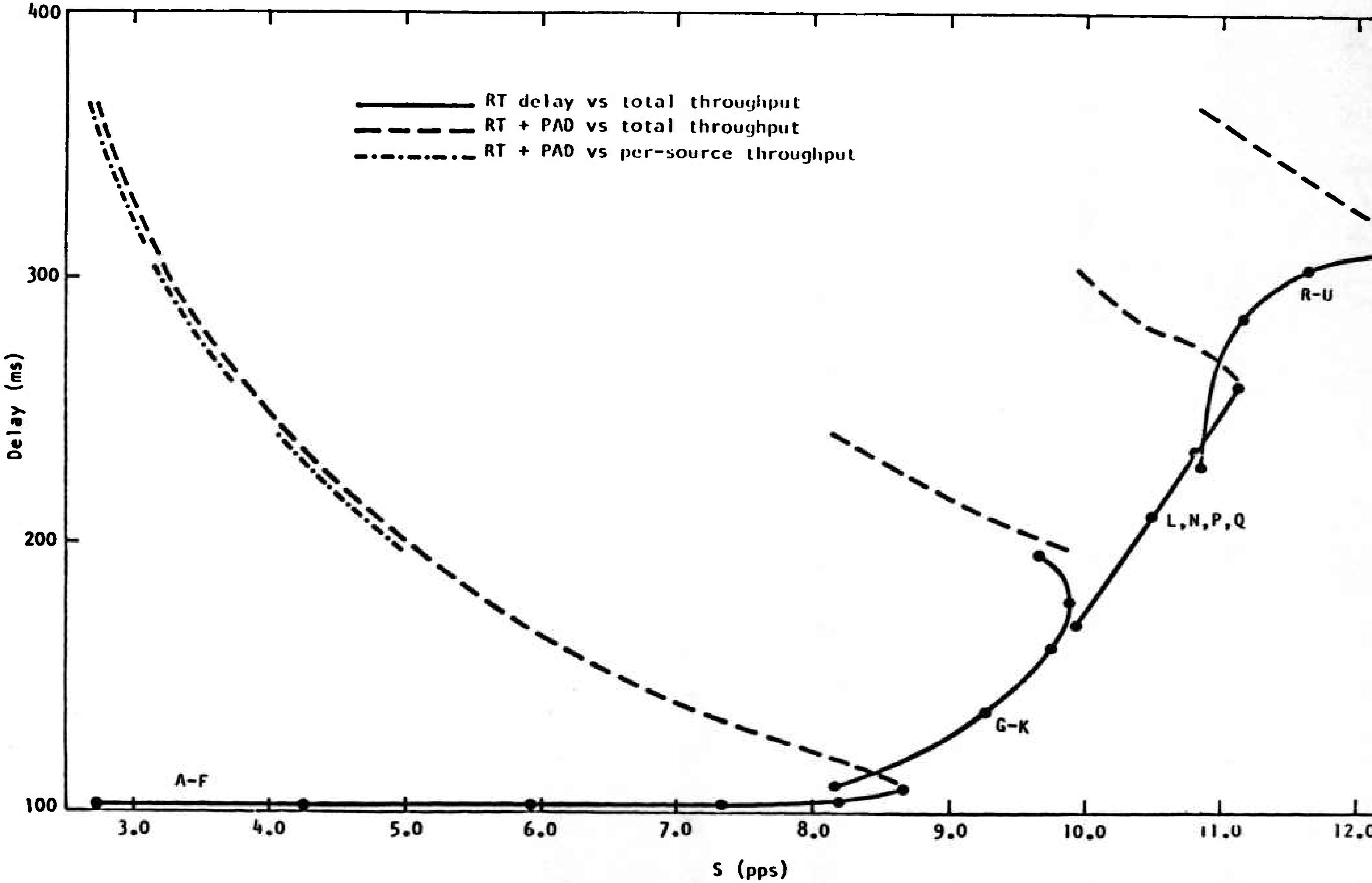

Series 2-2
DUPLICATE PACKETS
4000
3500
3000
2500
2000
1500
1000
500
0
E - G
I,J
A - D
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (PPS)

Figure 6



400
300
200
100
Delay (ms)
RT delay vs total throughput
RT + PAD vs total throughput
RT + PAD vs per-source throughput
A-F
G-K
L,N,P,Q
R-U
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
S (pps)

Figure 7

F. Tobagi
UCLA
January 13, 1978

On the Performance Analysis of Multihop Packet Radio Systems: Part I - The Design Problem

I. Introduction

The need for the sharing of computer resources by organizing these resources into computer networks has long been recognized and the feasibility of constructing such networks has been demonstrated by many successfully operating network systems. This economic sharing of computer resources has been made possible by the development of the packet switching technique [ROBE 70, KAHN 72, ABRA 73] whereby packet switches are interconnected by point-to-point data circuits according to some topological design. But when the number of communicating elements is sufficiently large and the overall traffic flow is small, the use of "packet broadcasting" techniques for interconnection becomes attractive, in that it simplifies the topological design considerably and provides very economic solutions. Moreover, economic studies [RODE 74] have clearly 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. 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 economic techniques to also bring the communication capabilities closer to the individual. Again, radio offers an attractive solution; numerous papers have already appeared in the literature which discuss the advantages of using these techniques over wire communication

[KLEI 75A, KLEI 75B]. The ALOHA system, at the University of Hawaii [ABRA 70] appears to have been the first computer communication system utilizing radio and is an excellent illustration of the feasibility of the technique. In line with the objectives set forth in the design of the ALOHA system, and to allow the support of many applications which require several added features not existing in the ALOHA system, (such features as direct communication by a ground radio network between users over a wide geographical area, coexistence with possibly different systems in the same frequency band, anti-jam protection, etc...) the Advanced Research Projects Agency of the Department of Defense undertook a new effort, the packet radio system, in order to demonstrate the applicability of the packet radio concept in organizing computer resources into a computer communications network. The target system requirements have been well assessed by R. Kahn [KAHN 75]: "The system consists of terminals and stations linked together by line-of-sight radio repeaters. The stations are minicomputers which provide system control; the terminals are hand-held devices, I/O consoles, computers, sensors, etc. The repeaters are simple relay devices which provide network area coverage for terminals and for one or more stations... The system should be capable of meeting the requirements of mobile communication with computers including real-time speech communication... The system should be able to share a common frequency band with other (possibly different) systems ...The packet radio technology should be able to serve mobile users whether on land, at sea or in the air... The individual elements of the system should be constructed so that they can be installed in the field with little delay...The packet radio network should not require the presence of any personnel for its normal operation...The system should provide essentially error-free performance for computer communications..."

In this series of notes, we concern ourselves with the design and performance evaluation of packet radio networks. For this, we start by defining, in this first part, the design problem in its generality, identifying all key system variables. It will be apparent from the discussion that the general design problem is a very complex and intractable one; this suggests that a viable approach consists of considering and analyzing specific configurations which are intuitively appealing. The results obtained will constitute a valuable feedback by means of which the design defficiencies are detected and subsequently corrected, and new configurations invented.

II. The Design Problem

The design of a packet radio system meeting a given set of target requirements and satisfying given performance and reliability constraints is a very complex task. At first a feasibility study is required which either demonstrates that the requirements can be met by the present technology or dictates that research and improvements need to be made in various specific areas; in either case, there can be a number of alternatives to choose among; this leads to a second design phase, an optimization phase, which consists of selecting the best combination of choices. In its most general form, the design problem consists of minimizing the system cost over the many system variables discussed thereafter.

A. Network Topology

In landbased networks, node locations are given and the topological design problem consists of determining the best configuration connecting those nodes; in packet radio systems, however, the question is to determine the number of devices (repeaters and stations), their locations, and their interconnections, needed to establish the required communications among sources and destinations. These variables are selected according to several factors: reliability, system load, frequency management... For example, the number of stations within the network is determined at first approximation by the amount of traffic to be handled; on the other hand, for reliability reasons the networks need to contain two or more stations; the allocation of responsibilities among the stations can be done either (i) by having one station assume the active role of controller while the other stations are kept in readiness for a backup mode, or (ii) by dynamically partitioning the control task mong them guaranteeing the backup of each station by the others in case of failure [KAHN 75]. The number of repeaters, on the other hand, is basically a function of: (i) the range of each repeater, essentially determined by its geographic setting and its effective radiated power, (ii) the repeater density required in achieving proper area coverage for communication with mobile hand-held terminals, and (iv) the frequency management and the communication mode adopted (point-to-point, broadcast, mixed...). The latter determines the interference pattern among the devices (it is discussed in more detail in the following paragraph).

B. Bandwidth Management

In digital nets using common carrier facilities, the system designer is given his choice of bit-rates and not of bandwidth. The common-carrier must worry about using his resources efficiently to provide a digital channel. It is not so with the packet radio network. The r.f. spectrum is a limited resource and the system designer has to guarantee efficient use of the bandwidth.

The bandwidth required for a packet radio network will largely depend on the channel configuration adopted. If the allocated bandwidth is divided into a number of low speed channels, dedicated to pairs of devices with directional antennas in such a way that the r.f. signal interference is completely avoided, then we would simply be creating a point-to-point network! Such a configuration can be appropriate if one is in the presence of a known and non-varying traffic pattern; a proper design allocation will guarantee an efficient utilization of all channels at all times. An important drawback of this approach, however, is that it requires careful alignment and tailoring of the devices; this provides too rigid a configuration when the traffic load is varying or should one require that the system be extended to handle larger geographical areas without a major redesign effort.

The most advantageous property one can gain in using radio as a communication media is perhaps the possibility of broadcast transmission. When devices are equipped with omnidirectional antennas, one can get rapid and convenient deployment as well as the desirable area coverage for mobile terminals.

Computer communication is often characterized by a high ratio of peak to average traffic rates. When the traffic handled by the devices is so characterized, then providing channels on a user-pair basis proves to be extremely wasteful. The provision of a single high-speed channel to be shared by a large number of users provides a better utilization of the bandwidth [TOBA 76]; it allows to take advantage of the "large number laws" whereby the demand at any instant can be approximated by the sum of the average demands of all users. With shared channels, however, we face the important problem of controlling the access to the channel by a geographically distributed set of message sources. The discussion of this issue is postponed until the next paragraph.

Traffic characteristics and other communications requirements may in some situations call for a network design with mixed modes. The bandwidth is partitioned into a number of channels some of which are shared while others are dedicated. The allocation of channels can be functional (channels used solely for a particular network function such as error control, monitoring...) and/or geographical to support the traffic requirements. In the latter case, the allocation has to be dynamic so that the statistical variations in traffic are accounted for. An illustration of a mixed mode network design is given by the "DAMN-FINE" concept introduced by Fralick [FRAL 73]. In this concept, the bandwidth is divided into a number of channels; each channel is time-shared among several mobile users and a repeater. Channel allocation is performed by an adaptive search and selection algorithm existing in each repeater and terminal, thus distributing the control and providing for an efficient dynamic allocation to match changing traffic and interference pattern.

C. Channel Access Policy

In conjunction with the proper selection of a channel configuration, the channel access procedure to be in use will have a great impact on the overall performance of the network.

With fixed bandwidth assignment, the time-bandwidth space is partitioned into slots which are allocated to the users in a static predetermined fashion. Fixed assignment takes two common forms: orthogonal, such as frequency division multiple access (FDMA) and synchronous time-division multiple access (TDMA), and "quasi-orthogonal", such as code division multiple access (CDMA). With fixed assignment the problem of accessing the channel is trivial. It is not so if the design calls for the usage of shared channels. Various methods are available. Some (such as contention and polling) require the presence of a central station performing the control. In a contention network, the terminal makes a request to transmit: if the channel is free, transmission goes ahead; if it is not free, the terminal must wait. The station schedules the transmissions either in a prearranged sequence (according to some scheduling scheme) or in the sequence in which the requests were made. In the polling technique, the station asks (polls) the terminals one by one as to whether they have anything to transmit. For this, the station may have a polling list giving the order in which terminals are polled. When a polling message is sent to the next terminal in sequence, and if the terminal has some data to transmit, it goes ahead; if not, a negative reply (or absence of reply) is received by the station and the next terminal is polled.

In the absence of central controllers, dynamic sharing can be accomplished through the so-called random access techniques. In this mode of sharing, however, one must be prepared to resolve conflicts which arise when more than one demand is simultaneously placed upon the channel; this is also referred to as interference. A simple scheme known as "pure ALOHA" [ABRA 70] permits users to transmit their packets anytime they desire

If within some appropriate time-out period they receive an acknowledgment from the destination, then they know that no conflicts occurred. Otherwise, they assume a collision occurred and they must retransmit. (To avoid continuously repeated conflicts, some scheme must be devised for introducing a random retransmission delay, spreading the conflicting packets over time; this parameter has been shown to have a great impact on the performance of the access mode, and its proper tuning is most crucial [KLEI 75B]. A second method for using the radio channel is to modify the completely unsynchronized use of the ALOHA channel by "slotting" time into segments whose duration is exactly equal to the transmission time of a single packet. If we require each user to start the transmission of his packets only at the beginning of a slot, then when two packets conflict, they will overlap completely rather than partially, providing an increase in channel efficiency. This method is referred to as "slotted ALOHA" [ROBE 72, ABRA 73, KLEI 73]. A third scheme, the "carrier sense multiple access" mode (CSMA) consists of reducing the level of interference (caused by overlapping packets) by allowing users to sense the carrier due to other users' transmissions; based on the information gained in this way about the state of the channel (busy or idle), the terminal takes an action prescribed by the particular CSMA protocol being used; (in particular, a terminal never transmits when it senses the channel busy). In [TOBA 74, KLEI 75B] two protocols referred to as nonpersistent and p-persistent CSMA were analyzed.

The selection of the proper channel access policy will largely depend on the behavior of these modes in the environment in question. It was shown [TOBA 76], for example, that if we are in the presence of a large population of bursty users contending on the same channel then random access exhibits a performance far superior to fixed assignment or polling. However, centrally controlled techniques become more attractive when the contending population consists of a small number (≤ 5) of devices with buffering capabilities.

Split-channel reservation multiple access [TOBA 74, TOBA 76] is yet another channel access technique, suitable for packet radio environments, which calls upon both random access and central control. The available channel bandwidth is divided into two parts: one used to transmit control information (such as requests), the second used for the messages themselves. The request channel is operated in a random access mode; the requests are handled by a central controller which controls the access to the message channel. This technique represents an interesting scheme in that it is simple and efficient over a large range of system parameters The performance of the above access schemes has been compared in single-hop environments in which all devices are within range and in line-of-sight. Their performance in multi-hop environments is yet to be determined.

D. Modulation schemes

Just as with channel access, the modulation technique used is an important element in the design of a packet radio network. It has an important impact on network performance as well as on the utilization of the r.f. spectrum. Indeed, for a given required probability of bit error and a given transmitted power, modulation schemes determine the tradeoff relation between the range of a packet radio device (which affects the network topology) and the bit rate achievable (which affects the network throughput). While the simplest schemes achieve a relatively inefficient use of power (which also constitutes a limiting resource when the equipments have to satisfy size and weight constraints), the most efficient ones require more complex receivers! From these points of view, the spread spectrum modulation scheme often represents a good compromise [FRAL 75]. In addition, it provides some kind of discrimination whereby one or more packets can be successfully detected in the presence of other packets; it also provides some degree of protection against unwanted interference and is desirable for jam-free operation. Finally, it renders coexistence with other systems feasible which may result in better utilization of the frequency spectrum. However, these advantages are obtained at the expense of a higher bandwidth requirement (possibly shared with other users). Discarding considerations of coexistence and vulnerability to unwanted interference (although in many design cases these may be the most critical factors), the designer is faced with a choice between spread-spectrum signalling and narrow-band (i.e., non-spread) systems: each presents different tradeoffs among some key system parameters such as the devices' radiated power, the r.f. bandwidth used, the channel data rate and the packet delay. A preliminary analysis made to that affect by Fralick [FRAL 74] has basically shown that the single-channel ALOHA spread-spectrum scheme is not an appealing choice in that the bandwidth required for given data-rate, throughput, and delay is much larger than that of the narrow-band system. The same result was also shown to be true for multi-channel spread-spectrum schemes with nonorthogonal ch nnels.

E. Operational protocols: Routing policy, error control procedures and flow control

The principal functions of a packet communication system are to provide the transport of packets efficiently and reliably. Packets originate at traffic sources and have to be routed through the network to reach their destination. It is apparent that, in addition to all afore mentioned design factors, network performance will strongly be affected by the routing and flow control policies adopted.

If a point-to-point channel configuration is in use (dedicated channels with directional antennas), then the routing algorithms long devised for landbased store-and-forward networks will apply. They usually are classified as either deterministic or stochastic according to whether route computations are based upon a deterministic or stochastic decision rule [FULT 71]. A number of studies have investigated many of these routing algorithms, their performance and applicability. With shared broadcast channels, however, the nature of the problem becomes significantly different: packet transmissions are received by all nodes within range of the transmitting device; unless fixed predetermined routes are assigned and followed (a scheme which would defeat the purpose of broadcast transmission), the transmitted packet should carry, at each transmission, the next node's address. Equivalently, the routing algorithm can be defined as the decision at a receiving node as to whether to relay or ignore the packet. A number of routing schemes suitable for broadcast radio networks have been discussed by Gitman et. al. [GITM 76]. An undirected routing algorithm, also called broadcast routing, is based on an unconditional repeat of a correctly received packet as long as its handover number did not exceed a specified maximum. The maximum handover number determines the lifetime of a packet in the net. Looping is prevented by saving at each receiving repeater a unique identifier of each packet for a specified length of time. This algorithm is inefficient in that it allows generation of a large number of duplicate packets and the delivery of the packet to its destination is guaranteed only to a certain extent. If the network structure is centralized*, then a directed routing algorithm can be used which utilizes the central station to periodically structure the network for efficient flow paths. A hierarchical point-to-point network can thus be formed by having the station assign to each repeater a label which defines its position in the hierarchy. The label carried in the packet header thus defines completely the path between the station and a repeater. Another class of routing is called the "directed broadcast" routing. Algorithms in this class are best suited to distributed radio networks in which communication has to be established between arbitrary source-destination pairs, and data need not flow through the station. To accomplish this task repeater nodes keep at all times routing information similar to that used in the ARPANET [KLEI 76]. The overhead incurred and the intelligence required by each node in performing the routing function may become significant.

The reliable transport of packets in a radio network can only be accomplished if proper error control procedures are in use. Indeed, in addition to random noise, errors in multi-access radio channels are due to multipath effects and interference caused by overlapping packets. Error detection code in conjunction with positive acknowledgment for each correct message is a reliable mechanism; however, the amount of overhead

---

*In a centralized network, all traffic passes through the station. Such a network architecture is best suitable when the radio network is used for local collection and distribution of traffic. The station then is the central computer facility or a gateway to other communications systems.

introduced and the degradation in network performance incurred still vary, here again, with the mode of operation. In [TOBA 78] for example, considering a single-hop system, we studied this degrading effect for two access schemes, slotted ALOHA and CSMA, and addressed ourselves to the problem of comparing two design alternatives: the common-channel configuration (a single channel for both information and error control traffic), and the split channel configurations. In particular, it was shown that the degradation in performance can be very significant when the acknowledgment packet size represents over 10% of the message packet!

In multi-hop networks we usually distinguish two types of acknowledgments: the hop-by-hop acknowledgment (HBH) and the end-to-end acknowledgment (ETE). HBH acknowledgments are issued by repeaters and guarantee the transport of a packet over a hop. They help decrease the delay incurred by end-to-end transmissions, but do so at the expense of added overhead. HBH acknowledging can be handled in several ways; HBH acknowledgments can be (i) piggy-backed (acknowledgment information is transmitted in the control portion of another packet destined to the sending repeater), (ii) passive (in that the relaying of a packet over a hop constitutes the acknowledgment for the transmission over the previous hop: this "echo-acknowledgment" mode is due to the omni-directional broadcast property), or (iii) active (whereby an acknowledgment packet is actually created and transmitted).

ETE acknowledging, in addition to guaranteeing a reliable communication between the end devices and the communicating processes, serves as a flow regulation mechanism; it has a great impact on network performance by preventing any source from overloading the network and causing serious congestion. The mechanism used also affects significantl end-to-end message delays and buffer utilization at end devices. It is common to see several mechanisms in use in the same network depending on the particular application being supported.

In real systems, the proper functioning of operational protocols is dictated by their ability to adapt to changing system states, which in turn calls for the existence of efficient network monitoring functions at the various levels of protocols. Dynamic routing procedures, for example, require a thorough knowledge and a continuous update of the network connectivity and of the traffic load distribution in order to effect the appropriate route changes. Such monitoring functions can be distributed whereby devices can collect data regarding their own locality. In centralized networks, the station can be given the capability of centrally collecting data regarding the entire network and thus can perform both local and global decisions; it can be made responsible for route assignments and route changes, as well as the proper value assignment to all of the parameters throughout the network.

D. Repeater Design

The repeater is perhaps the most important element of a packet radio network. Almost all traffic flows through a number of repeaters on its way to destination. In addition to selecting the proper hardware to perform all r.f. functions (transmission, reception, error detection, bit synchronization, etc...) and providing the proper implementation of all previously discussed operational protocols, there is a number of design factors which are believed to have a crucial effect on the overall network performance. They are: the repeater's transmit power, its processing speed, and the storage capacity and its management. When the application require the devices to be rugged, light weighted, portable and operational for a period of time without attendance, then a number of constraints are placed on these parameters. The storage capacity has to be minimal and the power consumption has to be conservative; the latter affects both the processing speed (which in turn impacts the achievable throughput) and the radiated power (which in turn affects the range of the device and therefore the network topology). On the other hand, for a given topology, repeater's range affects network connectivity and the interference pattern...

To summarize, the design problem in radio networks can be expressed as follows:

Minimize the network cost subject to given constraints on throughput, delay, and reliability over the system variables:

1) <u>Network topology</u>: # of devices and their geographical setting
2) <u>The bandwidth management</u>: dedicated channels, shared channels, mixed modes
3) <u>The channel access procedure</u>: fixed assignment, centrally controlled schemes, random access
4) <u>The modulation scheme</u>: spread-spectrum, narrow band...
5) <u>The operational protocols</u>: routing policy, error control procedures, flow control and monitoring functions
6) <u>Nodal design</u>: storage capacity, buffer management, power requirement, processing speed

It is all apparent from the above discussion that the design of a PR network involves a large number of variables which interact in a very complex fashion. In its present form the solution is extremely hard to come by. A reasonable approach to follow consists of first selecting those system variables which are obviously determined by some of the constraints. For example, for rapid deployment and easy communication among mobile devices, the entire system will employ omnidirectional antennas and will share a single high-speed channel via some random access scheme. Hopefully, this first step decreases significantly the variable space. The next step will consist of considering various specific configurations which are intuitively appealing; these configurations are then analyzed in order to

(i) identify the key parameters which affect the performance

(ii) determine the conditions under which the a priori specified constraints can be satisfied

This step represents an iterative process in that the results obtained from the analysis of some network configuration constitutes a valuable feed-back process by the means of which the design deficiencies are detected and subsequently corrected, and new configurations are invented.

In the forthcoming notes, we shall execute this second step by focusing on the analysis of specific network configurations (star networks, fully connected networks) operating under the slotted ALOHA and the nonpersistent CSMA modes.

References

1 ABRA 70 — Abramson, N. "Another Alternative for Computer Communications," AFIPS Conference Proceedings, FJCC 70, pp. 695-702.

2 ABRA 73A — Abramson, N. and F. Kuo (co-editors), Computer Communication Networks. New York: Prentice-Hall Pub. Co., 1973

3 ABRA 73B — Abramson, N., "Packet Switching with Satellites," in 1973 Nat. Comput. Conf., AFIPS Conference Proceedings, vol. 42, New York, 1973, pp. 695-702.

4 FRAL 74 — Fralick, S. and J. K. Leung, "Study of Throughput and Delay of Spread-Spectrum Multiple Access Modes," Packet Radio Note 2, SRI International, Menlo Park, California, April 1974.

5 FRAL 75 — Fralick, S. and J. Garrett, "A Technology for Packet Radio," AFIPS Conference Proceedings, vol. 44, 1975, AFIPS Press, Montvale, N.J.

6 FRAL 73 — Fralick, S., "The DAMN (-FINE) Concept," Packet Radio Temporary Note #82, SRI International, Menlo Park, California, Nov. 1973.

7 FULT 71 — Fultz, G. and L. Kleinrock, "Adaptive Routing Techniques for Store-and-Forward Computer Communications Networks," Proc. of the Internat. Conf. on Commun., pp. 39-1 to 39-8, Montreal, Canada, 1971.

8 GITM 76 — Gitman, I., R. M. Van Slyke and H. Frank, "Routing in Packet-Switching Broadcast Radio Networks," IEEE Trans. Commun., vol. COM-24, Aug. 1976.

9 KAHN 72 — Kahn, R. E., "Resource Sharing Computer Communication Networks," IEEE Proceedings, Nov. 1972, pp. 1397-1407.

10 KAHN 75 — Kahn, R. E., "The Organization of Computer Resources into a Packet Radio Network," AFIPS Conference Proceedings, NCC 75, pp. 177-186.

11 KLEI 73 — Kleinrock, L. and S. Lam, "Packet Switching in a Slotted Satellite Channel," in Nat. Comput. Conf., AFIPS Conference Proceedings, vol. 42, Montvale, N.J.: AFIPS Press, 1973, pp. 703-710.

12 KLEI 75A — Kleinrock, L. and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. Commun., vol. COM-23, pp. 410-423, Apr. 1975.

13 KLEI 75B — Kleinrock, L. 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.

14 KLEI 76 Kleinrock, L., Queueing Systems, vol. II, Computer Applications. New York: Wiley-Interscience, 1976.

15 ROBE 70 Roberts, L. G. and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, SJCC 70, pp. 543-549.

16 ROBE 72 Roberts, L. G., "ALOHA Packet System with and without Slots and Capture," ARPA Network Inform. Center, Stanford Res. Inst., Menlo Park, Calif., ASS Note 8 (NIC 11290), June 1972.

17 ROBE 74 Roberts, L. "Data by the Packet," IEEE Spectrum, February 1974.

18 TOBA 74 Tobagi, F., "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," Ph.D. dissertation, Comput. Sci. Dept., School of Engr. and Appl. Sci., Univ. of California, Los Angeles, rep. UCLA-ENG 7499, Dec. 1974.

19 TOBA 76 Tobagi, F. 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.

20 TOBA 78 Tobagi, F. and L. Kleinrock, "The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels," to appear in the IEEE Trans. Commun.

F. Tobagi
UCLA
January 17, 1978

On the Performance Analysis of Multihop Packet Radio Systems:
Part II - Star Configuration Employing Slotted ALOHA*

1. Tree-Structured Networks: Definitions

A number of papers have appeared which study various simple network topologies. Single-hop networks, where terminals communicate directly with a central station have been investigated extensively; access modes in such environments have been carefully evaluated [2,3,4]. A two-hop configuration involving a ring of repeaters around a station have been analyzed by Gitman [5]: network capacity was derived, but no consideration was made of network delays. In this note, we consider centralized networks characterized by an array of repeaters organized in a symmetric tree configuration with a (single) station at the root; all devices are provided with omnidirectional antennas and employ slotted ALOHA random access over a single shared channel. (See Figure 1.) Traffic originates at terminals located at the outskirts of the tree, and is destined to the station. Thus, we consider inbound traffic. The first transmission of a packet from a terminal is to a repeater located at a leaf of the tree. The routing of the packet through the network is completely specified by the tree structure, as is the connectivity pattern among the devices. The basic performance measure sought is the throughput-delay trade-off and its dependence on such key system parameters as the topology (degree of the tree, depth of the tree,** and the connectivity

---

*The results discussed in this note have already been published in [1], and distributed as PRTN #234. This note, however, is more complete in that all derivations are included.

**In this note, however, we shall restrict ourselves to trees of depth 1.

pattern among devices), repeaters' retransmission delays and repeaters' storage capacities.

2. Traffic Model and Transmission Protocol

The time axis is assumed to be universal and slotted into segments whose duration is equal to the transmission time of a packet.* All devices are assumed to be synchronized and start packet transmission at the beginning of a slot. Associated with each repeater located at a leaf of the tree is a population of terminals which generate new packets at an aggregate rate of s packets per slot, all destined to the station. Each repeater is provided with a finite storage capacity which can accomodate a maximum of M packets. The station has an infinite storage capacity. Packets are transmitted by repeaters on a first-come-first-served basis; when its buffer is non-empty, a repeater transmits the head of its queue with a probability p. When the packet transport is successful (i.e., the transmission is free of interference and storage is available at the receiving repeater), the packet is deleted from the sender's queue; otherwise, the packet incurs a retransmission delay geometrically distributed, with mean 1/p. A repeater learns about its success or failure instantaneously; that is, acknowledgments are assumed to be instantaneous and for free. At any one time, a repeater 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.

Let R denote a repeater located at some level k in the tree. Let S(R) denote the set of repeaters which are sons of R. Let F(R) denote the father of R. A packet successfully transmitted by the population of repeaters S(R)

---

*Packets in this study are all of a fixed size.

can be "blocked" at the immediate destination repeater R; blocking is due to two factors: (i) either R or F(R)* or both are in a transmit mode or (ii) R's receiver is on, F(R) is quiet, but R's buffer is full.

Due to the blocking of traffic at the receiving repeater, the rate of successful transmissions of packets to a leaf from its corresponding population of terminals is actually greater than s and is denoted by λ. Furthermore, this process of packet arrivals to a leaf is assumed to be a Bernoulli one.

In this note, we shall consider the special case of trees of depth 1 (and degree N), referred to as <u>star networks</u> (See Fig. 2), and we shall provide an analysis by means of which the throughput-delay performance can be numerically evaluated. In a forthcoming note [6], we shall consider configurations in which all repeaters are in line-of-sight and within range of each other. These are referred to as <u>fully-connected networks</u>.

3. Analysis of Star Networks Employing Slotted ALOHA

Let N denote the degree of the tree. (N also denotes the number of repeaters present in the configuration.) Given the transmission protocol adopted (as described in section 2 above), and given that the input process to each repeater is a Bernoulli process, the state of the system at slot t is entirely defined by the vector

$$\underline{n}^t = (n_1^t, n_2^t, \ldots, n_N^t)$$

where n_i^t is the number of packets present at the i^{th} repeater at slot t.

---

*Given that the hearing pattern among repeaters is assumed to be determined by the tree structure, a transmission by F(R) over a slot interferes with a transmission from S(R) over the same slot.

Moreover, $\underline{n}^t$ is a Markov chain. In this section, we shall analyze $\underline{n}^t$ by first considering the case where the buffer size at each repeater is limited to one packet (M=1) and then by treating the more general case M>1. In all cases, the configurations considered will be symmetric.

3.1 The Single-Buffer Case

3.1.1 Analysis

In this symmetric single-buffer configuration, the state of the system can be equivalently described by the number of repeaters with non-empty buffers, referred to as the number of "active" repeaters. Let n^t $(0 \leq n^t \leq N)$ denote that number at slot t. The Markov chain n^t has a transition matrix P whose $(i,j)^{th}$ element is given by

$$p_{ij} = \begin{cases} 0 & j < i - 1 \\ P_s(i)(1 - \lambda)^{N-i} & j = i - 1 \\ [1 - P_s(i)] \binom{N - i}{k} \lambda^k (1 - \lambda)^{N-i-k} & \\ \quad + P_s(i) \binom{N - i}{k + 1} \lambda^{k+1} (1 - \lambda)^{N-i-(k+1)} & j = i + k \\ & k = 0, 1, 2, \ldots, N - i \end{cases} \tag{1}$$

where $P_s(i)$ denotes the probability of a successful transmission given i active repeaters and is expressed as

$$P_s(i) = ip(1 - p)^{i-1} \tag{2}$$

Let $\pi_i \overset{\Delta}{=} \lim_{t \to \infty} Pr\{n^t = i\}$. We compute the stationary distribution $\Pi = \{\pi_0, \pi_2, \ldots, \pi_N\}$ by solving recursively the system $\Pi = \Pi P$.

Let $\bar{n}$ denote the average number of active repeaters. We have

$$\bar{n} = \sum_{k=0}^{N} k\pi_k \tag{3}$$

A packet successfully transmitted by a population of terminals can be "blocked" at the immediate destination repeater. As already mentioned in section 2, blocking is due to two factors: (i) the repeater is in a transmit mode* (and we denote by β the probability of such an event) or (ii) the repeater's receiver is on but its packet buffer is full (and we denote by α the probability of this event). Let $B = \alpha + \beta$. We have:

$$\begin{aligned}\alpha &= \Pr\{\text{queue at receiving repeater full}\}\ (1 - p) \\ &= (1 - p)\bar{n}/N\end{aligned} \tag{4}$$

$$\begin{aligned}\beta &= [1 - \Pr\{\text{queue at receiving repeater empty}\}]p \\ &= p\bar{n}/N\end{aligned} \tag{5}$$

$$B = \bar{n}/N \tag{6}$$

The total network throughput, denoted by S, is defined as the rate of successful packets received at the station; it is given by

$$S = (N - \bar{n})\lambda \tag{7}$$

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: (i) the <u>access delay</u> D_a, defined to be the time required for the packet to be correctly received at the leaf repeater, and (ii) the <u>network delay</u> D_n which consists of the time elapsed since the packet is accepted at the leaf repeater until it is successsully received at the

---

*Recall that the station's receiver is assumed to be always on.

station. By Little's result, the average network delay is given by

$$D_n = \overline{n}/S \tag{8}$$

The maximum value of λ, allowable in this model, is a function of the access mode in use by the terminals. Indeed, 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 buffer, 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. $B = \overline{n}/N$ represents the fraction of time a repeater is active; and D_n represents the total packet delay.

3.2 Numerical Results

The main difficulty in obtaining a closed form solution to the present problem lies in the fact that the outcome of the system in a slot (i.e., the amount of work completed) is dependent on the state of the system in that slot. Thus, the time spent by a packet until its success (i.e., the service time of the packet) is dependent on the evolution of the system over the period of time that constitutes its service. This situation is usually observed in the study of random-access techniques and renders the analysis of systems utilizing these techniques a rather tedious task. The numerical approach adopted here has helped overcome this difficulty (to a certain extent); its advantages over simulation are many: it is simpler, it is more economical, and the numerical results it provides are more accurate (if not exact). Accordingly it allows the generation of numerical results for a large range of system parameters. This is indeed a quality of great importance when one attempts to gain insight on the behavior of a system via experimental design and it is reflected in the number of cases studied and presented in this note. The technique, however, does present a major drawback: it amounts to performing measurements on a model of the real system which is based on a number of simplifying assumptions (acknowledgments are instantaneous and for free, processing time is negligible...). The relaxation of such assumptions may cause the analysis to become intractable; while it is more readily performed in a simulation model. Nevertheless, the assumptions made are not unreasonable when one's objective is, again, to gain insight into the behavior of packet radio systems and to determine the trends followed by the performance measures as a function of the system parameters.

Fixing N and λ, we observe that $\bar{n}$ is a <u>concave</u> function of p. Thus there exists a value of p which minimizes $\bar{n}$. From Equations (6), (7) and (8) which express D_n, B and S in terms of $\bar{n}$ we note that D_n and B are also concave functions of p while S is a convex function of p. Moreover, it is clear that the value of p which maximizes S, minimizes D_n and B. As an example we show in Figure 3 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 Figure 4, we plot the optimum delay versus the achieved throughput for various values of N. We note that as the degree of the tree increases, so does the network delay. The reverse behavior is observed for the probability of blocking B over a large range of S ($0 < S < 0.35$) as shown in Figure 5. The total packet delay D will be dealt with in a subsequent section. Next, we examine the network performance at saturation, i.e., at maximum attainable throughput.

3.1.3 Network Saturation

In terms of the feedback model, network behavior at saturation is obtained by driving λ to 1. With $\lambda = 1$, the Markov chain n^t reduces to one with only two states (state N-1 and state N) as depicted in Figure 6. Thus, at saturation, the stationary distribution π is such that $\pi_0 = \pi_1 = \ldots = \pi_{N-2} = 0$, and such that π_{N-1} and π_N are solution of the system

$$\pi_{N-1}\,(N-1)p(1-p)^{N-2} + \pi_N\,Np(1-p)^{N-1} = \pi_{N-1}$$
$$\pi_{N-1} + \pi_N = 1 \tag{9}$$

The latter yields

$$\pi_{N-1} = \frac{Np(1-p)^{N-1}}{1 - (N-1)p(1-p)^{N-2} + Np(1-p)^{N-1}} \tag{10}$$

π_{N-1} represents also the network throughput. Indeed, the average number of active repeaters $\overline{n}$ is given by

$$\overline{n} = (N-1)\pi_{N-1} + N\pi_N$$
$$= N - \pi_{N-1} \tag{11}$$

Given $\lambda = 1$, we see from Eq. (7) that S_{out} is precisely π_{N-1}. The network delay D_n is given by

$$D_n = \frac{N}{\pi_{N-1}} - 1$$
$$= \frac{1}{p(1-p)^{N-1}} - \frac{(N-1)p}{1-p} \tag{12}$$

Let us examine the special case of $N = 2$. The throughput and delay are expressed, as long as $p \neq 1$, as

$$S = \frac{2p}{1 + 2p} \tag{13}$$

$$D_n = 1 + \frac{1}{p} \tag{14}$$

The network capacity is obtained for $p \to 1$ and equals

$$\lim_{p \to 1} S = \frac{2}{3} \tag{15}$$

The minimum packet delay is then equal to 2 (see Fig. 4).

Let us now study the limiting case as $N \to \infty$. It is clear from Eq. (10) that, as N gets large, S is on the order of $Np(1-p)^{N-1}$. The maximum throughput is then achieved by assigning to p the value $1/N$, in which case it is expressed as

$$S = \left(1 - \frac{1}{N}\right)^{N-1} \tag{16}$$

This, in the limit as $N \to \infty$, approaches $1/e$, which, as expected, is the slotted ALOHA channel capacity with infinite population. For $p = \frac{1}{N}$ the network delay is given by

$$D_n = \frac{\quad}{\left(1 - \frac{1}{N}\right)^{N-1}} - 1 \tag{17}$$

which, as N gets large, follows the linear asymplotic behavior

$$D_n \approx Ne - 1 \tag{18}$$

The above results are summarized in Fig. 7 where we plot network capacity, network delay and the probability of blocking at saturation versus N.

Consider now the two-hop environment. In this case the maximum achievable value for λ is $1/e$. The system capacity is then expressed as

$$\max_{p} \left\{ \frac{N}{e} [1 - B(\frac{1}{e}, p)] \right\}$$

In Fig. 8 we plot the optimum network throughput S (maximized over p, λ kept constant) versus λ for various values of N. The system capacity is precisely the network throughput at $\lambda = 1/e$. We note the following. For the larger values of N ($N > 3$), the throughput S (which increases with increasing values of λ) levels off rather rapidly and gets really close to its maximum value for $\lambda < 1/e$. This is not so with the smaller values of N; for N = 2 and 3, values of λ greater than $1/e$ can produce system throughput noticeably higher than what can be achieved at $\lambda = 1/e$. This leads us to state that, for all practical purposes, 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 the system capacity for N = 2 is smaller than for N = 3, and that for $N \geq 3$, it is a decreasing function of N. In Fig. 9 we plot the system capacity versus N for the two-hop configuration.

3.1.4 The Throughput-Delay Tradeoff

To complete the delay analysis, we need to evaluate the access delay D_a for a given throughput S. Let us first examine the various states a terminal can be in and the possible transitions that exist among the states. Fig. 10 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 here by X. The terminal retransmits its packet and repeats this process until its transmission is free of collision by other terminals. When the latter is true, then the packet will be successfully received at the repeater if and only if the repeater is not transmitting 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 will repeat itself 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. 10 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, and which we denote by $T_{A_1A_5}$. Assuming that the blocking probability B is uniform over time and independent of the state that the population of terminals is in**, we can write (referring again to the diagram in Fig. 10),

$$D_a = T_{A_1A_5} = T_{A_1A_2} + (\frac{1}{1-B} - 1)(T_{A_3A_4} + T_{A_4A_2}) \tag{19}$$

To estimate $T_{A_1A_2}$ and $T_{A_3A_4}$, we call upon previously published results. Slotted ALOHA with an infinite population has been thoroughly analyzed by Kleinrock and Lam [7]. With the channel input from the infinite population

*In the case M = 1, this condition is equivalent to the repeater's buffer being empty.

**The analysis of these multi-hop environments has been simplified by the basic assumption that the processes governing the inner-hop (repeater-to-station) are stochastically independent from those governing the terminal-to-repeater hop; in particular we assume that the process of offered input to a repeater is independent from the state of the repeater (active or empty) and vice versa.

modeled as an independent Poisson process with an average of λ packets/slot, letting the maximum retransmission delay be an integer number K of packet slots (the retransmission delay being uniformly distributed over the K slots*), and neglecting the propagation delay, the delay $D_{S\text{-}ALOHA}(\lambda)$ is then given by [7], [8]

$$D_{S\text{-}ALOHA}(\lambda) = \left[1 + \frac{E(K+1)}{2}\right] \tag{20}$$

where

$$E = \frac{1-q_n}{q_t}$$

$$q_n = \left[e^{-\frac{G}{k}} + \frac{G}{k} e^{-G}\right]^k e^{-\lambda}$$

$$q_t = \frac{e^{-\frac{G}{k}} - e^{-G}}{1 - e^{-G}} \left[e^{-\frac{G}{k}} + \frac{G}{k} e^{-G}\right]^{k-1} e^{-\lambda}$$

$$\lambda = G \frac{q_t}{q_t + 1 - q_n}$$

The normalized delay (in slots) is shown in Fig. 11 versus λ for various values of K. For each value of λ, we note that an optimum value of K, K_{opt}, can be selected so as to achieve minimum delay. The lower envelope of all delay cuves provides the throughput-delay performance of slotted ALOHA with infinite population. Fig. 12 shows K_{opt} versus λ.

---

*The geometric retransmission randomization can be used as well. However, it was shown [8] that the slotted ALOHA channel performance is dependent primarily upon the average value of the retransmission delay and quite insensitive to its exact distribution.

The model used by Kleinrock and Lam in determing packet delay can be represented by the diagram of Fig. 13, where the delay $D_{S\text{-}ALOHA}(\lambda)$ corresponds to $T_{A'_1A'_2}$. Comparing the two diagrams in question, we note the following Both include representation of slotted ALOHA channels with throughput λ. However while in the diagram of Fig. 13 the entire input to the channel is produced by the independent Poisson process with rate λ, in the diagram of Fig. 10, only the fraction s is produced by an indep[illegible]t process; the remaining fraction $(\lambda - s)$ is generated by the terminals in the blocked state. Regarding transmissions by these terminals as "new" input to the channel, and neglecting the effect of the correlation in traffic introduced here by the rescheduling protocol, we approximate $T_{A_1A_2}$ and $T_{A_4A_2}$ by

$$T_{A_1A_2} = T_{A_4A_2} = D_{S\text{-}ALOHA}(\lambda) \tag{21}$$

$T_{A_3A_4}$ is the average rescheduling delay and is simply given by

$$T_{A_3A_4} = \frac{K_{opt}(\lambda)}{2} \tag{22}$$

and thus an estimate of D_a is given by

$$D_a = \frac{1}{1 - B} D_{S\text{-}ALOHA}(\lambda) + \frac{B}{1 - B} \frac{K_{opt}}{2} \tag{23}$$

In Fig. 14 we plot the packet delay $D_a + D_n$ versus S for various values of N. We basically note that for $N \geq 3$, the throughput-delay performance degrades slightly as N increases. As for the $N = 2$ throughput-delay curve, we note the following. First, the system capacity for $N = 2$ is, as indicated above, roughly equal to the one achieved with $N = 5$. Secondly, the curve exhibits the lowest delay for very small S, but as S increases to reach its maximum value, the curve intercepts both the N = 3 and N = 5 delay curves. To explain this behavior, we note that for a given throughput S, D_n is smaller for smaller N (see Fig. 4) while D_a is larger, due to the fact that for smaller N a larger value of λ is required to achieve that throughput, of course provided that the latter does not exceed the system capacity (see Fig. 8).

3.1.5 Dynamic Control for Improved Performance

Assuming that each repeater knows exactly the state of the system in each slot*, then one can improve the performance by maximizing the instantaneous throughput, given by $P_s(i)$ in Eq. (2), with respect to the transmission parameter p. Given that $n^t = i$, $P_s(i)$ is maximized for $p = 1/i$.

We illustrate the gain obtained via the dynamic control by plotting in Figures 15 and 16 the network delay D_n and the blocking probability B respectively. We also plot, in Fig. 17, the system capacity versus N for the two-hop configuration, and in Fig. 18, the throughput-delay curves for various values of N. One point is worth noting here. For N > 3, the system capacity is attained for optimum values of λ smaller than 1/e; as λ

---

*In practical situations, the assumption that each repeater knows the exact current state of the system clearly does not hold. The repeaters have no means of communicating among themselves other than the channel itself. However, each repeater may individually estimate the system state by observing the channel outcome over some period of time, and apply a control action based upon the estimate. In the context of a one-hop slotted ALOHA environment, Lam and Kleinrock [9] give some heuristic control-estimation algorithms which prove to be very satisfactory. The results obtained here assuming full knowledge of the system state will then represent the ultimate performance; a bound on the performance obtained via heuristic estimation algorithms.

increases beyond its optimum value the system throughput decreases, as shown in Fig. 19, and reaches the corresponding system capacity in the uncontrolled case. This explains the dashed portions of the $N = 5$ and $N = 10$ delay curves in Fig. 18.

3.2 The Multi-Buffer Case

3.2.1 Analysis

With $M > 1$, the state of the system is described by the vector $\underline{n}^t = (n_1^t, n_2^t, \ldots, n_N^t)$ where n_i^t $(0 \leq n_i^t \leq M)$ denotes the number of packets buffered at repeater i at time t. Let S denote the state space; we have

$$S = \{(n_1, n_2, \ldots, n_N)/0 \leq n_i \leq M, \forall i = 1, 2, \ldots, N\}$$

We are seeking the probability of the one-step transition from state $\underline{m} = (m_1, m_2, \ldots, m_N)$ to state $\underline{n} = (n_1, n_2, \ldots, n_N)$, which we denote by $Pr[\underline{n} / \underline{m}]$. We distinguish the following cases.

First, in any slot, the amplitude of change in n_i^t cannot exceed one, and there can be at most one departure. Therefore we have

(i) If there exists an index i such that $|m_i - n_i| > 1$ or if there exist i, j, $i \neq j$, such that $n_i = m_i - 1$ and $n_j = m_j - 1$ then $Pr\{\underline{n} / \underline{m}\} = 0$

If the above conditions are not satisfied, then either a successful transmission took place, or no packet was successfully transmitted. Thus, we have

(ii) If there exists an index i_0 such that $m_{i_0} = n_{i_0} + 1$ (indicating a successful transmission by repeater i_0) then

$$Pr\{\underline{n} / \underline{m}\} = p \prod_{j \neq i_0} (1 - p)^{x_j} \prod_{\ell \neq i_0} [\lambda \xi_\ell^- + (1 - \lambda + \lambda \zeta_\ell) \xi_\ell] \qquad (24$$

where

$$\chi_j = \begin{cases} 1 & \text{if } m_j > 0 \\ 0 & \text{if } m_j = 0 \end{cases}$$

$$\xi_j^- = \begin{cases} 1 & \text{if } m_j = n_j - 1 \\ 0 & \text{if } m_j \geq n_j \end{cases}$$

$$\xi_j = \begin{cases} 1 & \text{if } m_j = n_j \\ 0 & \text{if } m_j \neq n_j \end{cases}$$

$$\zeta_j = \begin{cases} 1 & \text{if } n_j = M \\ 0 & \text{if } n_j < M \end{cases}$$

The term $p \prod_{j \neq i_0} (1 - p)^{\chi_j}$ represents the probability that i_0 is the only transmitting repeater among all active ones. The second product term represents the probability of all changes (that is, the presence or absence of arrivals) which occured at the remaining queues in the current slot. The indicator ζ_ℓ accounts for the queue lengths being finite; an arrival which finds the queue full is rejected.

(iii) Otherwise no successful transmission took place. Letting $I_s = \{j \mid m_j = n_j\}$ we have

$$\Pr\{\underline{n} / \underline{m}\} = \prod_{j \varepsilon I_s} [p\chi_j + (1 - p)^{\chi_j}(1 - \lambda + \lambda\zeta_j)]$$

$$- \sum_{j \varepsilon I_s} p\chi_j \prod_{\substack{k \varepsilon I_s \\ k \neq j}} (1 - p)^{\chi_k}(1 - \lambda + \lambda\zeta_k) \prod_{j \notin I_s} (1 - p)^{\chi_j}\lambda \qquad (25)$$

where χ_j and ζ_j are as definec in (ii) 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 that slot. Thus, the number of packets queued at repeater $j \epsilon I_s$ remains unchanged with probability $p\chi_j + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)$, provided that any transmission (represented by the term $p\chi_j$) is unsuccessful. Since the repeaters are all independent, the probability of the event $\{m_j = n_j\}$ for all $j \epsilon I_s$ is then given by the expression in the first bracket, in which the summation

$$\left[\sum_{j \epsilon I_s} p\chi_j \prod_{\substack{k \epsilon I_s \\ k \neq j}} (1-p)^{\chi_k}(1-\lambda+\lambda\zeta_k)\right]$$

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.

The transition matrix P is computed numerically. Let $\Pi = \{\pi_{\underline{n}}\}_{\underline{n}\epsilon S}$ be the stationary distribution of $\underline{n}^t$. Π is evaluated by iteratively solving the system $\Pi = \Pi P$. Let q_i denote the queue length at repeater i. The marginal distribution of q_i is given by

$$Pr\{q_i = k\} = \sum_{\{\underline{n}\epsilon S | n_i = k\}} \pi_{\underline{n}} \tag{26}$$

The average queue length is then given by

$$\overline{q} = \sum_{k=0}^{M} k \, Pr\{q_i = k\} \tag{27}$$

The blocking probabilities α and β are expressed as

$$\alpha = \Pr\{q = M\} (1 - p) \tag{28}$$

$$\beta = [1 - \Pr\{q = o\}] p \tag{29}$$

The network throughput (in this symmetric case) is given by

$$S = N \lambda(1 - \alpha - \beta) \tag{30}$$

and by Little's result, the network delay is computed by

$$D_n = \frac{\overline{q}}{\lambda(1 - \alpha - \beta)} \tag{31}$$

3.2.3 Numerical Results

In Fig. 20 we plot on the (S,D_n) plane the constant λ contours (varying p) for 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 does not exactly correspond to the value of p which yields minimum blocking (and therefore maximum throughput). However, the probability of blocking at minimum delay is not significantly different from the minimum blocking achievable! The effect M has on network delay is shown in Fig. 21 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. We also note a slight decrease in network capacity. The effect M has on the probability of blocking is shown in Fig. 22 where we plot the minimum blocking as a function of S. Note the (slight) decrease achieved by going from M=1 to M=2. M=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? In Figures 23 and 24, we plot D versus S for N=2 and N=3, respectively, and various values of M. Again, we only note a slight improvement in performance by going from M=1 to M=2. No further significant improvement is gained beyond M=2. The increase in network capacity (observed particularly with N=2) is obviously due to the decrease in B.

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 the (α,β) plane on which we plot the constant λ contours. When p is small, α predominates: $\alpha>\beta$; As p increases, the inequality reverses. The locus of optima is displayed in Fig. 25 for various values of N. 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. This effect is due to the fact that, for the same throughput, the optimum p decreases as N increases in order to prevent conflict among a larger number of contending users. Is the system then storage bound when N is large, say 10 for example? It can be argued 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. 3) while D_a decreases as does S/N (for N=10, 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: $D_n \approx 10$, $B \approx 0.38$ and $D_a \approx 2.5$ yielding $D \approx 12.5$. By taking B=0, we can decrease D_a to 1.15 providing thus a lower bound on D of 11.15, a rather unimportant improvement. Moreover, due to the queueing effect, D_n increases with larger M.

4. Conclusion

The above analysis has provided a means by which the throughput and delay performance measures are evaluated for slotted ALOHA packet radio star-networks. In particular, we have shown the effect on the 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 lead us to believe that the system is channel bound (the processing time at repeaters assumed negligible); a slight improvement may be gained by increasing the buffer size to M=2, but no significant improvement is obtained beyond that. Moreover we considered a dynamically controlled transmission protocol which provides a tremendous improvement in system performance.

In two forthcoming notes [6,10] we shall report on the results of a similar study performed on fully-connected networks. Both slotted ALOHA and CSMA will be considered.

References

[1] F. Tobagi, "Performance Analysis of Packet Radio Communications Systems," Proceedings of the IEEE National Telecommunications Conference, December 1977, pp. 12:6-1 12:6-7.

[2] Kleinrock, L. and F. Tobagi, "Packet Switching in Radio Channels: Part I -Carrier Sense Multiple Access Modes and their Throughput Delay Characteristics," IEEE Transactions on Communications, vol. COM-23, No. 12, December 1975, pp. 1400-1416.

[3] Tobagi, F. 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 Transactions on Communications, Vol. COM-23, Vol. 12, December 1975, pp. 1417-1433.

[4] Tobagi, F.A. and L. Kleinrock, "Packet Switching in Radio Channels: Part III - Polling and (Dynamic) Split-channel Reservation Multiple Access," IEEE Transactions on Communications, Vol. COM-24, No. 8, August 1976, pp. 832-845.

[5] Gitman, I., "On the Capacity of Slotted ALOHA Networks and Some Design Problems," IEEE Transactions on Communications, Vol. COM-23 No. 3, March 1975, pp. 305-317.

[6] F. Tobagi, "On the Performance Analysis of Multihop Packet Radio Systems: Part III - Fully Connected Configurations Employing Slotted ALOHA," Packet Radio Temporary Note #248, SRI International Menlo Park, California,

[7] Kleinrock, L.and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Transactions on Communications, Vol. COM-23, pp. 410-423, April 1970.

[8] Lam, S., "Packet Switching in a Multiaccess Broadcast Channel with Applications to Satellite Communication in a Computer Network," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-Eng-7429. April 1974 (also published as Ph.D. Dissertation).

[9] Lam, S. and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," IEEE Transactions on Communications, Vol. COM-23, pp. 891-904, September 1975.

[10] Tobagi, F., "On the Performance Analysis of Multihop Packet Radio Systems: Part IV - Fully Connected Configurations Employing CSMA," Packet Radio Temporary Note #249, SRI International, Menlo Park, California.

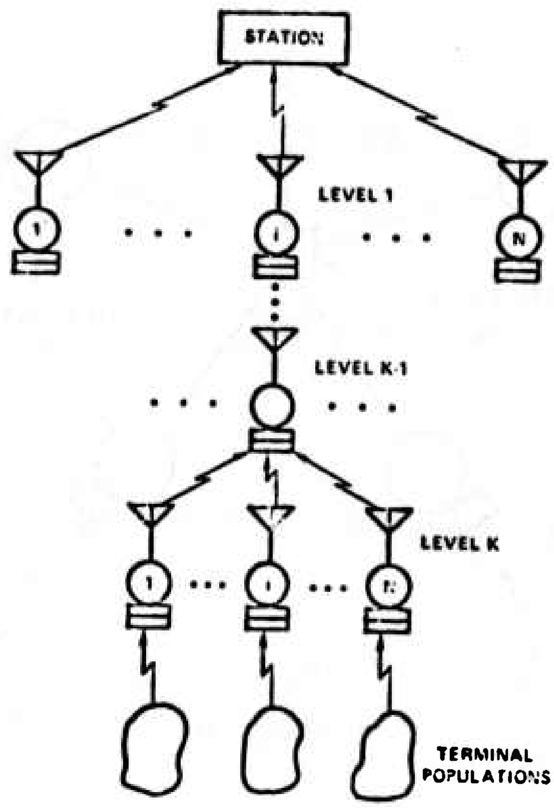


Figure 1 Tree-Structured Networks

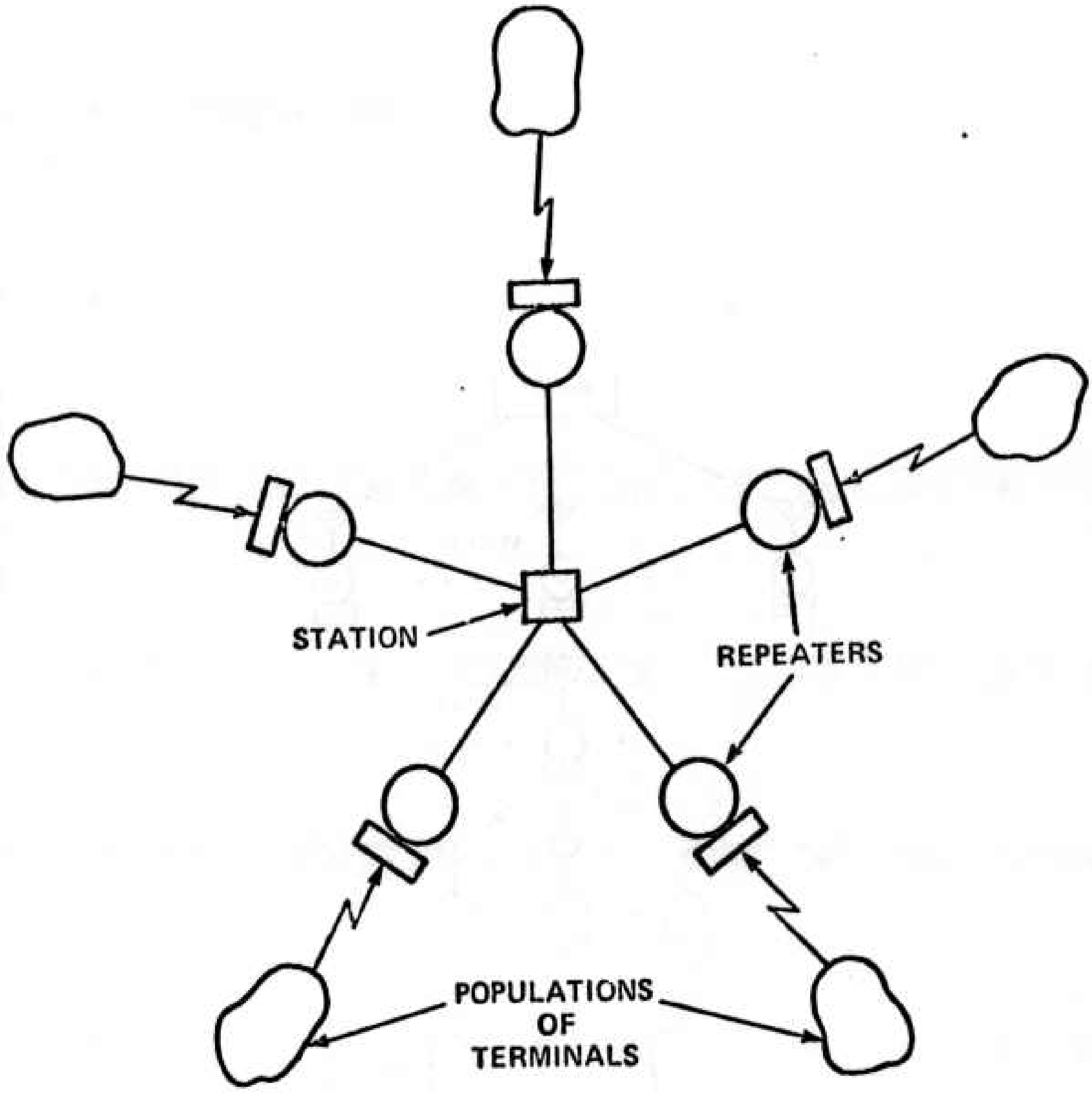


Figure 2 A Two-Hop Star Configuration

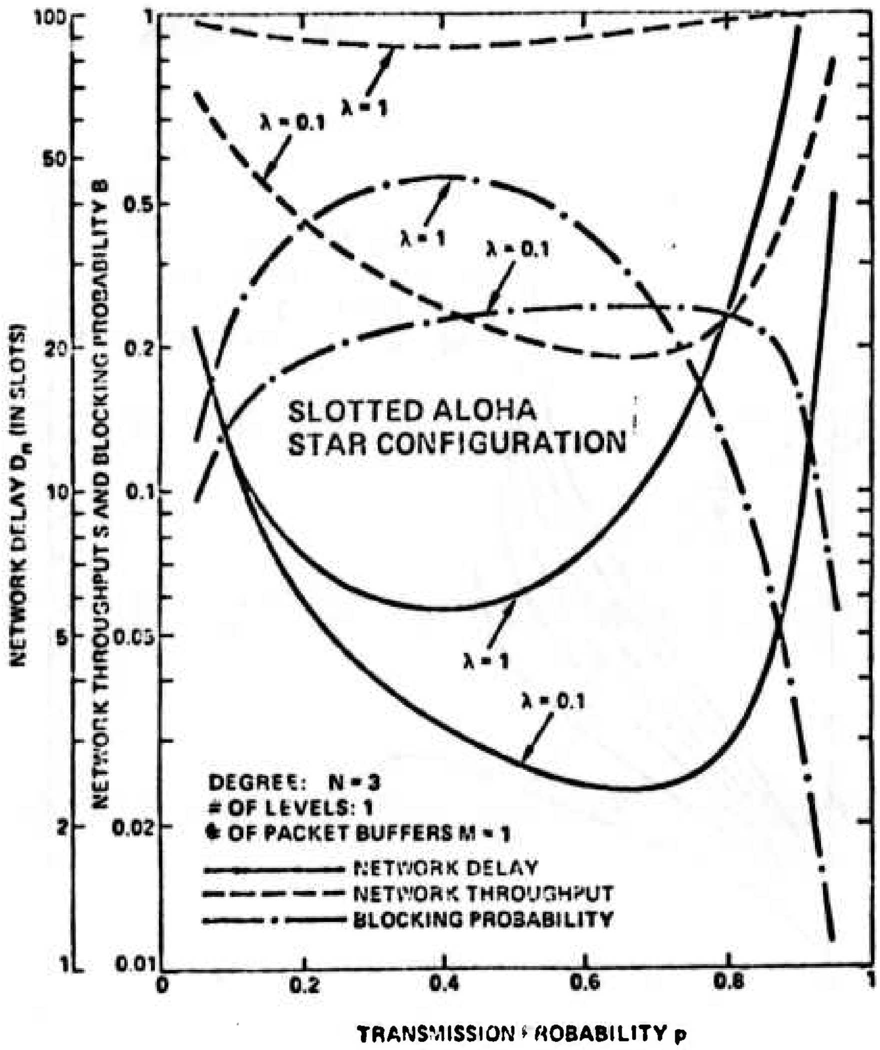


Figure 3 Slotted ALOHA Star Configuration: Network Delay, Throughput and Probability of Blocking versus p.

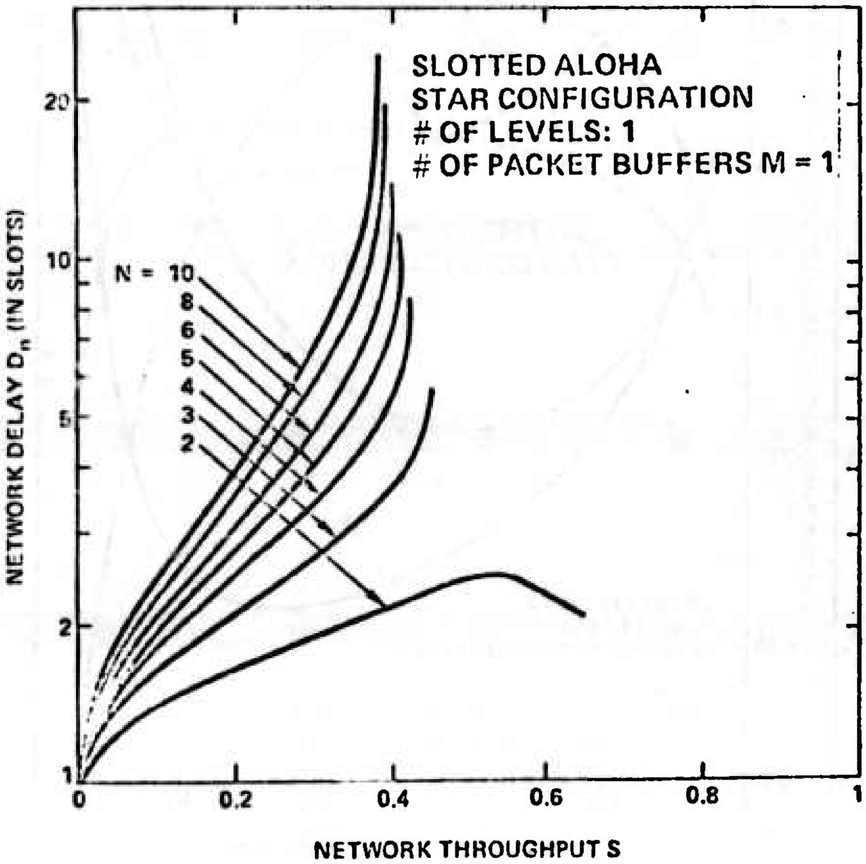


Figure 4 Slotted ALOHA Star Configuration: Optimum (Network) Throughput Delay Curves.

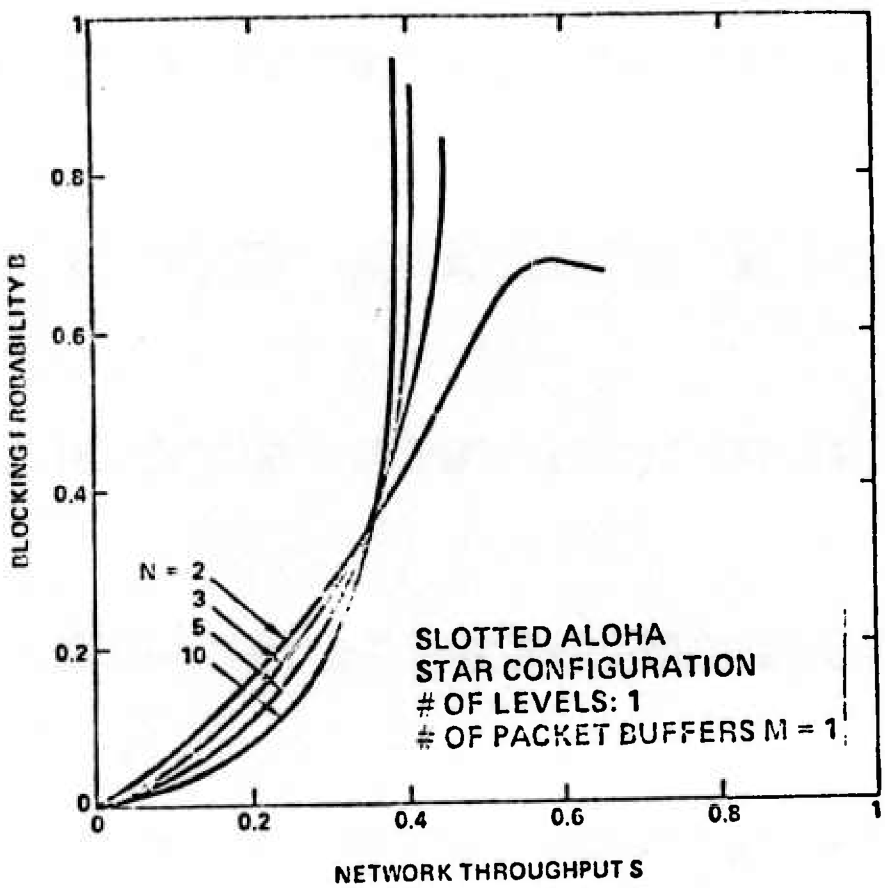


Figure 5 Slotted ALOHA Star Configuration: Minimum Blocking versus Throughput.

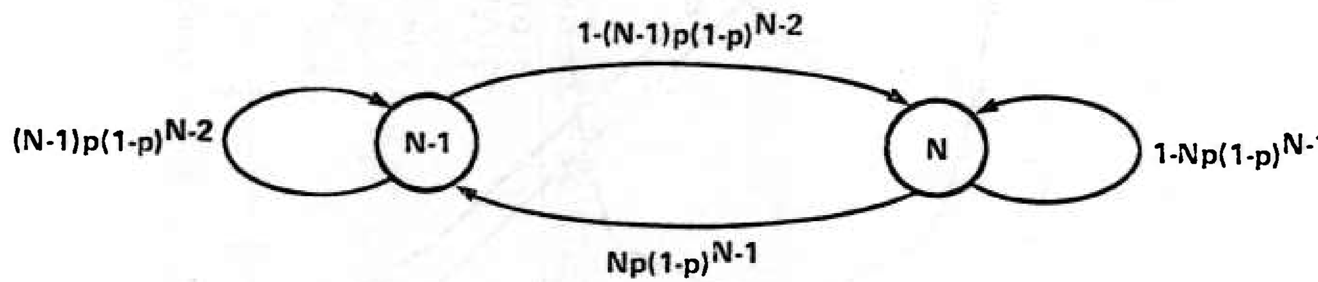


Figure 6 Markov Chain at Saturation ($\lambda = 1$).

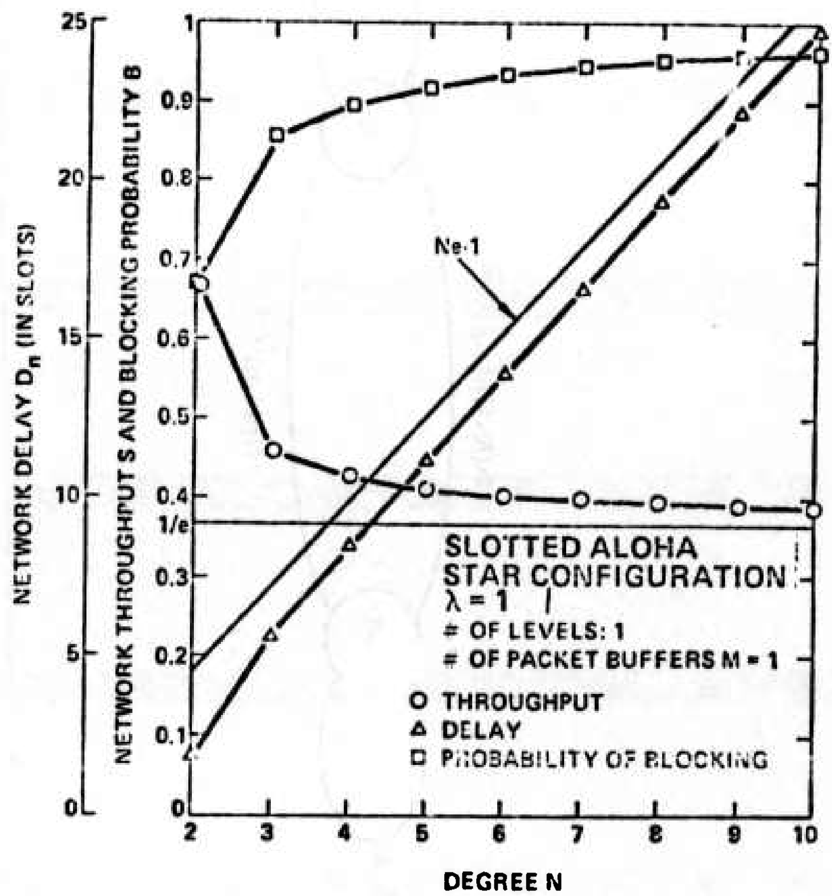


Figure 7 Slotted ALOHA Star Configuration: Performance at Saturation (λ = 1).

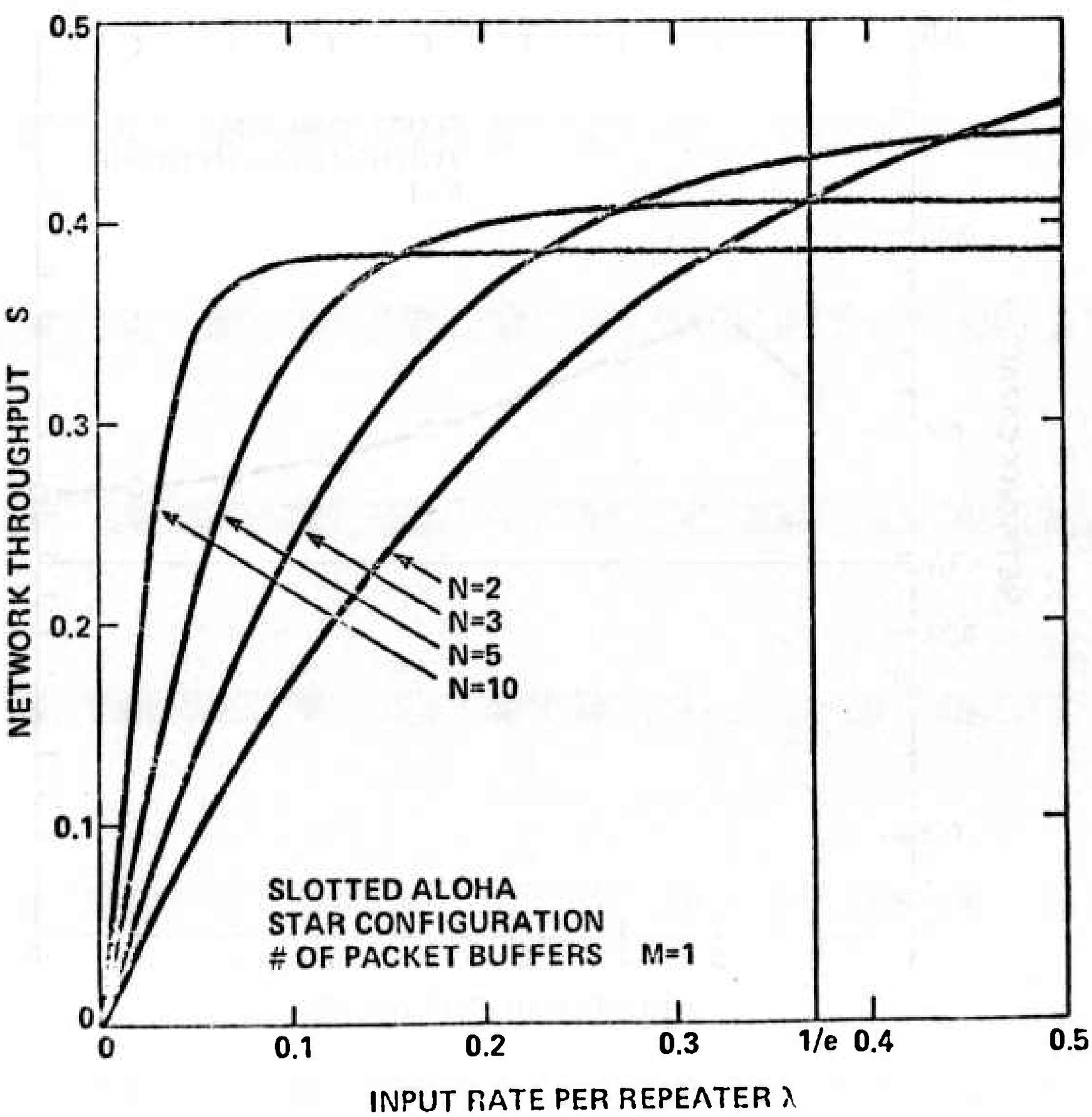


Figure 8 Slotted ALOHA Star Configuration: Throughput versus λ.

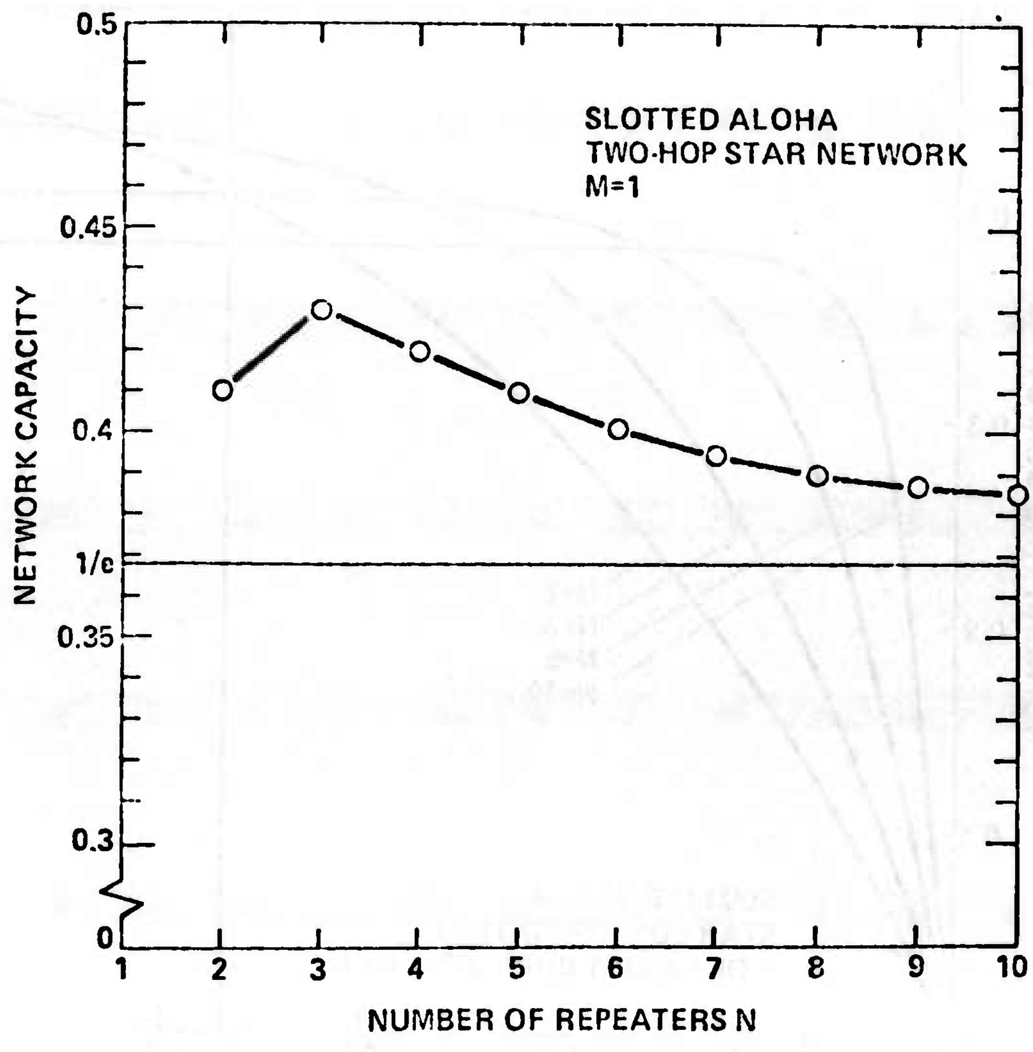


Figure 9 Slotted ALOHA Two-Hop Star Network: Network Capacity versus N.

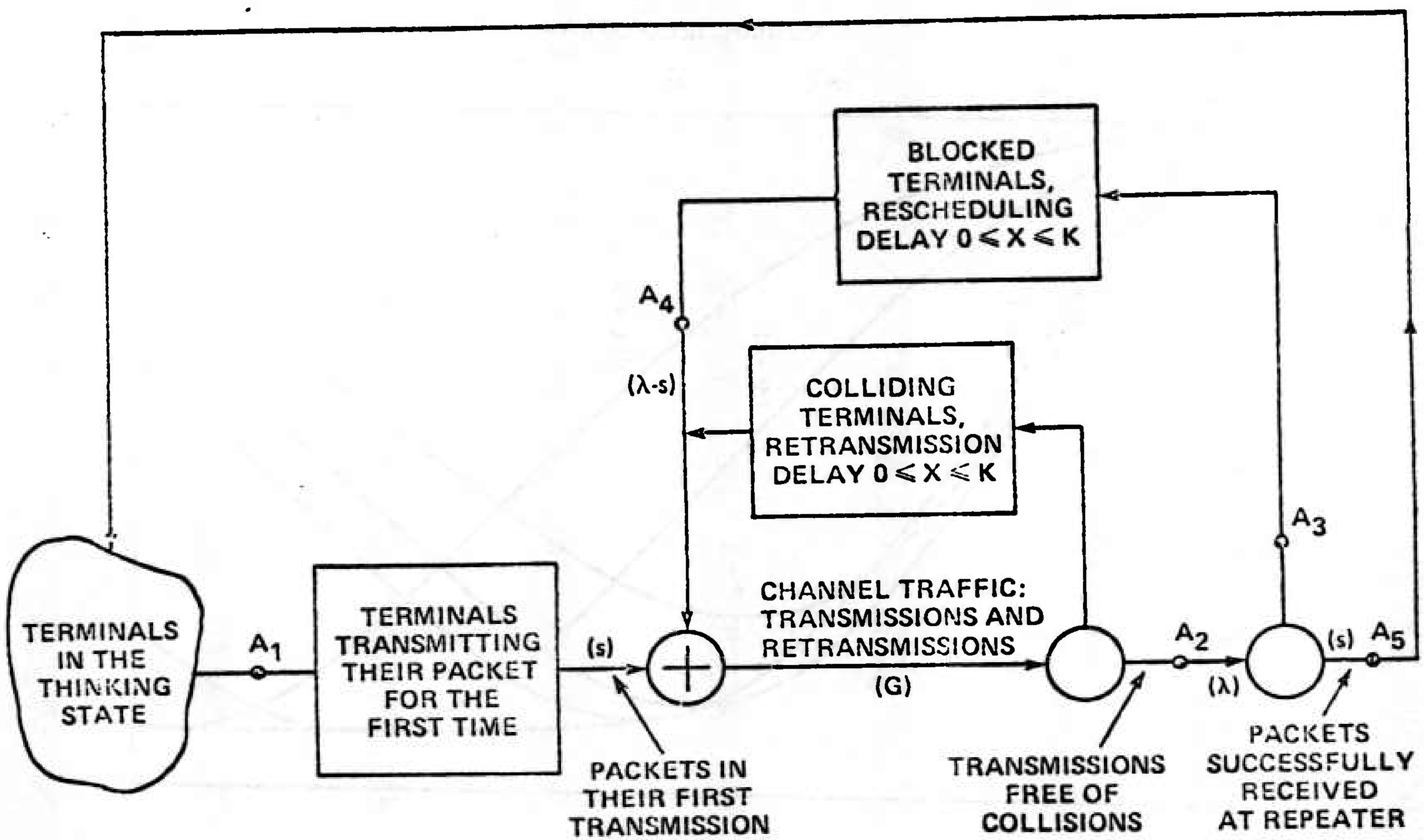


Figure 10 State Diagram for a Population of Terminals.

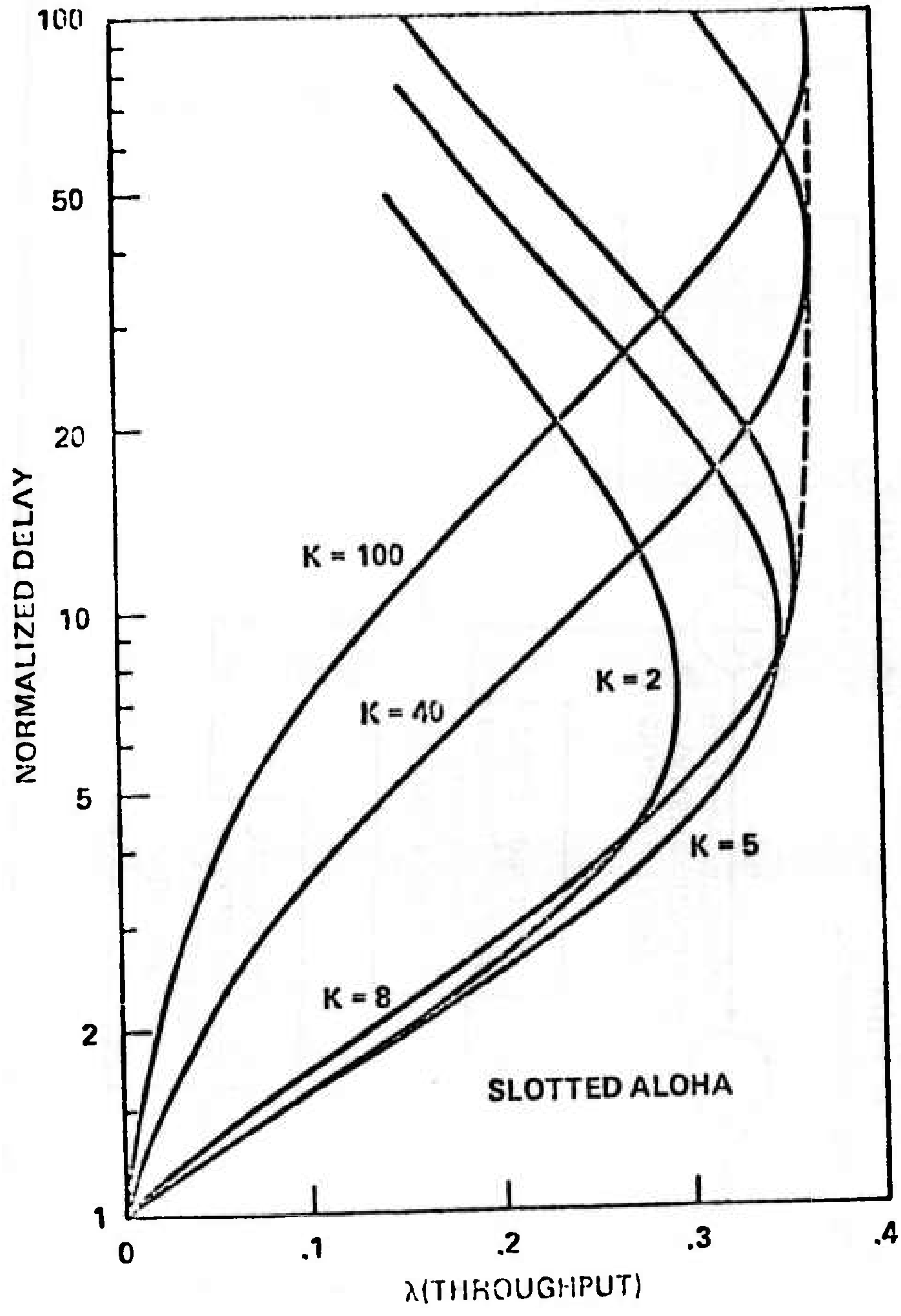


Figure 11 Delay in Slotted ALOHA Channels with Infinite Population.

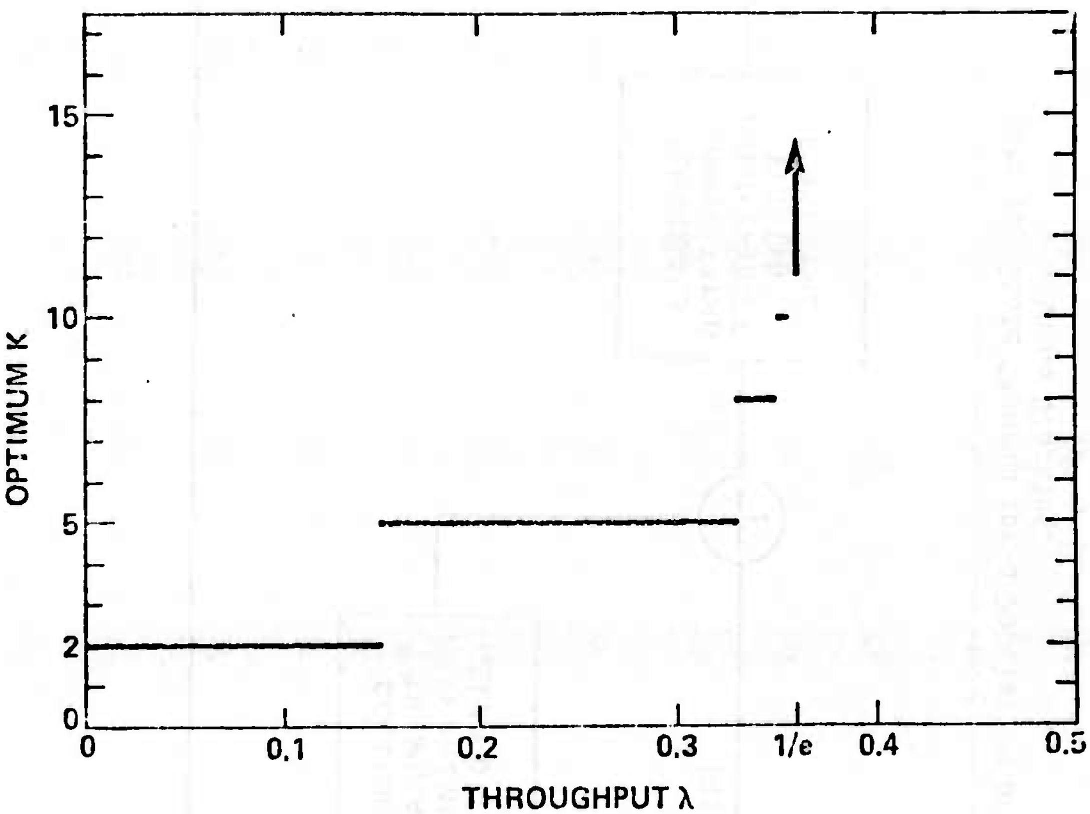


Figure 12 Optimum K versus Throughput

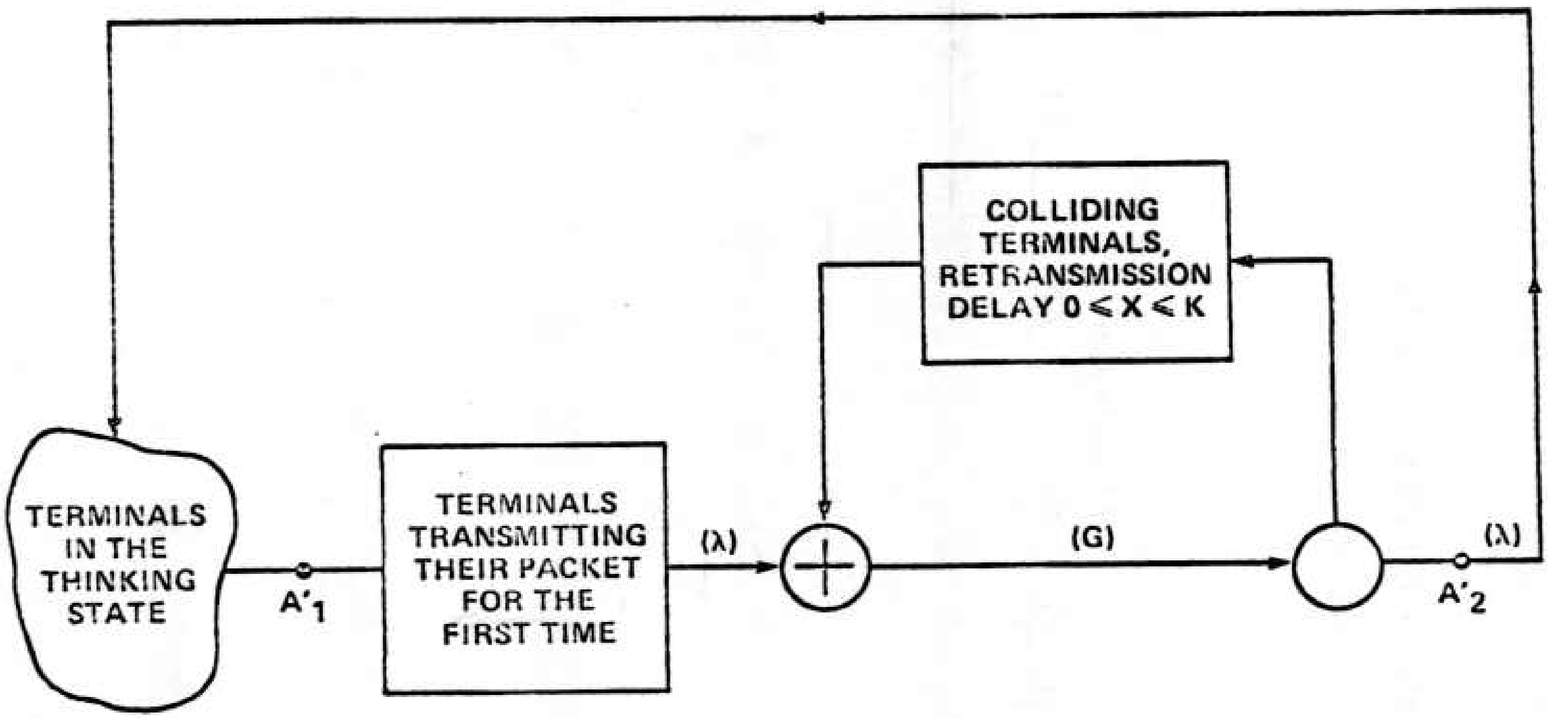


Figure 13 State Diagram for a Population of Terminals with no Blocking.

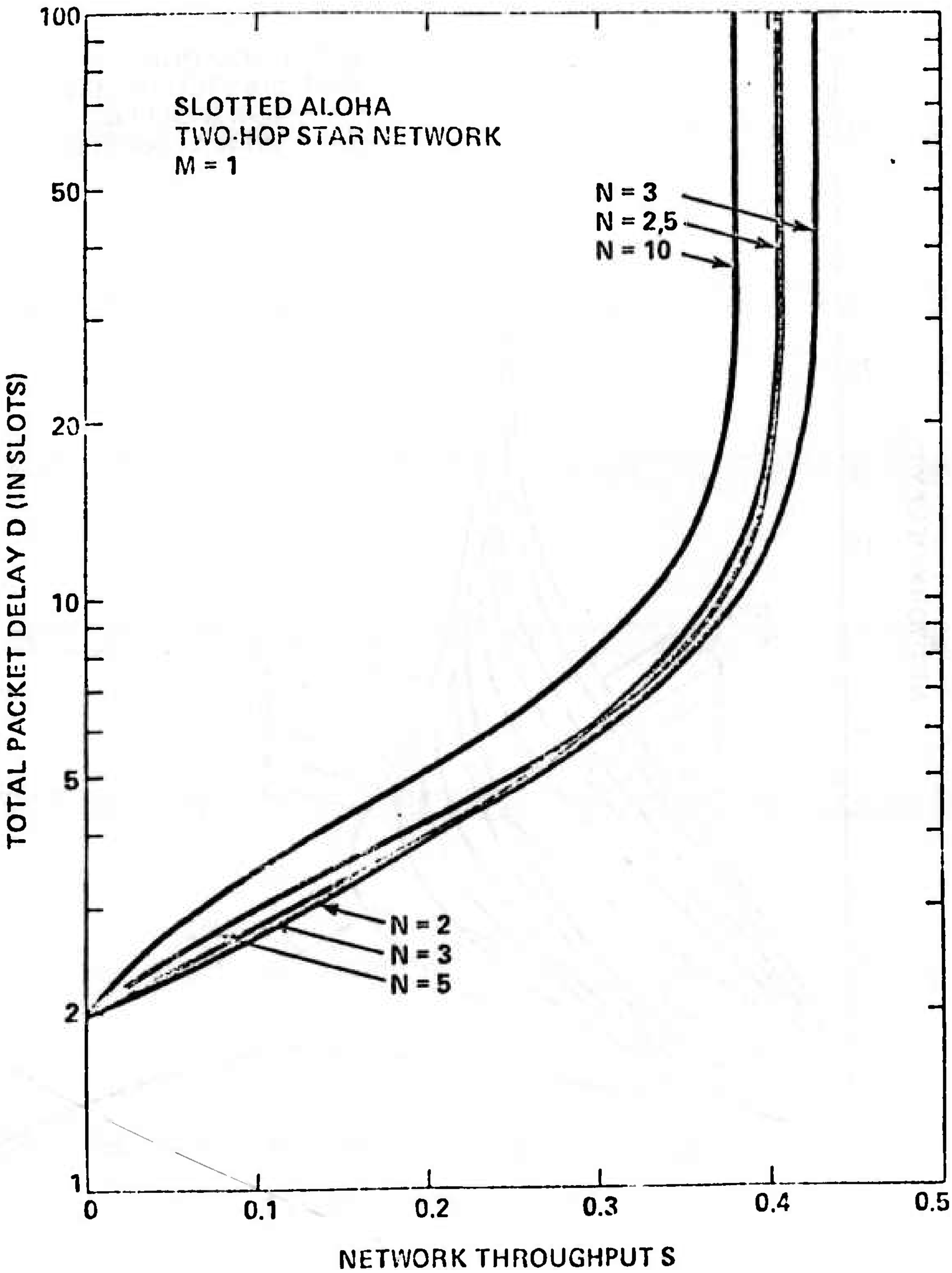


Figure 14 Throughput-Delay Tradeoff in Two-Hop Slotted ALOHA Star Networks

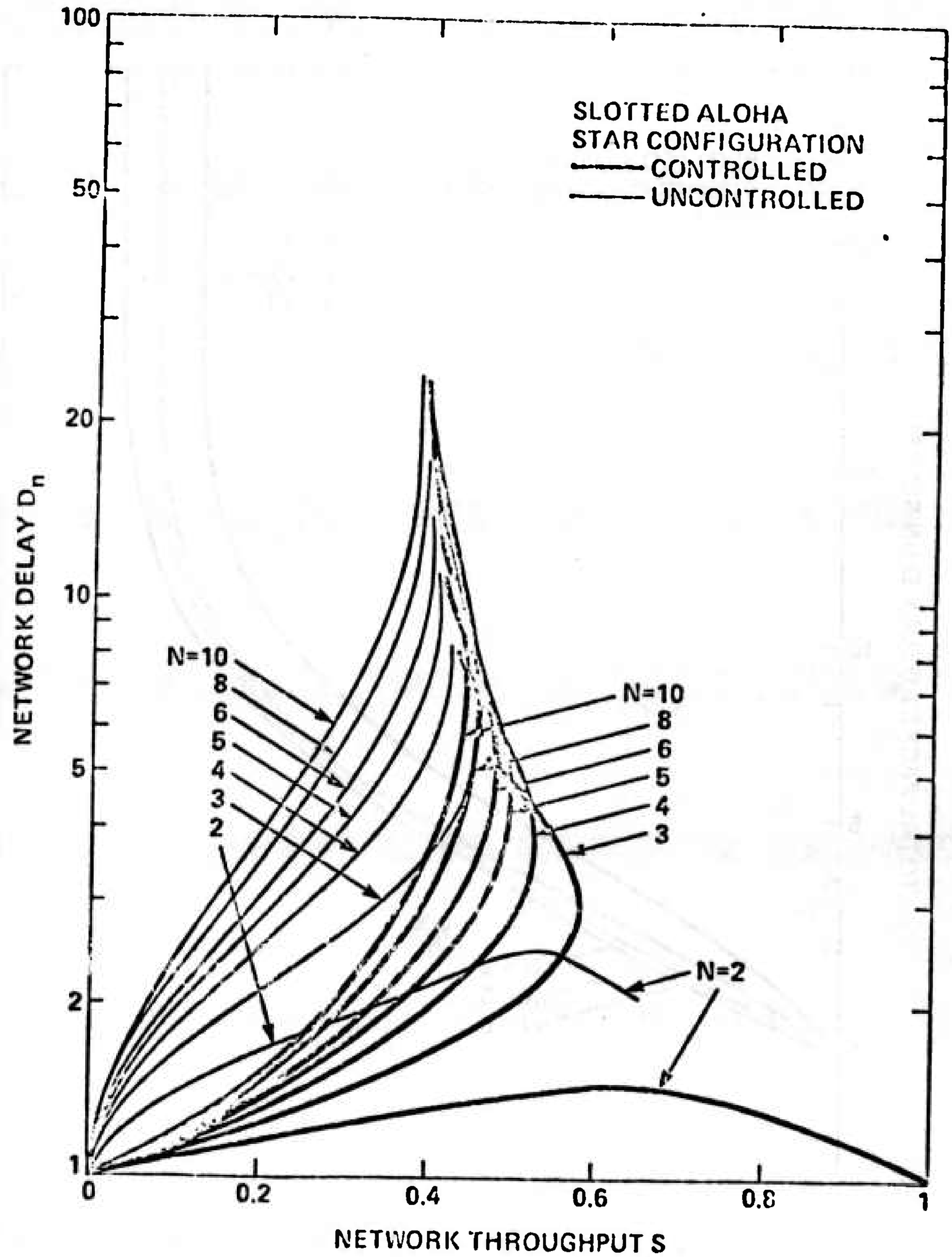


Figure 15 Slotted ALOHA Star Configuration: Optimum (Network) Throughput-Delay Curves for Controlled and Uncontrolled Systems.

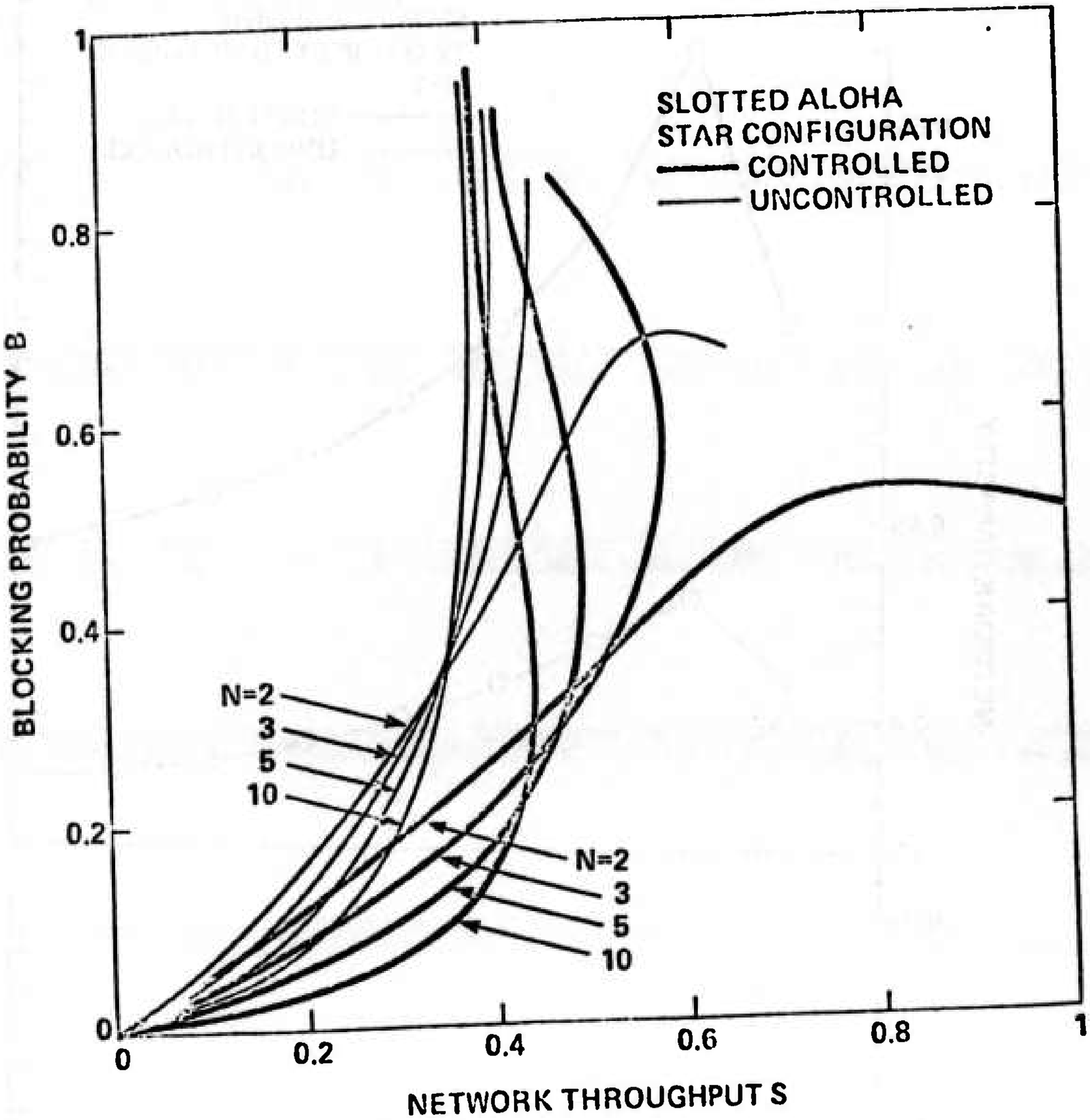


Figure 16 Slotted ALOHA Star Configuration: Minimum Blocking versus S for Controlled and Uncontrolled Systems.

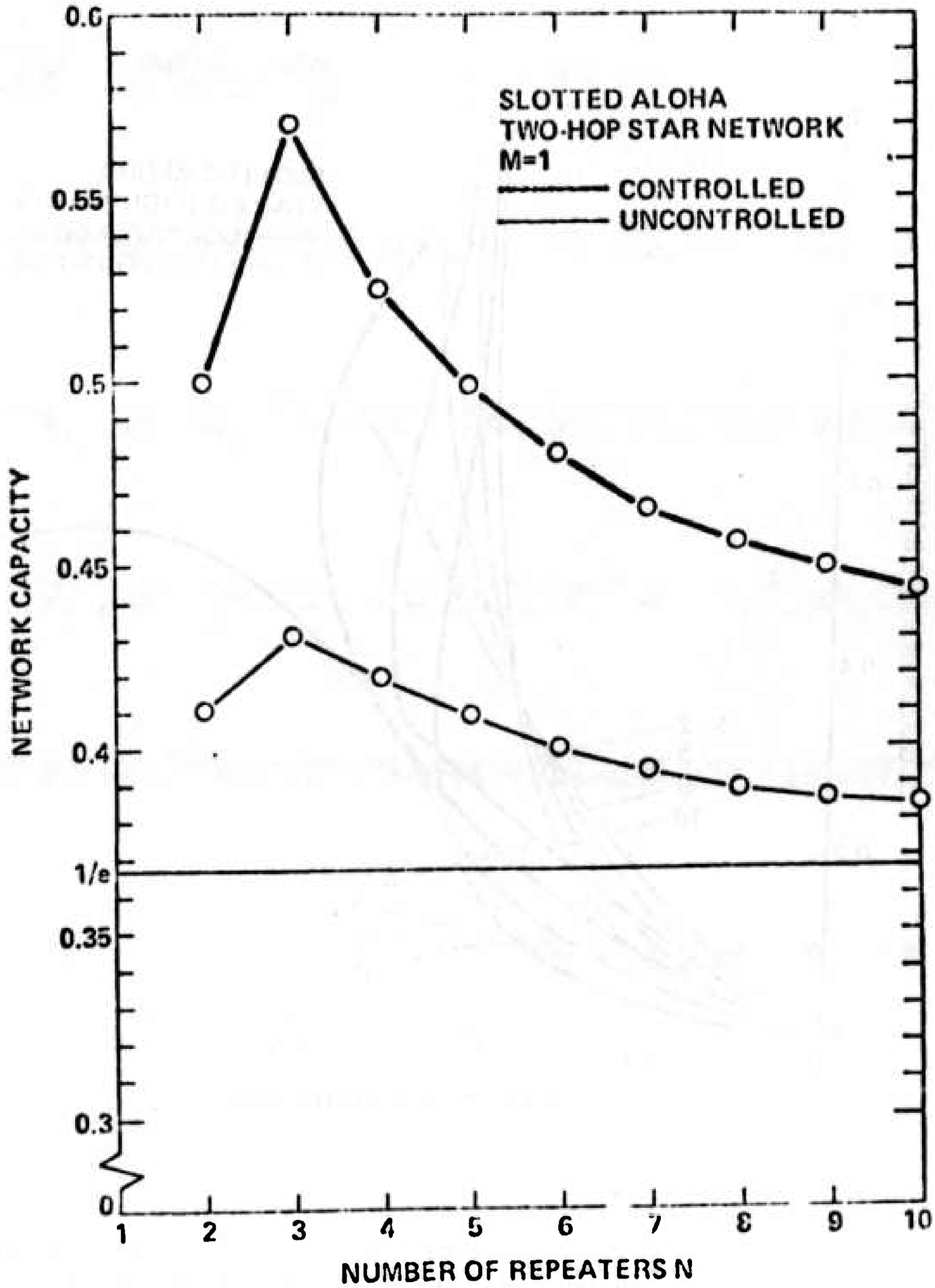


Figure 17 Slotted ALOHA Two-Hop Star Network: Network Capacity versus N.

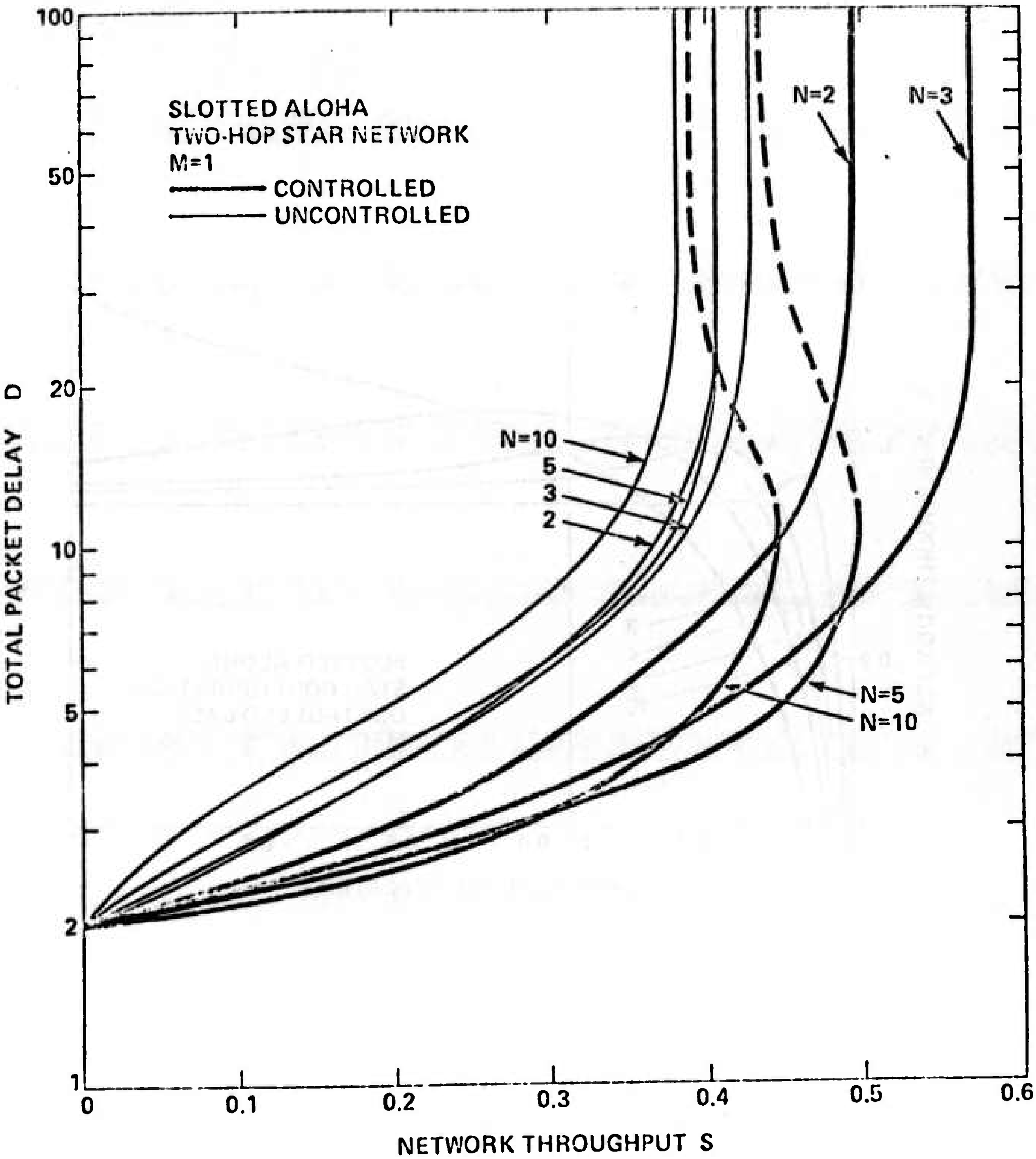


Figure 18 Throughput-Delay Tradeoff in Controlled Two-Hop Slotted ALOHA Star Networks.

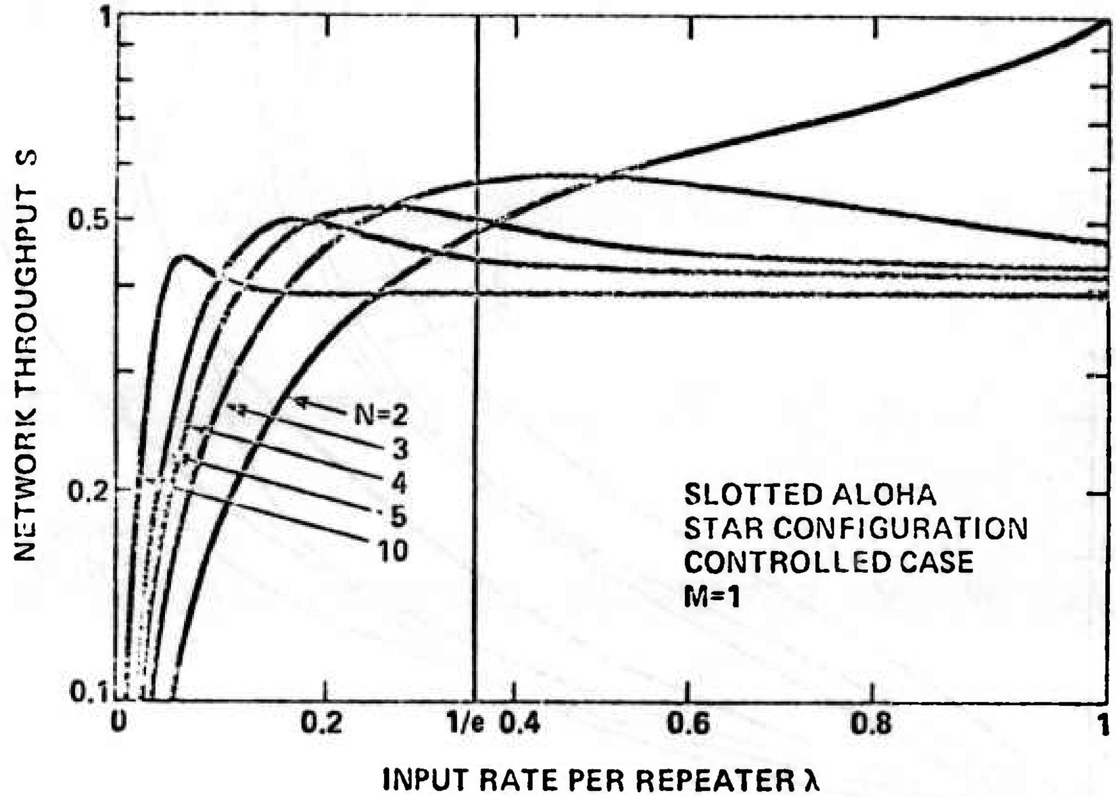


Figure 19 Controlled Slotted ALOHA Star Configurations: Throughput versus λ.

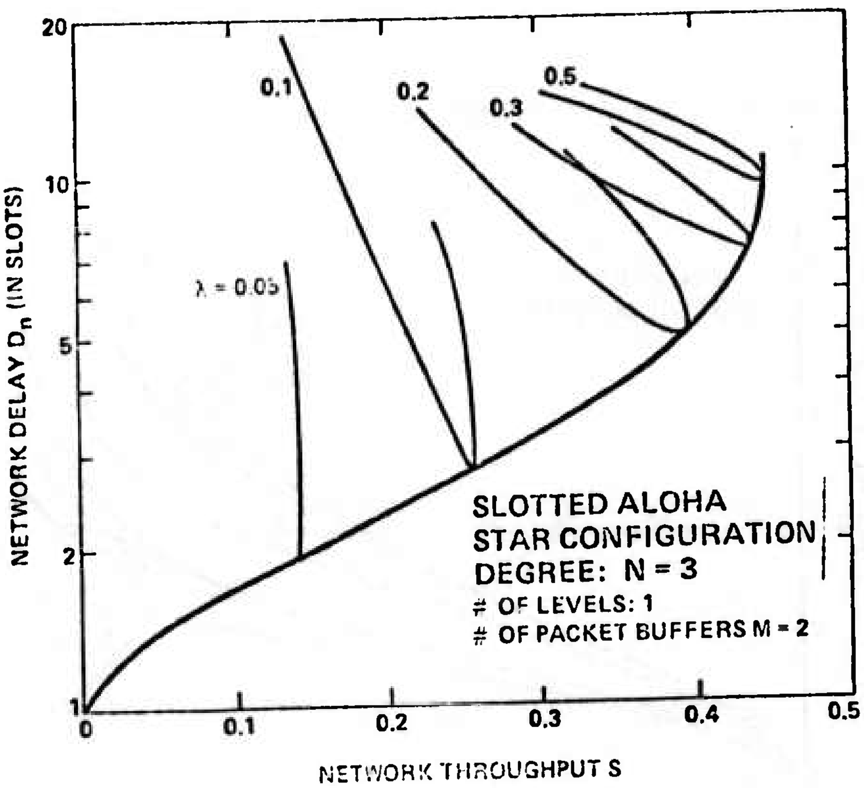


Figure 20 Slotted ALOHA Star Configuration: Delay versus Throughput with $M > 1$.

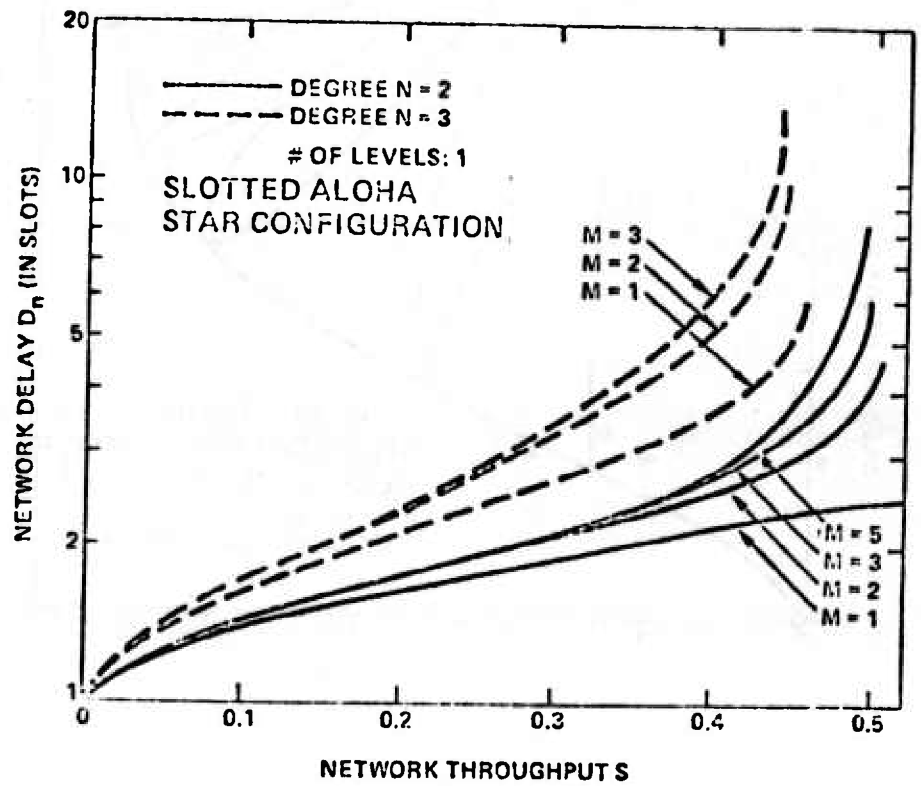


Figure 21 Slotted ALOHA Star Configuration: Minimum Delay versus S for Various Values of M.

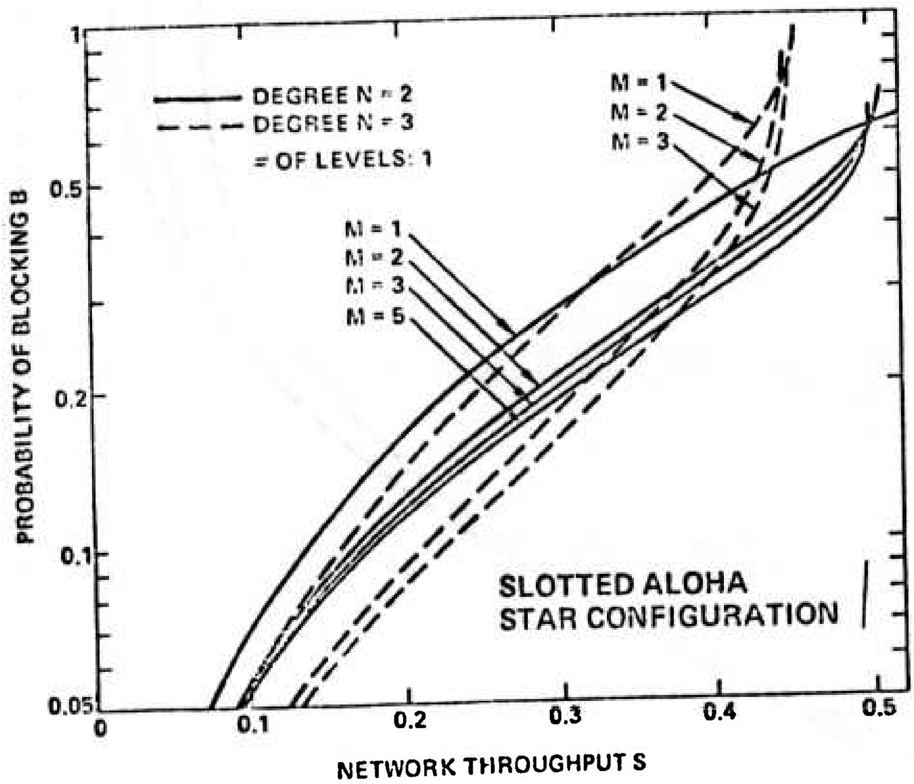


Figure 22 Slotted ALOHA Star Configuration: Minimum Blocking versus S for Various Values of M.

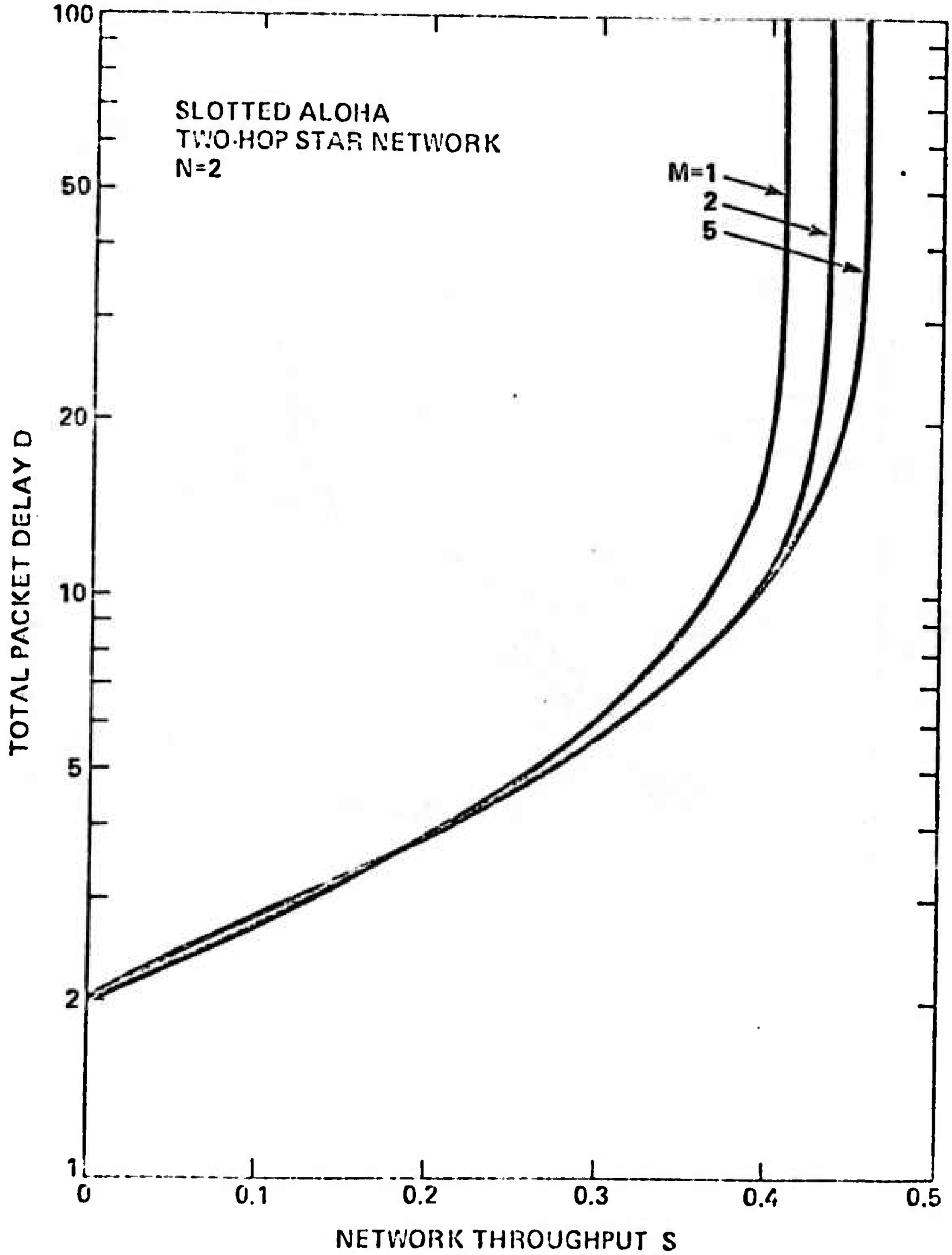


Figure 23 Two-Hop Slotted ALOHA Star Networks: Total Packet Delay versus S for N = 2 and M $\geq$ 1.

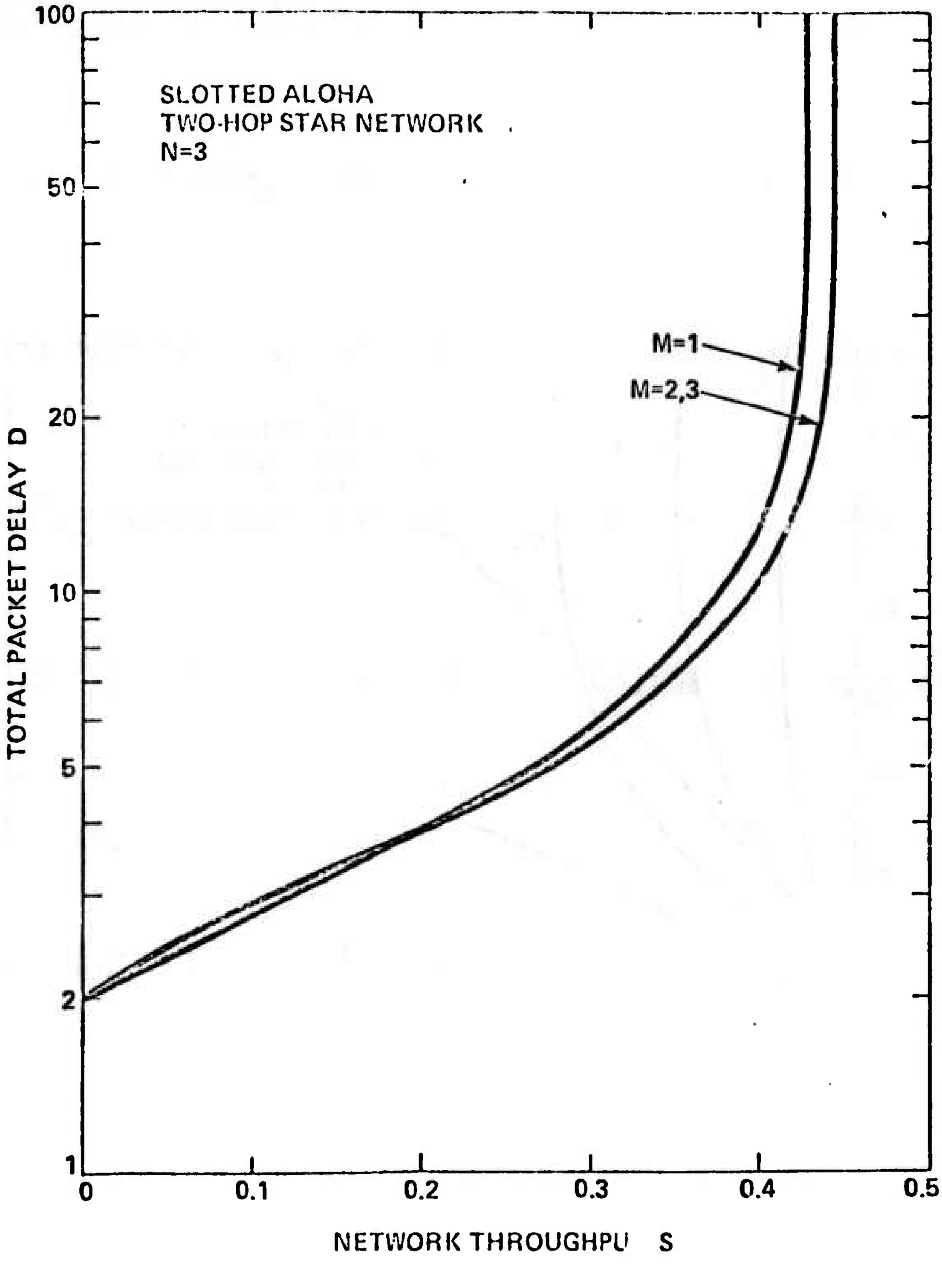


Figure 24 Two-Hop Slotted ALOHA Star Networks: Total Packet Delay versus S for N = 3 and $M \geq 1$.

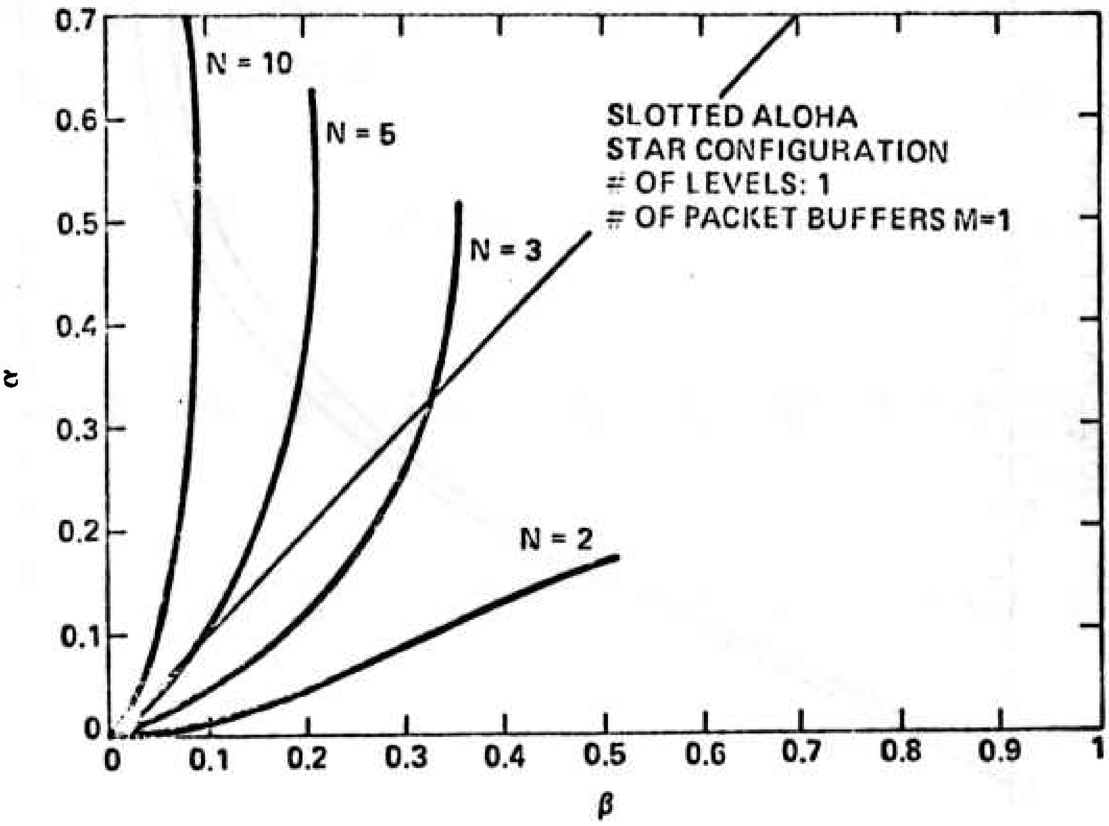


Figure 25 Slotted ALOHA Star Configuration: α versus β at Minimum Blocking.

PRTN #248

F. Tobagi
UCLA
January 13, 197?

On the Performance Analysis of Multihop Packet Radio Systems:
Part III - Fully Connected Configurations Employing Slotted ALOHA

I. Introduction

In part II of this series [1], we analyzed the so-called star configuration in which each repeater is in line-of-sight and within range of only the station. In this note we consider the fully connected (FC) network configuration in which all repeaters are within range and in line-of-sight of each other and of the station. (See Fig. 1) With each repeater is associated a population of terminals generating traffic which is destined to the station. The main difference that exists between this and the star configuration is that in the fully connected case an arrival to a repeater in a slot will not be successfully received if any of the repeaters is actively transmitting in that slot. Each repeater is provided with a finite storage capacity which can accomodate exactly one packet. No consideration will be made here of storage capacity greater than one since the results of Part II have shown that the system is mostly channel bound and not storage bound, and since it is even more so with fully connected configurations as illustrated below. Definitions, system assumptions, and the notation used are identical to those given in Part II; they will not be repeated here. The basic performance measures sought are the system capacity and the throughput-delay tradeoff. The results will be compared to those obtained with the star configuration.

2. Transmission Protocols

Just as in Part II, the time axis is assumed to be universal and slotted into segments whose duration is equal to the transmission time of a packet. All devices are synchronized and start their packet transmission at the beginning of a slot. When its buffer is non-empty, a repeater transmits its packet in a slot with probability p. When the packet transport is successful, the packet buffer is freed; otherwise the packet remains in the repeater's buffer and thus incurs a retransmission delay geometrically distributed with mean 1/p. In fact with this protocol even the first transmission of a newly received packet (at the repeater) incurs a geometrically distributed delay following its reception with mean 1/p. We shall refer to this transmission protocol as the delayed-first-transmission (DFT) protocol. It is precisely the DFT protocol that was considered in the analysis of star-configurations in Part II.

A slight variation of the above transmission protocol is also considered here, which consists of transmitting (with probability one) a newly received packet immediately following its reception. 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. The motivation in considering it is simply an expected decrease in packet network delay due to the avoidance of an initial delay at the first transmission of the packet. Numerical results will be discussed in section 4 below.

3. Analysis of Fully Connected Configurations

3.1. DFT Protocol

With single-packet buffers, the state of the system in a slot is entirely described by the number of active repeaters. We let n^t denote this number in slot t. n^t is a Markov chain with transition matrix P whose $(i,j)^{th}$ element is given by

$$p_{ij} = \begin{cases} 0 & j<i-1 \\ P_s(i) & j=i-1 \\ (1-p)^i (1-\lambda)^{N-i} + [1-(1-p)^i-P_s(i)] & j=i \\ (1-p)^i \binom{N-i}{j-i} \lambda^{j-i} (1-\lambda)^{N-j} & j>i \end{cases} \tag{1}$$

where $P_s(i)$ is the probability of a successful transmission, given i active repeaters, and is expressed as

$$P_s(i) = ip\,(1-p)^{i-1} \tag{2}$$

Let $\pi_i \overset{\Delta}{=} \lim_{t\to\infty} \Pr\{n^t=i\}$. We compute the stationary distribution $\Pi = \{\pi_0,\pi_1,\ldots,\pi_N\}$ by solving recursively the system $\Pi = \Pi P$. The average number of repeaters $\bar{n}$ is given by:

$$\bar{n} = \sum_{k=0}^{N} k\pi_k \tag{3}$$

Let β denote the probability that a terminal transmission is blocked due to transmission by one or more repeaters. Given that k repeaters are active, this probability is simply $1 - (1-p)^k$. Removing the condition we get

$$\beta = 1 - \sum_{k=0}^{N} \pi_k (1-p)^k \tag{4}$$

Let α denote the probability that a terminal transmission is blocked due to the repeater's buffer being full, assuming of course that no repeater is transmitting. Given k active repeaters, the probability is simply $\frac{k}{N} (1-p)^k$, 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

$$\alpha = \sum_{k=0}^{N} \pi_k \frac{k}{N} (1-p)^k \tag{5}$$

The total blocking probability is given by

$$B = \alpha + \beta \tag{6}$$

The network throughput S is expressed as

$$S = \sum_{k=0}^{N} \pi_k kp(1-p)^{k-1} = N\lambda (1 - B) \tag{7}$$

and the network delay is simply given by

$$D_n = \frac{\overline{n}}{S} \tag{8}$$

As in part II, the access delay D_a is estimated by

$$D_a = \frac{1}{1 - B} D_{S\text{-ALOHA}} (\lambda) + \frac{B}{1 - B} \frac{Kopt}{2} \tag{9}$$

3.2 IFT Protocol

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 or not new arrivals had occurred in slot t-1. 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 via a "cycle analysis".

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 k^{th} non-empty slot (see Fig. 1). d^k is a Markov chain. We derive its transition probabilities in the following.

Let $p_{ij}^k \overset{\Delta}{=} \Pr \{d^{k+1} = j / d^k = i\}$. Let $P = (p_{ij})$ be the transition matrix. (We drop the superscript k as we are only interested in steady-state conditions.) For i = 0, we have

$$p_{0j} = \begin{cases} \dfrac{N\lambda\ (1-\lambda)^{N-1}}{1-(1-\lambda)^N} & j=0 \\ 0 & j=1 \\ \dbinom{N}{j} \dfrac{\lambda^j\ (1-\lambda)^{N-j}}{1-(1-\lambda)^N} & j=2,3,\ldots,N \end{cases} \tag{10}$$

Given that $d^k = i$, let I_i denote the number of empty slot separating two consecutive non-empty 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 non-empty slots. Thus, for $i \neq 0$, N, we have

$$\Pr \{I_i = 0\} = 1 - (1-p)^i \tag{11}$$

$$\Pr \{I_i > 0\} = (1-p)^i \tag{12}$$

and the transition probabilities are given by ($i \neq 0, N$)

$$
p_{ij} = \begin{cases} 0 & j < i-1 \\[2ex] Pr\{I_i=0\}\,\dfrac{ip(1-p)^{i-1}}{1-(1-p)^i} + Pr\{I_i>0\}\,\dfrac{ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i-1 \\[2ex] Pr\{I_i=0\}\,\dfrac{1-(1-p)^i-ip(1-p)^{i-1}}{1-(1-p)^i} & \\ \quad + Pr\{I_i>0\}\,\dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i+(1-\lambda)^{N-i}[1-ip(1-p)^{i-1}-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i \\[2ex] Pr\{I_i>0\}\,\dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}[1-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i+1 \\[2ex] Pr\{I_i>0\}\,\dfrac{\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j}}{1-(1-\lambda)^{N-i}(1-p)^i} & j \geq i+2 \end{cases} \tag{13}
$$

Finally for $i = N$ we simply have

$$
p_{N,j} = \begin{cases} 0 & j<N-1 \\[2ex] \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j=N-1 \\[2ex] 1 - \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j=N \end{cases} \tag{14}
$$

Let $\pi_i^d \overset{\Delta}{=} \lim_{k\to\infty} Pr\{d^k = i\}$. The stationary distribution $\Pi^d = \{\pi_0^d, \pi_1^d, \ldots, \pi_N^d\}$ is obtained by solving recursively the system $\Pi^d = \Pi^d P$. We now derive the stationary performance measures. To do so, 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 $I_i + 1$. To compute the average cycle length, we need to determine the average length of I_i which we denote by $\bar{I}_i$. The probability density function of I_i is given by

$$\Pr\{I_i=\ell\} = \begin{cases} (1-\lambda)^{N(\ell-1)}[1 - (1-\lambda)^N] & i=0;\ \ell\geq 1 \\ 1 - (1-p)^i & i\neq 0,N;\ell=0 \\ (1-p)^i[(1-\lambda)^{N-i}(1-p)^i]^{\ell-1}[1-(1-\lambda)^{N-i}(1-p)^i] & i\neq 0,N;\ell>0 \\ (1-p)^{N\ell}[1-(1-p)^N] & i=N;\ell\geq 0 \end{cases} \tag{15}$$

Thus $\bar{I}_i$ is expressed as

$$\bar{I}_i = \begin{cases} \dfrac{1}{1-(1-\lambda)^N} & i=0 \\ \dfrac{(1-p)^i}{1-(1-p)^i(1-\lambda)^{N-i}} & i\neq 0,N \\ \dfrac{(1-p)^N}{1-(1-p)^N} & i=N \end{cases} \tag{16}$$

The average throughput over the cycle, which we denoted by S_i, is precisely the probability of a successful transmission and is given by

$$S_i = \Pr\{I_i=0\}\frac{ip(1-p)^{i-1}}{1-(1-p)^i}$$

$$+\Pr\{I_i>0\}\ \frac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i+ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-p)^i(1-\lambda)^{N-i}} \tag{17}$$

The average of the sum of active repeaters over the cycle is denoted by σ_i and is given by

$$\sigma_i = i\bar{I}_i + i + \Pr\{I_i>0\} \frac{(N-i)\lambda}{1-(1-p)^i(1-\lambda)^{N-i}} \qquad (18)$$

By renewal theory arguments, the stationary system throughput is expressed as

$$S = \frac{\sum_{i=0}^{N} \pi_i S_i}{\sum_{i=0}^{N} \pi_i (\bar{I}_i+1)} \qquad (19)$$

and the stationary average number of active repeaters is given by

$$\bar{n} = \frac{\sum_{i=0}^{N} \pi_i \sigma_i}{\sum_{i=0}^{N} \pi_i (\bar{I}_i+1)} \qquad (20)$$

By Little's result, the average network delay is

$$D_n = \frac{\bar{n}}{S} \qquad (21)$$

The probability of blocking B is simply

$$B = 1 - \frac{S}{N\lambda} \qquad (22)$$

The access delay is again estimated by the expression given in Eq (9).

4. Numerical Results and Discussion

We start by examining the fully-connected DFT case. Contrary to what was shown for the DFT star configuration (with M=1), given λ, we note that the value of p which yields minimum D_n does not correspond to the value of p 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 the optimum total delay D for a given throughput S.

In Fig 3 we plot the optimum D_n versus S for various values of N along with the corresponding curves obtained in the star configuration. Fig 4 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.

To see the importance of blocking due to repeaters transmissions (β) relative to the blocking due to lack of storage (α), we plot in Fig 5 α versus β for various values of N along with the curves corresponding to the star configuration; it is all too evident from this figure that β is the predominant factor and thus that the fully connected configuration is even more "channel bound" than the star configuration. Moreover, β becomes more and more important relative to α as N increases, due to the larger number of contending devices. This certainly justifies the absence of consideration in the present study for buffer sizes larger than 1.

For a two-hop fully connected environment, the system capacity is obtained for $\lambda = 1/e$. We plot in Fig 6 the DFT system capacity versus N for this and the star configuration. The fully-connected environment provides a smaller network capacity than the star configuration, especially for the smaller values of N ($N \leq 6$); however, 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 $D = D_a + D_n$ for the fully connected DFT system is plotted versus S in Figures 7, 8 and 9 for the cases N = 2, N = 5 and N= 10 respectively, along with, for comparison, the throughput delay curves corresponding to the star configuration. (The curves corresponding to the IFT protocol appearing in these figures will be discussed subsequently). We note that for both N = 2 and N = 5 the delay 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 Figure 3) overcomes the degrading effect of the larger blocking proability experienced (see Fig 4). The throughput delay curves for networks of arbitrary connectivity employing the DFT transmission protocol are expected to lie between the two displayed. However we have no formal proof for this claim!

Consider now the fully-connected IFT case*. The main focus here is to

*The comment we made earlier regarding minimum blocking and minimum delay in the fully-connected DFT case is also valid here.

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 10 and B versus S in Fig 11 (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 (and to a certain extent B) are 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 6 above; this capacity is not too sensitive to variations in the size of the network, N. The throughput-delay curves appear in Figures 7, 8 and 9 for N = 2, N = 5 and N = 10 respectively. 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.

5. Conclusion

In this note we studied fully connected configurations employing slotted ALOHA. We considered two transmission protocols, the delayed-first-transmission (DFT) protocol (also considered in Part II for the analysis of star configurations) and the immediate-first-transmission (IFT) protocol. The major conclusions follow. The DFT fully connected system provides a capacity which is smaller than the one achieved by the star configuration. The difference decreases as N approaches 10. The total packet delay is also larger in the fully connected case except with the larger values of N (namely N = 10), for which a slight improvement is gained over a significant range of S, due to a smaller achievable network delay. The sensitivity of the system performance to variations in the transmission protocol was observed by comparing the results obtained with the IFT protocol to those obtained with the DFT protocol. We basically noted a lower system capacity with IFT; this capacity is not too sensitive to changes in N. The packet delay, however, has slightly improved over a significant range of the throughput.

In the forthcoming Part IV [2], we shall be analysing the performance of packet radio networks employing the nonpersistent CSMA mode.

References

[1] Tobagi, F., "On the Performance Analysis of Multihop Packet Radio Systems: Part II - Star Configurations Employing Slotted ALOHA," Packet Radio Temporary Note #247, SRI International, Menlo Park, California, January 17, 1978.

[2] Tobagi, F., "On the Performance Analysis of Multihop Packet Radio Systems: Part IV - Fully Connected Configurations Employing CSMA," Packet Radio Temporary Note #249, SRI International, Menlo Park, California, January 31, 1978.

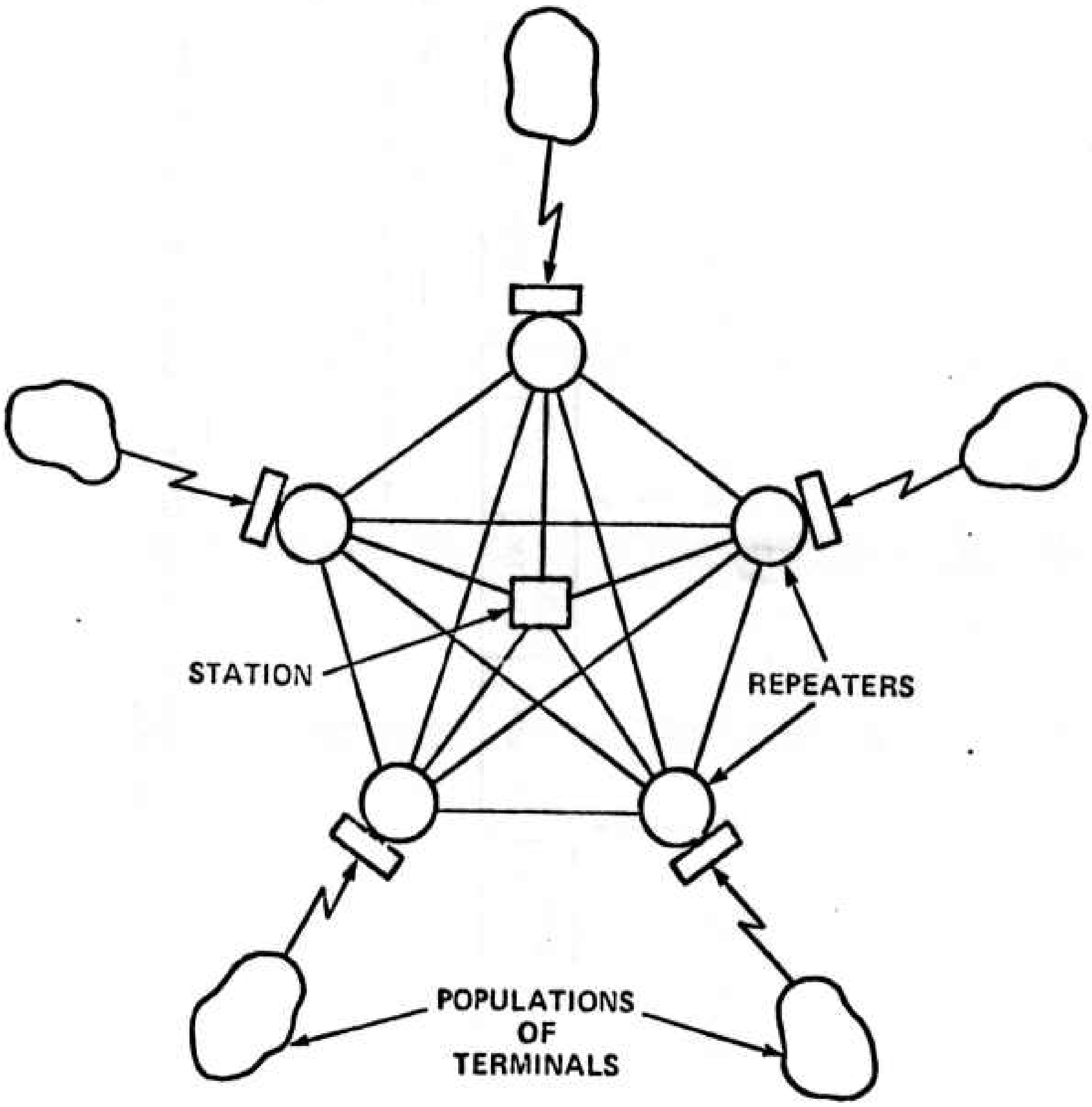


Figure 1 A Two-Hop Fully-Connected Configuration.

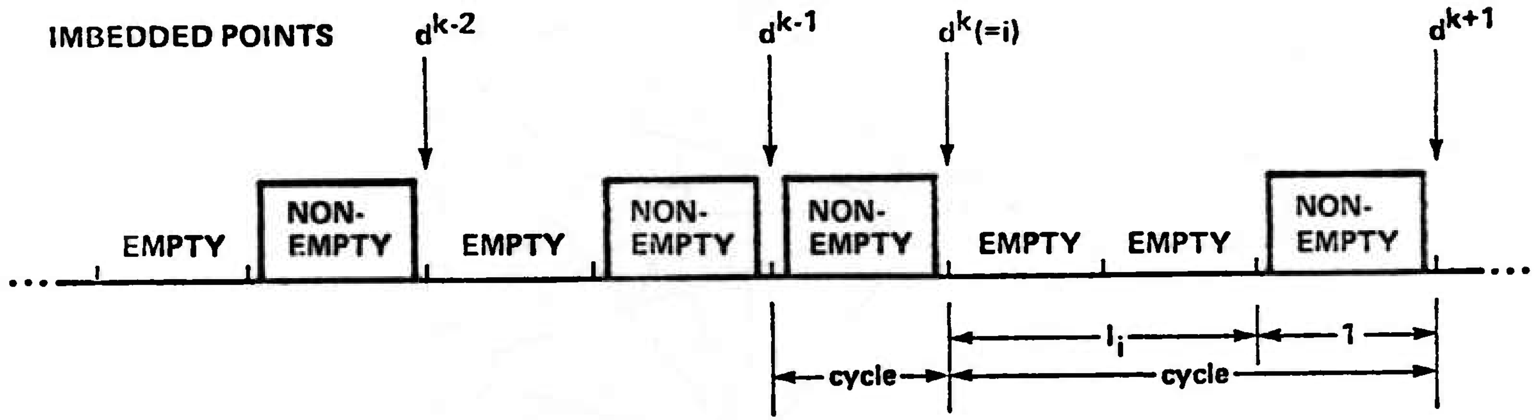


Figure 2 The Imbedded Markov Chain in the Slotted ALOHA IFT Protocol.

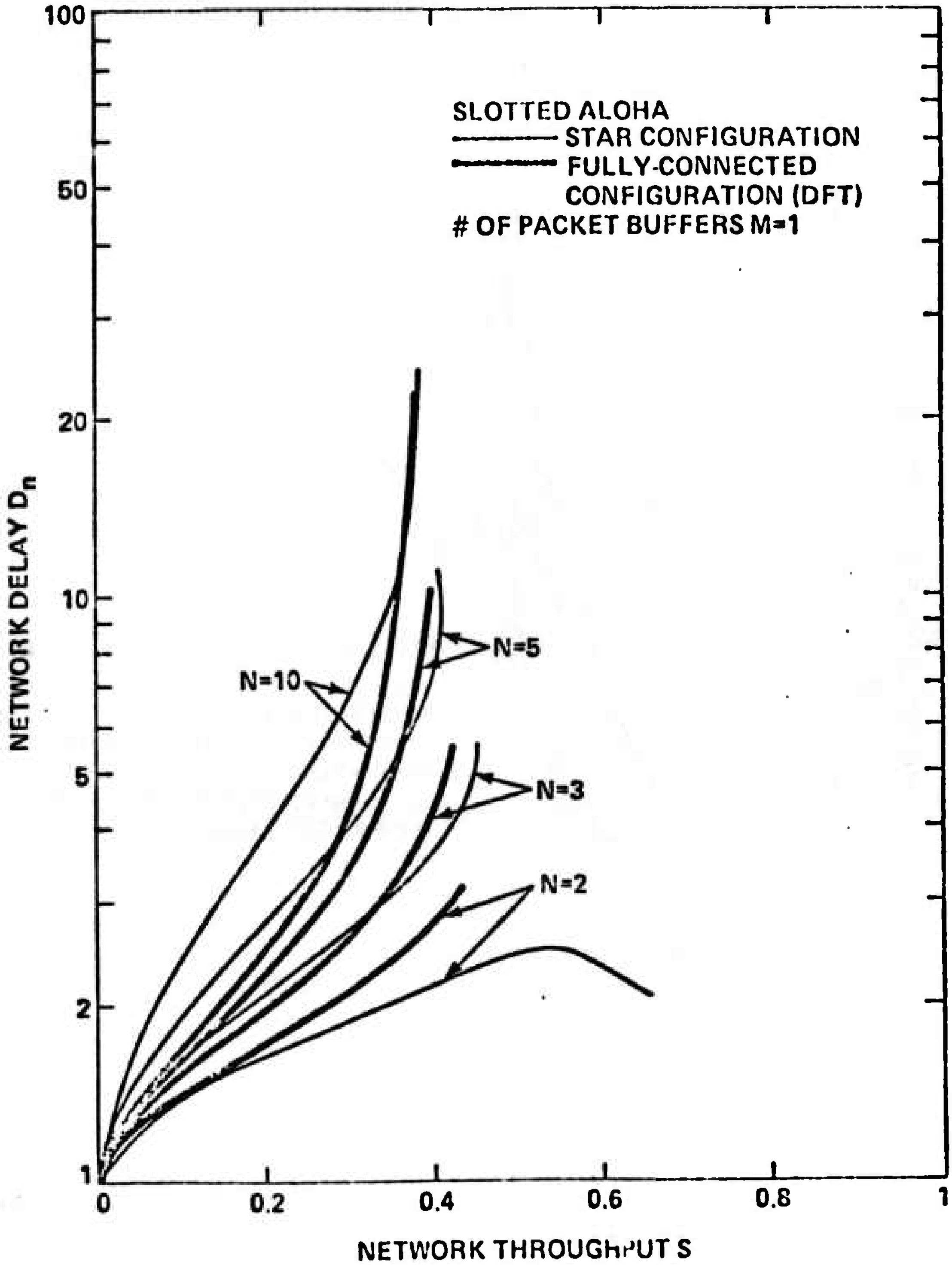


Figure 3 Slotted ALOHA Star and Fully-Connected Configurations: Optimum (Network) Throughput-Delay Curves.

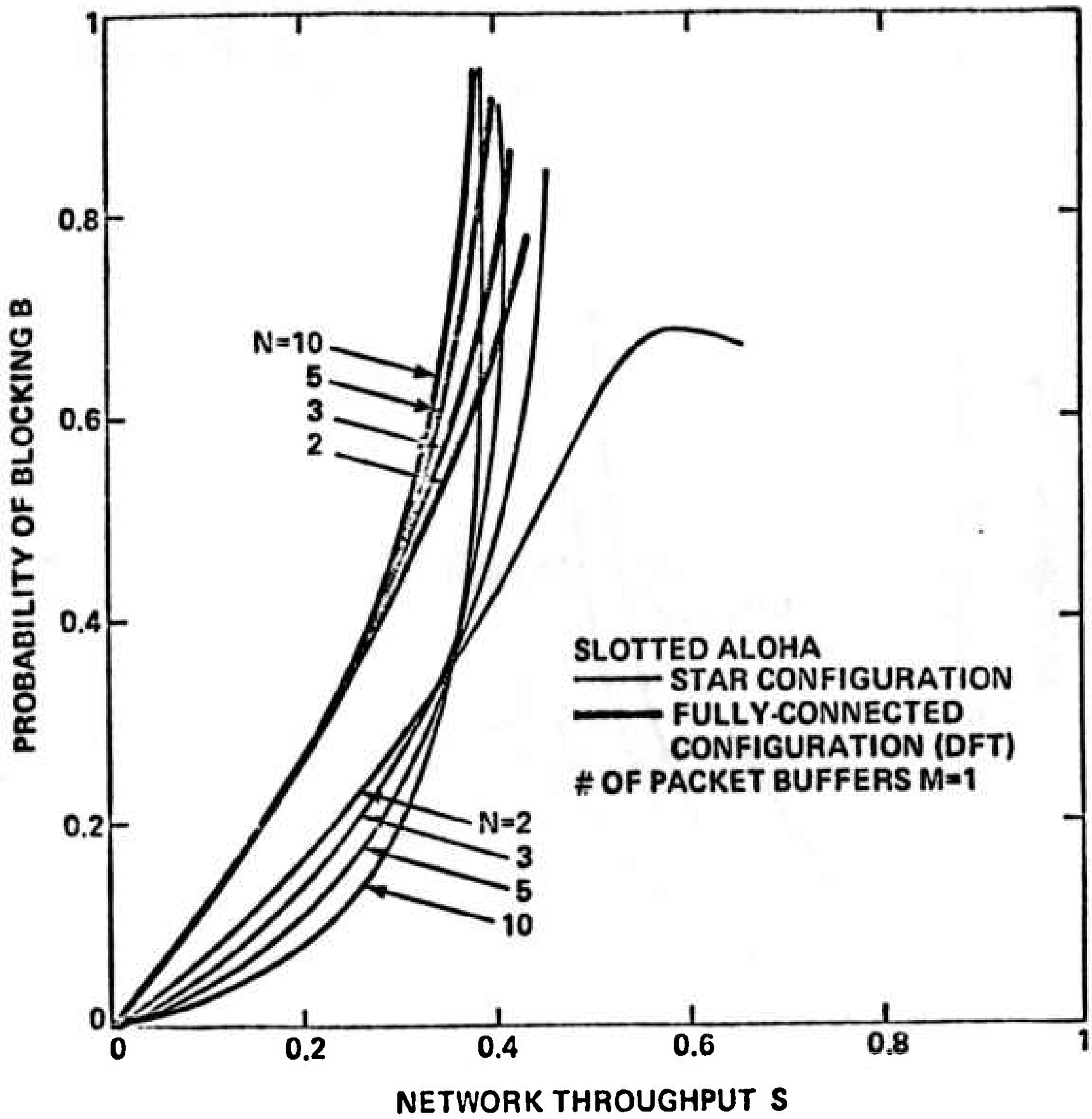


Figure 4 Slotted ALOHA Star and Fully-Connected Configurations: Minimum Blocking versus Throughput.

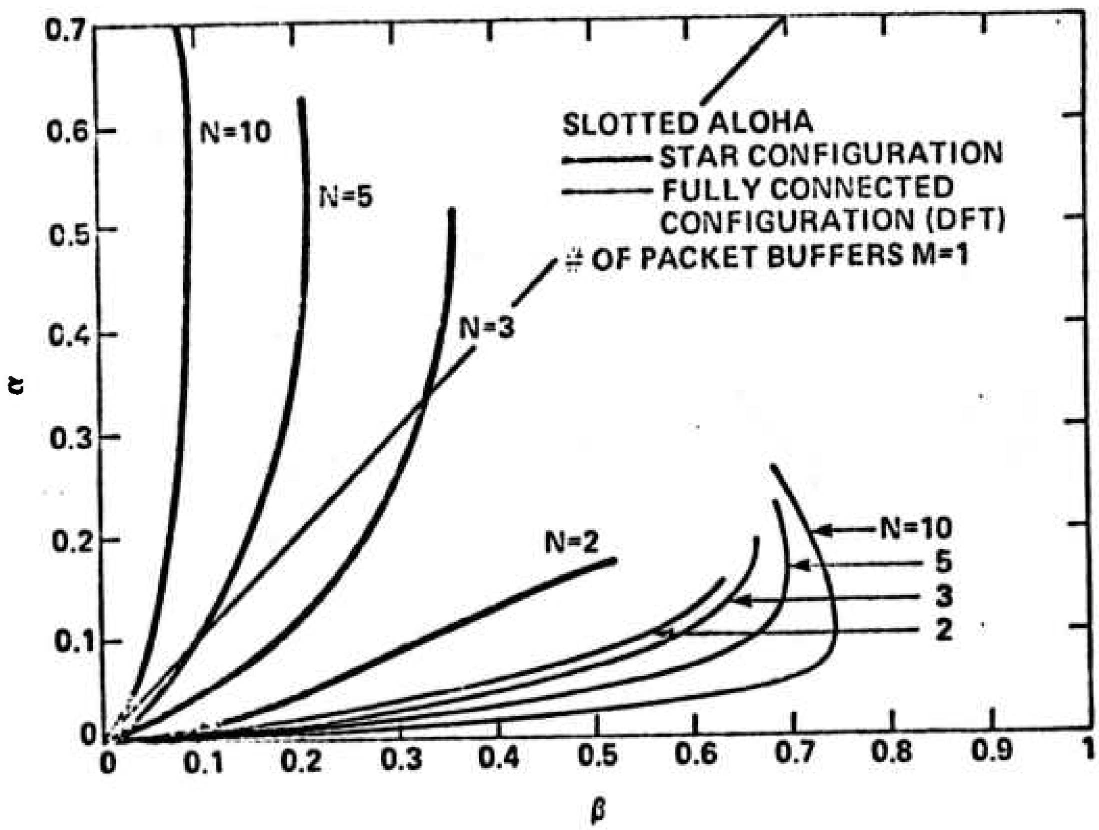


Figure 5 Slotted ALOHA Star and Fully-Connected Configurations: α versus β at Minimum Blocking.

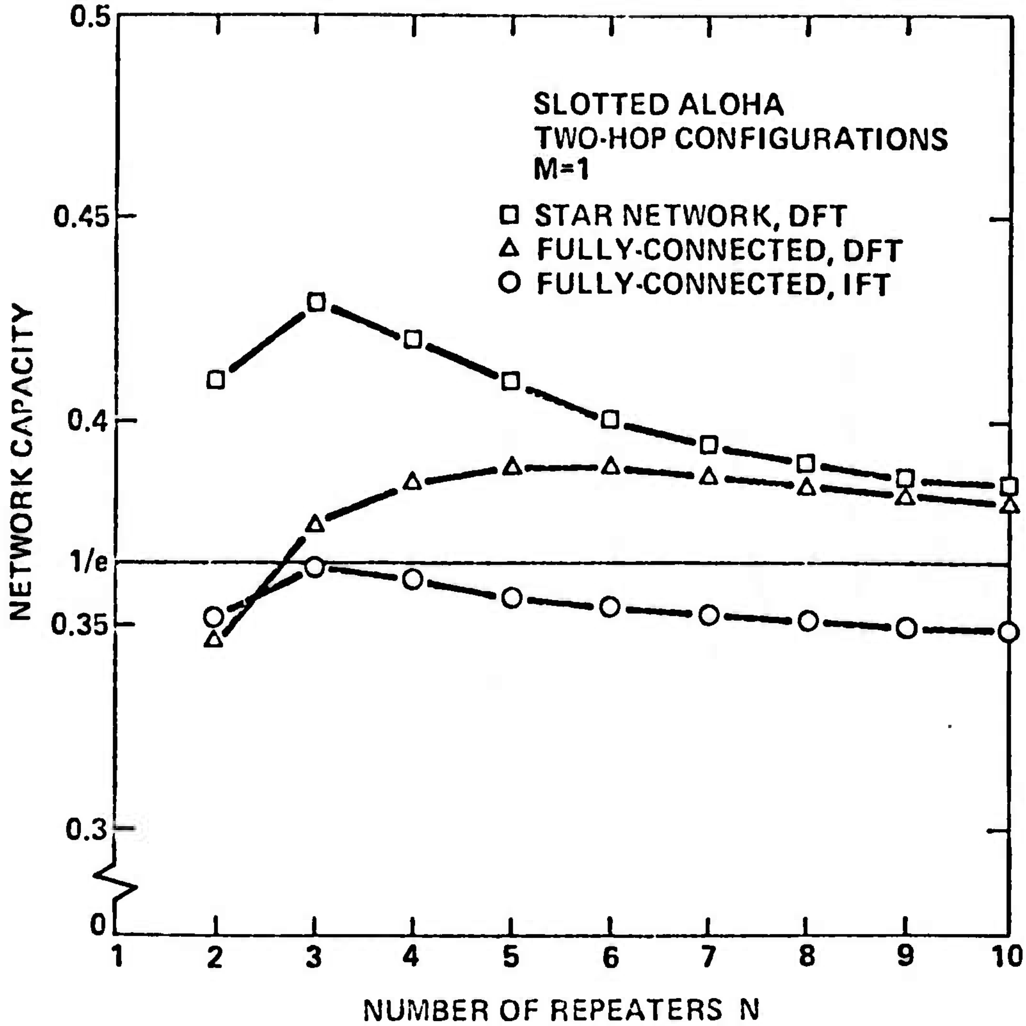


Figure 6 Slotted ALOHA Two-Hop Star and Fully-Connected Networks: Network Capacity versus N.

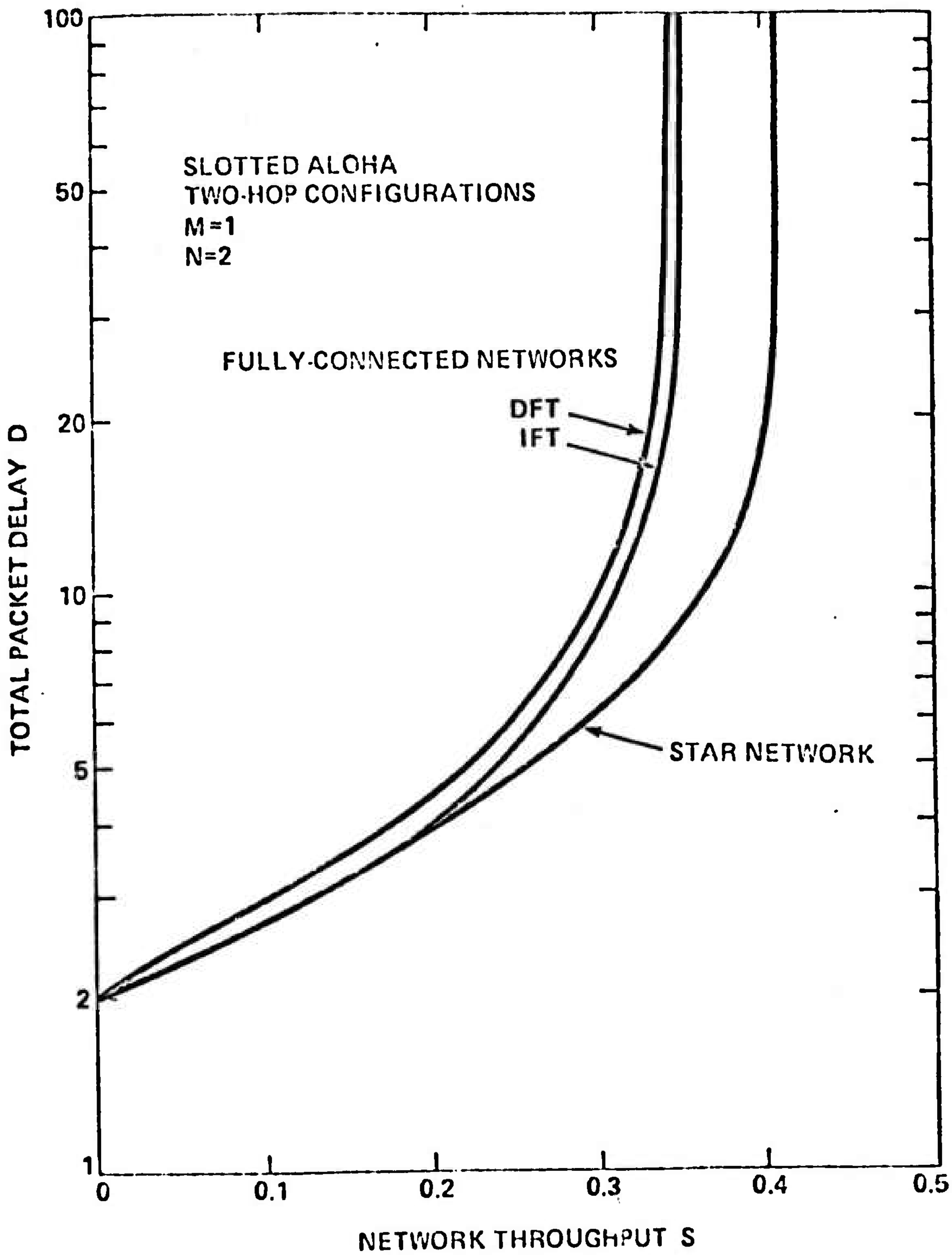


Figure 7 Throughput-Delay Tradeoff in Two-Hop Slotted ALOHA Star and Fully-Connected Networks. (N = 2)

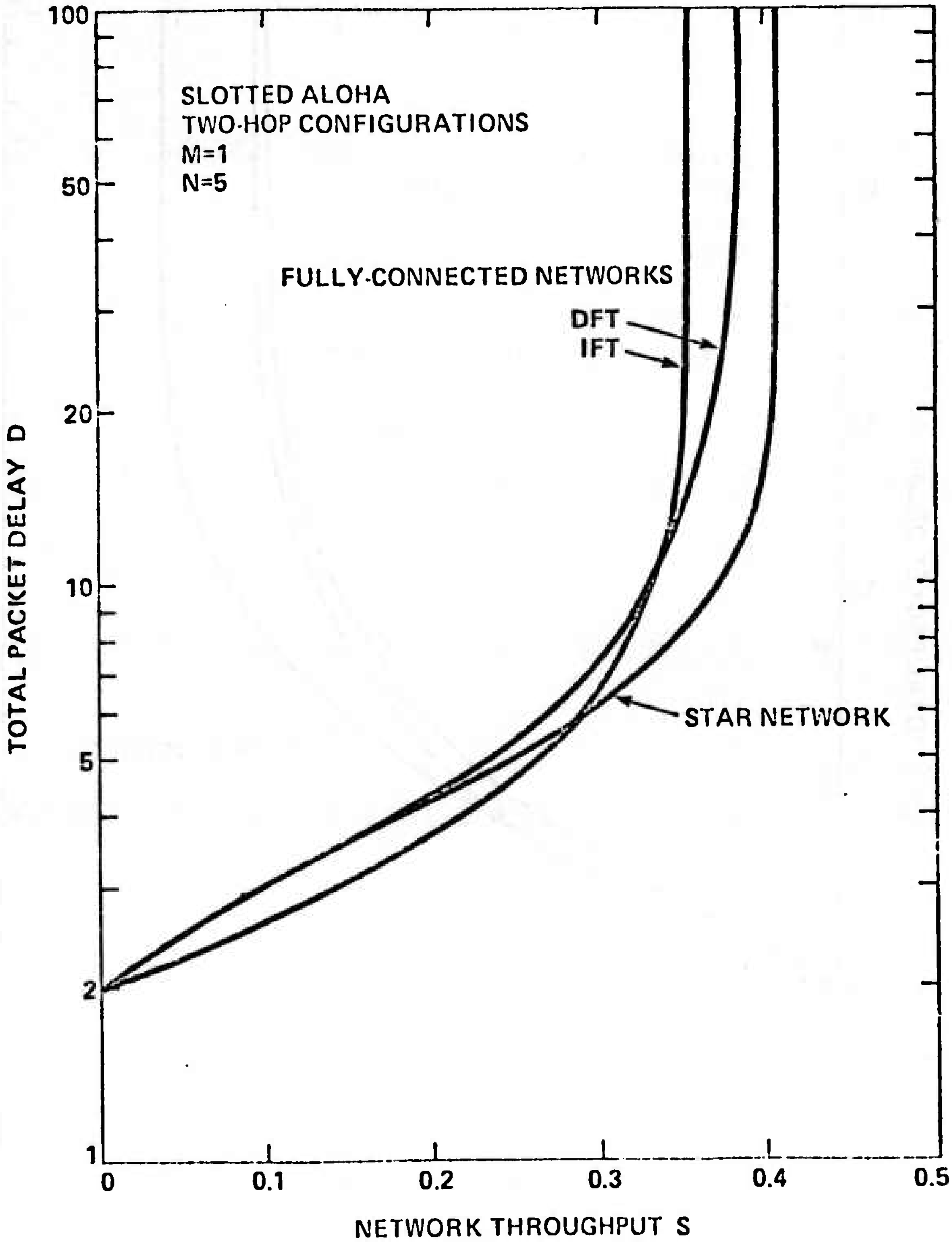


Figure 8 Throughput-Delay Tradeoff in Two-Hop Slotted ALOHA Star and Fully-Connected Networks. (N = 5)

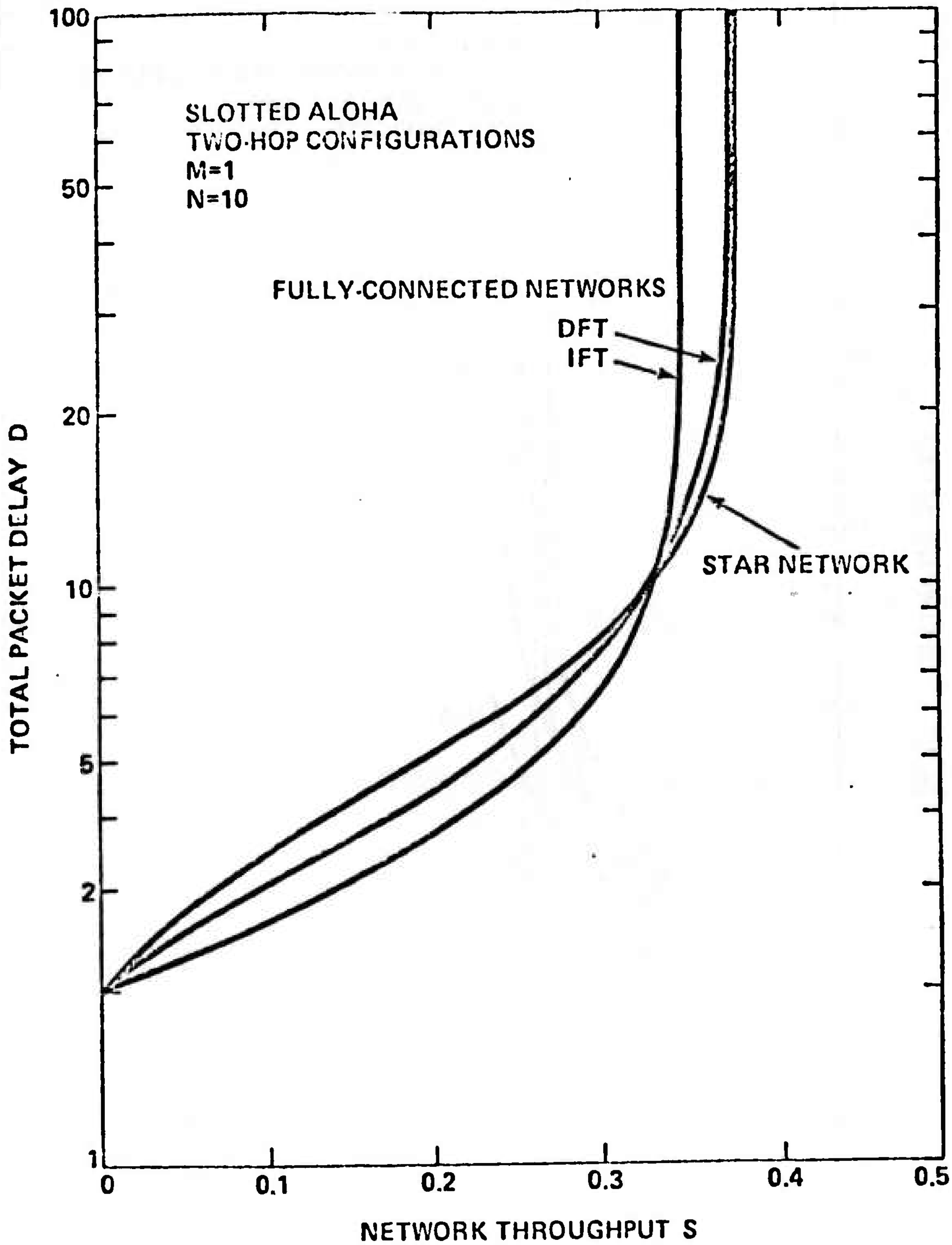


Figure 9 Throughput-Delay Tradeoff in Two-Hop Slotted ALOHA Star and Fully-Connected Networks. (N = 10)

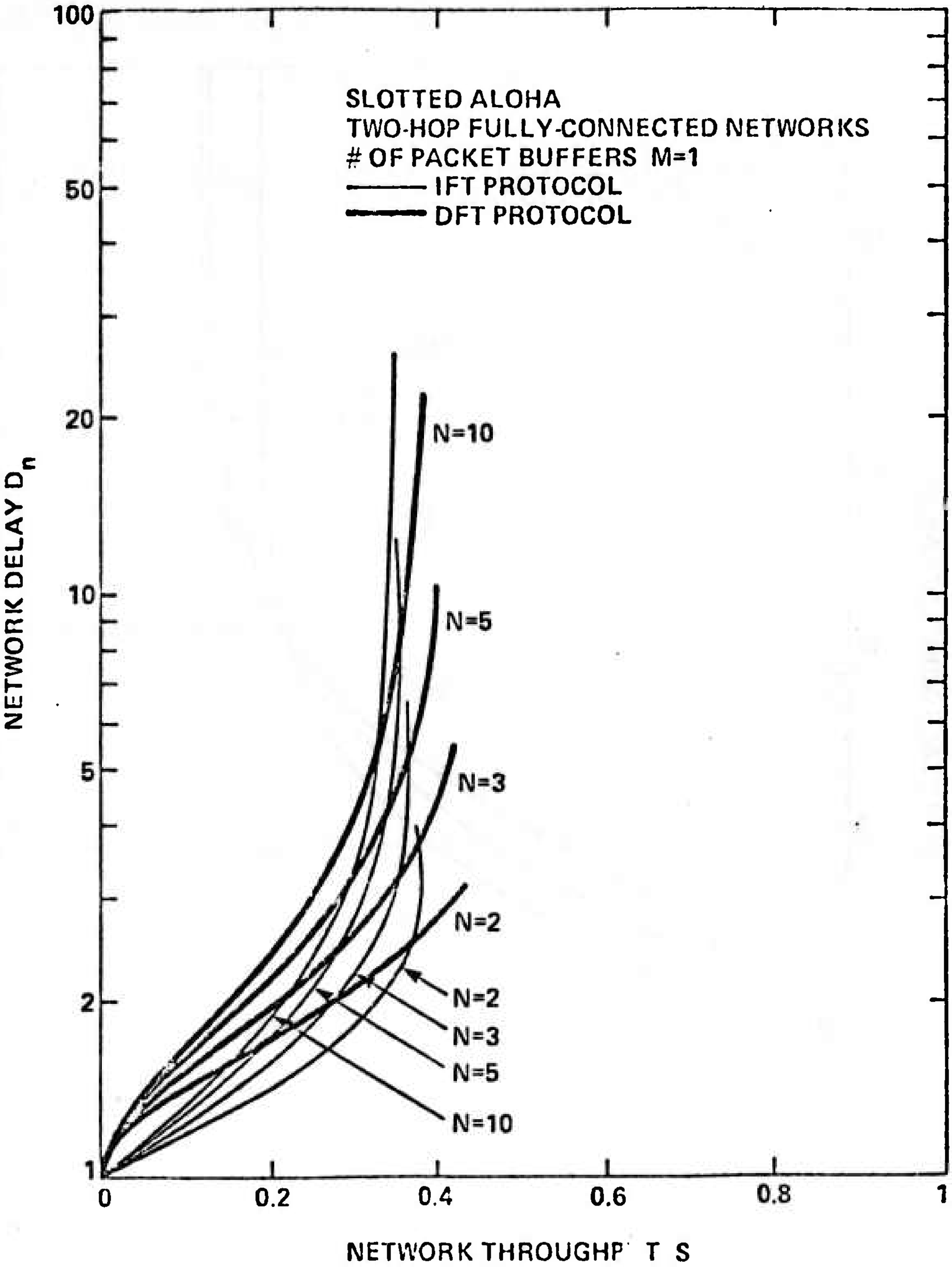


Figure 10 Slotted ALOHA Fully-Connected Configuration: Optimum (Network) Throughput-Delay Curves.

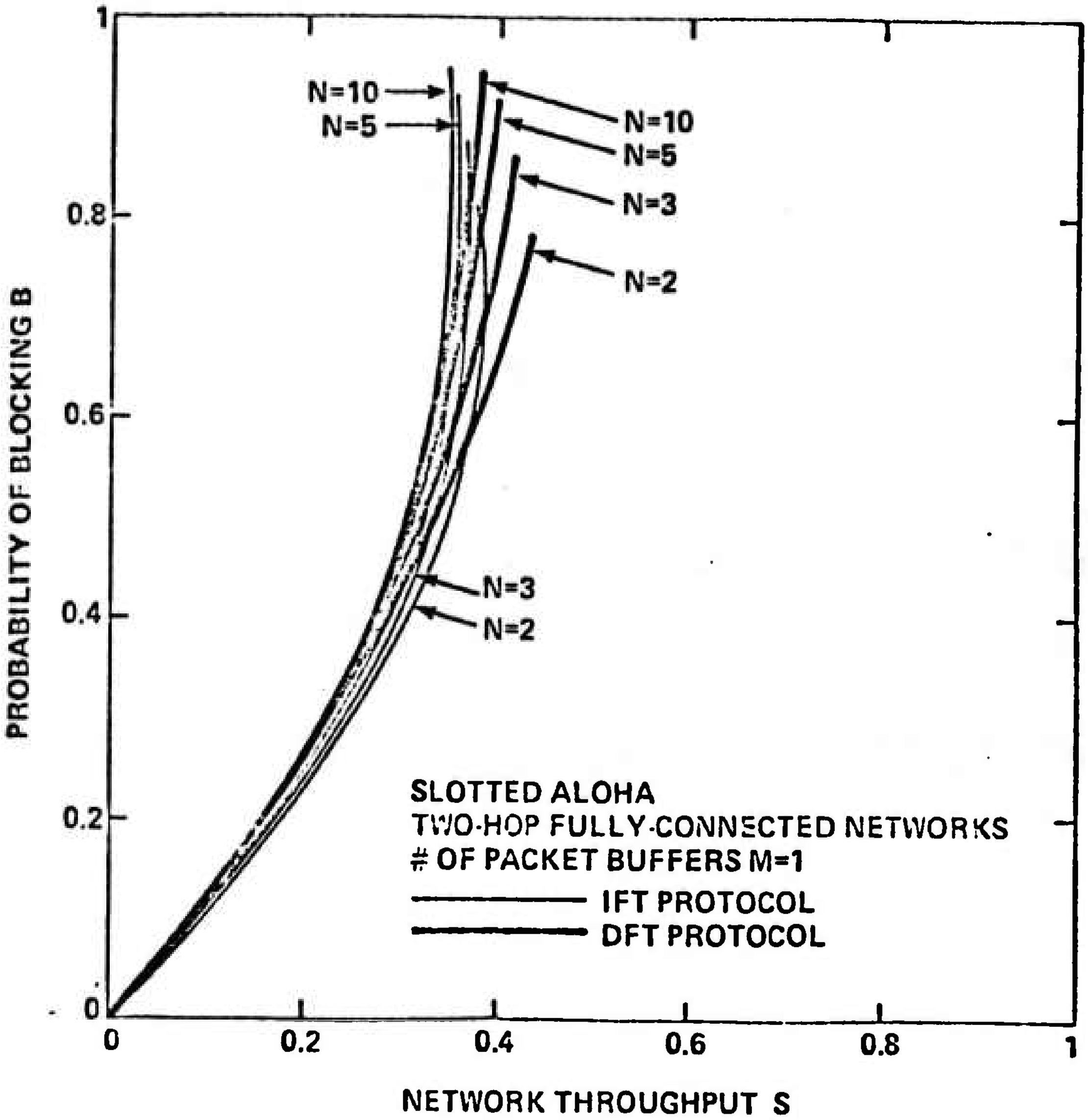


Figure 11 Slotted ALOHA Fully-Connected Configuration: Minimum Blocking Versus S.

PRTN #249

F. Tobagi
UCLA
January 31, 1978

On the Performance Analysis of Multihop Packet Radio Systems: Part IV - Fully Connected Configurations Employing CSMA

1. Introduction

Carrier Sense Multiple Access (CSMA) [1,2,3,4] has become a well known technique, which enables efficient sharing of a data communication channel by a large population of bursty users in a ground radio environment; this environment is characterized by a propagation delay between the devices which is very small compared to the transmission time of the packet (typically 1%). 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; based on this channel state information (busy or idle), the terminal takes an action prescribed by the particular CSMA protocol being used. In particular, a terminal never transmits when it senses that the channel is busy. In [1] we described and analyzed two protocols referred to as nonpersistent and p-persistent CSMA; the performance of these was given in terms of channel capacity and throughput-delay tradeoffs. It was shown that, for a ground radio single-hop environment in which all devices are within range and in line-of-sight of each other, CSMA offers a performance much superior to the ALOHA schemes. For a propagation delay equal to 1% of the transmission time of a packet, the nonpersistent protocol, for example, offers a channel capacity of over 80%, while we can only attain 36.8% for slotted ALOHA and 18.4% for pure ALOHA; packet delay is also significantly improved with CSMA. Now the questions of interest are: How does CSMA perform in multi-hop environments? How does it compare to the ALOHA modes? Do we still enjoy the improvement in performance over ALOHA experienced in single-hop environments?

The performance analysis of multi-access schemes in multi-hop environments is a difficult task; there is no simple approach yet to tackle the general problem, nor is there a way to translate the results obtained in single-hop systems to the more complex ones. For example, we are already certain that in multi-hop environments devices are not all within range and in line-of-sight of each other; this means that on one hand the hidden terminal problem prevails and, as shown in [2], this can significantly degrade the performance of CSMA; on the other hand, not all transmissions will affect the successful reception of a packet at an immediate destination.

In parts II and III of this series [5,6] we gave an analysis of slotted ALOHA in simple multihop environments, namely the star configuration and the fully-connected configuration; in these, packets originate at terminals, are destined to a central station, and require for their transport a successful repetition by repeaters which are in line-of-sight and within range of the station. In the fully-connected configuration, all repeaters are also within range and in line-of-sight of each other; a terminal however is within range of a number of terminals, but of only one repeater. Only inbound traffic was considered. We derived the system performance in terms of capacity and throughput-delay tradeoffs and determined its dependence on various system parameters such as the transmission protocol, the network configuration, and the repeater's storage capacity.

In this note we attempt to answer the above questions by considering the simple network configurations utilized in the context of slotted ALOHA;

in particular we restrict ourselves to the fully connected one (to be described in detail below) in order to gain much of the advantage of CSMA. Basically, we shall derive the fundamental throughput-delay performance, and thus give a comparison between this and the results obtained with slotted ALOHA. Only the nonpersistent CSMA protocol will be considered because of its relative simplicity in analysis and implementation, as well as its relatively high efficiency.

To render the analysis more tractable, a number of simplifying assumptions are made. These assumptions will be introduced as needed; and to the largest extent possible we will either justify them or evaluate the effect they have on the results.

II. The Nonpersistent CSMA Revisited* [1,3]

For the sake of clarity and completeness, we repeat here the description of the nonpersistent CSMA protocol as well as the infinite population model, along with its underlying assumptions, which was previously used in analyzing the scheme. We also give the throughput analysis, a report on the determination of packet delay and a discussion of simulation results. Moreover a simple approximate model is given which characterizes the traffic of successful packets from the infinite population.

2.1 The Nonpersistent CSMA Transmission Protocol [1]

A terminal with a packet ready for transmission senses the channel and operates as follows.

---

*This section consists basically of extracts of [1] and [3] which consitute the preliminaries and background necessary for the analysis presented in this note. The reader familiar with the results of [1] and [3] may skip this section perhaps with the exception of §2.5.

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 a retransmission delay distribution. At this new point in time, it senses the channel and repeats the algorithm described.

A slotted version of this can be considered in which the time axis is slotted and the slot size is τ seconds (the propagation among pairs of devices is assumed to be the same [1]). All terminals 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 terminal senses the channel at the beginning of the next slot and operates according to the protocol described above.

2.2 The Infinite Population Model [1]

Assume that our traffic source consists of an infinite number of users who collectively form an independent Poisson source with an aggregate mean packet generation rate of λ packets per packet transmission time T*. This is an approximation to a large but finite population in which each user generates packets infrequently and each packet can be successfully transmitted in a time interval much less than the average time between successive packets generated by a given user. Each user in the infinite population is assumed to have at most one packet requiring transmission at any time (including any previously blocked packet). Under equilibrium conditions, λ is also the channel throughput. Because of packet interference, the achievable throughput will always be less than 1. The traffic offered to the channel from our collection of users consists not only of new packets but also of previously

*We assume that each packet is of constant length requiring T seconds for transmission.

collided packets: this increases the mean offered traffic rate which we denote by G (packets per transmission time T) where $G \geq \lambda$. To avoid repeated conflicts, each user delays the transmission of a previously collided packet by some random time whose mean is $\bar{X}$ (chosen, for example, uniformly between 0 and $X_{max} = 2\bar{X}$).

Our two further assumptions are the following.

<u>Assumption 1</u>: The average retransmission delay $\bar{X}$ is large compared to T.

<u>Assumption 2</u>: The interarrival times of the point process defined by the start times of all the packets plus retransmissions (and reschedulings) are independent and exponentially distributed.

It is clear that Assumption 2 is violated. (We have introduced it for analytic simplicity.) However some simulation results are discussed below which show that performance results based on this assumption are excellent approximations. Moreover, in the context of slotted ALOHA it was analytically shown, in the limit as $\bar{X} \to \infty$, that Assumption 2 is satisfied.

So far, we have introduced the following notation: λ (throughput), G (offered channel traffic rate), T (packet transmission time), $\bar{X}$ (average retransmission delay) and τ (propagation delay). If we use T = 1 (for normalization), then we express τ as $a = \tau/T$ and $\bar{X}$ as $\delta = \bar{X}/T$.

<u>2.3 Throughput Analysis</u> [1]

Here we solve for λ in terms of G and a. The channel capacity is found by maximizing λ with respect to G. λ/G is the probability of a successful transmission or scheduling. G/λ is the average number of times a packet must be transmitted (or scheduled) until success.

Given the assumptions introduced above the (λ,G) relationship for the (unslotted) nonpersistent CSMA is given by

$$\lambda = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG}} \tag{1}$$

and for the slotted nonpersistent CSMA by

$$\lambda = \frac{aGe^{-aG}}{1 + a - e^{-aG}} \tag{2}$$

<u>Proof</u> (slotted case):

We consider the time axis and we define a transmission period (TP) to be the period of time required for transmission and reception of a packet and its (possible) overlapping packets. Thus we observe on the time axis transmission periods separated by idle periods, as depicted in Fig. 1. The length of a TP is $1 + a$. A TP is successful if only one packet is transmitted; the probability of this occurring is

$$P_s = \frac{aGe^{-aG}}{1 - e^{-aG}} \tag{3}$$

Due to the memoryless property of the Poisson process, the average idle period (normalized to T) is simply

$$\bar{I} = \frac{ae^{-aG}}{1 - e^{-aG}} \tag{4}$$

Using renewal theory arguments, the average channel utilization is given by

$$\lambda = \frac{P_s}{\bar{I} + 1 + a} \tag{5}$$

Substituting for P_s and $\bar{I}$ the expressions found above, we get Eq (2).

Q.E.D.

Curve A in Fig. 2 is a plot of λ versus G for the slotted nonpersistent CSMA when a = 0.01.

2.4 Discussion and Delay Analysis [1,3]

The above analysis is based on renewal theory and probabilistic arguments requiring independence of random variables provided by Assumption 2. Moreover steady state conditions are assumed to exist. However from the (λ,G) relationship derived above one can see that steady state may not exist because of inherent instability of these random-access techniques. This instability is simply explained by the fact that when statistical fluctuations in G increase the level of mutual interference among transmissions, then the positive feedback causes the throughput to decrease to 0. Extensive simulation runs performed on a slotted ALOHA channel with an infinite population [7], (a system known to exhibit a behavior similar to CSMA's), have indeed shown that the assumption of channel equilibrium is not valid; in fact, after some finite time period of quasi-stationarity conditions, the channel will drift into saturation with probability one. Nevertheless, the result above is useful for, as also supported by simulation, it is meaningful for this quasi-stationary finite (and possibly long) period of time.

In the simulation model considered [1] we relax Assumptions 1 and 2 concerning the retransmission delay and the independence of arrivals for the offered channel traffic. That is, only the newly generated packets are derived independently from a Poisson distribution; collisions and uniformly distributed random retransmissions are accounted for without further assumptions. In general, our simulation results indicate the following.

1) For each value of the input rate λ, there is a minimum value δ_0 for the average retransmission delay variable such that below that value it is

impossible to achieve a throughput equal to the input rate. The higher λ is, the larger δ_0 must be to prevent a constantly increasing backlog, i.e., to prevent the channel from saturating.

2) Recall that the throughput equations were based on the assumption that $\overline{X}/T = \delta >> 1$. Simulation shows that for finite values of δ, $\delta > \delta_0$, but not too large compared to 1, the system already "reaches" the asymptotic results ($\delta \to \infty$). That is, for some finite values of δ, Assumption 2 is excellent and delays are acceptable. Moreover, the comparison of the (λ,G) relationship as obtained from simulation and the results obtained from the analytic model exhibits an excellent match. Simulation experiments were also conducted to find the optimal delay; that is, the value of δ(S) which allows one to achieve the indicated throughput with the minimum delay. In Fig. 3 we plot in dashed line this delay for the nonpersistent CSMA. The values of δ used in obtaining this curve are:

$$\delta = \begin{cases} 1/2 & 0 \leq \lambda \leq 0.55 \\ 2 & 0.55 < \lambda \leq 0.65 \\ 4 & 0.65 < \lambda \leq 0.75 \end{cases}$$

In a subsequent paper [3], we formulated a Markovian model for the nonpersistent CSMA which allowed us to obtain analytically, in the case of a finite population, the throughput-delay performance as well as the effect of the retransmission delay and of the population size on the performance. With finite populations, the Markov chain in question is irreducible and ergodic and steady state is always reached; however, if the retransmission delay is not sufficiently large, then the stationary performance attained is significantly degraded (low throughput, very high delay), such that, for all practical purposes, the channel is said to have failed; it is then called an unstable channel. With an infinite population, the Markov chain is not ergodic and stationary conditions do not exist; the channel is always unstable. For unstable channels, a stability measure is defined which consists of the average time it takes the system, starting from an empty state, to reach a state determined to be critical*[3,8]. In fact, this critical state partitions the state space into two regions, a safe region and an unsafe region. 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 valid over an average finite period equal to FET [3]. In Fig. 3 we also plot the throughput-delay performance for the infinite population case with guaranteed FET of 1 minute, 1 hour, and 1 month. We note that the quasi-stationary results obtained by simulation constitute a rather conservative prediction of the "true" performance of CSMA channels for a period of time of at least 1 month.

2.5 A Model for Successful Traffic

It is often the case, in multihop environments, that the output of a population of terminals, defined as the process of successful packets on the channel, constitutes the input to some other system component, say a repeater. It is therefore useful to have a simple characterization of this process. In reference [9] we have approximated the output of a nonpersistent CSMA channel by a Poisson process with a mean of λ packets per T sec. The goodness of the approximation was verified by comparing the density function of the exponential

---

*In this model, the state is defined as the number of terminals with a non-empty packet buffer.

distribution to histograms of interdeparture times (i.e., time between the end of transmission of successive successful packets) obtained from simulation. Examples are shown in Fig. 4. We note that except for interdepartures in the range of one or two packet transmission times, the match is acceptable and that the samller the throughput is, the more valid is the assumption. The main criticism we have here, however, is that this approximation does not explicitly account, in the evaluation of interdeparture times, for the transmission time of the packet; the latter, as can be seen in the sequel, is an important factor in the analysis of multihop systems. Thus, we much prefer to use the following model. Let Y denote an arbitrary interdeparture time. We approximate Y by $1 + Z$ where Z is exponentially distributed with mean $1/\lambda - 1 = 1/\lambda'$. Thus we use

$$\Pr\{Y \leq y\} = 1 - e^{-\lambda'(y-1)} \qquad y \geq 1$$

The density function for this distribution is plotted in dashed lines in Fig. 4. The comparison with the interdeparture histograms shows that this approximation is very acceptable as well.

III. Analysis of Fully-Connected Configurations

3.1 System Configuration and Transmission Protocol

We consider in this note the fully-connected two-hop network configuration [6] in which all repeaters are within range and in line-of-sight of each other and of the station. With each repeater is associated an infinite population of terminals generating traffic which is destined to the station (see Fig. 5). Each repeater is provided with a finite storage capacity which can accomodate exactly one packet. Termina' follow the nonpersistent protocol described earlier in section 2.1. Repeaters also use the nonpersistent CSMA mode as described in the following. A repeater which has completed the successful reception of a packet from its associated population of terminals, transmits the packet <u>without delay</u>. 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. (This corresponds to the so-called immediate-first-transmission (IFT) protocol considered in [6] for the slotted ALOHA mode as well.) This first transmission of the packet by the repeater may be unsuccessful due to collisions with transmissions from other active repeaters. The rescheduling of the packet is assumed to be geometrically distributed: the unsuccessful repeater resenses the channel in the current slot with a fixed probability ν; of course a retransmission will result only if the channel is sensed idle. Just as in [5,6] we shall assume that devices learn about their success or failure instantaneously at the end of their transmission period. We define the network throughput S to be the average number of packets received at the station per T seconds. The packet delay D is defined to be the time since the packet originates at the terminal until it is successfully received at the station; as in [5,6], we distinguish the two components: D_a, the access delay and D_n the network delay. We are seeking here the throughput-delay performance of such CSMA networks. N denotes the number of repeaters present in the system.

3.2 The Single-Repeater Case

Before we proceed with the general model and its underlying assumptions, we consider here the simple case N = 1. It is clear that in this case repeater's transmissions are successful. Considering the infinite population model described in section 2.2, the analysis of this system differs from the single-hop one given in section 2.3 by the simple fact that a successful transmission here requires 2T seconds instead of only T. The probability of a successful transmission P_s and the average idle period $\overline{I}$ are as expressed in Eqs (3) and (4) respectively; the average busy period however is given by

$$\overline{B} = (1 - P_s)(1 + a) + P_s (2 + 2a) \tag{7}$$

The network throughput is then expressed as

$$S = \frac{P_s}{\overline{I} + \overline{B}} = \frac{aGe^{-aG}}{1 + a - e^{-aG} + aGe^{-aG}(1 + a)} \tag{8}$$

Curve B in Fig. 2 is a plot of S versus G for this simple case. We note here that the network capacity is about 46% of the total available bandwidth while, in single hop environments, we were able to achieve 85%; the channel utilization at the repeater, however, is 92%! This example shows that the introductio of a repeater, and thus an extra hop for each packet transmission, has significantly decreased the net throughput of a centralized network. Next, we consider the more general case $N > 1$.

3.3 Model Assumptions

Consider for each population of terminals T_i a time line which exhibits packet transmissions from T_i only. Consider 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. The processes definin these time lines are evidently dependent on each other in a rather complex way the dependence is determined by the particular system connectivity. To render the analysis tractable, we introduce here the following two assumptions.

Assumption 3. When sensing the channel, a repeater R_i can distinguish between the presence of carrier due to other repeaters and carrier due to transmissions by its associated population of terminals T_i*.

The simplification gained here lies in the fact that the decision made by a repeater to transmit its packet is solely dependent on the state of the repeaters population. The performance evaluation will be rather slightly optimistic; indeed, delaying transmission by a repeater because of terminals' transmission (whether successful or not**) would only degrade the system performance. It is to be noted, however, that the effect of this assumption will be smaller as N gets larger. Our next assumption is

Assumption 4. Similarly, terminals do not inhibit transmission when their associated repeater is transmitting.

It is clear that this assumption is violated in the environment under consideration. By introducing it, we simplify the problem in that we allow the processes defining each time line T_i to be independent of the repeaters time line; the successful transport of a packet from T_i to R_i, on the other hand, will be considered dependent on the state of R, as will be seen in the analysis below. The effect of Assumption 4 is to provide a pessimistic evaluation of system performance; indeed transmissions from T_i which start during a transmission period of R_i are useless and contribute to a higher traffic rate on time line T_i. However, it is to be noted here again that this effect

---

*The distinction between the two different carriers can be made possible in a real system if the discrepancy in power levels is significant (the repeater transmit power being for example much greater than terminals' transmit power) or if different data rates are used.

**Repeaters possess single packet buffers.

will be smaller as N gets larger. For N = 10, for example, T_i can only hear 10% of the repeaters' traffic! (In section 3.6 below, we evaluate this effect for the N = 1 case, which was analyzed exactly in 3.2 above.)

With the above two assumptions, the correlation among the time lines is reduced; Fig. 6 exhibits instances of these time lines. The analysis follows.

3.4 Analysis

Similarly to the analysis given in Parts II and III [5,6] for slotted ALOHA, we shall treat here the inner-hop and the terminal-to-repeater hop separately.

Consider time line R on which we observe an alternate sequence of transmission periods and idle periods. As in [3], we consider the <u>imbedded slots</u> defined to be the first slot of each idle period (see Fig. 7). The intervals between two consecutive imbedded slots are defined as <u>cycles</u>. 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 population of terminals is completed in that slot (since the repeater immediately relays it) or both. A transmission from T_i is said to be T_i-successful if it is free of collision from other terminals in T_i. It is clear that for a T_i-successful transmission to be successfully received at repeater R_i (considered inactive), this transmission should <u>entirely</u> take place during an <u>idle</u> period of time line R. Consider the imbedded slot t_e

---

*An active repeater is a repeater with a non-empty buffer.

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 idle). J_n is geometrically distributed; its density function is given by

$$\Pr\{J_n = k \text{ slots}\} = (1 - \nu)^{n(k-1)}[1 - (1 - \nu)^n] \qquad (9)$$

With $n^{t_e} = n$, there are N-n inactive repeaters. Let R_{i_j} again denote one of them. By the independence assumptions we regard time line T_{i_j} to be entirely governed by the processes of the infinite population model, where the rate of T_{i_j}-successful packets is λ and the total rate of sense points is G. By the same assumptions, we can reasonably assume that, relative to time line T_{i_j}, the end of a cycle represents a random look in time; accordingly, the probability that this point falls in a transmission period of T_{i_j} is $1 - \lambda/G$, and in an idle period, λ/G; these two instances are shown in Fig. 8. We let Y_{i_j} denote the time since ${}_e$ until the end of the transmission period; its distribution is then given by

$$\Pr\{Y_{i_j} \leq y\} = \frac{\lambda}{G} + \left(1 - \frac{\lambda}{G}\right)\frac{y}{T} \qquad 0 \leq y \leq T \qquad (10)$$

Given that a transmission from T_{i_j} requires T slots*, it is clear that no successful reception at repeater R_i can take place <u>before</u> slot $t_e + Y_{i_j} + T \triangleq t_e + Y'_{i_j}$. From the characterization of successful traffic introduced in section 2.5 above, we note that, following $t_e + Y'_{i_j}$ the arrival process from T_{i_j} to R_{i_j} can be represented by a Bernoulli process, whereby the probability

---

*Consider the slot in this section to represent the time unit; T then represents the number of slots per packet transmission time and a = 1/T; the performance measures however will still be given normalized to T.

of completion of a correct reception in a slot is $a\lambda'$, with $\lambda' = 1/(1/\lambda - 1)$. Without loss of generality, we let $0 \triangleq Y'_{i_0} \leq Y'_{i_1} \leq \ldots \leq Y'_{i_{N-n}} < \infty$. It is clear that for any slot t, $t_e + Y'_{i_j} \leq t < t_e + Y'_{i_{j+1}}$, (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

$$\Pr\{k \text{ packet receptions completed in } t,\ 0 \leq k \leq j\} = \binom{j}{k}(a\lambda')^k(1-a\lambda')^{j-k} \tag{11}$$

To avoid the great complexity involved in treating the problem exactly, we choose here to derive an upper and lower bound on performance by considering much simpler arrival processes. Let $Y'_{min} = Y'_{i_1}$ and $Y'_{max} = Y'_{i_{N-n}}$. The upper bound is obtained by considering the following:

$$\Pr\{k \text{ packet receptions completed in } t,\ 0 \leq k \leq N-n\}$$
$$= \begin{cases} 0 & t < t_e + Y'_{min} \\ \binom{N-n}{k}(a\lambda')^k\ (1-a\lambda')^{N-n-k} & t_e + Y'_{min} \leq t < \infty \end{cases} \tag{12}$$

The lower bound is obtained by substituting Y'_{max} for Y'_{min} in Eq. (12). Let Y'_m denote interchangeably Y'_{min} and Y'_{max} The subscript m will be replaced by min or max where 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. The system state does not vary over a transmission period of time line R.

With these considerations, the transition probabilities between consecutive imbedded points are simply given by

$$P_{0,j} = \begin{cases} \dfrac{N\,(a\lambda')\,(1-a\lambda')^{N-1}}{1-(1-a\lambda')^N} & j=0 \\ 0 & j=1 \\ \dfrac{\binom{N}{k}\,(a\lambda')^k(1-a\lambda')^{N-k}}{1-(1-a\lambda')^N} & j>1 \end{cases} \tag{13}$$

$$P_{N,j} = \begin{cases} 0 & j<N-1 \\ \dfrac{N\nu\,(1-\nu)^{N-1}}{1-(1-\nu)^N} & j=N-1 \\ 1-\dfrac{N\nu(1-\nu)^{N-1}}{1-(1-\nu)^N} & j=N \end{cases} \tag{14}$$

and for $1 \leq n \leq N-1$

$$P_{n,j} = \begin{cases} 0 & j<n-1 \\ \Pr\{J_n < Y'_m\}\dfrac{n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n} + \Pr\{J_n \geq Y'_m\}\dfrac{(1-a\lambda')^{N-n}n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n-1 \\ \Pr\{J_n < Y'_m\}\dfrac{1-n\nu(1-\nu)^{n-1}-(1-\nu)^n}{1-(1-\nu)^n} & \\ \quad +\Pr\{J_n \geq Y'_m\}\dfrac{(1-a\lambda')^{N-n}[1-n\nu(1-\nu)^{n-1}-(1-\nu)^n]+(N-n)a\lambda'(1-a\lambda')^{N-n-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n \\ \Pr\{J_n \geq Y'_m\}\dfrac{(N-n)a\lambda'(1-a\lambda')^{N-n-1}[1-(1-\nu)^n]}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n+1 \\ \Pr\{J_n \geq Y'_m\}\dfrac{\binom{N-n}{j-n}(a\lambda')^{j-n}(1-a\lambda')^{N-j}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j>n+1 \end{cases} \tag{15}$$

We now derive the expressions for $Pr\{J_n \geq Y'_{min}\}$ and $Pr\{J_n \geq Y'_{max}\}$. Given $n^{t_e} = n$, and the distribution for Y_{i_j} given in Eq. (10), we have

$$Pr\{Y'_{max} \leq T + y\} = \left[\frac{\lambda}{G} + (1 - \frac{\lambda}{G})\,\frac{y}{T}\right]^{N-n} \qquad 0 \leq y \leq T \tag{16}$$

$$Pr\{Y'_{min} > T + y\} = \left[(1 - \frac{\lambda}{G})\,(1 - \frac{y}{T})\right]^{N-n} \qquad 0 \leq y \leq T \tag{17}$$

From the distribution of J_n given in Eq. (9) we note that

$$Pr\{J_n \geq k\} = (1 - \nu)^{n(k-1)} \tag{18}$$

Using Eqs. (16) through (18) we have

$$Pr\{J_n \geq Y'_{max}\} = (1-\nu)^{n(T-1)} \left[\left(\frac{\lambda}{G}\right)^{N-n} + \int_{y=o}^{T} (1-\nu)^{ny}\, dPr\{Y'_{max} \leq T+y\}\right] \tag{19}$$

$$Pr\{J_n \geq Y'_{min}\} = (1-\nu)^{n(T-1)} \left[1-\left(1-\frac{\lambda}{G}\right)^{N-n} + \int_{y=o}^{T} (1-\nu)^{ny}\, dPr\{Y'_{min} \leq T+y\}\right] \tag{20}$$

It is shown in Appendix A, that the solution of the above integrals yields the following expressions:

$$\begin{aligned}
Pr\{J_n \geq Y'_{max}\} &= (1-\nu)^{n(T-1)} \left(\frac{\lambda}{G}\right)^{N-n} \\
&+ (1-\nu)^{n(2T-1)}(N-n) \left[\frac{1}{\alpha} + \sum_{k=1}^{N-n-1} (-1)^k \frac{(N-n-1)!}{(N-n-1-k)!}\,\frac{1}{\alpha^{k+1}}\right] \\
&- (1-\nu)^{n(T-1)}(N-n) \left[\frac{(\lambda/G)^{N-n-1}}{\alpha} + \sum_{k=1}^{N-n-1} (-1)^k \frac{(N-n-1)!}{(N-n-1-k)!}\,\frac{(\lambda/G)^{N-n-1-k}}{\alpha^{k+1}}\right]
\end{aligned} \tag{21}$$

$$\Pr\{J_n \geq Y'_{min}\} = (1-\nu)^{n(T-1)}\left[1-(1-\frac{\lambda}{G})^{N-n}\right]$$

$$+ (1-\nu)^{n(T-1)}(N-n)\left[(1-\nu)^{nT}\frac{(N-n-1)!}{\alpha^{N-n}} - \frac{(1-\lambda/G)^{N-n-1}}{\alpha}\right.$$

$$\left. - \sum_{k=1}^{N-n-1}\frac{(N-n-1)!}{(N-n-1-k)!}\,\frac{(1-\lambda/G)^{N-n-1-k}}{\alpha^{k+1}}\right] \qquad (22)$$

where $\alpha = -\dfrac{T\,\mathrm{Log}[(1-\nu)^n]}{(1-\frac{\lambda}{G})}$

Thus we have so far determined that n^{t_e} is a Markov chain and derived its transition probability matrix, which we denote here by P. The chain is irriducible and ergodic and a stationary distribution $\Pi = \{\pi_0, \pi_1, \ldots, \pi_N\}$ exists 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 + I; the cycle length is $I_n + T + 1$. The probability of a successful transmission by the repeaters over the cycle, which we denote by S_n is expressed as

$$S_n = \Pr\{J_n < Y'_m\}\,\frac{n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n}$$

$$+ \Pr\{J_n \geq Y'_m\}\,\frac{n\nu(1-\nu)^{n-1}(1-a\lambda')^{N-n} + (N-n)a\lambda'(1-a\lambda')^{N-n-1}(1-\nu)^n}{1-(1-\nu)^n\,(1-a\lambda')^{N-n}} \qquad (23)$$

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 arrivals may contend as well. Let σ_n denote the average sum of active repeaters over all slots in the cycle. $\overline{I}_n$ denoting the average length of I_n, it is expressed as

$$\sigma_n = (\overline{I}_n + T + 1)n + (T + 1) \Pr\{J_n \geq Y'_m\} \frac{(N-n)\, a\lambda'}{1-(1-\nu)^n\,(1-a\lambda')^{N-n}} \tag{24}$$

By renewal theory arguments, we write the stationary system throughput S and the stationary average number of active repeaters $\overline{n}$ respectively as

$$S = \frac{\sum_{n=o}^{N} \pi_n\, S_n\, T}{\sum_{n=o}^{N} \pi_n (\overline{I}_n + T + 1)} \tag{25}$$

$$\overline{n} = \frac{\sum_{n=o}^{N} \pi_n\, \sigma_n}{\sum_{n=o}^{N} \pi_n (\overline{I}_n + T + 1)} \tag{26}$$

By Little's result, the average network delay D_n is given by

$$D_n = \frac{\overline{n}}{S} \tag{27}$$

We are now left with the determination of $\overline{I}_n$. Given that $Y'_m = y$, the average idle period is simply given by

$$\begin{aligned} \overline{I}_{n/Y'_m=y} &= \Pr\{J_n < y\}\, \overline{I}_{n/J_n<y, Y'_m} = y \\ &+ \Pr\{J_n \geq y\} \left[y + \frac{1}{1-(1-\nu)^n (1-a\lambda')^{N-n}} \right] \end{aligned} \tag{28}$$

Let $n \neq 0,N$ we first have

$$\Pr\{J_n < y\} = 1 - \Pr\{J_n > y - 1\} = 1 - (1-\nu)^{n(y-1)} \tag{29}$$

For $0 \leq k \leq y-1$ we have

$$\Pr\{I_n = k \,/\, Y'_m = y,\ J_n < y\} = \frac{(1-\nu)^{n(k-1)}[1-(1-\nu)^n]}{1-(1-\nu)^{n(y-1)}} \tag{30}$$

The average idle period in this case is

$$\bar{I}_{n/Y'_m = y, J_n < y} = \sum_{k=0}^{y-1} \frac{k(1-\nu)^{n(k-1)}[1-(1-\nu)^n]}{1-(1-\nu)^{n(y-1)}}$$

$$= \frac{1-(1-\nu)^{ny} - y(1-\nu)^{n(y-1)}[1-(1-\nu)^n]}{[1-(1-\nu)^n][1-(1-\nu)^{n(y-1)}]} \tag{31}$$

Thus, for $n \neq 0,N$,

$$\bar{I}_{n/Y'_m = y} = \frac{1-(1-\nu)^{ny} - y(1-\nu)^{n(y-1)}[1-(1-\nu)^n]}{1-(1-\nu)^n} + y(1-\nu)^{n(y-1)} + \frac{(1-\nu)^{n(y-1)}}{1-(1-\nu)^n(1-a\lambda')^{N-n}}$$

$$= \frac{1}{1-(1-\nu)^n} + (1-\nu)^{n(y-1)}\left[\frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}} - \frac{(1-\nu)^n}{1-(1-\nu)^n}\right] \tag{32}$$

Removing the condition Y'_m, we finally have

$$\bar{I}_n = \frac{1}{1-(1-\nu)^n} + \Pr\{J_n > Y'_m\}\left[\frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}} - \frac{(1-\nu)^n}{1-(1-\nu)^n}\right] \tag{33}$$

where Y'_m can be replaced by either Y'_{min} or Y'_{max}.

When $n = 0$, $\Pr\{J_n < y\} = 0$ and Eq. (28) is written as

$$\bar{I}_{o/Y'_m = y} = y + \frac{1}{1-(1-a\lambda')^N} \tag{34}$$

Removing the condition on Y'_m, we get for the lower bound case

$$\bar{I}_{o,max} = \frac{1}{1-(1-a\lambda')^N} + T + N(1-\frac{\lambda}{G})\frac{1}{T}\int_0^T y\left[\frac{\lambda}{G} + (1-\frac{\lambda}{G})\frac{y}{T}\right]^{N-1} dy$$

$$= \frac{1}{1-(1-a\lambda')^N} + T + N(1-\frac{\lambda}{G})\, T\left[\sum_{k=o}^{N-1}\binom{N-1}{k}\frac{(\lambda/G)^k(1-\lambda/G)^{N-1-k}}{N-1-k+2}\right] \tag{35}$$

and for the upper bound case

$$\bar{I}_{o,min} = \frac{1}{1-(1-a\lambda')^N} + T + N(1-\frac{\lambda}{G})\frac{1}{T}\int_0^T y\left[(1-\frac{\lambda}{G}) - (1-\frac{\lambda}{G})\frac{y}{T}\right]^{N-1} dy$$

$$= \frac{1}{1-(1-a\lambda')^N} + T + N(1-\frac{\lambda}{G})^N\, T\left[\sum_{k=0}^{N-1}\binom{N-1}{k}\frac{(-1)^{N-1-k}}{N-1-k+2}\right] \tag{36}$$

For the case $n = N$, we simply have

$$\bar{I}_N = \frac{1}{1-(1-\nu)^N} \tag{37}$$

This completes the inner-hop analysis.

3.5 Calculation of D_a

As with slotted ALOHA networks in Part II [5], we estimate here the access delay D_a by

$$D_a = \frac{1}{1-B}\, D_{NPCSMA}(\lambda) + \frac{B}{1-B}\,\delta(\lambda) \tag{38}$$

where $D_{NPCSMA}(\lambda)$ 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 and is given in section 2.4, and B is the probability that a T_i-successful packet gets blocked at the receiving repeater and is expressed as

$$B = 1 - \frac{S}{N\lambda} \tag{39}$$

IV. Discussion and Results

4.1 Note on the Effect of Assumption 4

Consider the simple case of N = 1. We evaluate here the effect of assumption 4 by giving an approximate analysis and then compare the results to those obtained by the exact analysis given in section 3.2 above. With N = 1, the state of the system at all imbedded points is trivially $n^{t_e} = 0$. The average idle period is given by Eq. (35) or Eq. (36)

$$\bar{I}_0 = \frac{1}{a\lambda}T + T + \left(1 - \frac{\lambda}{G}\right)\frac{T}{2} = \left(\frac{1}{\lambda} + \frac{1-\lambda/G}{2}\right)T \tag{40}$$

The probability of success of a transmission on time line R is 1. The network throughput, using the normalized time units, is simply

$$S = \frac{1}{\frac{1}{\lambda} + \frac{1-\lambda/G}{2} + 1 + a} \tag{41}$$

where λ and G are related through Eq. (2). We plot S versus G in Fig. 2 above as curve C. With this analysis we have been able to predict a system capacity of only 0.32 versus 0.46 obtained by the exact analysis. Thus the independence assumption in this worse case of N = 1 has provided an error of 30% in predicting the capacity. As pointed out earlier, the error is due to the excessive transmissions by the terminal population which would not be present had we taken the proper connectivity into account; the error is represented by the term $\frac{1-\lambda/G}{2}$ which represents the additional waiting until an arrival at the repeater is successful; near capacity, this term approaches one half of a transmission time since λ/G is then small; that is, the random look falls during a transmission period with a high probability. It is not surprising to note that the removal of this term will lead to an expression for S in terms of G which is identical to the one obtained through the exact analysis, namely Eq. (8)! The effect of assumption 4 will be much smaller for $N = 2$, since normally terminals in a given population will be out of range of one repeater, that is of half of the repeaters' traffic. With $N = 10$, the results will be even more accurate. At any rate, these results represent a pessimistic evaluation of performance.

4.2 Discussion of Numerical Results

We show in Table 1 numerical results obtained for various values of the parameters N, λ, and ν. The numerical results have shown the following.

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. The access delay, on the other hand, is the predominant factor in packet delays as the throughput increases.

We explain these points by noting 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. Moreover, 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 other protocols than IFT.)

3. The difference between the lower bound and upper bound on S and B becomes important as λ increases. The discrepancy is even more important for larger values of N. Thus, no good estimate is obtained for S (and B). Network delay, however, is not affected by the approximation and an accurate estimate is obtained by this analysis.

Table 1

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	[illegible]
		0.01	1.013	1.013	0.1535	0.1513	0.232	[illegible]
		0.1	1.012	1.012	0.1535	0.1513	0.232	[illegible]
		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 numberical results, we observe that the stationary distributions Π_{min} and Π_{max} are "identical"* for the optimum $(\nu = 0.1)$ the probability of success $[(P_s)_n]_{min}$ and $[(P_s)_n]_{max}$ are also very close to each other and close to 1; the average idle periods $[\bar{I}_n]_{min}$ and $[\bar{I}_n]_{max}$, on the contrary, show important differences affecting significantly the performance evaluation. To overcome this difficulty we recourse to simulation and estimate separately, $(P_s)_n$ and $\bar{I}_n$ for $n = 0,1,\ldots,N$; then using Π_{min} or Π_{max} we derive the performance measures. The simulation of the subprocesses I_n (and $(P_s)_n$) is a much simpler task than a complete simulation of the system. The details follow.

4.3 Simulation of I_n and $(P_s)_n$

Let $n^{te} = n$. The algorithm used to generate one sample of I_n, $(P_s)_n$ and σ_n is as follows.

1. Generate N-n random variables $\{Y'_{i_j}\}_{j=1}^{N-n}$ according to the distribution given in Eq. (10). Without loss of generality, we assume that

$$0 = Y'_{i_0} \leq Y'_{i_1} \leq Y'_{i_2} \ldots \leq Y'_{i_{N-n}} \leq Y'_{i_{N-n+1}} = \infty$$

2. $j \leftarrow 0$

3. Generate a random variable J_n^j such that

$$Pr\{J_n^j = k\} = \left[(1-\nu)^n (1-a\lambda')^j\right]^{(k-1)} \left[1 - (1-\nu)^n(1-a\lambda')^j\right] \tag{42}$$

If $J_n^j < Y'_{i_{j+1}} - Y'_{i_j}$ then do;

$$I_n = Y'_{i_j} + J_n^j \tag{43}$$

---

*Accurate within 4 decimals (the accuracy used in printing the results).

$$(P_s)_n = \frac{n\nu(1-\nu)^{n-1}(1-a\lambda')^j + j\ a\lambda'(1-a\lambda')^{j-1}(1-\nu)^n}{1-(1-\nu)^n\ (1-a\lambda')^j} \tag{44}$$

$$\sigma_n = (I_n + T + 1)n + \frac{a\lambda'(T+1)}{1-(1-\nu)^n(1-a\lambda')^j} \tag{45}$$

stop;

else $j \leftarrow j + 1$; repeat this step.

Let L be the number of samples needed. The algorithm is repeated L times. The estimates for $\overline{I}_n$, $(P_s)_n$, σ_n , denoted by $(\overline{I}_n)_{sim}$, $[(P_s)_n]_{sim}$, $(\sigma_n)_{sim}$ 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 Eqs. (25), (26) and (27) in which we substitute $(\overline{I}_n)_{sim}$, $[(P_s)_n]_{sim}$, $(\sigma_n)_{sim}$ for I_n , $(P_s)_n$ and σ_n respectively.

4.4 The Throughput-Delay Tradeoff; CSMA versus Slotted ALOHA

We plot in Fig. 9 the throughput-delay tradeoff for N=2, 5 and 10. We note a slight improvement as N increases. Contrary to the slotted ALOHA case [5,6] 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 also 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; the access delay D_a is also smaller with smaller λ.

As for the comparison between the performances of slotted ALOHA and CSMA networks, we summarize in Figs. 10, 11 and 12 the throughput-delay tradeoff for all systems considered in this series of notes for N = 2, 5 and 10 respectively. We note that CSMA offers an improvement over slotted ALOHA, which becomes more significant as N increases. Fig. 13 displays the system capacity.

Appendix A

Derivation of $Pr\{J_n \geq Y'_{max}\}$ and $Pr\{J_n \geq Y'_{min}\}$

$$\text{(i)}\qquad Pr\{J_n \geq Y'_{max}\} = (1-\nu)^{n(T-1)}\left[\left(\frac{\lambda}{G}\right)^{N-n} + \int_{y=o}^{T} (1-\nu)^{ny}\, dPr\{Y'_{max} \leq T + y\}\right]$$

$$= (1-\nu)^{n(T-1)}\left[\left(\frac{\lambda}{G}\right)^{N-n}\right.$$

$$\left. + (N-n)\left(1 - \frac{\lambda}{G}\right)\frac{1}{T}\int_0^T (1-\nu)^{ny}\left[\frac{\lambda}{G} + \left(1 - \frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-n-1} dy\right] \tag{A.1}$$

We note that the integral is of a known form

$$\int x^m \,\ell^{\alpha x}\, d = e^{\alpha x}\left[\frac{x^m}{\alpha} + \sum_{k=1}^{m} (-1)^k \frac{m(m-1)\ldots(m-k+1)}{\alpha^{k+1}} x^{m-k}\right] \tag{A.2}$$

Define the following:

$$\mu = (1-\nu)^n \tag{A.3}$$

$$m = N-n-1 \tag{A.4}$$

$$\beta = \lambda/G \tag{A.5}$$

$$\gamma = (1-\lambda/G)/T \tag{A.6}$$

$$x = \beta + \gamma y \tag{A.7}$$

$$\alpha = \frac{\text{Log}\ \mu}{\gamma} \tag{A.8}$$

then the integral in Eq. (A.1) above becomes

$$\int_0^T (1-\nu)^{ny} \left[\frac{\lambda}{G} + (1-\frac{\lambda}{G}) \frac{y}{T}\right]^{N-n-1} dy$$

$$= \frac{e^{-\alpha\beta}}{\gamma} \int_\beta^1 x^m e^{\alpha x} dx$$

$$= \frac{e^{-\alpha\beta}}{\gamma} \left\{ e^{\alpha} \left[\frac{1}{\alpha} + \sum_{k=1}^{m} (-1)^k \frac{m!}{(m-k)!} \frac{1}{\alpha^{k+1}} \right] - e^{\alpha\beta} \left[\frac{\beta}{\alpha} + \sum_{k=1}^{m} (-1)^k \frac{m!}{(m-k)!} \frac{\beta^{m-k}}{\alpha^{k+1}} \right] \right\} \quad \text{(A.9)}$$

Substituting this expression in Eq. (A.1), and replacing m, β, and γ by their expressions given in Eqs. (A.4), (A.5) and (A.6) respectively, we get Eq. (21)

(ii) $$\Pr\{J_n \geq Y'_{min}\} = (1-\nu)^{n(T-1)} \left[1-(1-\frac{\lambda}{G})^n + \int_0^T (1-\nu)^{ny} \, d\Pr\{Y'_{min} \leq T + y$$

$$= (1-\nu)^{n(T-1)} \left[1-(1-\frac{\lambda}{G})^n \right.$$

$$\left. + (N-n)(1-\frac{\lambda}{G}) \frac{1}{T} \int_0^T (1-\nu)^{ny} \left[1-\frac{\lambda}{G} - (1-\frac{\lambda}{G}) \frac{y}{T}\right]^{N-n-1} dy \right]$$

(A.10)

Define $\beta' = 1 - \beta$ (A.11)

$\gamma' = -\gamma$ (A.12)

$x = \beta' + \gamma' y$ (A.13)

$\alpha' = -\alpha$ (A.14)

then the integral in Eq. (A.10) above is again of the form given in Eq. (A.2)

$$\int_0^T (1-\nu)^{ny} \left[1-\frac{\lambda}{G} - \left(1-\frac{\lambda}{G}\right)\frac{y}{T}\right]^{N-n-1} dy$$

$$= \frac{e^{-\alpha'\beta'}}{\gamma'} \int_\beta^0 x^m e^{\alpha' x}\, dx$$

$$= \frac{e^{-\alpha'\beta'}}{\gamma'} \left\{(-1)^m \frac{m!}{(\alpha')^{m+1}} - e^{\alpha'\beta'} \left[\frac{(\beta')^m}{\alpha'} + \sum_{k=1}^{m} (-1)^k \frac{m!}{(m-k)!} \frac{(\beta')^{m-k}}{(\alpha')^{k+1}}\right]\right\}$$

(A.15)

Substituting this for the integral in Eq. (A.10) and replacing m, β', and γ' by their expressions as defined in Eqs. (A.4), (A.11) and (A.12) respectively, we obtain Eq. (22).

References

[1] Kleinrock, L. and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and their Throughput Delay Characteristics," IEEE Transactions on Communications, Vol. COM-23, No. 12, December 1975, pp. 1400-1416.

[2] Tobagi, F. 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 Transactions on Communications, Vol. COM-23, Vol. 12, December 1975, pp. 1417-1433.

[3] Tobagi, F. and L. Kleinrock, "Packet Switching in Radio Channels: Part IV - Stability Considerations and Dynamic Control in Carrier Sense Multiple Access," IEEE Transactions on Communications, Vol. COM-25, No. 10, pp. 1103-1120, October 1977.

[4] Tobagi, F., "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," Ph.D. Dissertation, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, rep. UCLA-ENG-7499, December 1974.

[5] Tobagi, F., "On the Performance Analysis of Multihop Packet Radio Systems: Part II - Star Configurations Employing Slotted ALOHA," Packet Radio Temporary Note #247, SRI International, Menlo Park, California.

[6] Tobagi, F., "On the Performance Analysis of Multihop Packet Radio Systems: Part III - Fully Connected Configurations Employing Slotted ALOHA," Packet Radio Temporary Note #248, SRI International, Menlo Park, California.

[7] Lam, S., "Packet Switching in a Multiaccess Broadcast Channel with Applications to Satellite Communication in a Computer Network," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-Eng-7429. April 1974 (also published as Ph.D. Dissertation).

[8] Kleinrock, L. and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Transactions on Communications, Vol. COM-23, pp. 410-423, April 1970.

[9] Tobagi, F.A. and L. Kleinrock, "Packet Switching in Radio Channels: Part III - Polling and (Dynamic) Split-Channel Reservation Multiple Access," IEEE Transactions on Communications, Vol. COM-24, No. 8, August 1976, pp. 832-845.

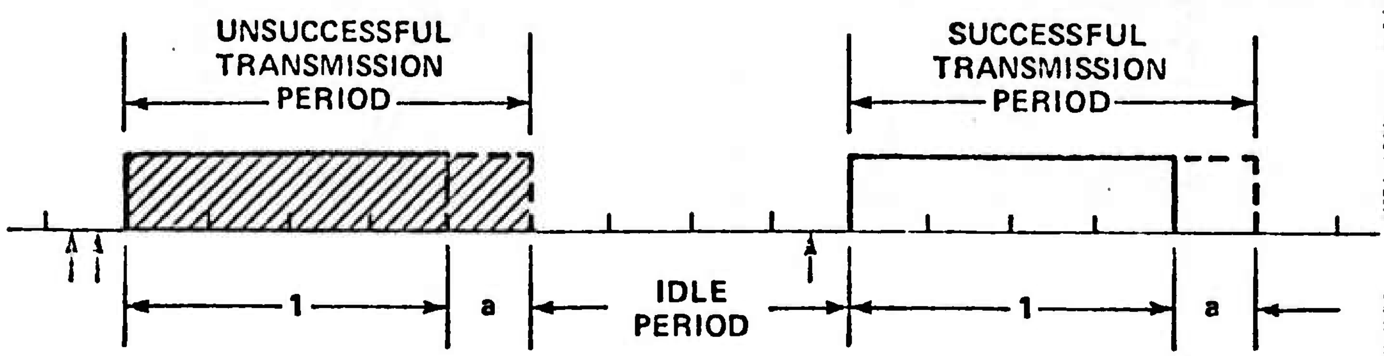


Figure 1 Slotted Nonpersistent CSMA: Transmission and Idle Periods.

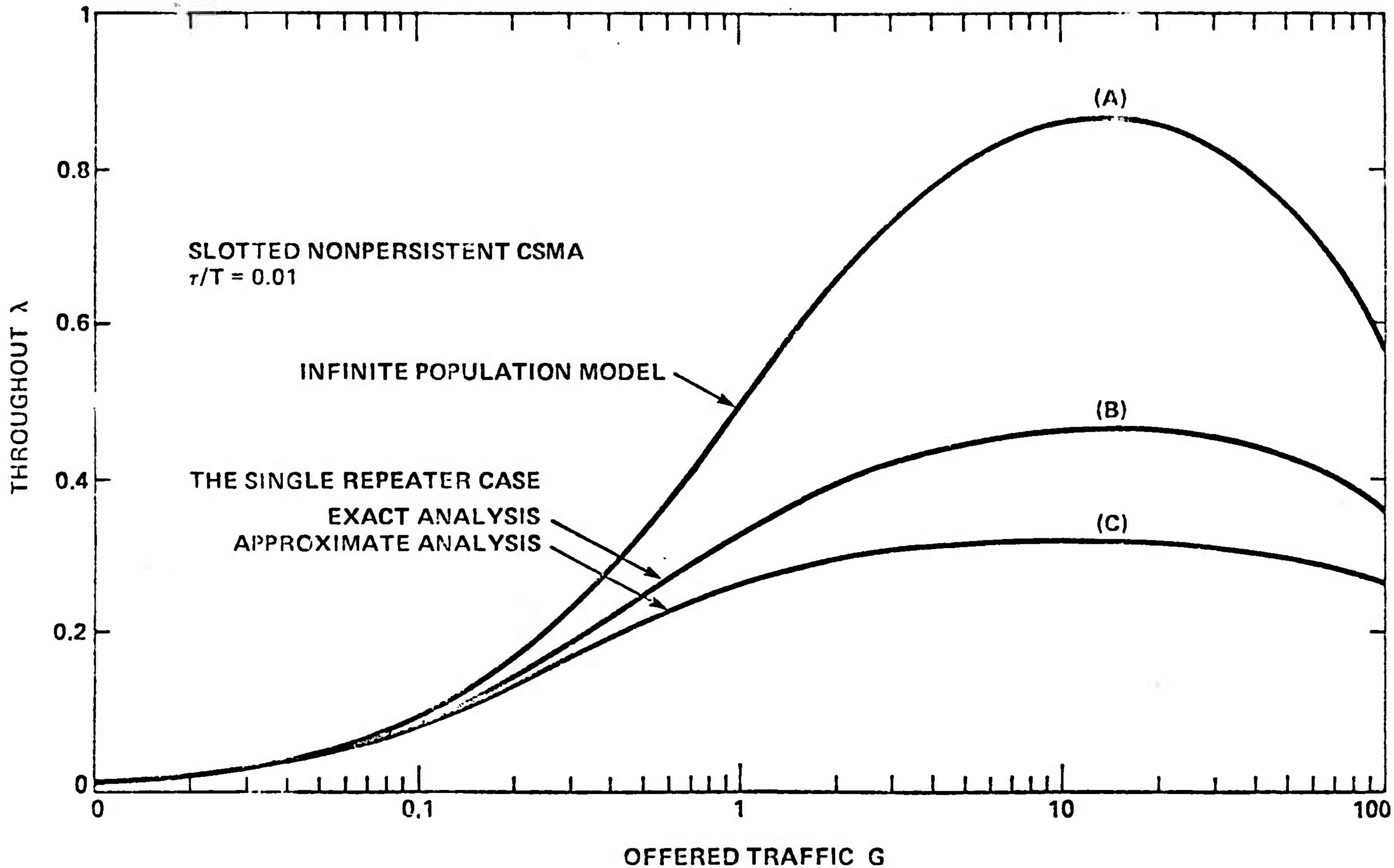


Figure 2 Nonpersistent CSMA: Throughput versus Offered Channel Traffic.

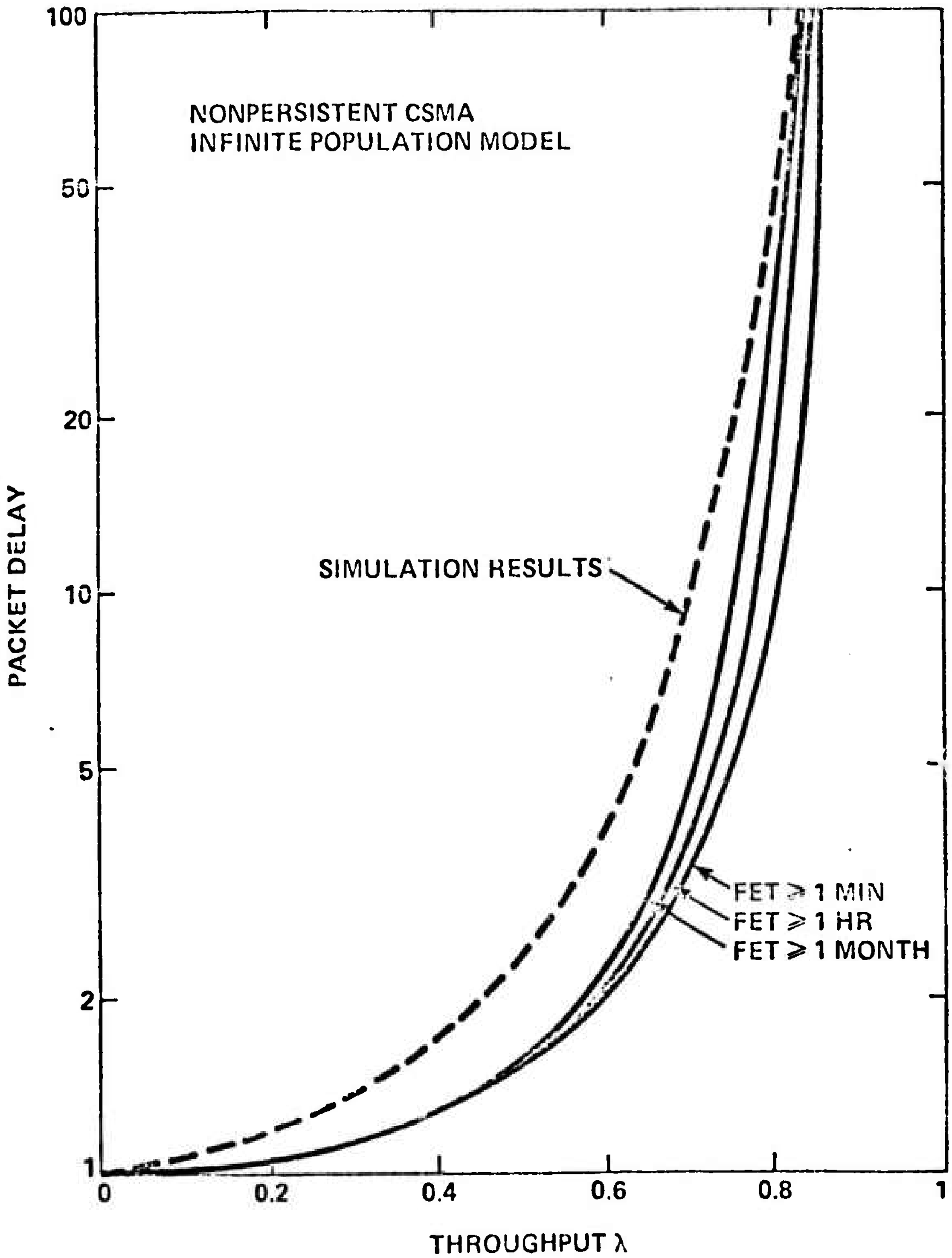


Figure 3 Nonpersistent CSMA: Throughput-Delay Tradeoff for the Infinite Population (a = 0.01).

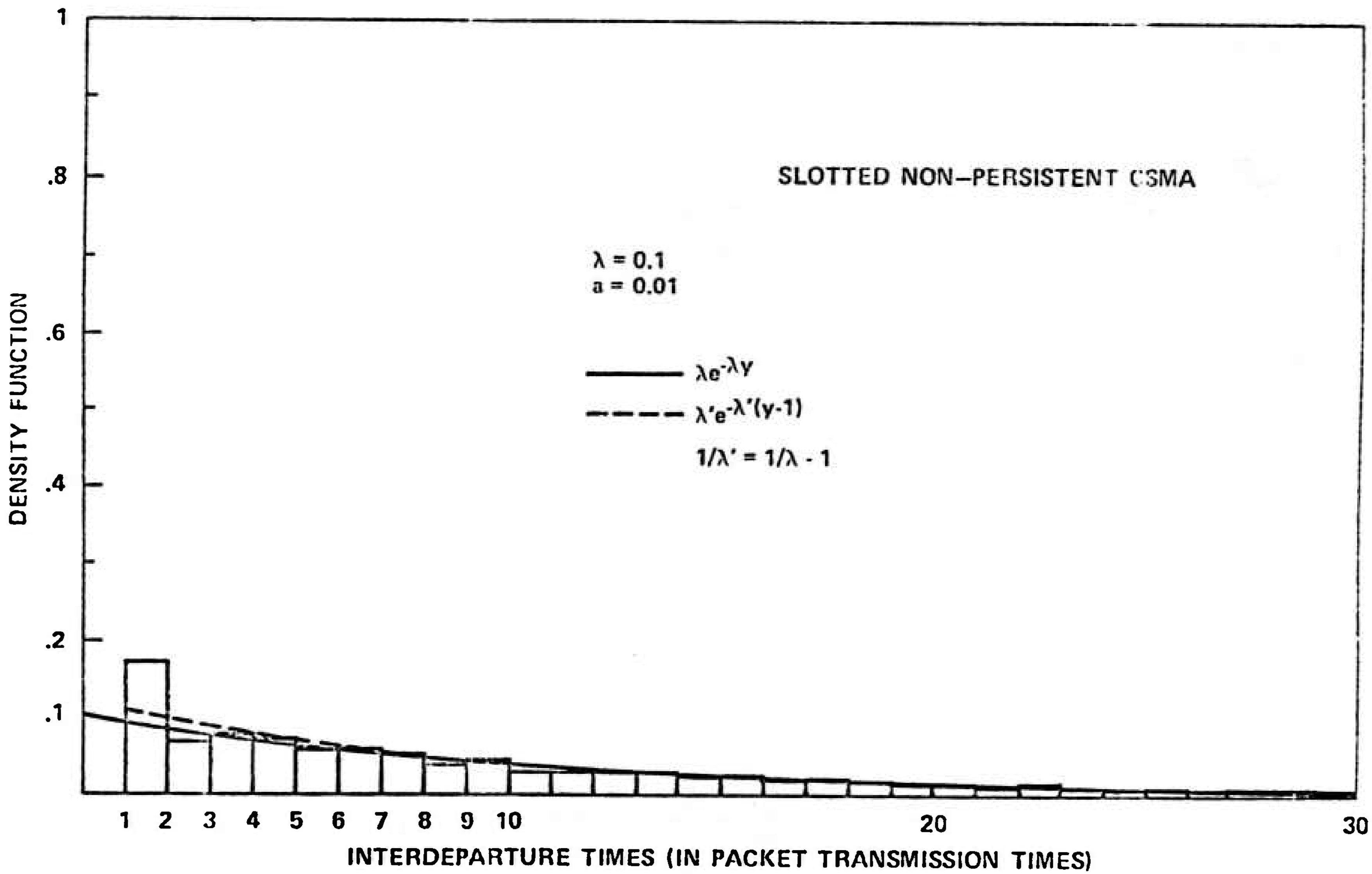


Figure 7-6(a) Histograms of Interdeparture Times in Slotted Non-Persistent CSMA (S_r=0.1).

Figure 4(a) Histograms of Interdeparture Times in Slotted Nonpersistent CSMA (λ = 0.1).

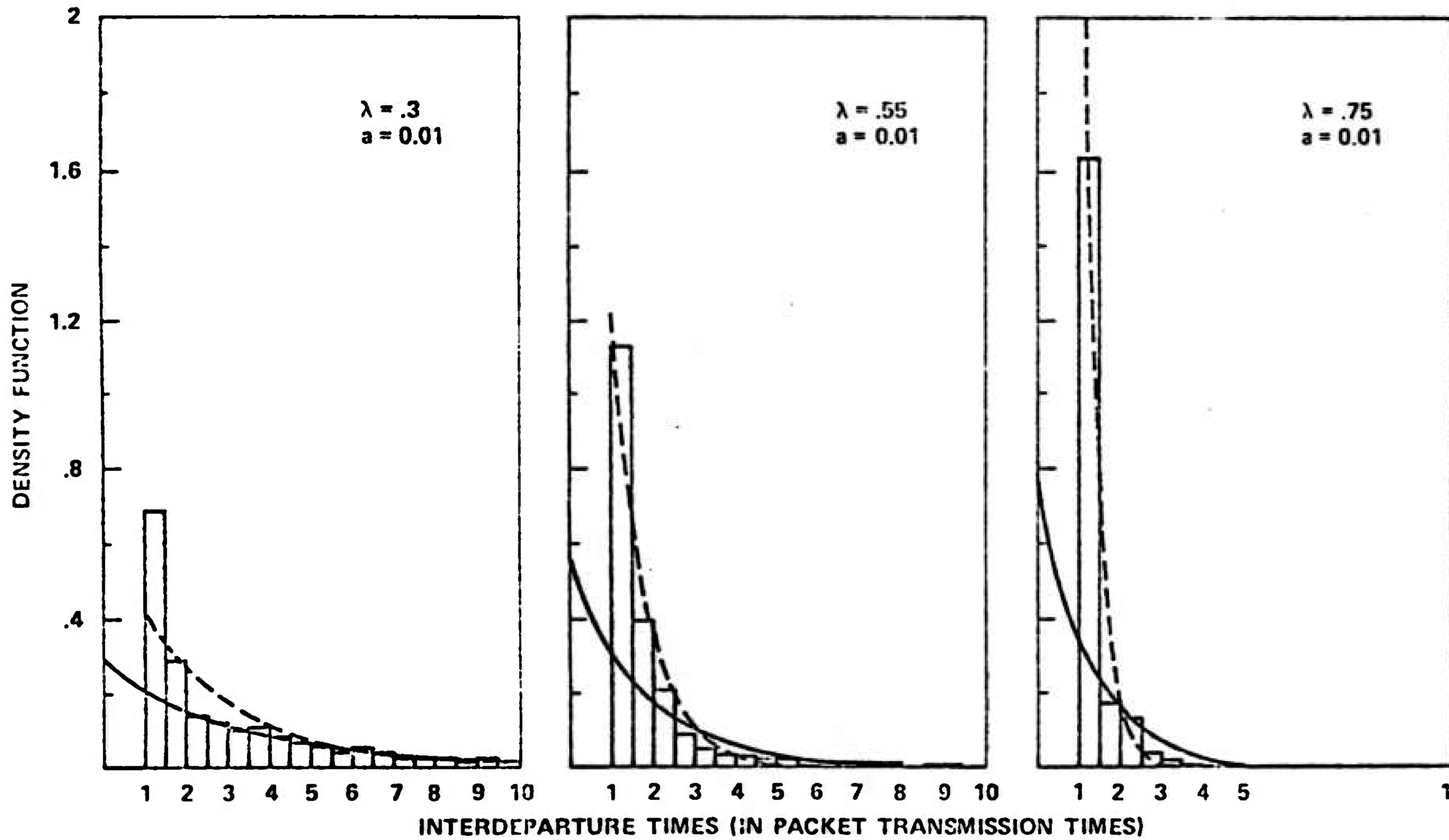


Figure 4(b) Histograms of Interdeparture Times in Slotted Nonpersistent CSMA (λ = 0.3, 0.55 and 0.75).

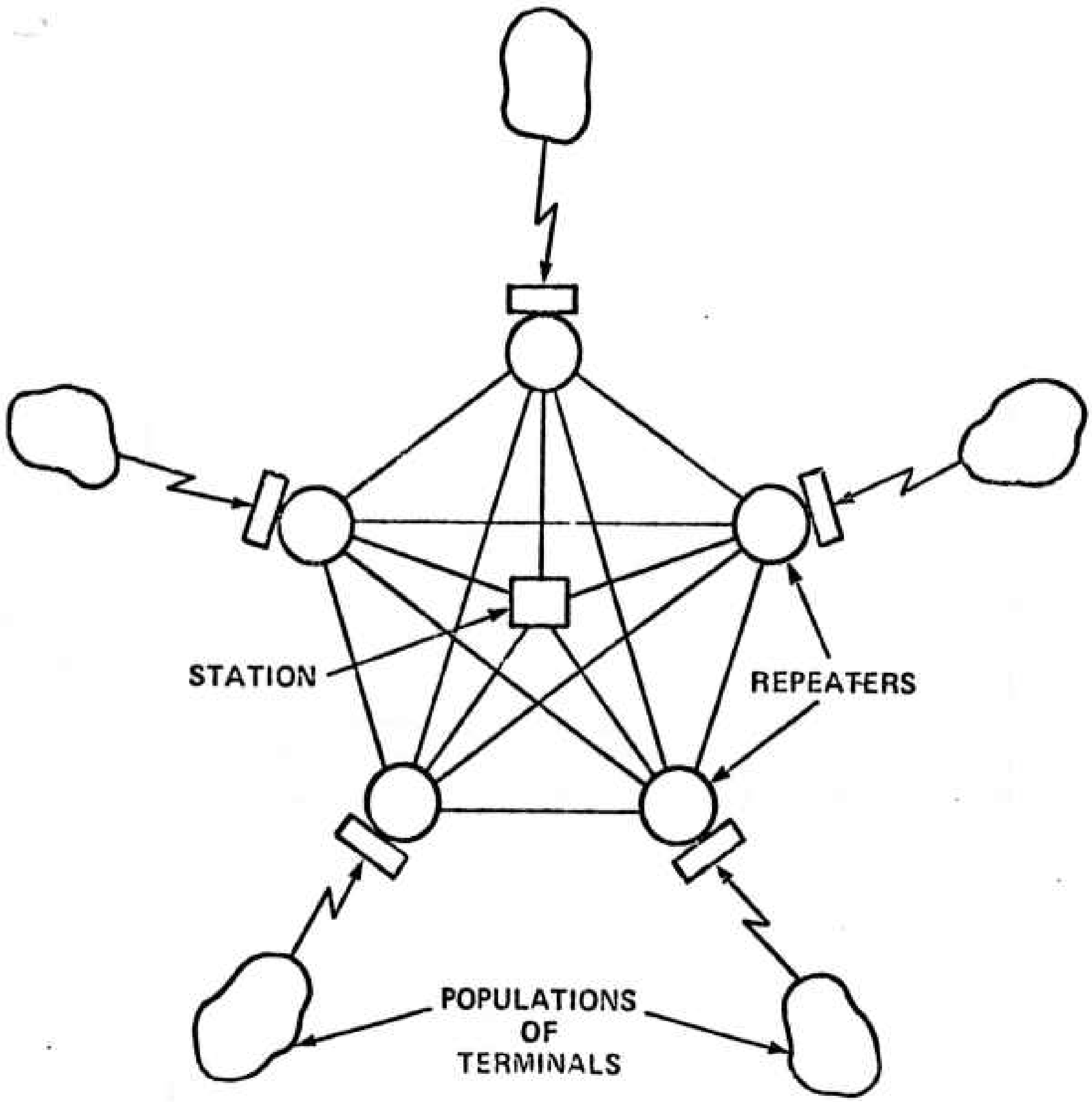


Figure 5 A Two-Hop Fully-Connected Configuration.

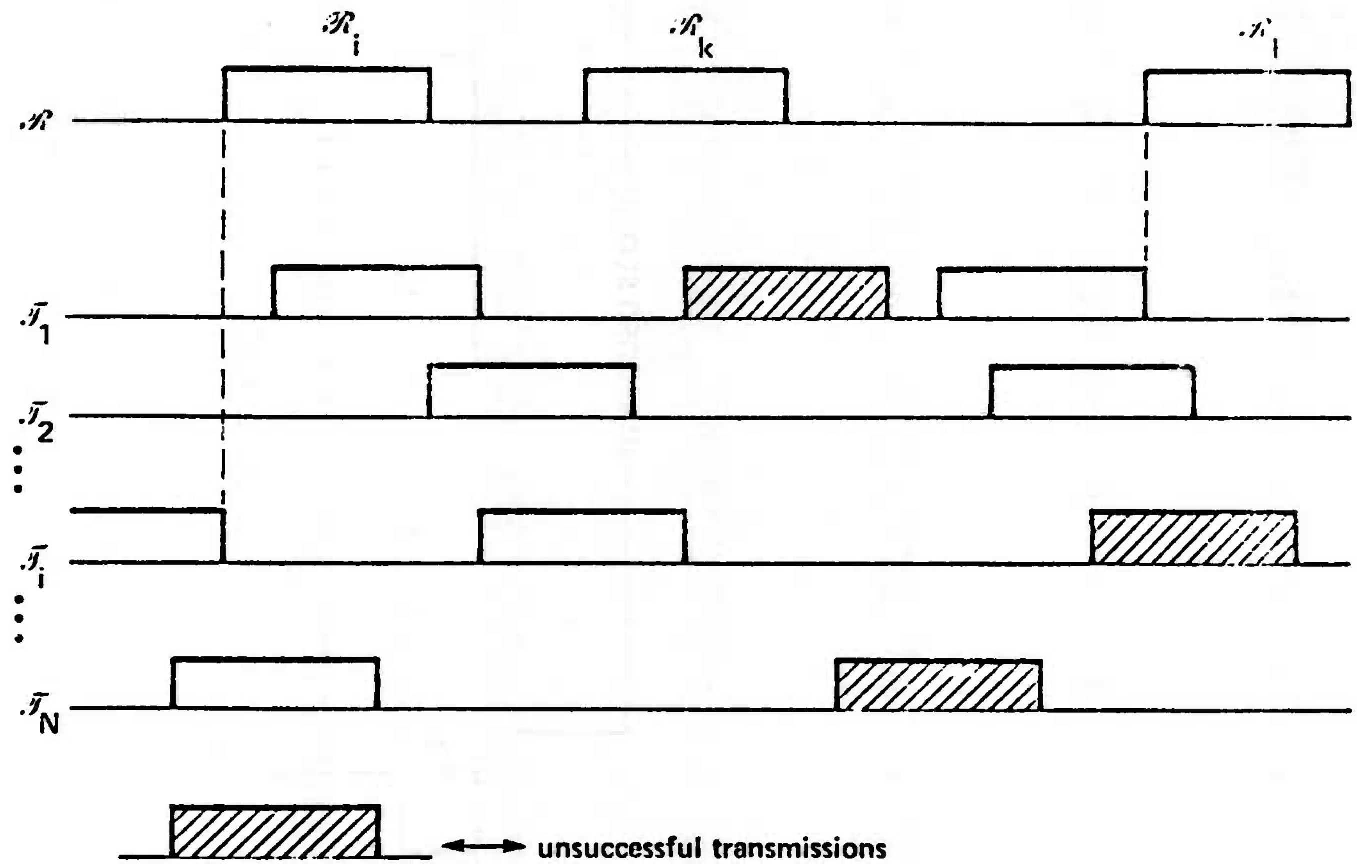


Figure 6 R and T_i Time Lines Exhibiting Packet Transmissions.

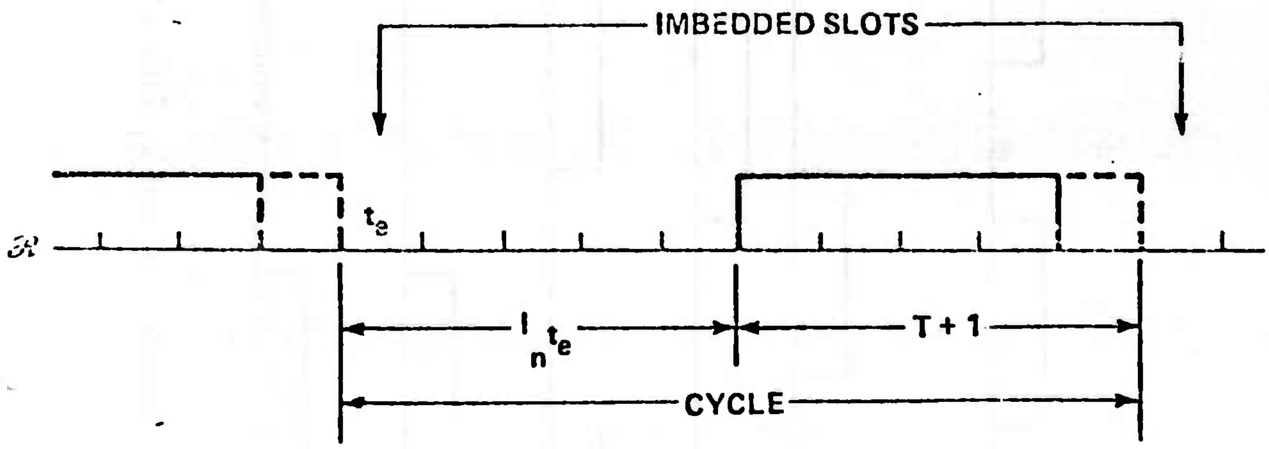


Figure 7 The Imbedded Slots on ' ime Line R.

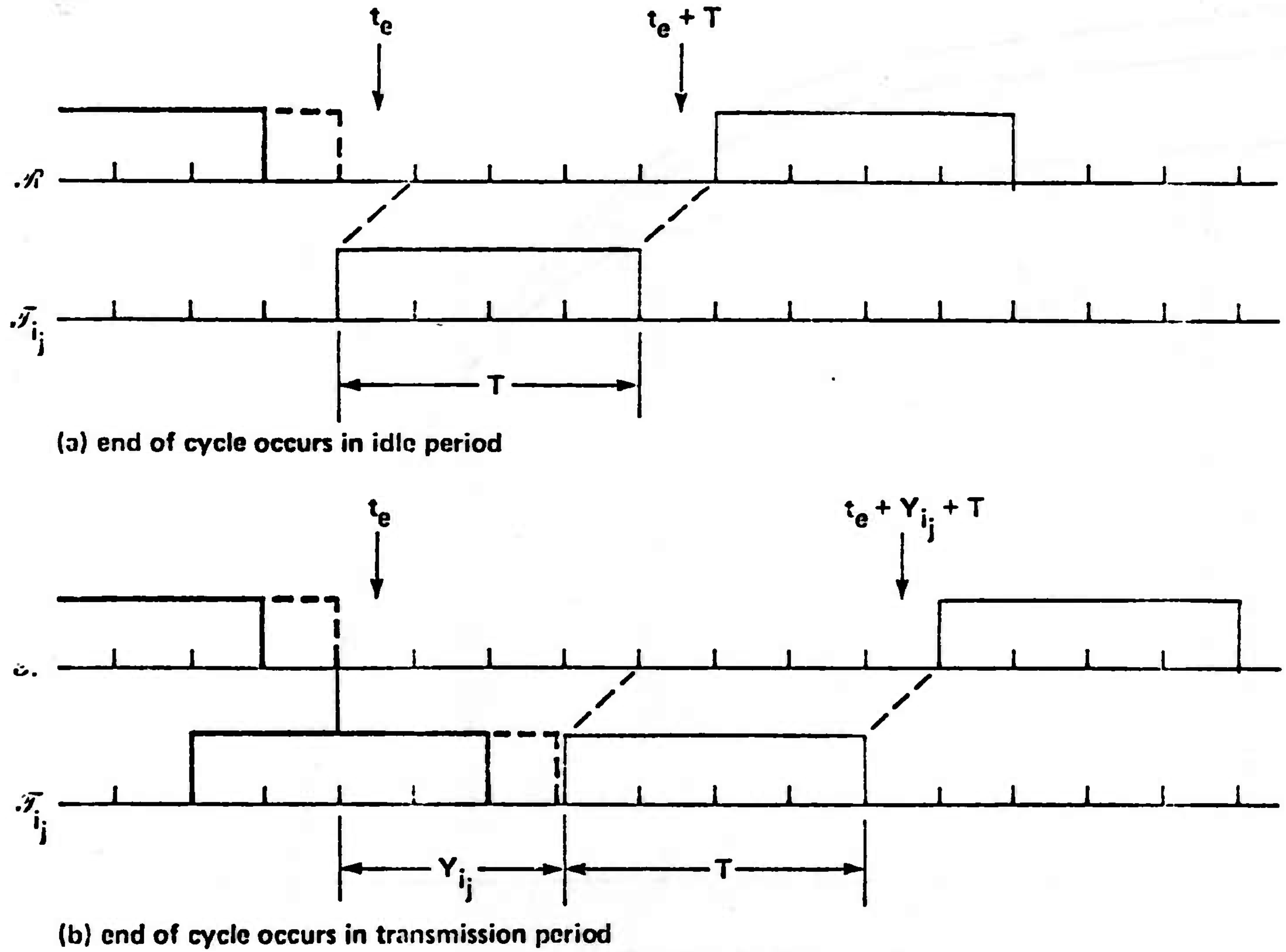


Figure 8 Determination of Minimum Idle Time for Possible Completion of Packet Reception.

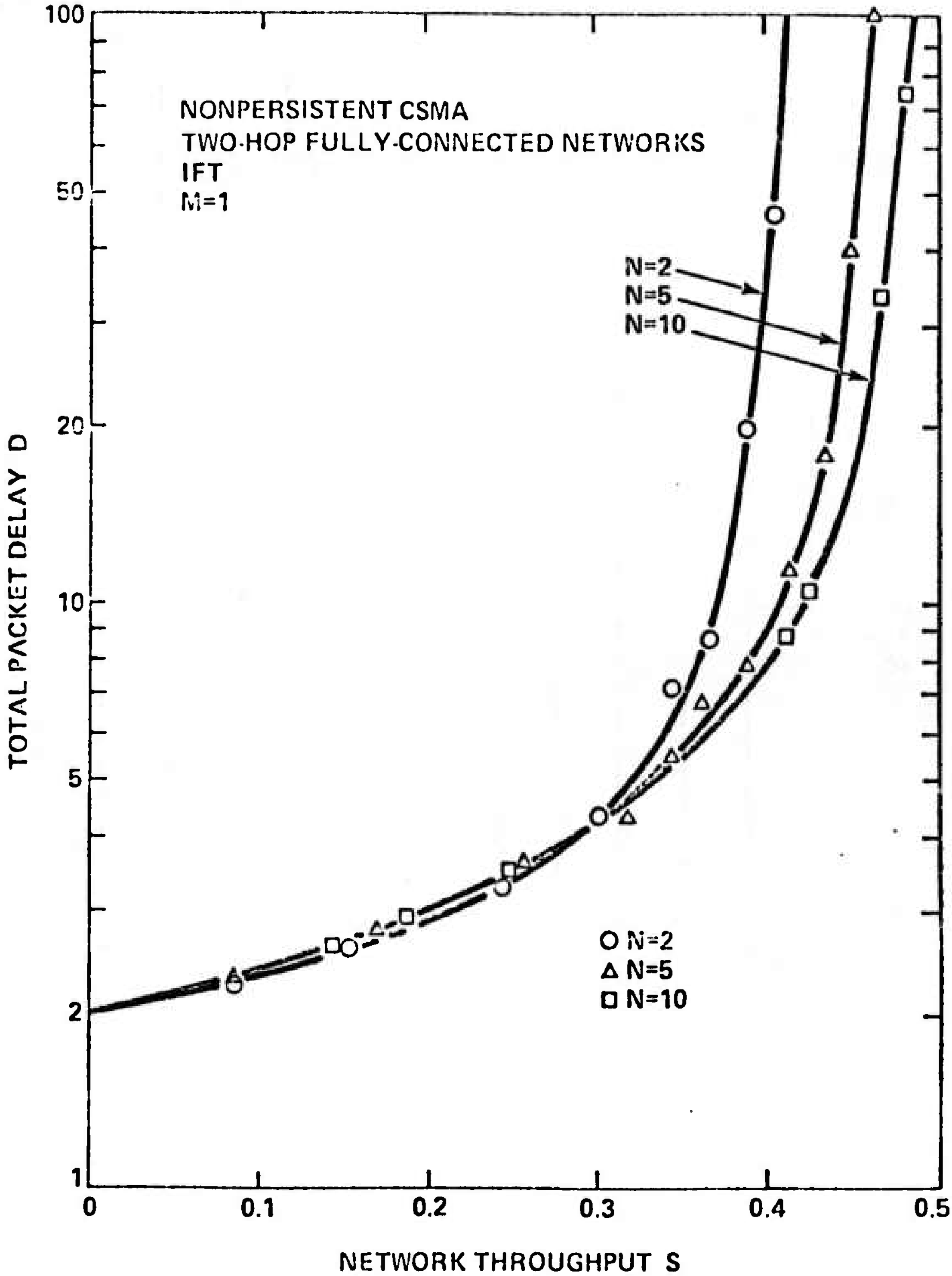


Figure 9 Throughput-Delay Tradeoffs in Nonpersistent CSMA Two-Hop Fully-Connected Networks (a = 0.01).

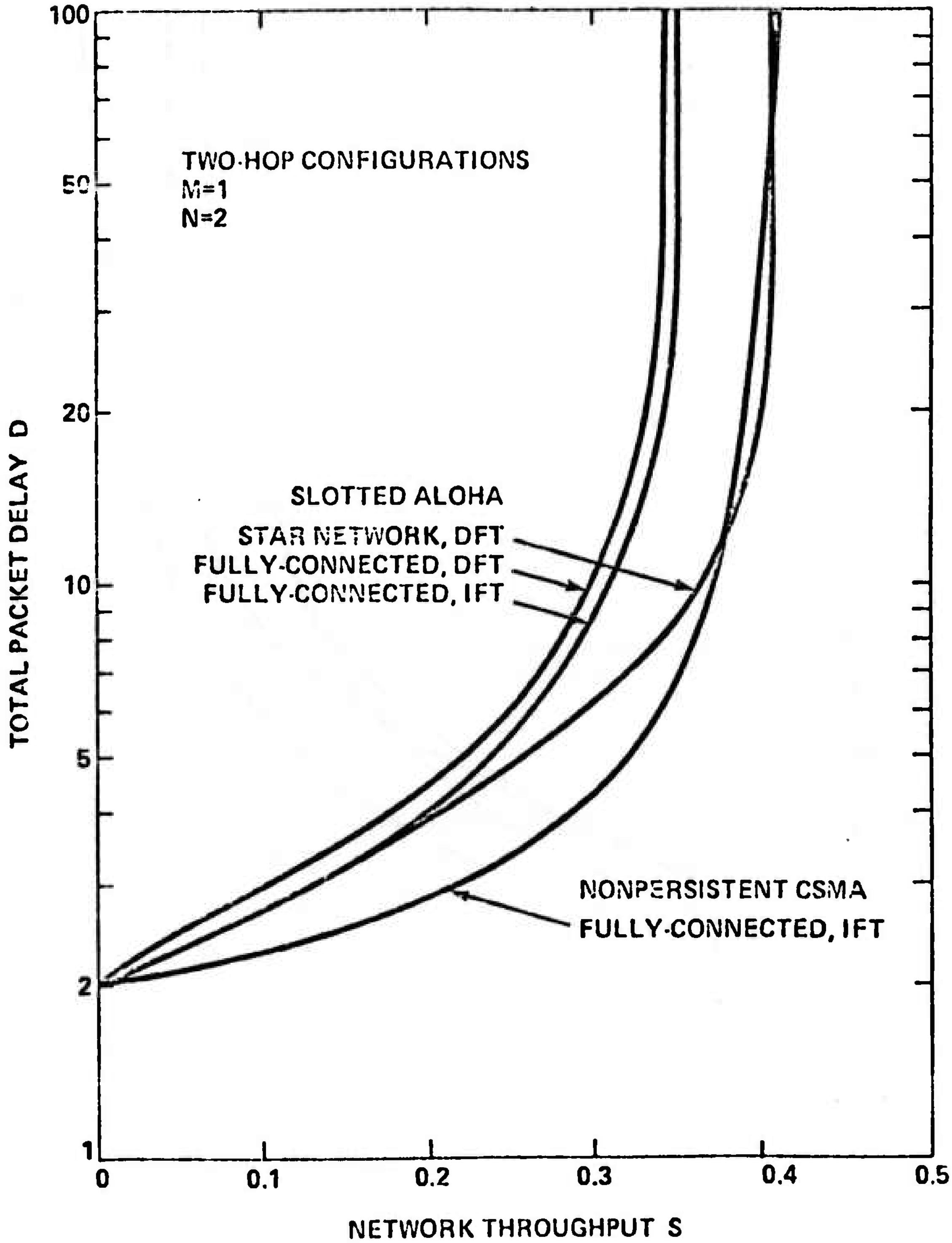


Figure 10 Comparison between Slotted ALOHA and Nonpersistent CSMA (a = 0.01) for N = 2.

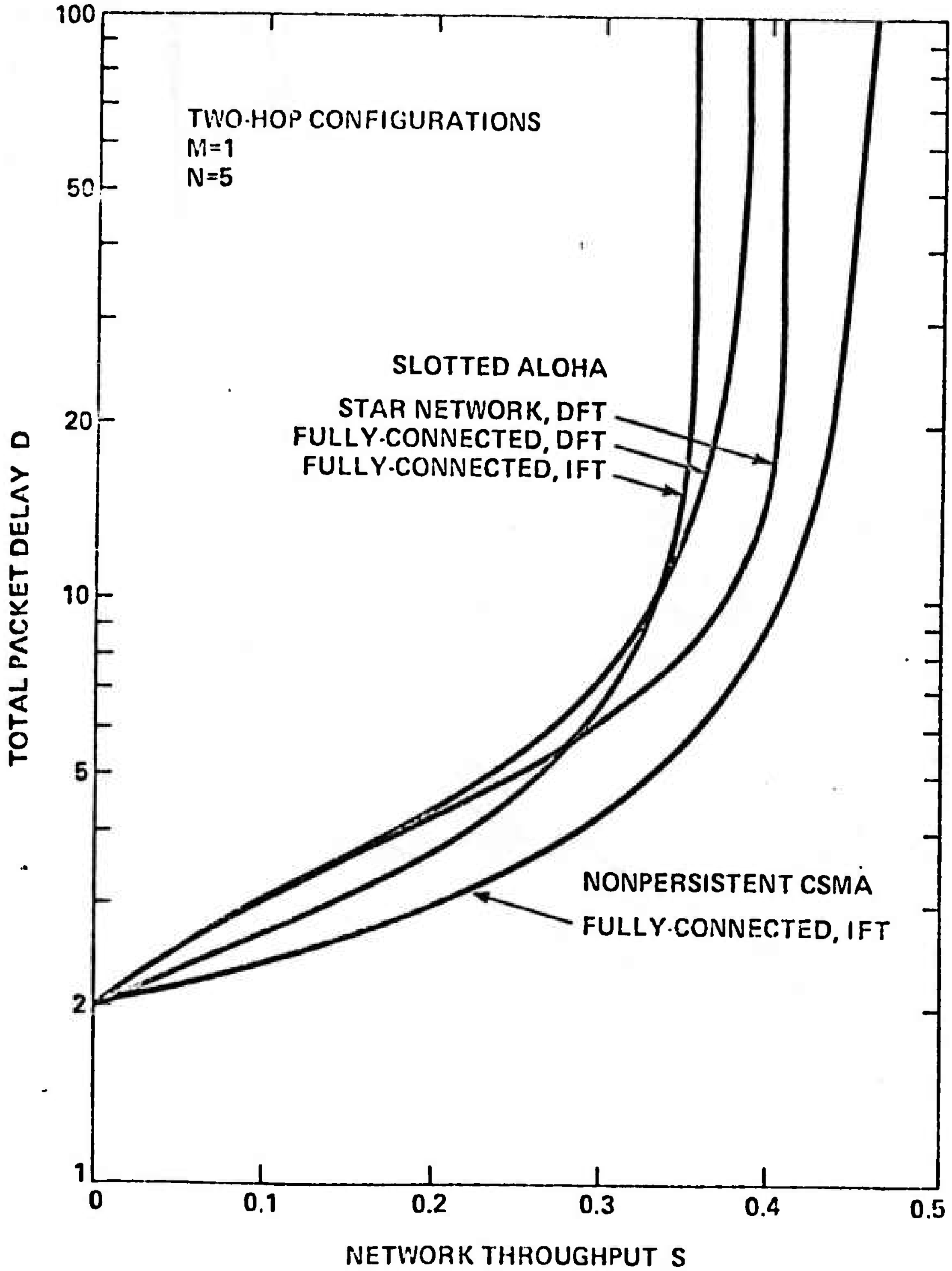


Figure 11 Comparison between Slotted ALOHA and Nonpersistent CSMA (a = 0.01) for N = 5.

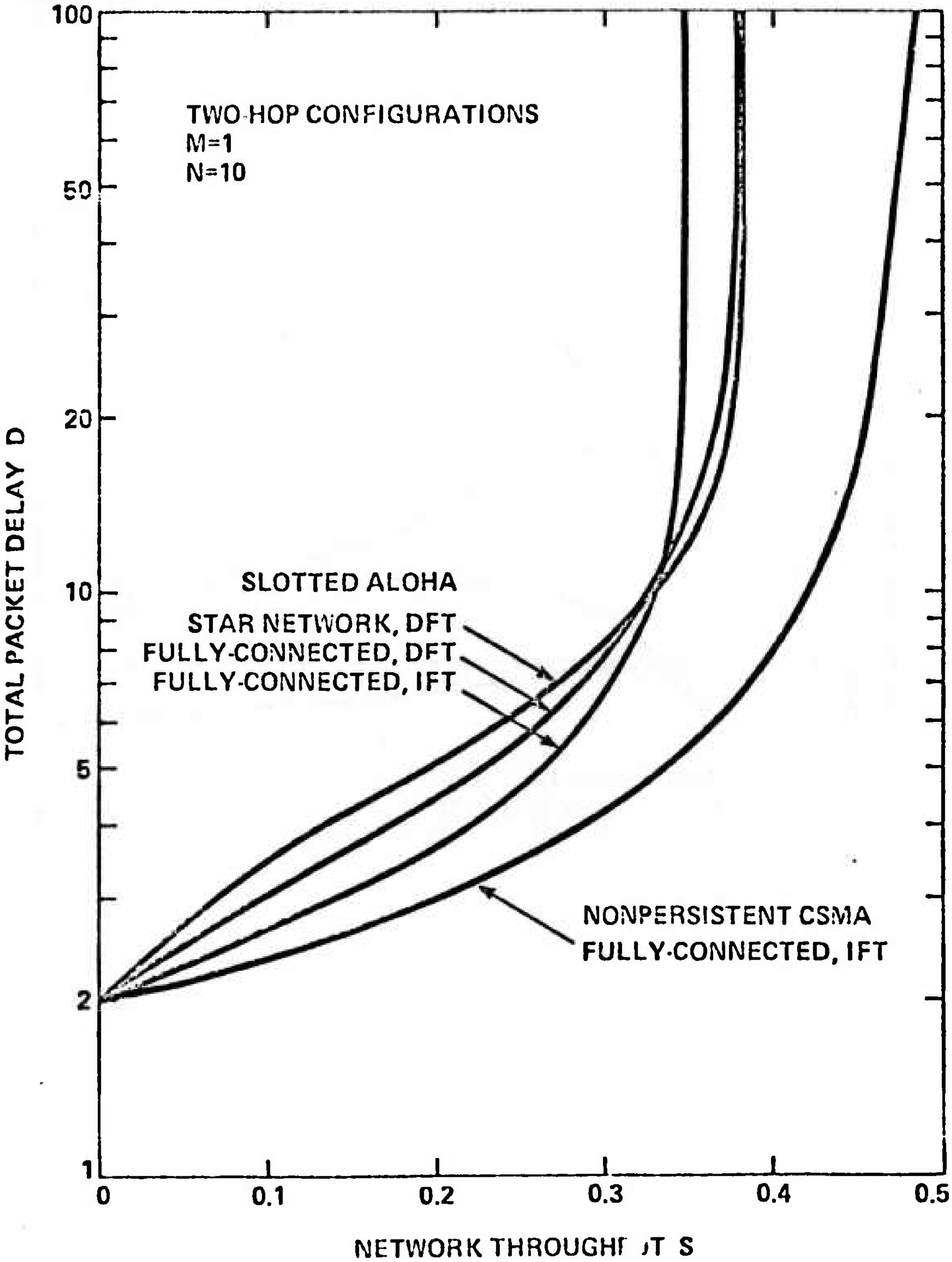


Figure 12 Comparison between Slotted ALOHA and Nonpersistent CSMA (a = 0.01) for N = 10.

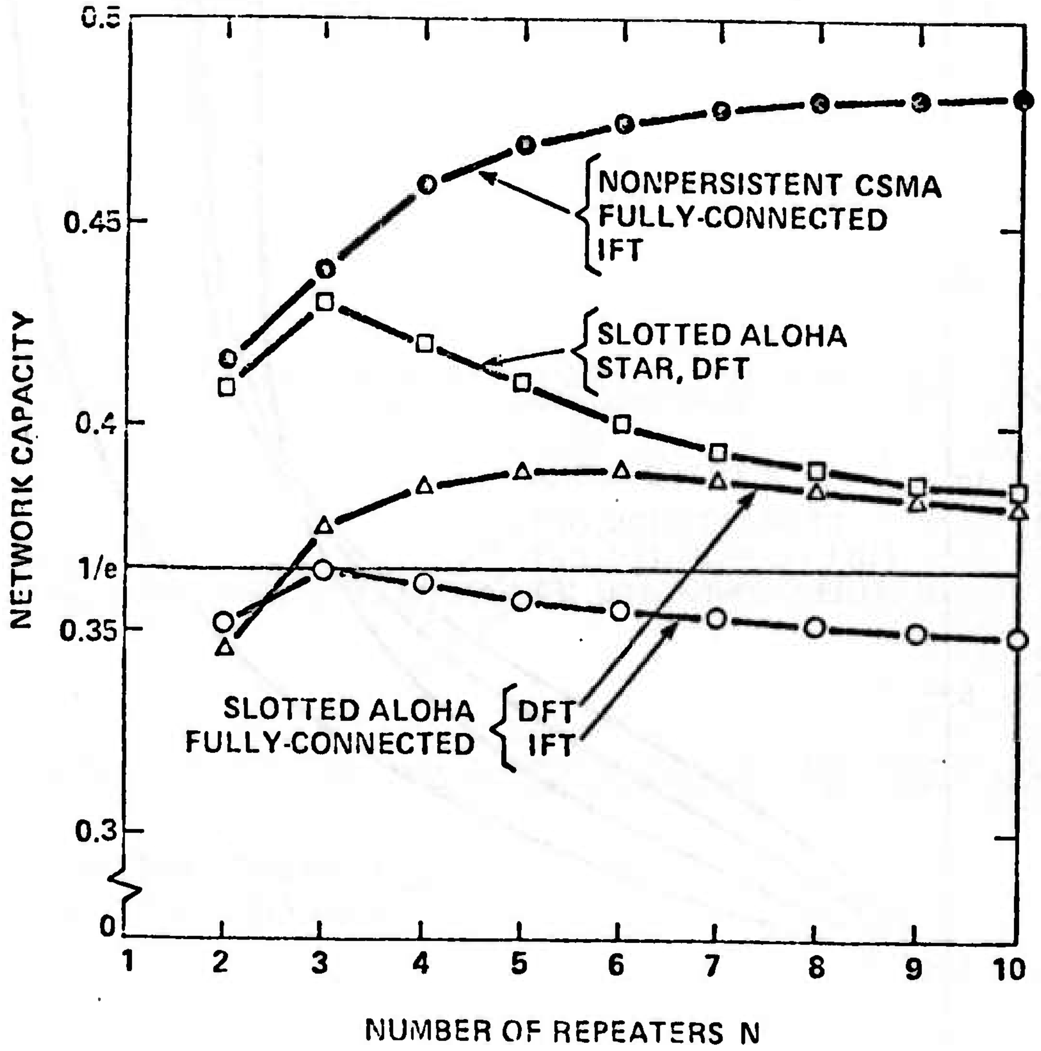


Figure 13 Network Capacity versus N for Slotted ALOHA and Nonpersistent CSMA Netowrks ($a = 0.01$).

PERFORMANCE ANALYSIS OF PACKET RADIO COMMUNICATION SYSTEMS*

by

Fouad A. Tobagi
Computer Science Department
University of California, Los Angeles
Los Angeles, California 90024

Summary

We study the performance of packet radio systems characterized by an array of store-and-forward repeaters organized in a tree configuration with a station at its root; all devices share a common radio channel via the slotted ALOHA access scheme. Conclusions are drawn regarding the effect on performance of such key system parameters as the topology, the transmission rate, and the nodal storage capacity.

1. Introduction

The economic sharing of computer resources has been made possible by the development of the packet-switching technique[1] whereby packet switches are interconnected by point-to-point data circuits according to some topological design. When the number of communicating elements is sufficiently large and the overall traffic flow is small, the use of "packet broadcasting" techniques for interconnection becomes attractive in that it considerably simplifies the topological design and provides very economic solutions. Moreover, economic studies[2] have clearly shown that, for geographically distributed networks, a significant part of the overall system cost is incurred by local collection and dissemination of data. With the proliferation of computer applications, computer resources have to be brought increasingly close to the individual; this makes it also extremely desirable to create more economic techniques to bring the communications capabilities closer to the individual. Again, radio systems offer an attractive solution. The ALOHA system, at the University of Hawaii[3] appears to have been the first computer communication system utilizing radio, and is an excellent illustration of the feasibility of the technique. In line with the objectives set forth in the design of the ALOHA system and to allow the support of many applications which require several additional features not existing in the ALOHA system (such as direct communication by a ground radio network between users over a wide geographical area, coexistence with possibly different systems on the same frequency band, anti-jam capability, etc.) the Advanced Research Projects Agency undertook a new effort, the development of a packet radio system; in particular, it will serve to demonstrate the applicability of the packet radio concept in organizing computer resources into a computer communication network. Basically, the system consists of terminals and stations linked together by line-of-sight radio repeaters. The stations are minicomputers which provide system control; the terminals are hand-held devices, I/O consoles, computers, sensors, etc.; the repeaters are simple relay devices which provide network area coverage for terminals and for one or more stations. The target system requirements have been well assessed by R. Kahn[4]. A prototpye system has already been deployed in the Palo Alto, California area, and experimentation is in progress.

The design of a packet radio system meeting a given set of target requirements and satisfying given performance and reliability constraints is a very complex task. At first a feasibility study is required which either demonstrates that the requirments can be met by the present technology or dictates that research and improvements need to be made in various specific areas; in either case, there can be a number of alternatives to choose among; this leads to a second design phase, an optimization phase, which consists of selecting the best combination of choices. In its most general form, the design problem consists of minimizing the system cost subject to given constraints on throughput, delay, and reliability, over the many system variables:

A. Network Topology: the number of devices (repeaters and stations), their locations and their interconnections, needed to establish the required communication among sources and destinations;

B. Bandwidth Management, i.e., the channel configuration adopted: low-speed channels dedicated to pairs of devices with directional antennas; a single high-speed channel to be shared by a large number of users; or mixed mode configurations whereby the bandwidth is partitioned into a number of channels, some of which are shared while others are dedicated.

C. Channel Access Policy (non-trivial when in presence of shared channels): polling techniques[5]; random access techniques such as pure ALOHA[3], slotted ALOHA[6], and carrier sense multiple access[7,8]; contention techniques such as split-channel reservation multiple access[5].

D. Modulation Scheme: spread-spectrum; narrow-band modulation,...(For a given required probability of bit error, and a given transmitted power, the modulation scheme determines the tradeoff relation between the range of a packet radio device (which affects the network topology) and the bit rate achievable (which affects the network throughput.)

E. Operational Protocols: consisting of the routing policy (point-to-point, undirected routing, directed routing, directed broadcast routing[9]), the error control procedure (of significant importance since, in addition to random noise, errors in multi-access radio channels are due to multipath effects and interference caused by overlapping packets), the flow control algorithms (flow control mechanisms which prevent any source from overloading the network and causing serious congestion) and the network monitoring functions (which allow the proper functioning of the operational protocols by providing them with the ability to adapt to changing system states.)

F. Repeater Design: the repeater's transmit power, its processing speed, its storage capacity and the scheme used to manage it.

It is apparent that the design of a packet radio network involves a large number of variables which interact in a very complex fashion. In its present form the solution is extremely hard to come by. A reasonable approach to follow consists of first selecting those system variables which are obviously determined by some of the constraints. For example, for rapid deployment and

*This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC15-73-C-368.

easy communication among mobile devices, the entire system will employ omnidirectional antennas and will share a single high-speed channel via some random access scheme. Hopefully, this first step decreases significantly the variables' space. The next step will consist of considering various specific configurations which are intuitively appealing; these configurations are then analyzed in order to (i) identify the key parameters which affect the performance and (ii) determine the conditions under which the a priori specified constraints can be satisfied. This step represents an iterative process in that the results obtained from the analysis of some network configuration constitutes a valuable feedback process by means of which the design deficiencies are detected and subsequently corrected, and new configurations are invented.

In the next section, we shall study the performance of systems characterized by an array of repeaters organized in a tree configuration with a station at its root, and draw some conclusions about the effect on performance of such key system parameters as the topology, the transmission protocol, and the nodal storage capacity.

2. Performance Analysis of Tree-Structured Networks

A number of papers have appeared which study various simple network topologies. Single-hop networks, where terminals communicate directly with a central station, have been investigated extensively; access modes in such environments have been carefully evaluated[5,7,8]. A two-hop configuration involving a ring of repeaters around a station have been analyzed by Gitman[10]: network capacity was derived, but no consideration was made of network delays. In this paper, we consider centralized networks characterized by an array of repeaters organized in a symmetric tree configuration with a (single) station at the root; all devices are provided with omnidirectional antennas and employ slotted ALOHA random access over a single shared channel (see Fig.1).

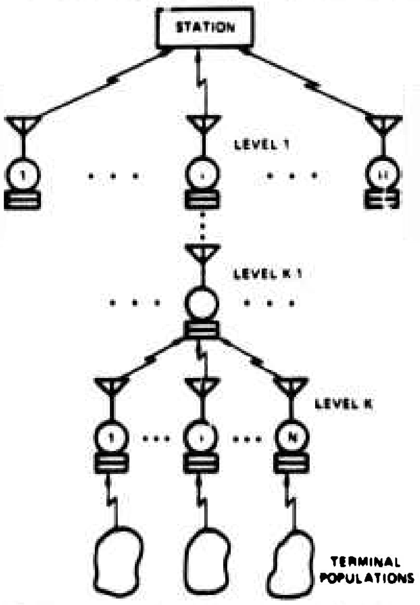


Fig.1 - Tree-Structured Networks

Traffic originates at terminals located at the outskirts of the tree, and is destined to the station. Thus, we only consider inbound traffic. The first transmission of a packet from a terminal is to a repeater located at a leaf of the tree. The routing of the packet through the network is completely specified by the tree structure, as is the connectivity pattern among the devices. The basic performance measure sought is the throughput-delay trade-off and its dependence on such key system parameters as topology (degree of the tree, depth of the tree, and the connectivity pattern among devices), retransmission delays and repeater's storage capacities.

2.1 Traffic Model and Transmission Protocol

The time axis is assumed to be universal and slotted into segments whose duration is equal to the transmission time of a packet. Packets in this study are all of a fixed size. All devices are assumed to be synchronized and start packet transmission at the beginning of a slot. Associated with each repeater located at a leaf of the tree is a population of terminals which generate new packets at an aggregate rate of s packets per slot, all destined to the station. 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 transmitted by repeaters on a first-come-first-served basis; when its buffer is non-empty, a repeater transmits the head of its queue with a probability p. When the packet transport is successful (i.e., the transmission is free of interference and storage is available at the receiving repeater), the packet is deleted from the sender's queue; otherwise, the packet incurs a retransmission delay geometrically distributed, with mean $1/(1-p)$. A repeater learns about its success or failure instantaneously; that is, acknowledgements are assumed to be instantaneous and for free. At any one time, a repeater 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.

Due to the blocking of traffic at the receiving repeater, the rate of successful transmissions to a leaf from its corresponding population of terminals is actually greater than s and is denoted by λ. Furthermore, this process of packet arrivals to a leaf is assumed to be a Bernouilli one (an assumption, which has been proven to be reasonable).

2.2 Single-level Single-buffer Networks.

The state of the system is determined by the number of packets buffered at each repeater. In this simple symmetric case where M=1, the state description can equivalently be given by the number of "active" repeaters (repeaters with non-empty buffers). Let n^t $(0 \leq n^t \leq N)$ denote that number at slot t. It is simple to note that n^t is a Markov chain with a transition matrix P whose $(i,j)^{th}$ element is given by

$$P_{ij}=\begin{cases} 0 & j < i-1 \\ P_s(1-\lambda)^{N-i} & j = i-1 \\ [1-P_s]\binom{N-i}{k}\lambda^k(1-\lambda)^{N-i-k} + P_s\binom{N-i}{k+1}\lambda^{k+1}(1-\lambda)^{N-i-(k+1)} & j = i+k \end{cases}$$

$$k = 0, 1, 2, \ldots, N-i \quad (1)$$

where P_s denotes the probability of a successful transmission given i active repeaters and is expressed as

$$P_s = ip(1-p)^{i-1} \quad (2)$$

Let $\pi_i \triangleq \lim_{t\to\infty} \Pr\{n^t = i\}$. We compute the stationary distribution $\Pi = \{\pi_0, \pi_1, \ldots, \pi_N\}$ by solving recursively the system $\Pi = \Pi P$. Let $\bar{n}$ denote the average number of active repeaters. We have

$$\bar{n} = \sum_{k=0}^{N} k\pi_k \quad (3)$$

A packet successfully transmitted by a population of terminals can be "blocked" at the immediate destination repeater. Blocking is due to two factors: (i) the repeater is in a transmit mode (and we denote by β the probability of such an event) or (ii) the repeater's receiver is on but its packet buffer is full (and we denote by α the probability of this event). Let $B = \alpha+\beta$. We have:

$$\alpha = (1-p)\ \overline{n}/N \qquad (4)$$

$$\beta = p\ \overline{n}/N \qquad (5)$$

$$B = \overline{n}/N \qquad (6)$$

The total network throughput, defined as the rate of successful packets received at the station, and denoted by S, is given by

$$S = (N-\overline{n})\ \lambda \qquad (7)$$

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: (i) the access delay D_a, defined to be the time required for the packet to be correctly received at the leaf repeater, and (ii) the network delay D_n which consists of the time elapsed since the packet is accepted at the leaf repeater until it is successfully received at the station. By Little's result, the average network delay is given by

$$D_n = \overline{n}/S \qquad (8)$$

Note: The maximum value of λ allowable in this model is a function of the access mode in use by the terminals. With slotted ALOHA, $\lambda \leq 1/e$. However, given the memoryless property of the Bernouilli input process, the above analysis corresponds also to a "feedback" model whereby, following the successful transmission of its buffer, a repeater is assumed to generate a new packet after a geometrically distributed time with mean $1/\lambda$. In the feedback model, λ can take any value between 0 and 1. $B = \overline{n}/N$ represents then the fraction of time a repeater is active; and D_n represents the total packet delay.

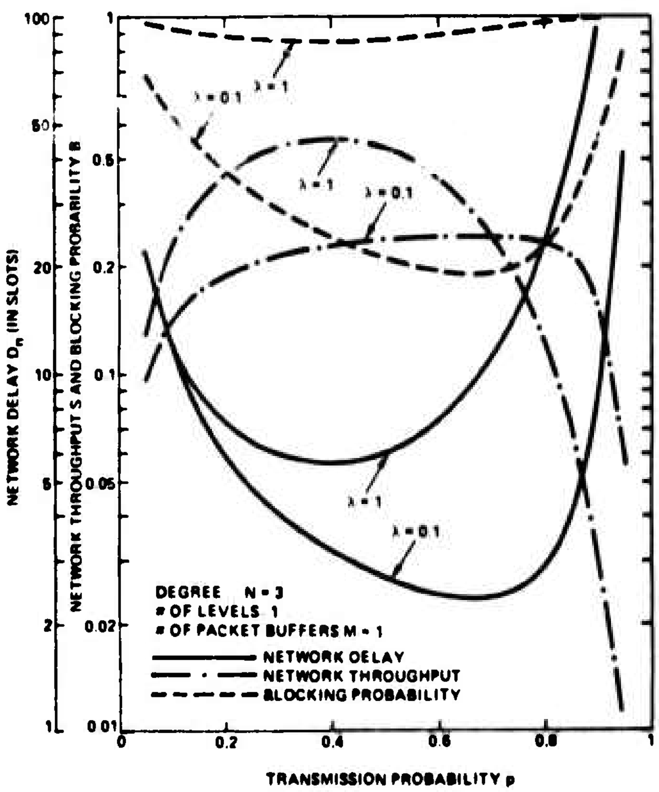


Fig.2 Network Delay, Throughput and Probability of Blocking Versus p.

Fixing N and λ, we observe that $\overline{n}$ is a concave function of p; an optimum value of p minimizes $\overline{n}$. From Eqs. (6-8) we note that D_n and B are also concave functions of p while S is a convex one, and that the same value of p maximizes S and minimizes both D_n and B. As an example, we show in Fig.2 D_n, S, and B versus p for N=3 and two values of λ. We also observe that the throughput S is not as sensitive to p as are D_n and B. Thus, if p is improperly tuned, while the system can maintain the desired throughput, the network delay and the probability of blocking (and thus the access delay) may suffer significant increases! In Fig.3, we plot the optimum delay versus S for various values of N. We note that as the degree of the tree increases, so does D_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.4.

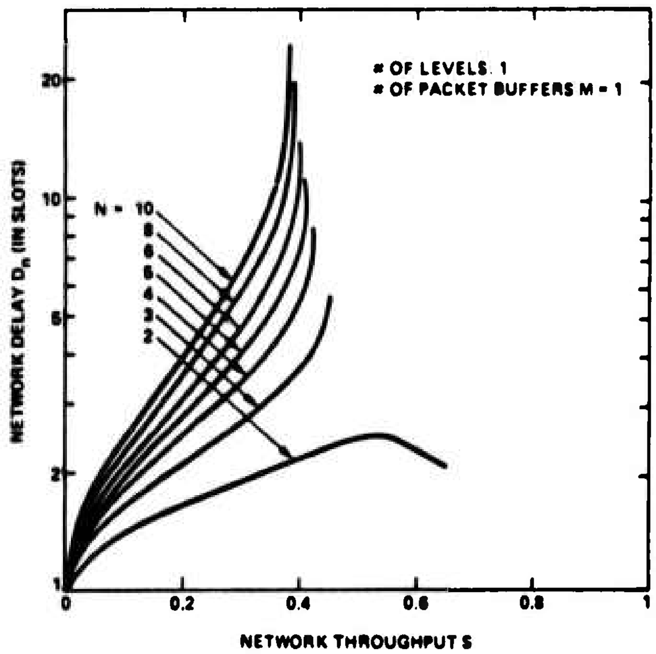


Fig.3 Optimum (Network) Throughput-Delay Curves.

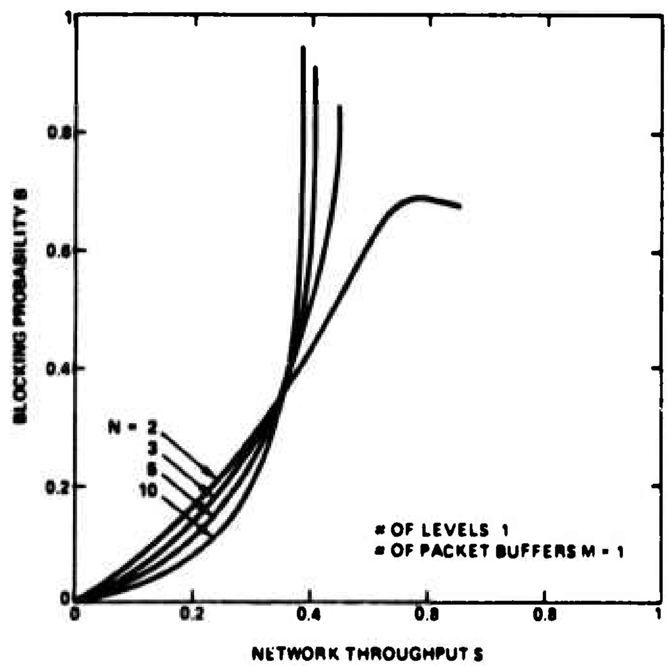


Fig.4 Minimum Blocking Versus Throughput

Network Saturation. In terms of the feedback model, network behavior at saturation is obtained by driving λ to 1. At saturation, $\pi_0 = \pi_1 = \ldots = \pi_{N-2} = 0$ and $\pi_{N-1} = 1-\pi_N \neq 0$. We have

$$\pi_{N-1} = \frac{Np(1-p)^{N-1}}{1-(N-1)p(1-p)^{N-2} + Np(1-p)^{N-1}} \tag{9}$$

π_{N-1} represents also the network throughput. The network delay is simply given by

$$D_n = \frac{1}{p(1-p)^{N-1}} - \frac{(N-1)p}{1-p} \tag{10}$$

The network capacity is obtained by maximizing S with respect to p. The optimum p will also minimize the delay. In Fig.5 network capacity, network delay, and the probability of blocking at saturation are plotted versus N. The network capacity decreases with increasing values of N, and approaches 1/e as $N\to\infty$. D_n increases and approaches $Ne-1$.

In the two hop configuration case, the maximum λ is 1/e; The system capacity is then expressed as $\max_p \left\{\frac{N}{e}\,[1-B(\frac{1}{e},p)]\right\}$; some numerical results are shown in Fig.9 below.

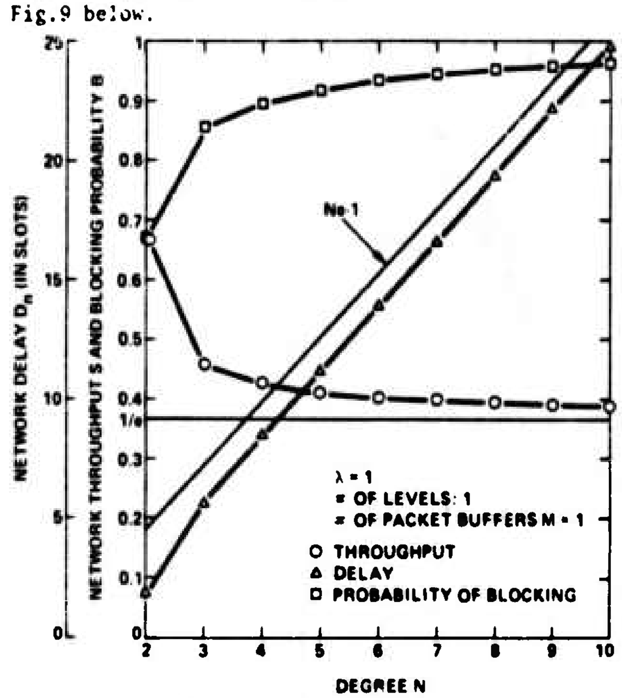


Fig.5 Performance of Single-level Single-buffer Networks at Saturation.

2.3 Single-level Multi-buffer Networks

With $M > 1$, the state of the system is described by the vector $\underline{n} = (n_1, n_2, \ldots, n_N)$ where $0 \le n_i \le M$ denotes the number of packets buffered at repeater i. The one-step transition probability from state $\underline{m} = (m_1, m_2, \ldots, m_N)$ to state $\underline{n} = (n_1, n_2, \ldots, n_N)$, denoted by $Pr[\underline{n}/\underline{m}]$, is determined by the following:

(i) 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 $Pr[\underline{n}/\underline{m}] = 0$

(ii) otherwise, if $\exists\, i_0$ such that $m_{i_0} = n_{i_0} + 1$ indicating a successful transmission), then

$$Pr[\underline{n}/\underline{m}] = p \prod_{j \neq i_0} (1-p)^{\chi_j} \prod_{\ell \neq i_0} [\lambda \xi_\ell^- + (1-\lambda+\lambda\zeta_\ell)\,\xi_\ell] \tag{11}$$

(iii) otherwise, (letting $I_s = \{\, j \mid m_j = n_j \,\}$),

$$Pr[\underline{n}/\underline{m}] = \Big\{ \prod_{j\in I_s} [p\chi_j + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)] - \sum_{j \in I_s} p\chi_j \prod_{\substack{k\in I_s \\ k\neq j}} (1-p)^{\chi_k}(1-\lambda+\lambda\zeta_k) \Big\} \prod_{j \notin I_s} (1-p)^{\chi_j}\lambda \tag{12}$$

where

$$\chi_j = \begin{cases} 0 & \text{if } m_j \le 0 \\ 1 & \text{if } m_j > 0 \end{cases} \qquad \xi_j = \begin{cases} 1 & \text{if } m_j = n_j \\ 0 & \text{if } m_j \neq n_j \end{cases}$$

$$\xi_j^- = \begin{cases} 1 & \text{if } m_j = n_j - 1 \\ 0 & \text{if } m_j \ge n_j \end{cases} \qquad \zeta_j = \begin{cases} 1 & \text{if } n_j = M \\ 0 & \text{if } n_j < M \end{cases}$$

The transition matrix P is numerically computed and the stationary distribution Π is evaluated by iteratively solving the system $\Pi = \Pi P$. Given Π, network throughput, network delay and the probability of blocking are easily computed.

In Fig.6 we plot on the (S, D_n) plane the constant λ contours (varying p) for 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 does not exactly correspond to the value of p which yields minimum blocking (and therefore maximum throughput). However, the probability of blocking at minimum delay is not significantly different from the minimum blocking achievable! The effect M has on network delay is shown in Fig.7 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. We also note a slight decrease in network capacity. The effect M has on the probability of blocking is shown in Fig.8 where we plot the minimum blocking as a function of S. Note the (slight) decrease achieved by going from M=1 to M=2. M=3, however, offers no further significant improvement!

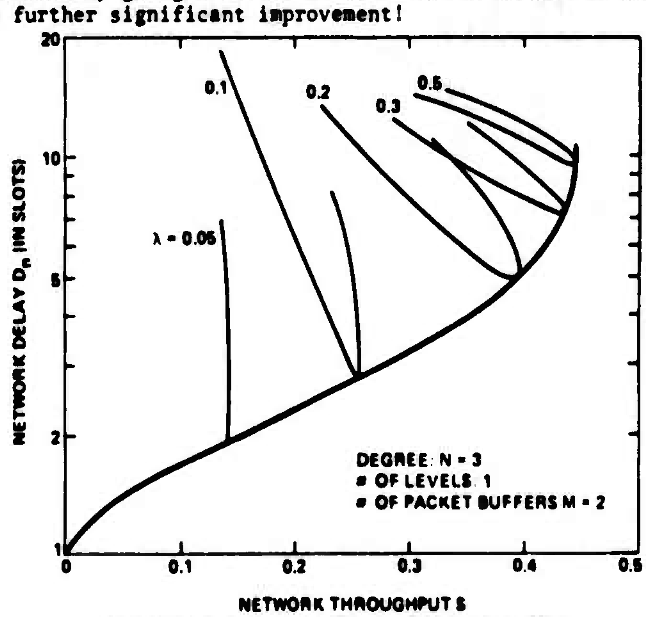


Fig.6 Delay Versus Throughput with M>1

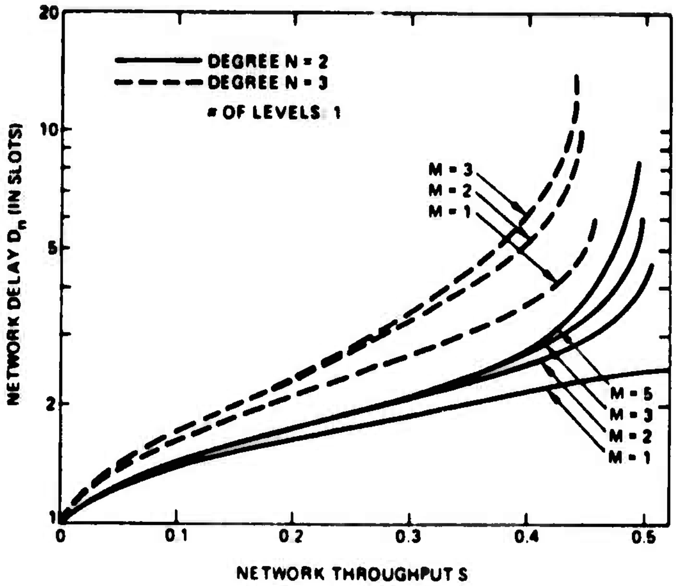


Fig.7 Minimum Delay Versus
S for Various Values of M.

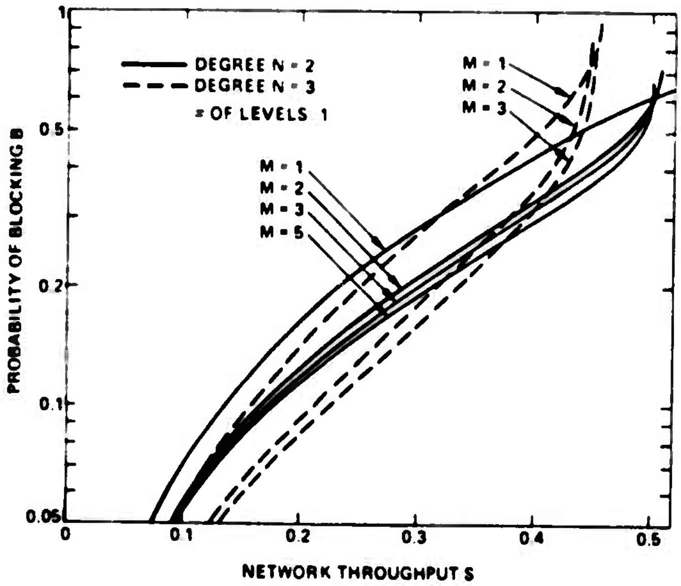


Fig.8 Minimum Blocking Versus S
for Various Values of M.

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? Given the throughput-delay characteristics of an infinite population employing slotted ALOHA (which we denote here by $D_{S\text{-}ALOHA}$ (throughput) and which we have displayed in Fig.9 [5]), we estimate the access delay by

$$D_a = \frac{1}{1-B} \; D_{S\text{-}ALOHA}\left(\frac{S}{N(1-B)}\right) \qquad (13)$$

In Fig.9, we plot D versus S for N=2 and 3 and various values of M. Again, we only note a slight improvement in performance by going from M=1 to M=2. No further significant improvement is gained beyond M=2. The increase in network capacity (observed particularly with N=2) is obviously due to the decrease in B.

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 the (α,β) plane on which we plot the constant λ contours. When p is small, α predominates: $\alpha>\beta$; As p increases, the inequality reverses. The locus of optima is displayed in Fig.10 for various values of N. 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. This effect is due to the fact that, for the same throughput, the optimum p decreases as N increases in order to prevent conflict among a larger number of contending users. Is the system then storage bound when N is large, say 10 for example? It can be argued that there is still no

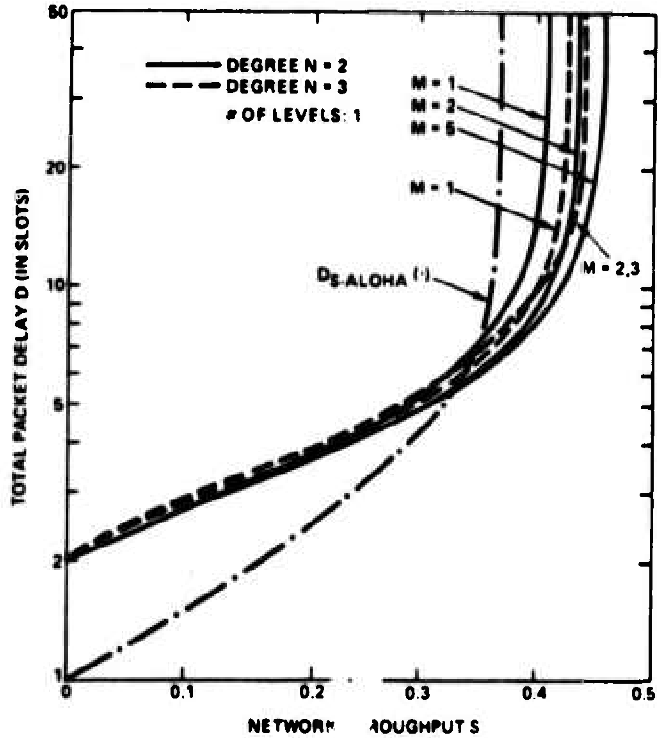


Fig.9 Total Packet Delay Versus Throughput.

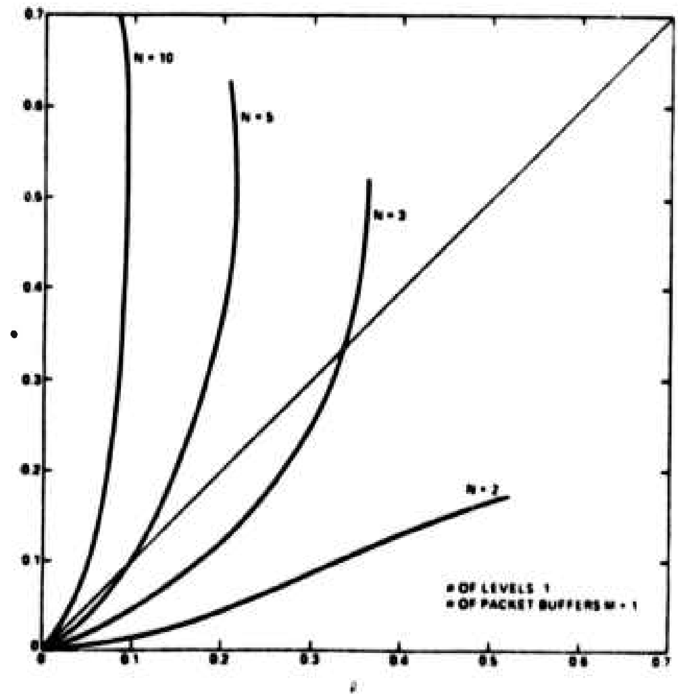


Fig.10 α Versus β at Minimum Blocking

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.3) while D_a decreases as does S/N (for N=10, 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: $D_n \simeq 10$, $B \simeq 0.38$ and $D_a \simeq 2$ yielding $D \simeq 12$. By taking B=0, we can decrease D_a to 1.15 providing thus a lower bound on D of 11.15, a rather unimportant improvement. Moreover due to the queueing effect, D_n increases with larger M.

2.4 Multilevel Networks

The larger the number of repeaters and the number of levels in the network, the larger is the state space and the more complex is the generation of the transition matrix P. Although we are not saying that it is entirely infeasible to analyze multilevel configurations exactly, the exact approach becomes intractable as the number of levels in the tree increases and the interaction among repeaters becomes more complex. Aside from simulation, approximate methods are the only recourse. The method used in this paper, called the "decomposition" method, consists of treating the levels in the tree separately and in sequence, assuming they are stochastically independent. First, we note that the success of a transmission from a repeater located at level k is not affected by the state or behavior of those at levels $\ell > k$. Let R denote a repeater located at level k. Let $S(R)$ denote the set of repeaters which are sons of R. Now, the following simplifying assumptions are made: (A1). The process defined by the arrival of packets to R from $S(R)$ is a Bernouilli process, and (A2). The probability of blocking at repeater R is independent of the system state and uniform over time.

Let $B_k = \alpha_k + \beta_k$ denote the probability of blocking at R. Let n^t, $0 \leq n^t \leq N$, be the number of active repeaters in $S(R)$ at time t. Given that $n^t = i$, the probability of successfully transporting a packet from level k + 1 (that is, from $S(R)$) to level k (that is, to R) is given by

$$p_s^{k+1} = ip(1-p)^{i-1}(1-\alpha_k - \beta_k)\ (1-\beta_{k-1}) \qquad (14)$$

The process n^t is again a Markov chain whose transition matrix is defined by Eq. (1), and where P_s is now expressed as in Eq. (14). The steady state solution at level k+1 is obtained as with single-level systems, provided that $\alpha_k + \beta_k$ and β_{k-1} have already been computed. Thus, given the network throughput S, the approach consists of solving for the steady state level by level, starting with k=1 and $\alpha_0 = \beta_0 = \beta_{-1} = 0$.

Validation of the approach: This decomposition method was first motivated by a study of the packet transport process in single-level networks with M=1, which showed that this process can be approximated by a Bernouilli process for a large range of the system parameters N, λ and p. A chi-square test was performed in comparing the exact distribution governing the transport process (obtained analytically) to a Bernouilli one with the same rate. The results show that the Chi-square 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%; (degree of freedom = 100).

In an attempt to further validate the above approach, a simple test case, a tree of depth 2 and degree 2 (M=1) was analyzed exactly. In Table I, we compare the results obtained from the exact analysis to those obtained by the decomposition method. The two are extremely close! Further validation is in progress using simulation techniques and more complex configurations. (At any rate, it is to be noted that even though the approximate method may not yield very accurate estimates of the performance measures, the trends observed are believed to correspond to reality and should provide fairly valuable insight into the behavior of tree-structured networks and into the effect of system parameters on these trends.)

λ	P_1	P_2	S		$\alpha_1+\beta_1$		$\alpha_2+\beta_2$		D_1		D_2	
			exac	appr	exac	appr	exac	appr	exac	appr	exac	appr
0.024	0.85	0.9	0.093	0.093	0.082	0.065	0.032	0.032	1.39	1.39	1.38	1.36
0.046	0.85	0.86	0.170	0.172	1.175	0.136	0.014	0.070	1.58	1.58	1.73	1.65
0.068	0.8	0.83	0.238	0.238	0.245	0.206	0.124	0.120	1.74	1.73	2.08	1.99
0.088	0.8	0.80	0.289	0.291	0.305	0.271	0.177	0.172	1.86	1.86	2.45	2.37
0.109	0.75	0.77	0.333	0.334	0.352	0.330	0.234	0.234	1.97	1.97	2.81	2.80
0.134	0.75	0.75	0.372	0.370	0.399	0.428	0.304	0.308	2.06	2.06	3.27	3.32
0.164	0.75	0.72	0.404	0.400	0.439	0.500	0.383	0.390	2.15	2.14	3.78	3.90
0.292	0.75	0.66	0.453	0.444	0.517	0.590	0.611	0.619	2.30	2.28	5.39	5.57

Table I . Comparison of Exact Results and Approximate Results in the Case of a Tree of Degree 2 and Depth 2.

Numerical Results. Let K be the number of levels in the tree; let d_k denote the delay incurred by a packet at level k, $1 \leq k \leq K$. The total network delay is given by

$$D_K = \sum_{k=1}^{K} d_k.$$

First we inquire as to how d_k varies with k. We show in Fig.11 d_k versus k for N=2 and 5, and fixed values of the network throughput S. The curves show that there exists a critical hop, a level with the highest delay; this critical hop, k_c, is a function of S and N. When the network is lightly loaded (small S) then $k_c = 1$. As the throughput increases, k_c increases. Its highest value is $k_c = 4$ for N=2, $k_c = 2$ for N>2. Beyond the critical level, d_k decreases rapidly, reaching asymptotically the value 1. This behavior is explained by the fact that the throughput handled by a repeater at level k decreases geometrically as k increases, and that for $k \leq k_c$, the counter-effect of the blocking factors $\alpha_{k-1} + \beta_{k-1}$ and

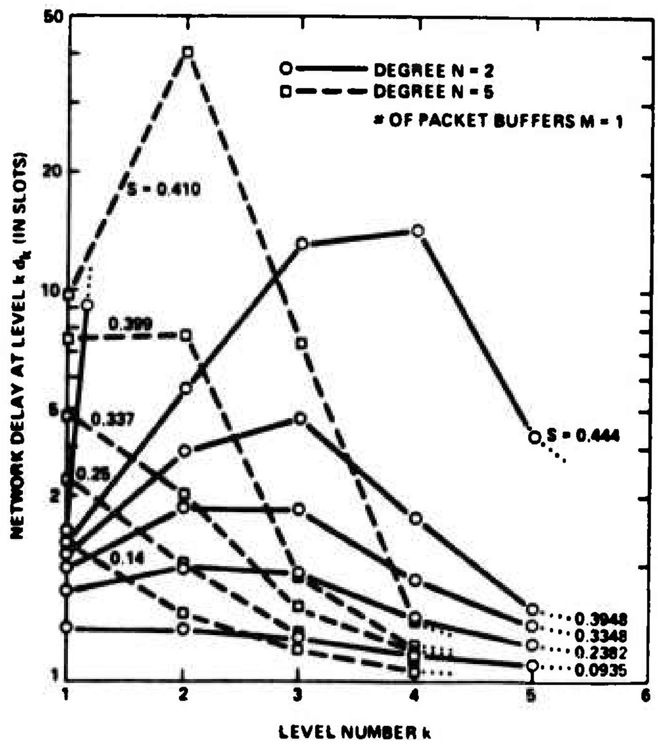


Fig.11 Network Delay at Level k Versus k

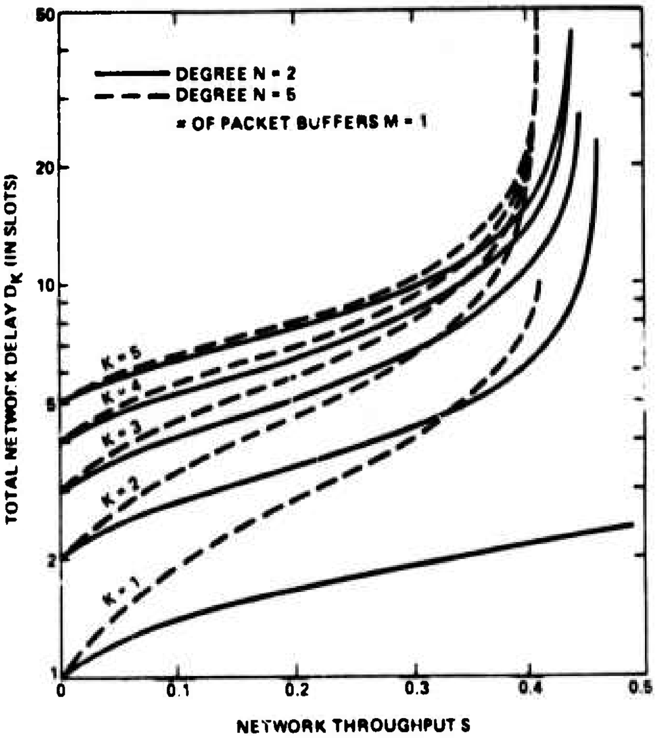


Fig.12 Network Delay D_K Versus S for Various K.

ε_{k-2} may be stronger for larger S.

The total network delay D_K is plotted versus S in Fig.12 for N=2 and 5 and $1\le K\le 5$. Note that for moderate values of S (not too close to the network capacity) and as K exceeds 4 the total network delay becomes not too sensitive to N. It is clear however, that the network capacity decreases as N increases, approaching 1/e.

Now we inquire as to the effect that an increase in buffer size at the repeaters will have on the overall network throughput-delay performance. Let us, towards that end, increase the number of buffers at level 1 to M=2. In Fig.13 we plot the total network delay for a tree of degree 3 illustrating that buffer size of M=1 is again an adequate design. (Such a result is not surprising here since given the conclusions of section 2.3 d_k with finite N is smaller than D_a of an infinite population with the same throughput).

3. Conclusion

The above analysis have provided a means by which the throughput and delay performance measures can be evaluated for slotted ALOHA, tree-structured packet radio networks. We have shown the effects of various systems parameters: the degree of the tree, the number of levels, the buffer size. The analysis was concerned with inbound traffic only. If we were in the presence of outbound traffic alone, it is strongly believed that a similar behavior would be observed. If the single channel is used to support both inbound and outbound traffic, it is clear that to avoid deadlocks, each repeater should be provided with two packet buffers, each dedicated to traffic in one direction. Since a packet buffer is released only when the transmission of the packet is acknowledged to be successful, an additional buffer is required for the reception and processing of the acknowledgement packets; this buffer can be shared by both directions The results lead us to believe that the system is channel bound (the processing time at repeaters assumed negligible) and that exactly three packet buffers at each component constitute an adequate design. A slight improvement may be gained by going to M=5 (two packet buffers for each direction, and one for

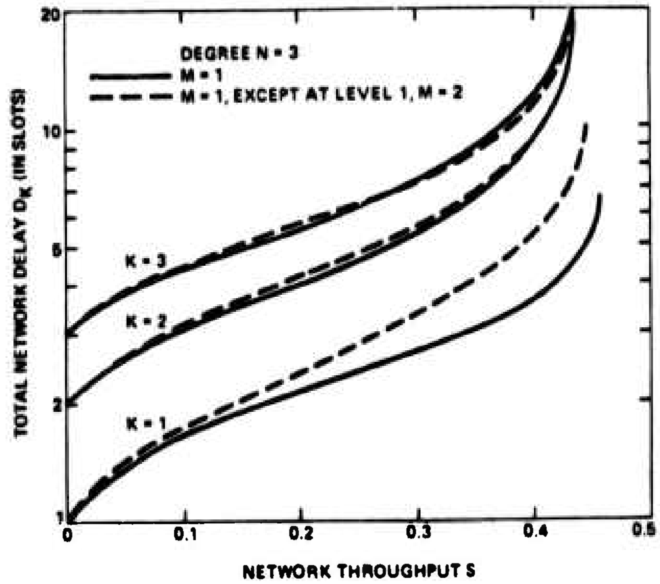


Fig.13 Effect of M>1 on Network Delay D_K

acknowledgements); but no significant improvement is obtained beyond that.

References

1. Roberts, L.G. and B.D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, SJCC 70, pp. 543-549.
2. Roberts, L., "Data by the Packet," IEEE Spectrum, February 1974, pp. 46-51.
3. Abramson, N., "Another Alternative for Computer Communications," AFIPS Conference Proceedings, FJCC 70, pp. 695-702.
4. Kahn, R., "The Organization of Computer Resources into a Packet Radio Network, "AFIPS Conference Proceedings, NCC 75, pp. 177-186.
5. Tobagi, F.A. and L. Kleinrock, "Packet Switching in Radio Channels: Part III - Polling and (Dynamic) Split-channel Reservation Multiple Access," IEEE Transactions on Communications, Vol. COM-24, No. 8, August 1976, pp. 832-845.
6. Roberts, L., "APPANET Satellite System," Notes 8 (NIC Document 11290) and 9 (NIC Document 11291), available from the ARPA Network Information Center, Stanford Research Institute, Menlo Park, Calif.
7. Kleinrock, L. and F. Tobagi, "Packet Switching in Radio Channels: Part I--Carrier Sense Multiple Access Modes and their Throughput Delay Characteristics," IEEE Transactions on Communications, Vol. COM-23, No. 12, December 1975, pp. 1400-1416.
8. Tobagi, F. 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 Transactions on Commun., Vol. COM-23, Vol. 12, December 1975, pp. 1417-1433.
9. Gitman, I., R.M. Van Slyke and H. Frank, "Routing in Packet-Switching Broadcast Radio Networks," IEEE Transactions on Communications, Vol. COM-24, No. 8, August 1976, pp. 926-930.
10. Gitman, I., "On the Capacity of Slotted ALOHA Networks and Some Design Problems," IEEE Transactions on Communications, Vol. COM-23, No.3, March 1975, pp. 305-317.

ANALYSIS OF SLOTTED ALOHA IN A CENTRALIZED TWO-HOP PACKET RADIO NETWORK.*

Fouad A. Tobagi

Digital Systems Laboratory
Stanford University
Stanford, California 94305

Abstract

The concept of packet broadcasting has brought together the advantages of both packet switching and broadcast communication. Unfortunately, the optimum design of a packet radio system is a very difficult task. The complexity of the problem leads us to consider and analyze some simple but typical configurations in an attempt to understand the behavior of these systems and to derive their performance. In this paper we consider a two-hop centralized environment, employing slotted ALOHA, 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 forming a ring around the station. The basic performance sought is the throughput-delay tradeoff and its dependence on such key system parameters as the network topology (number of repeaters, network connectivity) and the repeaters' transmission protocol.

1. Introduction

The concept of packet broadcasting has brought together the advantages of both packet switching and broadcast communications. 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, easily accessible and particularly suitable for mobile communications. The applications are indeed numerous and the resulting product is of great impact on the future trends of communications systems. 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. 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[1]. Another prominent example is typified by the Packet Radio system of the Advanced Research Projects Agency, Department of Defense. The target requirements of the system have been well assessed by R. Kahn in [2]. These requirements 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, anti-jam protection... The key requirement of direct communication over wide geographical areas leads the repeaters to become integral components of the system: "The system consists of terminals and stations linked together by line-of-sight radio repeaters."[2]

With the advent of packet broadcasting emerged a large class of new problems of the most challenging kind. In Reference [3], we briefly defined the design problem of packet radio systems and identified the key system variables and protocols which affect the performance; in summary, these are: the network topology (number of devices and their geographical setting), the bandwidth management (dedicated channels, shared channels, mixed modes...), the channel access policy (fixed assignment, centrally controlled schemes, random access modes...), the modulation scheme (spread-spectrum, narrowband,...), the operational protocols (routing policy, error control procedures, flow control, monitoring functions...) and finally the nodal design (storage capacity, buffer management, power requirement, processing speed...). The discussion shows that the design of a packet radio network involves a large number of variables which interact in a very complex fashion. 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 some constraints. For example, for rapid deployment in military applications, and for easy communication among mobile terminals, it is advantageous that the entire system employs omnidirectional antennas and shares 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 variables' space, and need to only focus on packet radio systems with the above characteristics.

We are now still faced with deciding among several design choices and this requires that we have a means by which we are able to evaluate the

*This research was supported by the Advanced Research Projects Agency, Department of Defense, Contract No. MDA 903-77-C-0272. This work was performed while the author was with the Computer Science Department of the University of California, Los Angeles, California 90024

performance of the system under each choice. This task is still of a very high caliber. 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 of these systems, and to derive their performance. In this paper, just as in Reference [3], we opt for the latter approach.

A number of papers have already appeared in the literature which study various simple network topologies. Single-hop networks, where terminals communicate directly with each other or with a central station have been investigated extensively[4,5,6]. A two-hop configuration involving a ring of repeaters around a station has been analyzed by Gitman[7]; network capacity was studied, but no consideration was made of network delays. In Reference [3], we considered centralized networks characterized by an array of repeaters organized in a symmetric tree configuration with a (single) station at the root; all devices are provided with omnidirectional antennas and employ the slotted ALOHA random access over a single shared channel. Traffic originates at terminals located at the outskirts of the tree, and is destined to the station. Thus, only <u>inbound traffic</u> is considered. The first transmission of a packet from a terminal is to a repeater located at a leaf of the tree. The routing of the packet through the network is completely specified by the tree structure, as is the connectivity pattern among the devices. The basic performance measure obtained is the throughput-delay trade-off and its dependence on such key system parameters as the topology (degree of the tree, depth of the tree), repeaters' retransmission delays and repeaters' storage capacities. Of importance is the particular simple case of single-level trees, constituting thus a two-hop network referred to as the <u>star configuration</u> and depicted in Fig. 1. In a star configuration, each repeater is in line-of-sight and within range of only the station.

In this paper we continue this study by considering the two-hop <u>fully connected</u> (FC) network configuration in which all repeaters are within range and in line-of-sight of each other and of the station. This configuration is depicted in Fig. 2. The main difference that exists between this and the star configuration is that in the fully connected case an arrival to a repeater in a slot will not be successfully received if any of the repeaters is actively transmitting in that slot. Each repeater is provided with a finite storage capacity which can accomodate exactly <u>one</u> packet. No consideration will be made here of storage capacity greater than one since the results of [3] have shown that the system is mostly channel bound and not storage bound, and since it is even more so with fully connected configurations as illustrated below. The station is assumed to have infinite storage capacity.

2. Traffic Model and Transmission Protocols

The time axis is assumed to be universal and slotted into segments whose duration is equal to the transmission time of a packet. Packets in this study are all of a fixed size. All devices are assumed to be synchronized and start packet transmission at the beginning of a slot. Associated with each repeater is a population of terminals, in line-of-sight and within range of only that repeater, which generates new packets at an aggregate rate of s packets per slot, all destined to the station. Packets are transmitted by repeaters on a first-come-first-served basis; when its buffer is non-empty, a repeater transmits the head of its queue with a probability p. When the packet transport is successful (i.e., the transmission is free of interference and storage is available at the receiving repeater), the packet is deleted from the sender's queue; otherwise, the packet incurs a retransmission delay geometrically distributed, with mean 1/p. A repeater learns about its success or failure instantaneously; that is, acknowledgments are assumed to be instantaneous and for free. At any one time, a repeater 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. With this protocol the first transmission of a newly received packet (at the repeater) incurs a geometrically distributed delay following its

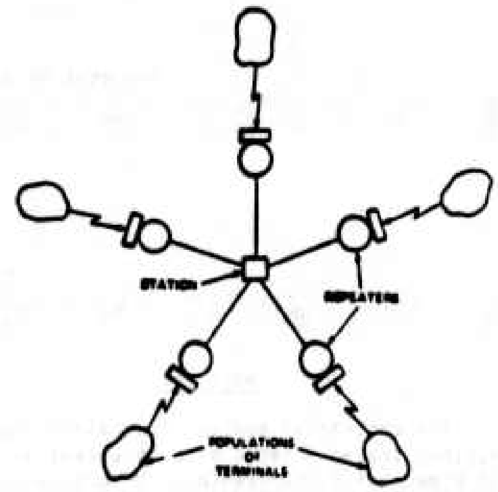


Figure 1 A Two-Hop Star Configuration

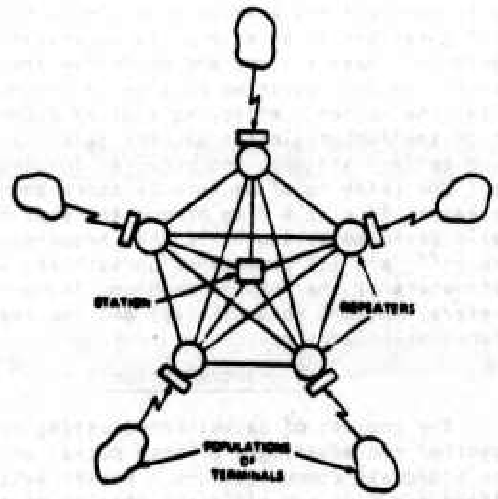


Figure 2 A Two-Hop Fully-Connected Configuration

reception with mean $1/p$. We shall refer to this transmission protocol here as the delayed-first-transmission (DFT) protocol. It is precisely the DFT protocol which was considered in the analysis of the star configurations in [3]. In addition to the DFT protocol we consider here a slight variation of it which consists of transmitting (with probability one) a newly received packet immediately following its reception. 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. The motivation in considering it is simply an expected decrease in packet network delay due to the avoidance of an initial delay at the first transmission of the packet. Numerical results will be discussed below.

A packet successfully transmitted by a given population of terminals can be "blocked" at the immediate destination repeater; blocking is due to two factors: (i) the repeater (or any other repeater in the FC-configuration) is in a transmit mode or (ii) the repeater's receiver is on (and all other repeaters are quiet in the FC-configuration), but the repeater's buffer is full.

Due to the blocking of traffic at the receiving repeater, 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.*

The structure of the remainder of the paper is as follows. In section 3, we briefly review the analysis of star networks; and the basic results obtained in [3]. In section 4, we examine the fully-connected case, in which we consider both transmission protocols described above. A comparison of all these systems and configurations will then follow in section 5.

*This assumption is introduced here for analytic simplicity. Its validity 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 the results obtained from a separate analytic study of the packet transport process from N repeaters (or terminals) to a station, $2 \le N \le 10$, 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. A Chi-square test was performed in comparing the exact distribution governing the transport process (obtained analytically) to a Bernoulli one with the same rate; the Chi-square value of a sample of 1000 inter-arrival times (at the station) is below 67, which corresponds to a level of confidence of over 99.9%, (degree of freedom = 100).

3. Analysis of Star Networks Employing Slotted ALOHA (DFT Protocol)[3]

Let N denote the number of repeaters present in the configuration. Given the DFT transmission protocol 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 number of repeaters with non-empty buffers, referred to as the number of "active" repeaters. Let n^t $(0 \le n^t \le N)$ denote that number at slot t. n^t is a Markov chain; its transition matrix P is given by

$$P_{ij} = \begin{cases} 0 & j < i-1 \\ P_s(i)(1-\lambda)^{N-1} & j = i-1 \\ [1-P_s(i)]\binom{N-1}{k}\lambda^k(1-\lambda)^{N-1-k} & \\ \quad +P_s(i)\binom{N-i}{k+1}\lambda^{k+1}(1-\lambda)^{N-i-(k+1)} & j = i+k \\ & k = 0,1,2,\ldots,N-i \end{cases} \tag{1}$$

where $P_s(i)$ denotes the probability of a successful transmission given i active repeaters and is expressed as

$$P_s(i) = ip(1-p)^{i-1} \tag{2}$$

Let $\pi_i \triangleq \lim_{t\to\infty} \Pr\{n^t = i\}$. We compute the stationary distribution $\underline{\pi} = \{\pi_0, \pi_1, \ldots, \pi_N\}$ by solving recursively the system $\underline{\pi} = \underline{\pi}P$. Let $\bar{n}$ denote the average number of active repeaters. We have

$$\bar{n} = \sum_{k=0}^{N} k\pi_k \tag{3}$$

A packet successfully transmitted by a population of terminals can be "blocked" at the immediate destination repeater. As already mentioned in section 2, blocking is due to two factors: (i) the repeater is in a transmit mode (and we denote by β the probability of such an event) or (ii) the repeater's receiver is on but its packet buffer is full (and we denote by α the probability of this event). Let $B = \alpha + \beta$. We have:

$$\alpha = (1-p)\bar{n}/N \tag{4}$$

$$\beta = p\bar{n}/N \tag{5}$$

$$B = \bar{n}/N \tag{6}$$

The total network throughput is defined as the rate of successful packets received at the station; it is given by

$$S = (N - \bar{n})\lambda \tag{7}$$

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: (i) the access delay D_a, defined to be the time required for the packet to be correctly received at the repeater, and (ii) 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

$$D_n = \bar{n}/S \tag{8}$$

Results in [3] have shown that for given N and λ, there exists a single value of p which maximizes S and minimizes D_n and B, thus providing the optimum performance. The analysis was further pursued to accomodate the cases where the buffer size at the repeater is greater than one. Numerical results have shown that a slight improvement may be gained by increasing the buffer size to two packets, but that no significant improvement is obtained beyond that point. It is shown that the system is "channel bound" rather than storage bound. For small N, β predominates α, showing that blocking is mostly due to the receiver being shut off. With larger N, α predominates β (see Fig. 8 below); but it can still be argued that no improvement can be gained by increasing the buffer size since for larger N the predominant factor of delay is D_n and not D_a [3].

4. Analysis of Fully Connected Configurations

A. Analysis of the DFT Protocol in Fully Connected Configurations

With single-packet buffers, the state of the system in a slot is again entirely described by the number of active repeaters, n^t. The process n^t is a Markov chain with transition matrix P given by

$$P_{ij} = \begin{cases} 0 & j<i-1 \\ P_s(i) & j=i-1 \\ (1-p)^i(1-\lambda)^{N-i}+[1-(1-p)^i-P_s(i)] & j=i \\ (1-p)^i\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j} & j>i \end{cases} \tag{9}$$

where $P_s(i)$ is as expressed in Eq. (2). We compute the stationary distribution $\Pi=\{\pi_0,\pi_1,\ldots,\pi_N\}$ by solving recursively the system $\Pi = \Pi P$. The average number of repeaters $\bar{n}$ is given again by Eq. (3). Let β denote the probability that a terminal transmission is blocked due to transmission by *one* or more repeaters. Given that k repeaters are active, this probability is simply $1-(1-p)^k$. Removing the condition we get

$$\beta = 1-\sum_{k=0}^{N}\pi_k(1-p)^k \tag{10}$$

Let α denote the probability that a terminal 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 $\frac{k}{N}(1-p)^k$. Removing the condition we get

$$\alpha = \sum_{k=0}^{N}\pi_k\frac{k}{N}(1-p)^k \tag{11}$$

The total blocking probability is given by $B=\alpha+\beta$. The network throughput S is expressed as

$$S = \sum_{k=0}^{N}\pi_k\,kp(1-p)^{k-1} = N\lambda(1-B) \tag{12}$$

and the network delay is again simply given by Eq. (8).

B. Analysis of the IFT Protocol in Fully Connected Configurations

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 or not new arrivals had occurred in slot t-1. 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 via a "cycle analysis."

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 k^{th} non-empty slot (see Fig. 3); d^k is a Markov chain. We derive its transition probabilities in the following.

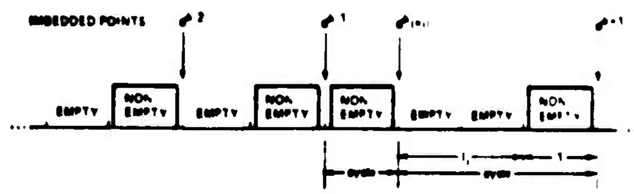


Figure 3 The Imbedded Markov Chain in the Slotted ALOHA IFT Protocol

Let $p_{ij}^k \triangleq Pr\{d^{k+1} = j/d^k = i\}$. Let $P = (p_{ij})$ be the transition matrix. (We drop the superscript k as we are only interested in steady-state conditions.) For i = 0, we have

$$P_{0j} = \begin{cases} \dfrac{N\lambda(1-\lambda)^{N-1}}{1-(1-\lambda)^N} & j=0 \\ 0 & j=1 \\ \dbinom{N}{j}\dfrac{\lambda^j(1-\lambda)^{N-j}}{1-(1-\lambda)^N} & j=2,3,\ldots,N \end{cases} \tag{13}$$

and for i = N we simply have

$$P_{N,j} = \begin{cases} 0 & j<N-1 \\ \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j=N-1 \\ 1-\dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j=N \end{cases} \tag{14}$$

Given that $d^k = i$, let l_i denote the number of empty slots separating two consecutive non-empty 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 non-empty slots. Thus, for $i \neq 0$, N, we have

$$Pr\{l_i=0\}=1-(1-p)^i \text{ and } Pr\{l_i>0\}=(1-p)^i \tag{15}$$

and the transition probabilities are given by

$$P_{ij}=\begin{cases} 0 & j<i-1 \\ Pr\{I_i=0\}\dfrac{ip(1-p)^{i-1}}{1-(1-p)^i}+Pr\{I_i>0\}\dfrac{ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i-1 \\ Pr\{I_i=0\}\dfrac{1-(1-p)^i-ip(1-p)^{i-1}}{1-(1-p)^i}+Pr\{I_i>0\}\dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i+(1-\lambda)^{N-i}[1-ip(1-p)^{i-1}-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i \\ Pr\{I_i>0\}\dfrac{(N-i)\lambda(1-\lambda)^{N-i-1}[1-(1-p)^i]}{1-(1-\lambda)^{N-i}(1-p)^i} & j=i+1 \\ Pr\{I_i>0\}\dfrac{\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j}}{1-(1-\lambda)^{N-i}(1-p)^i} & j\geq i+2 \end{cases} \tag{16}$$

Let $\pi_i^d \triangleq \lim_{k\to\infty} Pr\{d^k=i\}$. The stationary distribution $\Pi^d = \{\pi_0^d, \pi_1^d, \ldots, \pi_N^d\}$ is obtained by solving recursively the system $\Pi^d = \Pi^d P$. We now derive the stationary performance measures. To do so, 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 $I_i + 1$. To compute the average cycle length, we need to determine the average length of I_i which we denote by $\overline{T}_i$. The probability density function of I_i is given by

$$Pr\{I_i=\ell\}=\begin{cases} (1-\lambda)^{N(\ell-1)}[1-(1-\lambda)^N] & i=0;\ \ell\geq 1 \\ 1-(1-p)^i & i\neq 0,N;\ \ell=0 \\ (1-p)^i[(1-\lambda)^{N-i}(1-p)^i]^{\ell-1}\cdot[1-(1-\lambda)^{N-i}(1-p)^i] & i\neq 0,N;\ \ell>0 \\ (1-p)^{N\ell}[1-(1-p)^N] & i=N;\ \ell\geq 0 \end{cases} \tag{17}$$

Thus $\overline{T}_i$ is expressed as

$$\overline{T}_i=\begin{cases} \dfrac{1}{1-(1-\lambda)^N} & i=0 \\ \dfrac{(1-p)^i}{1-(1-p)^i(1-\lambda)^{N-i}} & i\neq 0,N \\ \dfrac{(1-p)^N}{1-(1-p)^N} & i=N \end{cases} \tag{18}$$

The average throughput over the cycle, which we denoted by S_i, is precisely the probability of a successful transmission and is given by

$$S_i=Pr\{I_i=0\}\frac{ip(1-p)^{i-1}}{1-(1-p)^i} + Pr\{I_i>0\}\frac{(N-i)\lambda(1-\lambda)^{N-i-1}(1-p)^i+ip(1-p)^{i-1}(1-\lambda)^{N-i}}{1-(1-p)^i(1-\lambda)^{N-i}} \tag{19}$$

The average of the sum of active repeaters over the cycle is denoted by σ_i and is given by

$$\sigma_i = i\overline{T}_i + 1 + Pr\{I_i>0\}\frac{(N-i)\lambda}{1-(1-p)^i(1-\lambda)^{N-i}} \tag{20}$$

By renewal theory arguments, the stationary system throughput is expressed as

$$S=\frac{\sum_{i=0}^{N}\pi_i S_i}{\sum_{i=0}^{N}\pi_i(\overline{T}_i+1)} \tag{21}$$

and the stationary average number of active repeaters is given by

$$\overline{n}=\frac{\sum_{i=0}^{N}\pi_i \sigma_i}{\sum_{i=0}^{N}\pi_i(\overline{T}_i+1)} \tag{22}$$

By Little's result, the average network delay is as in Eq. (8) and the probability of blocking B is simply $B = 1 - S/N\lambda$.

C. Access Delay and the Throughput-Delay Tradeoff

To complete the delay analysis, we need to evaluate the access delay D_a for a given throughput S. Let us first examine the various states a terminal can be in and the possible transitions that exist among the states. Fig. 4 represents the state diagram for the population of terminals associated with a repeater. It is clear from the diagram in Fig. 4 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, and which we denote by $T(A_1A_5)$. Assuming that the blocking probability B is uniform over time and independent of the state that the population of terminals is in, we can write

$$D_a=T(A_1A_5)=T(A_1A_2)+\left(\frac{1}{1-B}-1\right)[T(A_3A_4)+T(A_4A_2)] \tag{23}$$

To estimate $T(A_1A_2)$ and $T(A_3A_4)$, we call upon previously published results. Slotted ALOHA with an infinite population has been thoroughly analyzed by Kleinrock and Lam [6]. With the channel input from the infinite population modeled as an independent

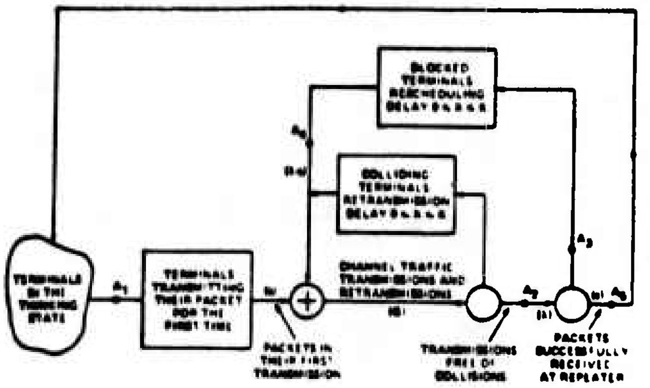


Figure 4 State Diagram for a Population of Terminals

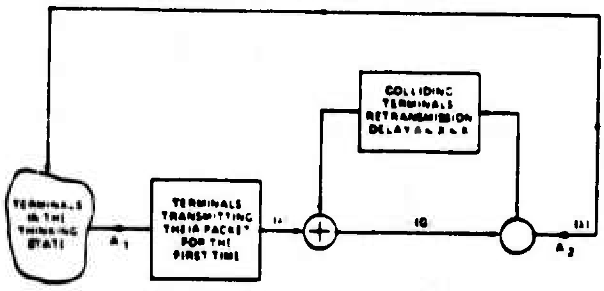


Figure 5 State Diagram for a Population of Terminals with no Blocking

Poisson process with an average of λ packets/slot, letting the maximum retransmission delay be an integer number K of packet slots (the retransmission delay assumed uniformly distributed over the K slots), and neglecting the propagation delay, the delay $D_{S\text{-}ALOHA}(\lambda)$ has been explicitly obtained in Ref.[6]. For each value of λ, we note that an optimum value of K, K_{opt}, can be selected so as to achieve minimum delay. The lower envelope of all delay curves provides the throughput-delay performance of slotted ALOHA with infinite population.

The model used by Kleinrock and Lam in determining packet delay can be represented by the diagram of Fig. 5, where the delay $D_{S\text{-}ALOHA}(\lambda)$ corresponds to $T(A_1A_2)$. Comparing the two diagrams in question, we note the following. Both include representation of slotted ALOHA channels with throughput λ. However, while in the diagram of Fig. 5 the entire input to the channel is produced by the independent Poisson process with rate λ, in the diagram of Fig. 4, only the fraction s is produced by an independent process; the remaining fraction $(\lambda - s)$ is generated by the terminals in the blocked state. Regarding transmissions by these terminals as "new" input to the channel, and neglecting the effect of the correlation in traffic introduced here by the rescheduling protocol, we approximate $T(A_1A_2)$ and $T(A_4A_2)$ by

$$T(A_1A_2) = T(A_4A_2) = D_{S\text{-}ALOHA}(\lambda) \tag{24}$$

$T(A_3A_4)$ is the average rescheduling delay and is simply given by $K_{opt}(\lambda)/2$, and thus an estimate of D_a is given by

$$D_a = \frac{1}{1-B} D_{S\text{-}ALOHA}(\lambda) + \frac{B}{1-B}\frac{K_{opt}}{2} \tag{25}$$

5. Numerical Results and Discussion

We start by examining the fully-connected DFT case. Contrary to what was shown for the DFT star configuration in [3], given λ, we note that the value of p which yields minimum D_n does not correspond to the value of p 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 the optimum total delay D for a given throughput S.

In Fig. 6 we plot the optimum D_n versus S for various values of N along with the corresponding curves obtained in the star configuration. Fig. 7 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.

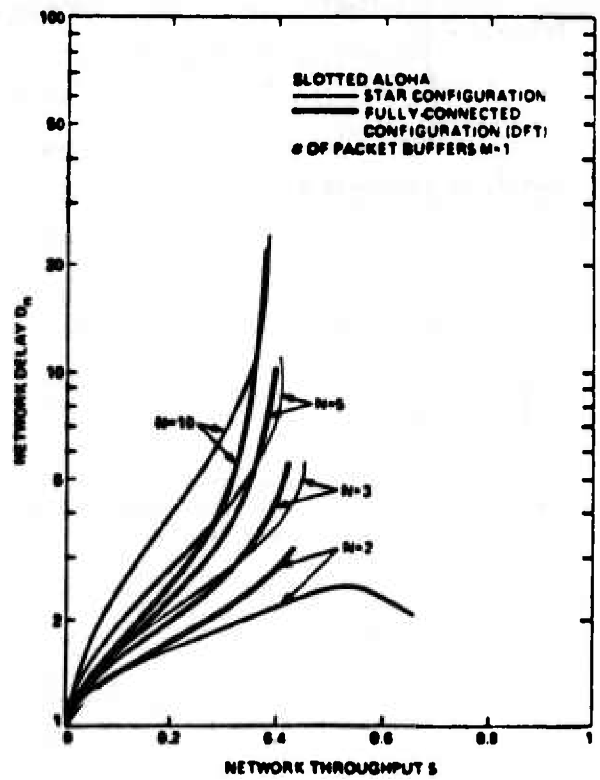


Figure 6 Optimum (Network) Throughput-Delay Curves.

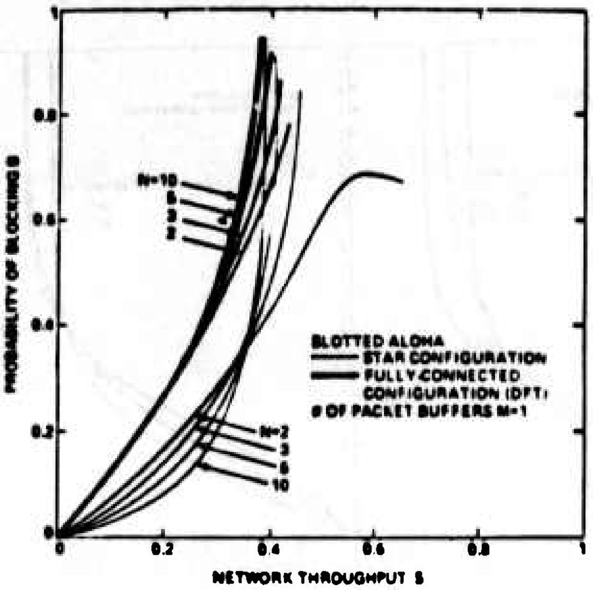


Figure 7 Minimum Blocking versus Throughput

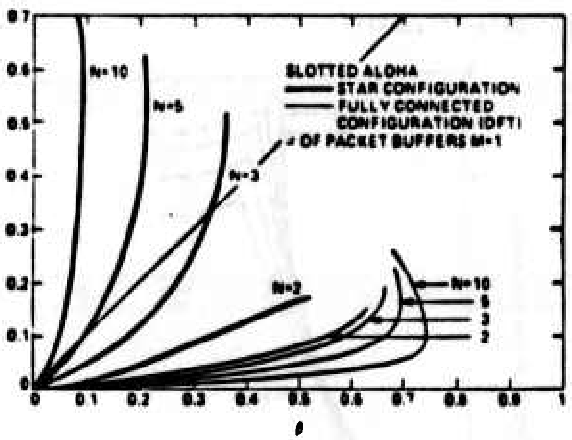


Figure 8 α versus β at Minimum Blocking

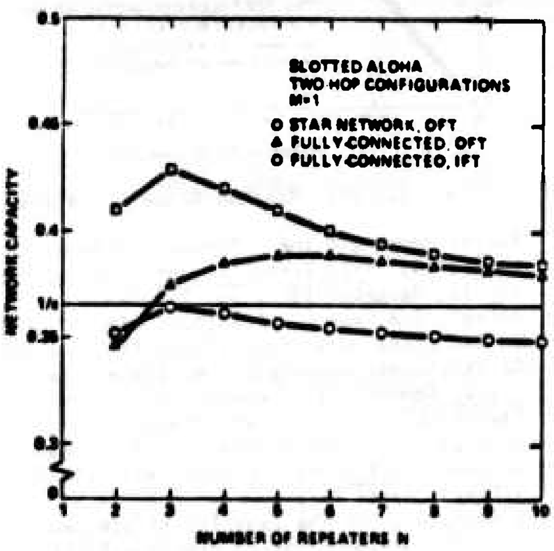


Figure 9 Network Capacity versus N

To see the importance of blocking due to repeaters' transmissions (β) relative to the blocking due to lack of storage (α), we plot in Fig. 8 α versus β for various values of N along with the curves corresponding to the star configuration; it is all too evident from this figure that β is the predominant factor and thus that the fully connected configuration is even more "channel bound" than the star configuration. Moreover, β becomes more and more important relative to α as N increases, due to the larger number of contending devices. This certainly justifies the absence of consideration in the present study for buffer sizes larger than 1.

For a two-hop centralized environment, the system capacity is obtained for $\lambda=1/e$. We plot in Fig. 9 the DFT system capacity versus N for this and the star configuration. The fully-connected environment provides a smaller network capacity than the star configuration, especially for the smaller values of N ($N<6$); however, 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 $D = D_a+D_n$ for the fully connected DFT system is plotted versus S in Fig. 10 for N=2, 5 and 10 along with, for comparison, the throughput delay curves corresponding to the star configuration. (The curves corresponding to the IFT protocol appearing in these figures will be discussed subsequently). We note that for both N=2 and N=5 the delay 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 slightly 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 predominent component of D; the improvement in D_n observed for N=10 (See Fig. 6) overcomes the degrading effect of the larger blocking probability experienced (see Fig. 7). The throughput delay curves for networks of arbitrary connectivity employing the DFT transmission protocol are expected to lie between the two displayed. However we have no formal proof for this claim!

Consider now the fully-connected IFT case. The comment we made earlier regarding minimum blocking and minimum delay in the fully-connected DFT case is also valid here. 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. 11 and B versus S in Fig. 12 (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 (and to a certain extent B) 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. 9 above; this capacity is not too sensitive to variations in the size of the network, N. The throughput-delay curves appear in Fig. 10 for N=2, 5, and 10. 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

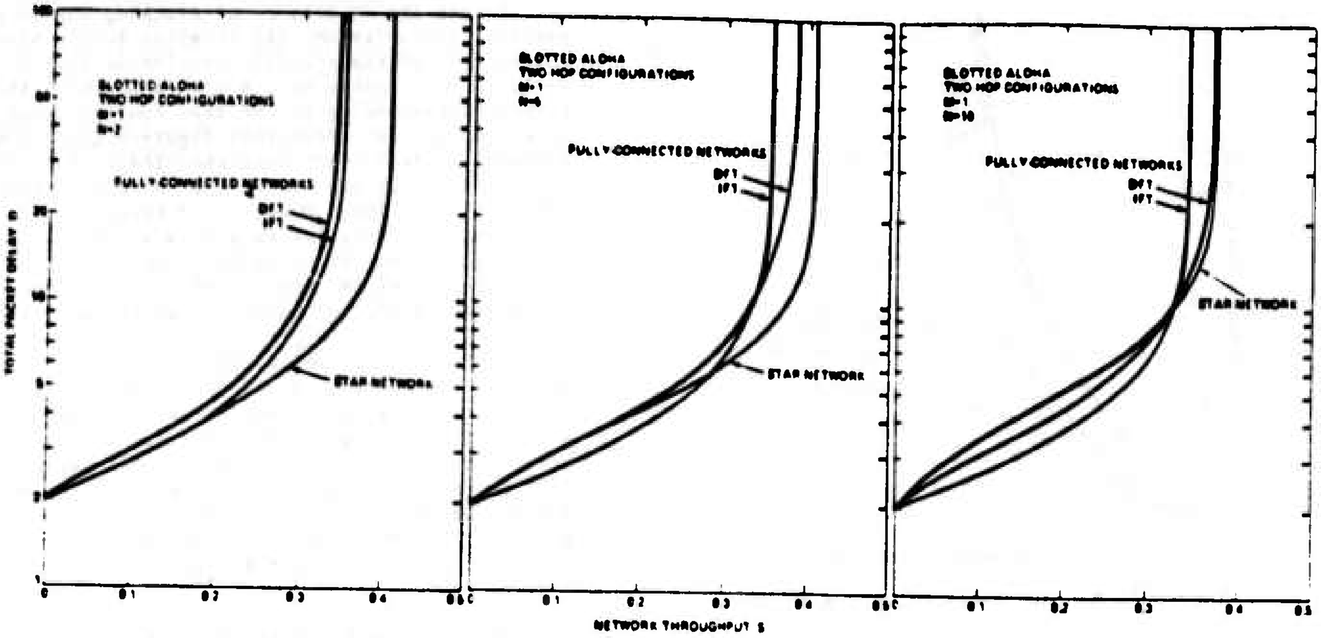


Figure 10 Throughput-Delay Tradeoff

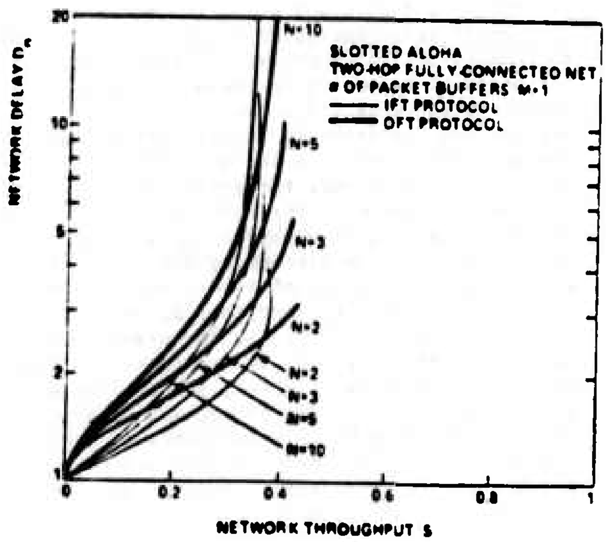


Figure 11 IFT: Network Throughput-Delay Curves

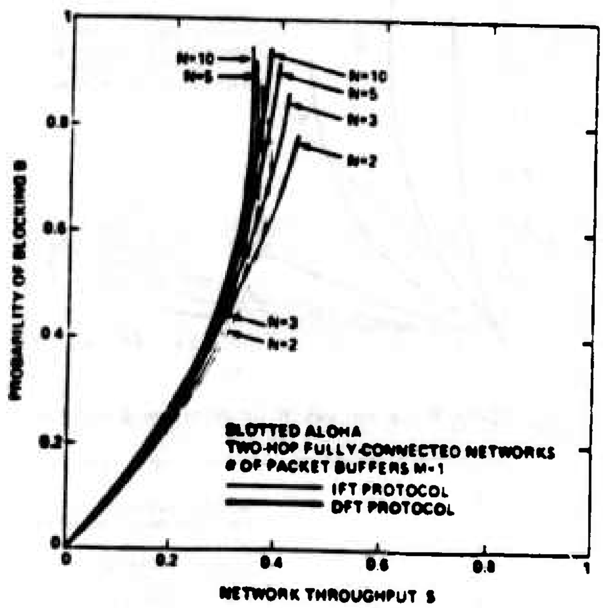


Figure 12 IFT: Minimum Blocking Versus S

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.

References

1. Abramson, N., "The Throughput of Packet Broadcasting Channels," IEEE Trans. Commun., Vol. COM-25, No. 1, January 1977, pp. 117-128.
2. Kahn, R.E., "The Organization of Computer Resources into a Packet Radio Network," AFIPS Conf. Proc., NCC 75, pp. 177-186.
3. Tobagi, F., "Performance Analysis of Packet Radio Communication Systems," Proc. of the IEEE Nat. Telecom. Conf., December 1977, pp. 12:6-1 to 12:6-7.
4. Kleinrock, L. 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, No. 12, December 1975, pp. 1400-1416.
5. Tobagi, F.A. and L. Kleinrock, "Packet Switching in Radio Channels: Part III-Polling and (Dynamic) Split Channel Reservation Multiple Access," IEEE Trans. Commun., Vol. COM-24, No. 8, August 1976.
6. Kleinrock, L. and S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. Commun., Vol. COM-23, pp. 410-423, April 1970.
7. Gitman, I., "On the Capacity of Slotted ALOHA Networks and Some Design Problems," IEEE Trans. Commun., Vol. COM-23, No. 3, March 1975, pp.305-317.

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.

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.

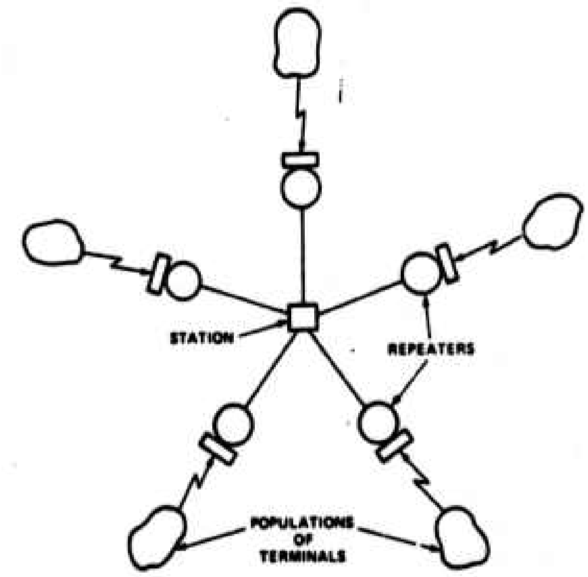


Fig. 1. A two-hop star configuration.

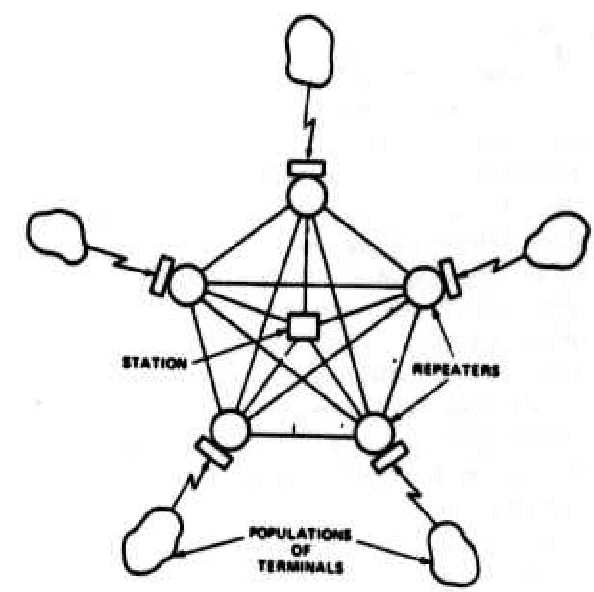


Fig. 2. A two-hop fully connected configuration.

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, $0 < n^t < N$, 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

$$p_{ij} = \begin{cases} 0 & j < i-1 \\ P_s(i)(1-\lambda)^{N-i} & j = i-1 \\ [1 - P_s(i)]\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j} & \\ \quad + P_s(i)\binom{N-i}{j-i+1}\lambda^{j-i+1}(1-\lambda)^{N-j-1} & \\ & j > i \end{cases} \tag{1}$$

where $P_s(i)$ denotes the probability of a successful transmission given i active repeaters and is expressed as

$$P_s(i) = ip(1-p)^{i-1}. \tag{2}$$

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

$$\bar{n} = \sum_{k=0}^{N} k\pi_k. \tag{3}$$

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

$$\alpha = (1-p)\bar{n}/N \tag{4}$$

$$\beta = p\bar{n}/N, \tag{5}$$

and the total probability of blocking is given by

$$B = \alpha + \beta = \bar{n}/N \tag{6}$$

The total network throughput, denoted by S, is defined as the rate of successful packets received at the station; it is given by

$$S = (N - \bar{n})\lambda. \tag{7}$$

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

$$D_n = \bar{n}/S. \tag{8}$$

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

$$p_{ij} = \begin{cases} 0 & j < i-1 \\ P_s(i) & j = i-1 \\ (1-p)^i(1-\lambda)^{N-i} + [1-(1-p)^i - P_s(i)] & \\ & j = i \\ (1-p)^i\binom{N-i}{j-i}\lambda^{j-i}(1-\lambda)^{N-j} & j > i \end{cases} \tag{9}$$

[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, $2 < N < 10$, 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.

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]

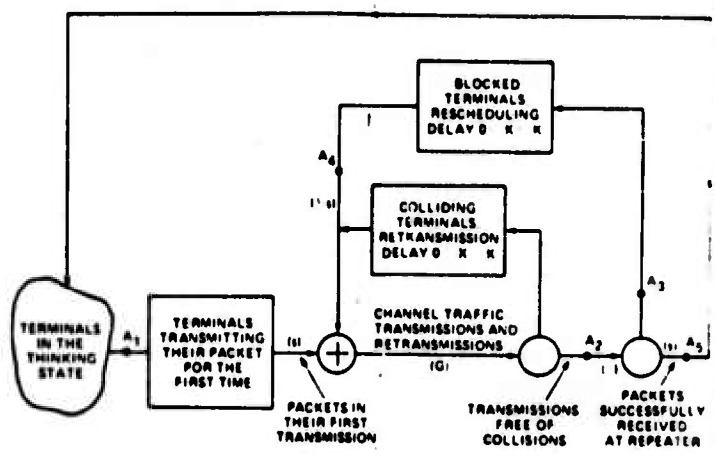


Fig. 3. State diagram for a population of terminals.

$$D_{S-\text{ALOHA}}(s, K, B) = 1 + \frac{E(K+1)}{2} + \frac{1}{2} \tag{13}$$

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.

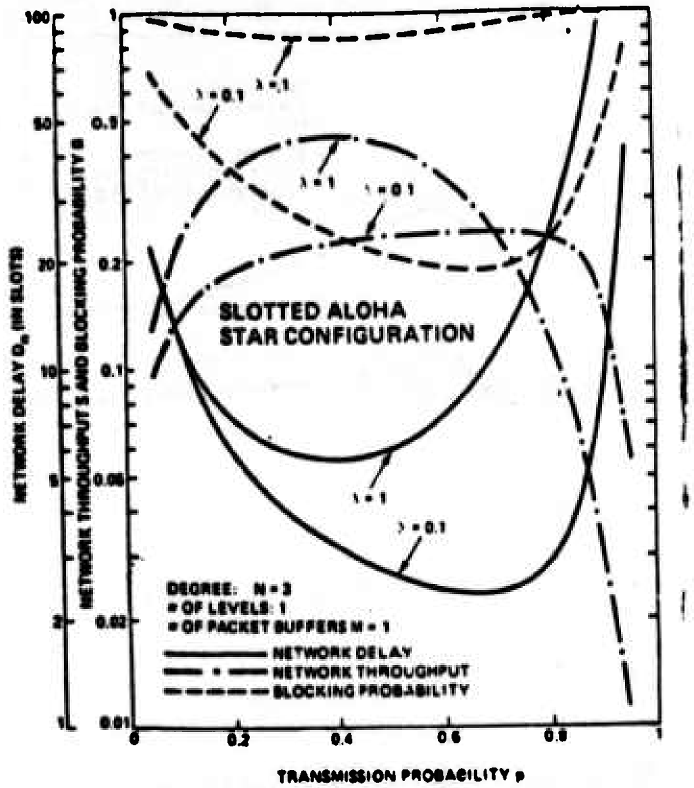


Fig. 4. Slotted ALOHA star configuration: Network delay, throughput and probability of blocking versus p.

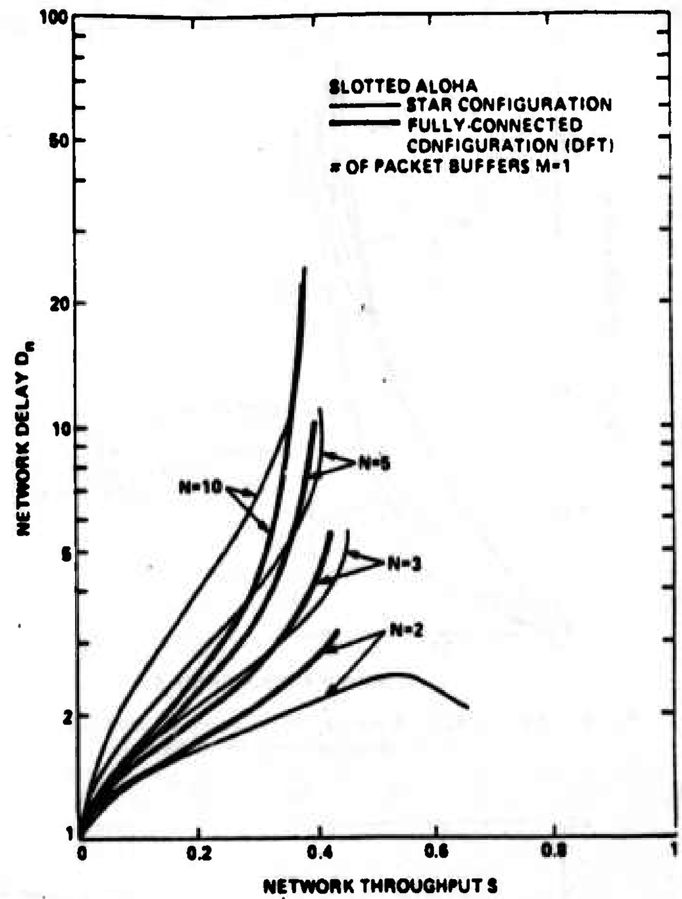


Fig. 5. Slotted ALOHA star and fully connected configurations: Optimum (network) throughput-delay curves.

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 $N (N > 3)$, 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 $N > 3$, 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.

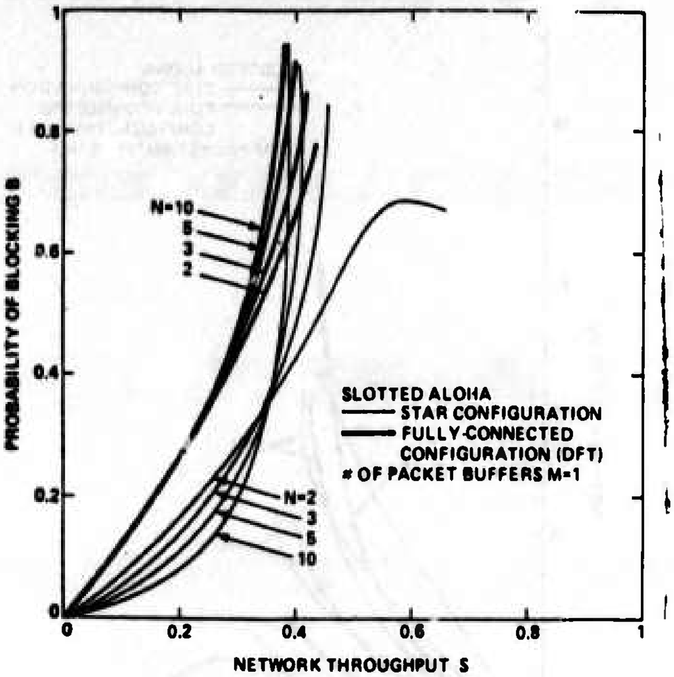


Fig. 6. Slotted ALOHA star and fully connected configurations: Minimum blocking versus throughput.

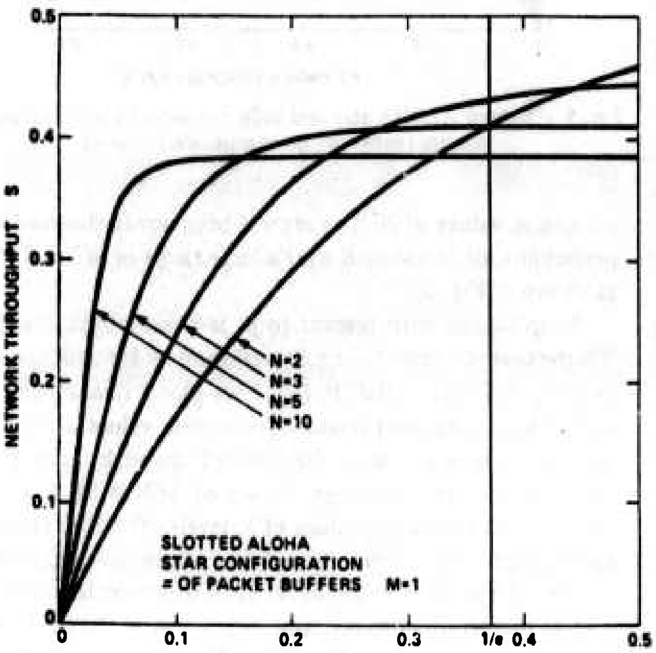


Fig. 7. Slotted ALOHA star configuration: Throughput versus λ.

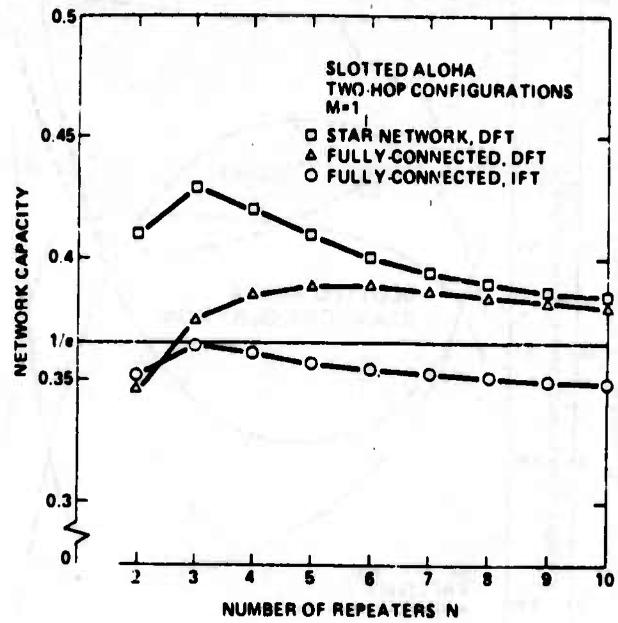


Fig. 8. Slotted ALOHA two-hop star and fully connected networks: Network capacity versus N.

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. 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 ($N < 6$), as shown in Fig. 8; but as N gets larger ($7 < N < 10$), 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, $S = \{(n_1,$

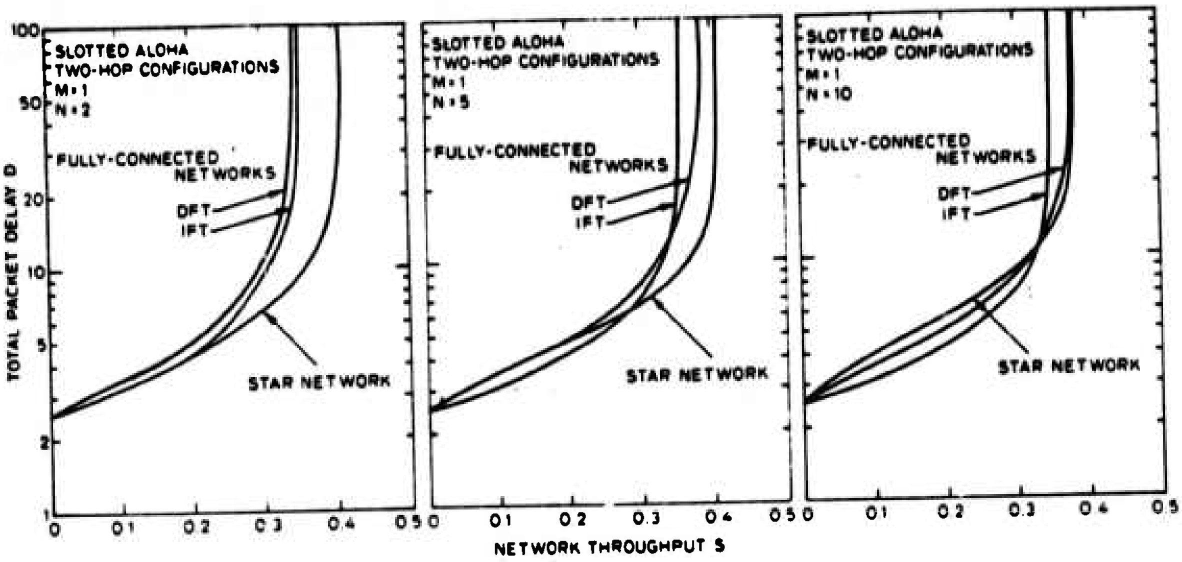


Fig. 9. Throughput-delay tradeoffs in two-hop slotted ALOHA star and fully connected networks.

$n_2, \cdots, n_N) \mid 0 \le n_i \le M, \forall i = 1, 2, \cdots, N\}$. 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

$$\Pr\{n_i = k\} = \sum_{\{\underline{n} \in S \,|\, n_i = k\}} \pi_{\underline{n}}. \tag{15}$$

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

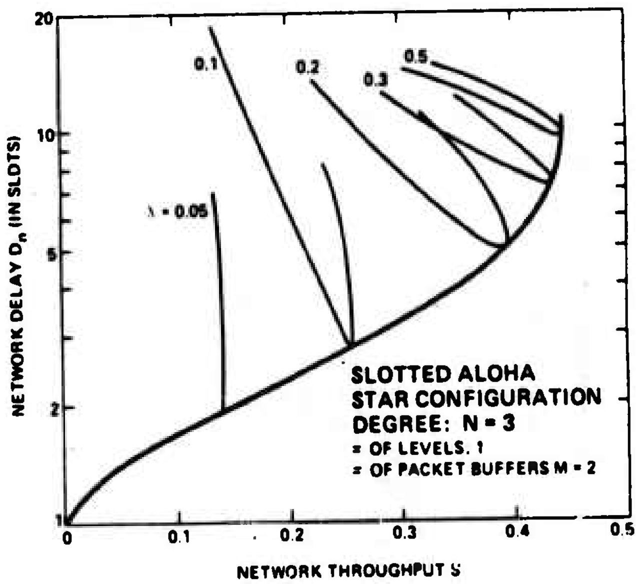


Fig. 10. Slotted ALOHA star configuration: Network delay versus throughput with $M > 1$.

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-

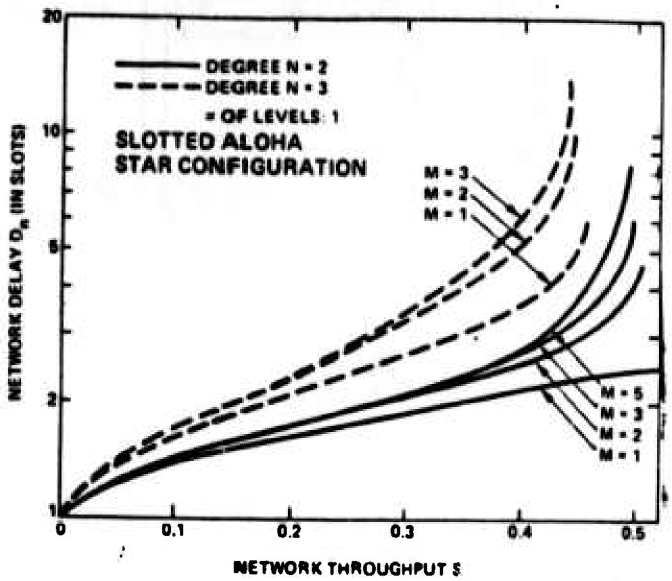


Fig. 11. Slotted ALOHA star configuration: Minimum network delay versus S for various values of M.

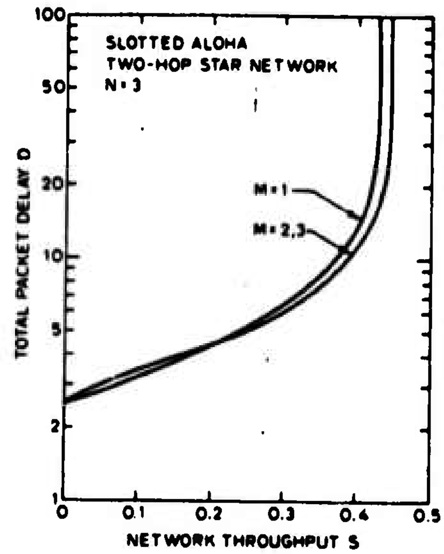


Fig. 13. Two-hop slotted ALOHA star networks: Total packet delay versus S for $N = 3$ and $M > 1$.

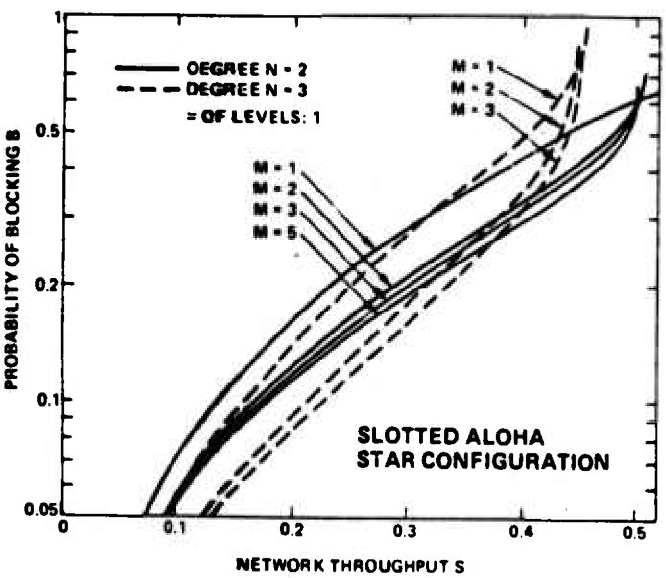


Fig. 12. Slotted ALOHA star configuration: Minimum blocking versus S for various values of M.

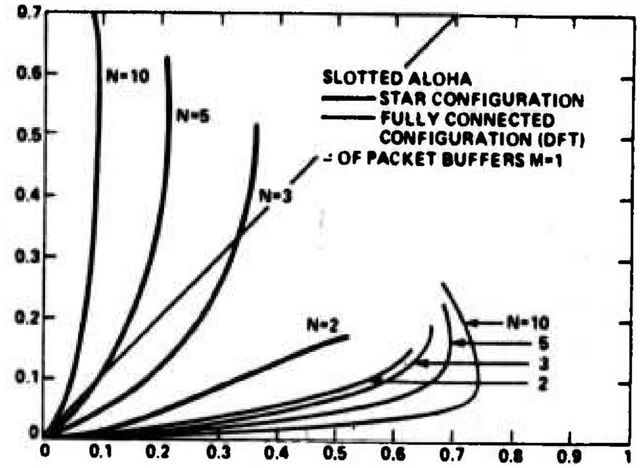


Fig. 14. Slotted ALOHA star and fully connected configurations: α versus β at minimum blocking.

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

$$S = \frac{\sum_{i=0}^{N} \pi_i S_i}{\sum_{i=0}^{N} \pi_i(\bar{I}_i + 1)} \tag{23}$$

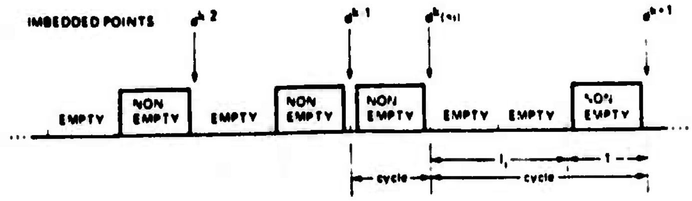


Fig. 15. The imbedded Markov chain in the slotted ALOHA IFT protocol.

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.

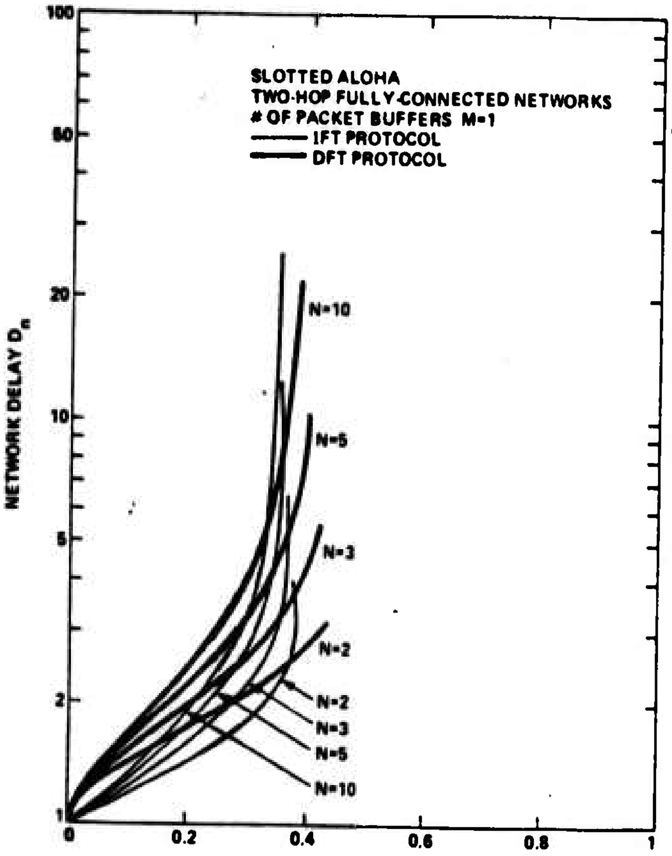


Fig. 16. Slotted ALOHA fully connected configuration: Optimum (network) throughput-delay curves.

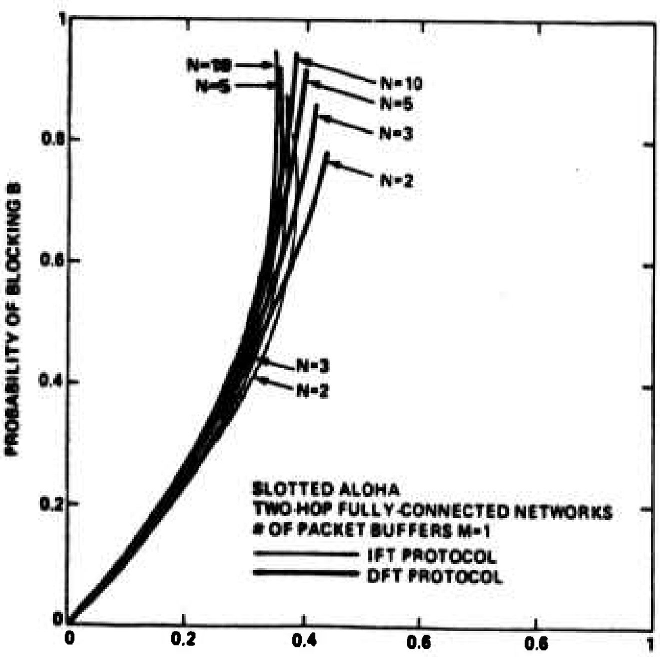


Fig. 17. Slotted ALOHA fully connected configuration: Minimum blocking versus S.

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 second 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 $I_0 = \{j \mid m_j = n_j\}$ we have

$$\Pr\{n \mid m\} = \left[\prod_{j\in I_0} [p^{\chi_j} + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)] - \sum_{j\in I_0} p\chi_j \prod_{\substack{k\in I_0\\ k\neq j}} (1-p)^{\chi_k}(1-\lambda+\lambda\zeta_k)\right] \cdot \prod_{j\notin I_0} (1-p)^{\chi_j}\lambda$$

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 $j \in I_0$ remains unchanged with probability $p\chi_j + (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_j)$, *provided* that any transmission (represented by the term $p\chi_j$) is unsuccessful. Since the repeaters are all independent, the probability of the event $\{m_j = n_j\}$ for all $j \in I_0$ is then given by the expression in the first bracket, in which the summation

$$\left[\sum_{j\in I_0} p\chi_j \prod_{\substack{k\in I_0\\ k\neq j}} (1-p)^{\chi_j}(1-\lambda+\lambda\zeta_k)\right]$$

represents the probability of all possible *successful* transmissions. Now for all $j \notin I_0$, 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

$$p_{N,j} = \begin{cases} 0 & j < N-1 \\ \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j = N-1 \\ 1 - \dfrac{Np(1-p)^{N-1}}{1-(1-p)^N} & j = N. \end{cases} \tag{B.4}$$

The probability density function of I_i is given by

$$Pr\{I_i = l\} = \begin{cases} (1-\lambda)^{N(l-1)}[1-(1-\lambda)^N] & i = 0; l \geq 1 \\ 1-(1-p)^i & i \neq 0, N; l = 0 \\ (1-p)^i[(1-\lambda)^{N-i}(1-p)^i]^{l-1} \cdot [1-(1-\lambda)^{N-i}(1-p)^i] & i \neq 0, N; l > 0 \\ (1-p)^{Nl}[1-(1-p)^N] & i = N; l \geq 0. \end{cases} \tag{B.5}$$

Thus $\bar{I}_i$ is expressed as

$$\bar{I}_i = \begin{cases} \dfrac{1}{1-(1-\lambda)^N} & i = 0 \\ \dfrac{(1-p)^i}{1-(1-p)^i(1-\lambda)^{N-i}} & i \neq 0, N. \\ \dfrac{(1-p)^N}{1-(1-p)^N} & i = N. \end{cases} \tag{B.6}$$

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) [redacted], [redacted] 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 © 1980 IEEE

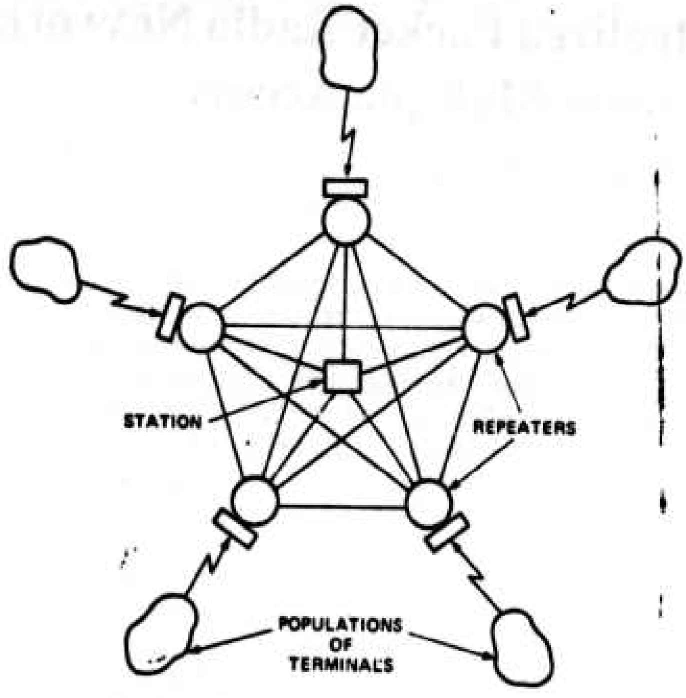


Fig. 1. A two-hop fully connected configuration.

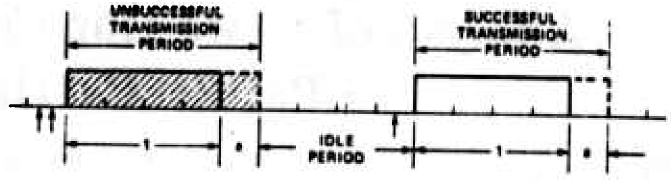


Fig. 2. Slotted nonpersistent CSMA: Transmission and idle periods (vertical arrows represent terminals becoming ready to transmit).

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 *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

$$\Pr\{Y \leq y\} = 1 - e^{-\lambda'(y-1)} \qquad y \geq 1. \tag{2}$$

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

[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]

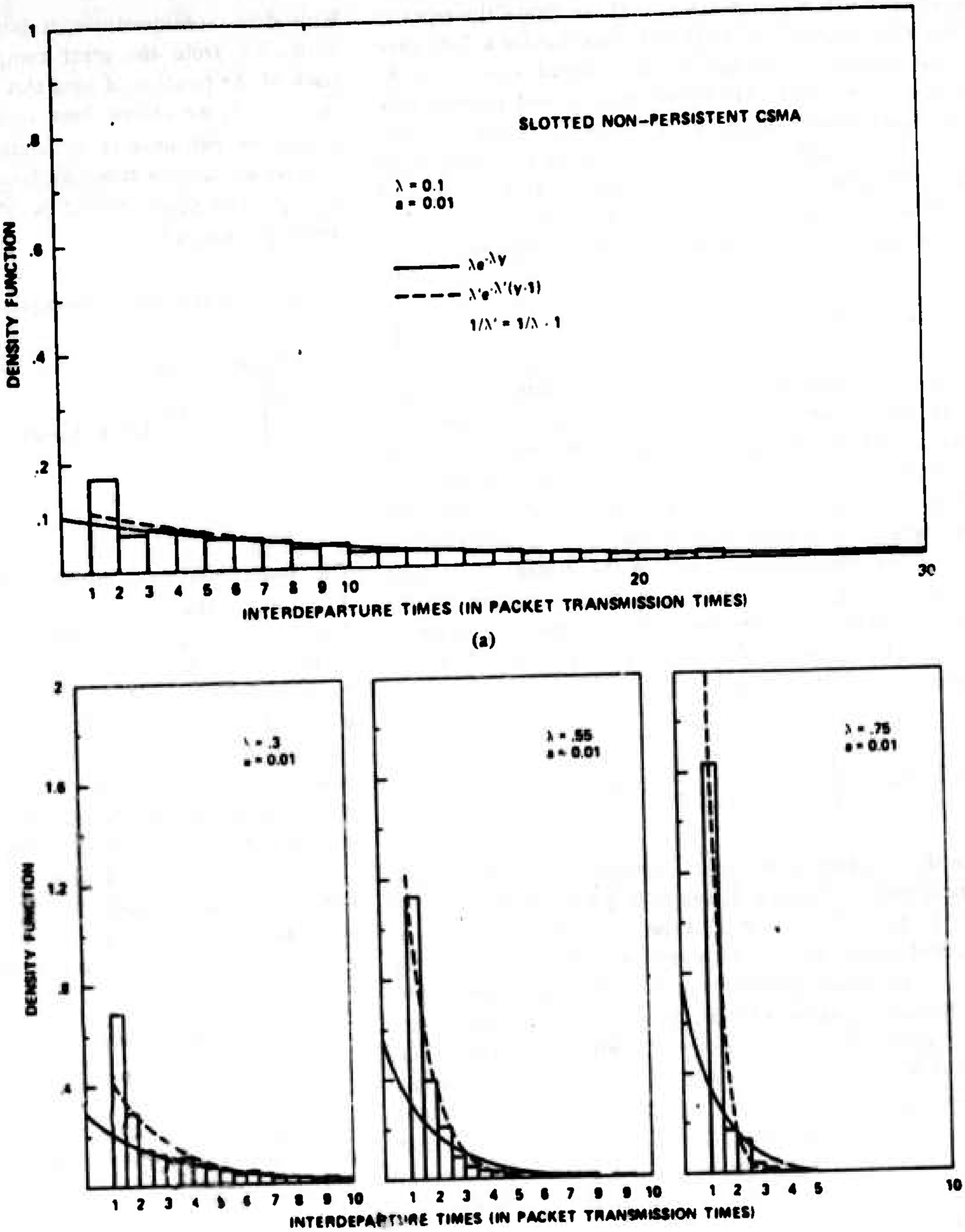


Fig. 3. (a) Histograms of interdeparture times in slotted nonpersistent CSMA ($\lambda = 0.1$). (b) Histograms of interdeparture times in slotted nonpersistent CSMA ($\lambda = 0.3$, 0.55 and 0.75).

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-

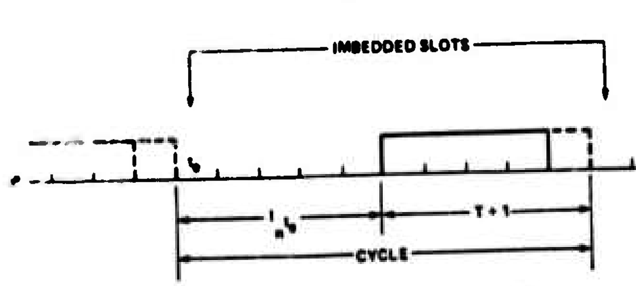


Fig. 4. The imbedded slots in time line R.

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

$$\Pr\{Y_i \leq y\} = \frac{\lambda}{G} + \left(1 - \frac{\lambda}{G}\right)\frac{y}{T} \qquad 0 \leq y \leq T. \tag{4}$$

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

$$\sigma_n = (\bar{I}_n + T + 1)n + (T + 1)\Pr\{J_n \geq Y_m'\} \cdot \frac{(N - n)a\lambda'}{1 - (1 - \nu)^n(1 - a\lambda')^{N-n}}. \tag{8}$$

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

$$S = \frac{\sum_{n=0}^{N} \pi_n S_n T}{\sum_{n=0}^{N} \pi_n(\bar{I}_n + T + 1)} \tag{9}$$

$$\bar{n} = \frac{\sum_{n=0}^{N} \pi_n \sigma_n}{\sum_{n=0}^{N} \pi_n(\bar{I}_n + T + 1)}. \tag{10}$$

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

$$D_a = \frac{1}{1 - B} D_{\mathrm{NPCSMA}}(\lambda) + \frac{B}{1 - B}\,\delta(\lambda) \tag{11}$$

where $D_{\mathrm{NPCSMA}}(\lambda)$ 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 $D_{\mathrm{NPCSMA}}(\lambda)$; B is the probability that a T_i-successful packet gets blocked at the receiving repeater and is expressed as

$$B = 1 - \frac{S}{N\lambda}. \tag{12}$$

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 ($\nu < 0.001$) 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 ($N < 10$), 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 $n^{t_e} = n$. 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

$$0 = Y_0' < Y_1' < Y_2' - \cdots < Y_{N-n}' < Y_{N-n+1}' = \infty.$$

2) $j \leftarrow 0$.

3). Generate a random variable J_n^j such that

$$\Pr\{J_n^j = k\} = [(1 - \nu)^n(1 - a\lambda')^j]^{(k-1)} \cdot [1 - (1 - \nu)^n(1 - a\lambda')^j] \tag{13}$$

If

$J_n^j < Y_{j+1}' - Y_j'$ then do:

$$I_n = Y_j' + J_n^j \tag{14}$$

[3] Accurate within four decimals (the accuracy used in printing the results).

$$S_n = \frac{n\nu(1-\nu)^{n-1}(1-a\lambda')^j + ja\lambda'(1-a\lambda')^{j-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda')^j} \tag{15}$$

$$\sigma_n = (\bar{I}_n + T + 1)\nu + \frac{a\lambda'(T+1)}{1-(1-\nu)^n(1-a\lambda')^j} \tag{16}$$

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 $(\bar{I}_n)_{\text{sim}}$, $(S_n)_{\text{sim}}$, $(\sigma_n)_{\text{sim}}$, 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 $(\bar{I}_n)_{\text{sim}}$, $(S_n)_{\text{sim}}$, $(\sigma_n)_{\text{sim}}$ for $\bar{I}_n$, 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

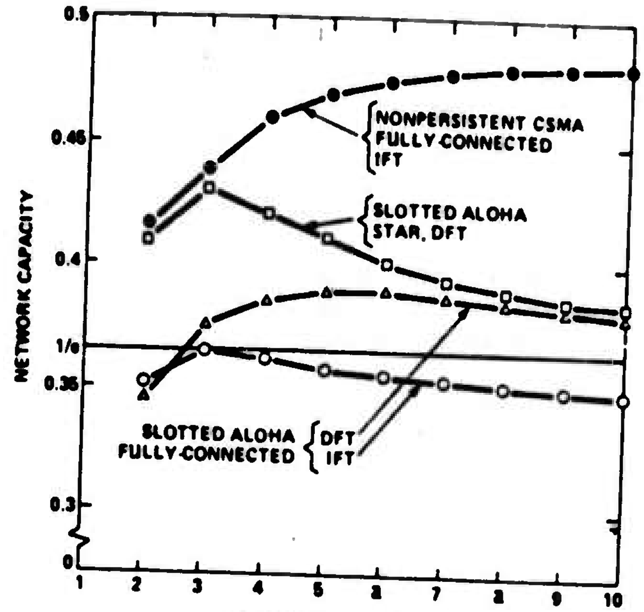


Figure 5. Network capacity versus N for slotted ALOHA and nonpersistent CSMA networks ($a = 0.01$).

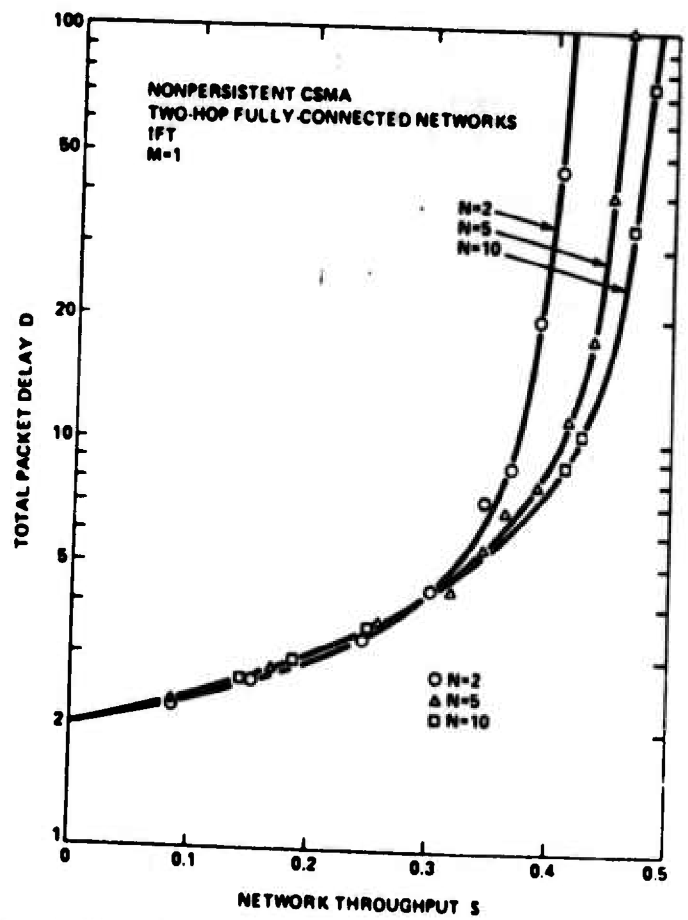


Fig. 6. Throughput-delay tradeoffs in nonpersistent CSMA two-hop fully connected networks ($a = 0.01$).

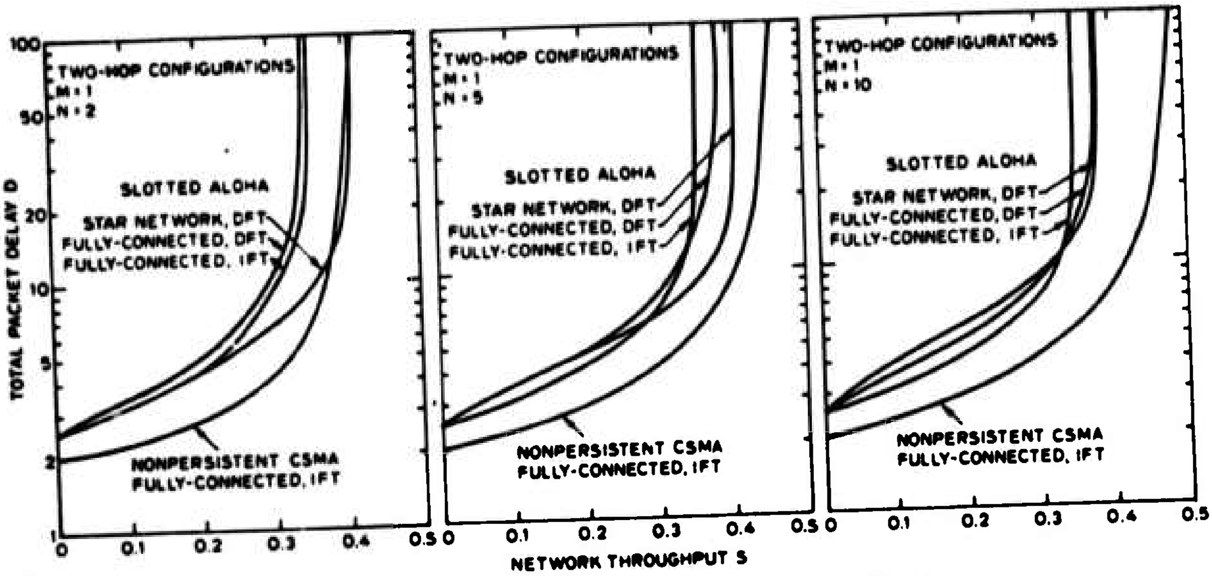


Fig. 7. Comparison between slotted ALOHA and nonpersistent CSMA ($a = 0.01$) for various values of N.

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 $1 \leq n \leq N-1$

$$p_{n,j} = \begin{cases} 0 & j<n-1 \\ \Pr\{J_n < Y_m'\}\dfrac{n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n} + \Pr\{J_n \geq Y_m'\}\dfrac{(1-a\lambda')^{N-n}n\nu(1-\nu)^{n-1}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n-1 \\ \Pr\{J_n < Y_m'\}\dfrac{1-n\nu(1-\nu)^{n-1}-(1-\nu)^n}{1-(1-\nu)^n} & \\ \quad + \Pr\{J_n \geq Y_m'\}\dfrac{(1-a\lambda')^{N-n}[1-n\nu(1-\nu)^{n-1}-(1-\nu)^n]+(N-n)a\lambda'(1-a\lambda')^{N-n-1}(1-\nu)^n}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n \\ \Pr\{J_n \geq Y_m'\}\dfrac{(N-n)a\lambda'(1-a\lambda')^{N-n-1}[1-(1-\nu)^n]}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j=n+1 \\ \Pr\{J_n \geq Y_m'\}\dfrac{\binom{N-n}{j-n}(a\lambda')^{j-n}(1-a\lambda')^{N-j}}{1-(1-\nu)^n(1-a\lambda')^{N-n}} & j>n+1. \end{cases} \tag{A.3}$$

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

$$\Pr\{J_n = k \mid Y_m' = y, J_n < y\} = \frac{(1-\nu)^{n(k-1)}[1-(1-\nu)^n]}{1-(1-\nu)^{n(y-1)}}. \tag{A.13}$$

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 $n \neq 0, N$,

$$\bar{I}_{n|Y_m'=y} = \frac{1-(1-\nu)^{ny} - y(1-\nu)^{n(y-1)}[1-(1-\nu)^n]}{1-(1-\nu)^n} + y(1-\nu)^{n(y-1)} + \frac{(1-\nu)^{n(y-1)}}{1-(1-\nu)^n(1-a\lambda')^{N-n}}$$

$$= \frac{1}{1-(1-\nu)^n} + (1-\nu)^{n(y-1)} \cdot \left[\frac{1}{1-(1-\nu)^n(1-a\lambda')^{N-n}} - \frac{(1-\nu)^n}{1-(1-\nu)^n}\right]. \quad \text{(A.15)}$$

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.