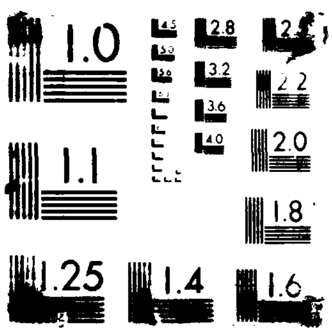
AD-A198 090 RANDOM ACCESS ALGORITHMS FOR MULTIUSER COMPUTER 1/2
COMMUNICATION NETWORKS..(U) VIRGINIA UNIV
CHARLOTTESVILLE DEPT OF ELECTRICAL ENGINEERING.
UNCLASSIFIED P PAPANTONI-KAZAKOS ET AL. JUL 88 F/G 12/7 NL


1.0
2.8
3.2
3.6
4.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

A Technical Report
Grant No. N00014-86-K-0742
September 1, 1986 - August 31, 1988

RANDOM ACCESS ALGORITHMS FOR MULTIUSER COMPUTER COMMUNICATION NETWORKS

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, VA 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

M. Paterakis
Graduate Assistant

Report No. UVA/525415/EE89/115
July 1988

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

A Technical Report

Grant No. N00014-86-K-0742

September 1, 1986 - August 31, 1988

RANDOM ACCESS ALGORITHMS FOR MULTIUSER COMPUTER COMMUNICATION NETWORKS

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, VA 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

M. Paterakis
Graduate Assistant

Department of Electrical Engineering
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/525415/EE89/115

Copy No. ____

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release: distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UVA/525415/EE89/115	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION University of Virginia Dept. of Electrical Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Resident Representative
6c. ADDRESS (City, State, and ZIP Code) Thornton Hall Charlottesville, VA 22901	7b. ADDRESS (City, State, and ZIP Code) 818 Connecticut Ave., N.W. Eighth Floor Washington, D.C. 20006
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N0C014-86-K-0742
8c. ADDRESS (City, State, and ZIP Code) 800 North Quincy Street Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Random Access Algorithms for Multiuser Computer Communication Networks

12. PERSONAL AUTHOR(S)

P. Papantoni-Kazakos, M. Paterakis

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 9/1/86 TO 8/31/88	1988 July	112

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

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

We consider a computer communication system with many geographically distributed stations accessing a common channel. Stations receive only channel feedback information. The problem is to design efficient algorithms for dynamically allocating the channel resource among the stations. The main performance measures of such algorithms are throughput (channel utilization), induced delays, and feedback error sensitivity.

For many communication applications with time constraints (e.g., transmission of packetized voice messages), a critical performance measure is the percentage of messages which are transmitted within a given amount of time after their generation at the transmitting station. In this talk we present a random access algorithm (RAA) suitable for time-constrained applications. Performance analysis demonstrates that significant message delay improvement is attained, at the expense of minimal traffic loss.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION

SECURITY CLASSIFICATION OF THIS PAGE

We also consider the case of noisy channels. The noise effect appears at erroneously observed channel feedback. Error sensitivity analysis shows that the proposed random access algorithm is insensitive to feedback channel errors.

Window Random Access Algorithms (RAAs) are considered next. These algorithms constitute an important subclass of Multiple Access Algorithms (MAAs); they are distributive and they attain high throughput and low delays by controlling the number of simultaneously transmitting users. We propose a method for delay distribution analysis of Window RAAs. It is shown that a recently proposed method for computing bounds on the moments of the delay process~~,~~ can be extended to provide bounds on the distribution of the delays. The quantities of interest are related to the solution of a denumerable system of linear equations. The methodology is applied to the delay distribution analysis of two well known Window RAAs.

RANDOM ACCESS ALGORITHMS FOR MULTIUSER COMPUTER COMMUNICATION NETWORKS

A Dissertation

presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy

by

Michael Paterakis

August 1988

APPROVAL SHEET

This Dissertation is submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Michael Paterakis
(Author)

This Dissertation has been read and approved by the examining committee:

Dissertation adviser

Accepted for the school of Engineering and Applied Science:

Dean, School of Engineering
and Applied Science

August 1988

Dedication

To my family

ACKNOWLEDGEMENTS

I am grateful to my advisor, Professor P. Papantoni-Kazakos, for motivating my interest in the topic of my research and for her encouragement and support throughout my stay at both the University of Virginia and the University of Connecticut.

I would also like to thank Dr. L. Georgiadis for giving me the chance to work with him during my stay at the University of Connecticut, and during the year he spent at the University of Virginia. It was a rewarding experience. I am indebted to him for the fruitful and enlightening discussions on my research topic, and above all his true friendship.

Special appreciation is due to:

Professor S. Wilson for serving as the Chairman of my dissertation committee, and for giving me the chance to attend three of his courses that helped my understanding in communication systems, and to:

Professor L. Pitt for serving as a member of my dissertation committee, and for teaching me rigorous mathematics in a simple manner.

I also want to thank Professor G. Giannakis for his interest in my research, and the enriching conversations we had. Above all, I want to thank him for his true friendship, and for the fun we had together.

Additionally, I want to express my appreciation to all the colleagues and staff of the

During the last four years a number of friends and colleagues have influenced my thinking and supported my effort. Special thanks go to Dr. C. Bisdikian, Prof. M. Georgiopoulos, Prof. L. Merakos, Prof. I. Stavrakakis, Dr. H. Tsaknakis,Prof. K. Zografos, and Mr. C. Loucopoulos.

Above all, I want to thank my family in Greece for their love, understanding, and support.

The support by the U.S. Office of Naval Research under the contract N00014-86-K-0742, and by the National Science Foundation under the grant ECS-85-06916, is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

I.1 Overview of the Multiple Access Problem

We consider the situation where a number of geographically distributed, independent users wish to communicate with each other, or with a central station, by transmitting messages over a single channel.

The problem then is the design of an algorithm, that is a set of rules which determine the time instants when a user can transmit his messages.

This problem is referred to as the **Multiple Access Problem,** and the algorithm employed is called a **Multiple Access Algorithm (MAA).** Typical examples of multi - user communication systems are the following [1] :

(1) **Geosynchronous Satellite Systems,** where many ground stations transmit to a common satellite receiver, with the received messages being relayed to the ground stations.

(2) **Packet Radio Networks.** Here a message transmitted by a radio transmitter may be received over a wide area by any number of receivers. This is referred to as the **broadcast capability.** Thus a ground packet radio channel provides a completely connected network topology for a large number of users within range of each other.

It is assumed that at most one message can be successfully transmitted at a time over the common communication channel. If more than one messages are transmitted simultaneously over the common channel we say that a **collision** occurred, then the received signal is the sum of attenuated transmitted signals, perhaps corrupted by distortion and noise. The information contained in the original messages is assumed lost, thus each message involved in a collision must be retransmitted at some later time, with further retransmissions possible until the message is successfully transmitted.

The MAA may be **centralized** (when a central controller dictates the action of each user), **distributed** (when each user acts according to prespecified rules without the intervention of a central controller), or a combination of both.

The performance of a MAA depends on the statistical nature of the message generating process. The Frequency Division Multiple Access (FDMA) , and Time Division Multiple Access (TDMA) algorithms are the first designed to deal with the problem. They are both Collision - Free algorithms.

In the **FDMA** algorithm, the available bandwidth is divided in bands, and each band is dedicated to a user. In the **TDMA** algorithm, the time is divided in frames, each frame is subdivided into slots, and each user is allowed to transmit only once per frame, and only within the slot dedicated to him. When the users transmit messages continuously, both the FDMA and the TDMA are very efficient, permitting 100 % utilization of the channel. However, if the number of users assigned channels is large and possibly time varying, and if each user transmits bursts of data infrequently, both the above algorithms induce unnecessarily long delays. Consider for example, the TDMA algorithm and, assume that there are M (M>>1), users in the network. A message generated by some user will have to wait an average of M/2 time slots for its transmission, even if no other user has messages for transmission.

Measurements conducted on time - sharing systems indicate that both computer and terminal data streams are bursty, [2]. Depending on channel speed the ratio between the peak and the average data rates may be as high as 2000 to 1, [3]. From the above discussion it becomes clear that, if the number of users is large, and if the users generate bursts of messages infrequently, different channel sharing techniques are needed.

The **Random Access Algorithms (RAAs),** are designed exactly for this purpose. The main difference between a RAA and a perfectly scheduled MAA (e.g TDMA, or FDMA) is that the former allows simultaneous transmission attempts by different users in the same frequency band. Therefore, collisions are possible. Our objective then is to design **Collision Resolution Algorithms (CRAs),** which resolve the resulting collisions and maintain stable system operation and low delays (time that elapses from the instant when a message is generated until this message is successfully transmitted). Due to the fact that a portion of the channel capacity is used for the resolution of collisions, the channel utilization is in general less than 100 %. However, the message delays at low traffic rates are generally low, and the channel utilization depends on the total traffic rate, and not on the number of users.

Only **slotted, synchronous** Random Access Communication Systems will be considered. In the sequel we list the modeling assumptions and briefly discuss their implications.

(1) **Slotted, Synchronous Systems.** Messages are divided into constant size packets, and each packet requires one time slot for transmission. All users are synchronized to the beginning of the slot intervals. Packet transmission may start only at the beginning of some slot.

(2) **Poisson Arrivals.** It is assumed that the cumulative packet generating process is a Poisson random variable, independent from slot to slot. The **Throughput** of the RAA is then defined as the supremum of all Poisson rates λ, such that, with probability one, every packet is successfully transmitted with finite delay.

(3) **Collision and Perfect Reception.** It is assumed that if two or more users attempt to transmit their packets in a given time slot, then a collision occurs and all the information contained in the transmitted packets is destroyed, thus retransmission is then necessary. If just one user sends a packet in a given time slot, then the packet is successfully transmitted.

(4) **Immediate Feedback Information.** At the end of each slot, each user or perhaps a subset of the users, obtain feedback information about the transmission activity in that slot. This feedback information is not necessarily error-free.

(5) **Infinite Number of Users.** The system has an infinite number of users and each newly arriving packet arrives at a new user. This assumption leads to a worst case model. The design of algorithms under this assumption is of interest, since their performance will not be greatly affected by the number of users in the system.

The slotted system assumption (1) leads to a discrete - time system, thus simplifying the analysis. Synchronizing the users is not entirely trivial, but can be accomplished with relatively stable clocks and some guard time between the end of a packet transmission and the beginning of the next slot. The assumption of Poisson arrivals (2) approximates well the case of a large number of independent bursty users. The assumption of Collision and Perfect Reception (3) ignores the possibility of errors due to noise, and also ignores the possibility of **Capture** techniques, by which a receiver can capture one transmission

Feedback (4) is quite unrealistic, particularly in the case of satellite channels. Fortunately delayed feedback although complicates random access algorithms, causes no fundamental problems.

A **RAA** is the combination of the **CRA** and the **First Time Transmission Rule** (FTTR). It is distributed in nature, since action to be taken by each user is not based on the knowledge of the status of the other users.

The oldest RAA is the **ALOHA** [4] algorithm. It was developed around 1970 to provide radio communication between the central computer and various data terminals at the campuses of the University of Hawaii. Each user with a new packet transmits this packet at the beginning of the slot immediately after the packet arrival time. The feedback information is of the acknowledgment type. Therefore only the users that are involved in a collision are informed about it. To resolve the collision the above users select random time instants for the retransmission of their packets. As a result, the system contains two types of users, at each time slot: those that have new packets to transmit, and those whose packets have previously collided, and are attempting to retransmit (Blocked Users). The Aloha algorithm was first analyzed by Abramson [4]. The number of users was assumed infinite, and the number of new packets generated by the users at a given time slot, was assumed to be Poisson distributed with rate λ packets per slot. However, to simplify the analysis, some additional assumptions were made on the distribution of the blocked users, and a steady - state equilibrium condition was assumed. Unfortunately, a careful examination of the algorithm [5] shows that the system is unstable for any $\lambda > 0$, in the sense that the output rate eventually decreases to zero.

The ALOHA system with a fixed number M of users has also been studied [6]. It tion of the system is ergodic, if and only if the per packet retransmission probability p can be chosen to satisfy the inequality

$$\lambda < pM(1-p)^{M-1}$$

From this inequality the following conclusions can be drawn:

(a) If $M \rightarrow \infty$, then λ cannot exceed e^{-1}.

(b) If λ, p and M are such that the above inequality is satisfied, then there exists M_0, such that the system is not stable for all $M \geq M_0$. In other words the ALOHA algorithm is sensitive to the number of users in the system.

Since the ALOHA algorithm is unstable under the infinite population model, the natural question is whether the design of stable RAAs is possible. The existence of a stable RAA, for the Poisson packet generating process, was shown independently [7] and [8]. In those studies the channel is assumed slotted and the feedback informs **all the users** in the system whether that slot contained zero, one, or more than one packet transmissions (ternary feedback). The actions of the users depend only on the received feedback. It is required that every user observes the feedback channel independently if he has a packet to transmit, from the time the system started operating **(continuous feedback sensing).** The algorithm can operate even if the feedback is binary (informing whether a slot contained more than one packet transmissions or not, or in other words collision versus noncollision feedback), and it can be shown that it is stable if $\lambda < 0.345$. A modification of this algorithm increases its maximum throughput to 0.42. Under ternary feedback, algorithms whose maximum throughput is 0.487 [9],[10] have been devised. It can be shown however, that in the presence of feedback errors the latter algorithms become unstable [11], while those with the binary feedback are still operational at the expense of some throughput reduction

The requirement of continuous feedback sensing is usually unrealistic. Commonly in practice, a user becomes active only when he has a packet to transmit. Also in the case of a failure, the continuity of channel sensing is interrupted. It is thus desirable to devise algorithms which require that each user observes the feedback only when he is active. Those algorithms are called **Limited Feedback Sensing RAAs.**

In [12] the first such algorithm was presented. Modifications that improve its performance can be found in [13], [14] and [15]. The feedback is assumed to be either binary or ternary, the algorithms have good delay characteristics for low arrival rates and their maximum throughput varies from 0.36 to 0.4076. The analysis in [13], also shows that the behavior of those algorithms in the presence of feedback errors is quite satisfactory. In [16], [17] and [18] another class of limited feedback sensing algorithms was presented. These algorithms can achieve throughput as high as the highest throughput achievable by the known continuous-feedback-sensing algorithms, at the expense of a small increase in delays and complication. They are also not blocked (i.e., they do not lead to deadlocks), in the presence of feedback errors.

Three things are worth noting here:

(i) Both continuous and limited feedback sensing RAAs need more feedback sensing than that assumed by the ALOHA algorithm. Indeed, the ALOHA algorithm requires that each user know only the outcomes of his own transmissions. The existence of stable algorithm for the infinite population model subject to an acknowledgement type of feedback is still an open problem.

(ii) The algorithms developed in [7], [8], [9] and [10] are called **blocked access algorithms.** In that case, all users are able to identify the end of a Collision Resolution feedback continuously, (a CRI is defined as the time period between the time a collision occurred and the time when all the packets involved in the collision have been successfully transmitted). On the other hand the algorithms developed in [12], [13], [14] and [15] , are called **free access algorithms.** The latter algorithms do not need the channel feedback history before the packet generation to determine the time of the first transmission attempt. (A user with a new packet, transmits this packet in the very next slot following packet's generation time).

(iii) The algorithms in [16], [17] and [18] exploit the fact that after a user becomes active, a finite number of time slots is needed to determine the status of the channel, by observing the feedback.

For a more detailed overview of the Random Access Problem and the proposed algorithms, the interested reader is referred to [1], [9], [11] and [19].

I.2 OUTLINE OF THE Ph.D DISSERTATION

The work can be divided in two parts. The first part deals with Random Access Systems where the objective is not only throughput maximization and mean packet delay minimization.

In the second part of the work, a method for delay distribution analysis of window RAAs is presented. In more detail, the Dissertation is organized as follows:

Multiple Access Algorithms are generally designed to allow the successful transmission of all the generated messages with finite delays. There are applications, threshold.

In Chapter II, we propose and analyze a window type RAA for time-constrained applications. The algorithm guarantees that the **total delay** of each successfully transmitted packet does not exceed a prespecified threshold, and it requires continuous feedback sensing. The algorithm can be easily modified to operate under limited feedback sensing. For various thresholds on the transmission delays, and various input rates, we computed the percentage of the successfully transmitted traffic and the expected delay of a successfully transmitted packet. We found that significant improvement of the expected packet delay is attained at the expense of minimal traffic loss.

In Chapter III, we consider the RAA proposed in Chapter II without imposing any delay constraints. We perform throughput, delay, and feedback error sensitivity analyses. When the packet generating process is Poisson, the algorithm attains the same throughput with that attained by the Capetanakis's Dynamic Algorithm [11], and it induces lower mean packet delays for arrival rates above 0.30. The algorithm is very insensitive to feedback errors. It operates in systems where the packet generating process is not Poisson, (e.g where more than one packets can be generated at a given time instant). It is worth noting here, that the algorithms in [9], [10], are blocked when more than one packets can be generated at the same time instant. In the last section of this chapter, we analytically evaluate the steady-state distribution of the distance between two consecutive, successful transmissions, when the proposed algorithm is employed. The latter distribution is important when one studies interconnected systems which employ the algorithm for their internal transmissions. A typical example is the multi-hop problem, where part of the output *(internally successfully transmitted packets) of a random*

In Chapter IV, a method for delay distribution analysis of **Window RAAs** is presented. Window RAAs constitute an important subclass of RAAs; they are distributive and attain high throughputs and low delays by controlling the number of simultaneously transmitting users. The throughput analysis of algorithms in this class is a relatively easy task. The delay analysis, however, presents difficulties, due mainly to the fact that the window size is variable (we only impose a maximum window size), and the complicated state space that the algorithms create. This restriction prohibits the application of results from standard queueing theory in the delay analysis.

We show that the methodology employed in [20], can be extended to provide **bounds on the distribution of the delays.** The quantities of interest are related to the solution of a **denumerable system of linear equations.** Methods for the computation of the constant terms and the coefficients of the unknowns of the system are developed. The methodology is applied to the delay distribution analysis of the Capetanakis's Dynamic algorithm [11], and the Part-and-Try algorithm [9] and [10], both under binary (collision versus noncollision) feedback. It can also be applied directly to other Window RAAs with different feedback. An interesting result of the analysis is that as the arrival rate increases, the **tails** of the delay distribution become longer, but the **median** grows much slower than the expected delay.

In the last Chapter, Chapter V, we consider a two-cluster packet radio network. In each cluster a single forward channel is available for packet transmission. The frequency bands dedicated to each cluster are different. Consequently, a packet transmission over the forward channel in one cluster does not interfere with a packet transmission over the forward channel of the other cluster. Each cluster contains local and marginal users. The

The marginal users are located in the overlapping region of the two clusters. They are capable of transmitting their packets over either one of the forward channels and, receiving feedback information from either one of the feedback channels. Due to the double exposure the marginal users have a choice: they can join either cluster 1 or cluster 2 for the transmission of their packets. In this Chapter, we propose a dynamic protocol according to which a marginal user can select the forward channel over which he will transmit his packet. The protocol requires that each marginal user with a packet for transmission observes both the feedback channels from the packet's generation time. No a priori knowledge of the input traffic rates or the state of the system is required.

CHAPTER II

A CONTINUOUS SENSING WINDOW RANDOM-ACCESS ALGORITHM FOR MESSAGES WITH STRICT DELAY CONSTRAINTS

II.1 INTRODUCTION

We consider the case where a number of geographically distributed stations wish to communicate by transmitting messages over a single channel. The problem then is the design of efficient algorithms for allocating the channel among the users. This problem is referred to as the Multiple-Access problem, and the algorithm employed is called a Multiple-Access Algorithm (MAA). The main performance measures of an algorithm are generally throughput, induced delays, and error sensitivity. In addition, Multiple-Access algorithms are generally designed to allow the successful transmission of all the messages generated in the system with finite delays.

There are applications, however, where a message is considered lost if its delay exceeds a certain threshold. Typical examples of such applications are the following:

a) The transmission of control messages that are required to monitor remote devices within a specified amount of time.

b) Distributed target-tracking, where a user attempts to track a moving target by using local observations and the information received from the other users.

Recently, there has been considerable interest in the design and analysis of algorithms suitable for time-constrained applications. Those algorithms disregard the transmission of a message if its delay exceeds the given threshold. Therefore, a percentage of the generated traffic is lost.

Random Access Algorithms constitute an important subclass of Multiple Access Algorithms, and it is of considerable interest to design such algorithms that will operate efficiently and reliably under strict constraints on the delays of the generated messages. As discussed in [21], the advantages of the Random Access Algorithms for time constrained communications are "guaranteed connectivity, the transfer of temporary overloading into packet delay rather than blocked connections, and the ability to trade message loss for message delay". The major disadvantage is that a Random Access Algorithm induces variable message delay.

Simulation and experimental studies of the performance of the Ethernet Protocol under time constraints can be found in [32], [33]. Window Access protocols for time constrained communication have been considered in [22]. In [22], it was assumed that constraints are imposed only on the time elapsed from the generation time of a message to the time of its first transmission attempt. No restriction was imposed on the additional delays experienced by a message when its first transmission attempt is not successful. Furthermore, an approximate model of the system was developed and analyzed. This model, however, fails to take into account the fact that window sizes are necessarily variable in a real system.

In this chapter, we propose and analyze a window Random-Access algorithm for that the users observe the channel feedback continuously (continuous feedback sensing). We point out that the algorithm can be easily modified to operate under limited feedback sensing, where the users are required to observe the feedback only whenever they have a message to transmit. However, the analysis of the algorithm is much more involved in the latter case.

This chapter is organized as follows: In section II.2, we present the system model. In section II.3, we present the operation of the algorithm. In section II.4, we extend the basic ideas presented in [20] to provide an exact analysis of the time-constrained algorithm. The performance of the algorithm is discussed in section II.5.

II.2 SYSTEM MODEL

The transmission delay of a message in a communication network is defined as the time period between the instant that the message is generated and the instant when its successful transmission has been completed. When a strict upper bound on the transmission delays is imposed, some messages are lost. Important performance criteria in this case are the fraction of the lost message traffic and the expected transmission delays of the transmitted messages.

In this chapter,we consider a random access packet network with upper bounds on the packet delays. In particular, we assume that a common slotted channel is shared by packet transmitting independent users, that the feedback per slot is binary CNC (collision versus noncollision), and that the feedback channel is errorless. We also assume that collisions result in total loss of all the involved packets, a packet transmission may start only at the beginning of some slot, and that there are no propagation delays. Finally, it is assumed that the packet generating process is Poisson. This is a reasonable assumption since we assume a large number of independent bursty users. Furthermore in [28] we prove that for a large class of Random Access algorithms (RAAs), and given a fixed value for the cumulative packet generation rate, as the user population increases the stability of an algorithm in the class is determined by its throughput under the Poisson user model. For the above system model we propose and analyze a continuous feedback sensing random access algorithm. Additionally the method we develop is also applicable to the case where the number of packets generated per slot forms an i.i.d process with general distribution (this may be the case in some synchronized systems), and to the finite-population user model.

II.3 THE ALGORITHM

Let T be the upper bound on the packet transmission delay, and let it be common to all users. Let t be a time instant corresponding to the beginning of some slot, such that, for some $t_1 < t$, all the packet arrivals in $(0, t_1]$ have been either successfully transmitted or rejected by the algorithm, and there is no information regarding the arrival interval $(t_1, t]$, (Fig.II.1). Such a point t is called a "collision resolution point" (CRP). Let time be measured in slot units. Then, in slot t, the arrivals in $(t_2, t_3]$ transmit, where for Δ being an algorithmic constant we have: $t_2 = \max(t-(T-1), t_1)$ and $t_3 = \min(t_2+\Delta, t)$. At the same time, the arrivals in $(t_1, t_2]$ are automatically rejected, because their transmission delay has exceeded T. The arrival interval $(0, t_1]$ above is called "resolved interval," the interval $(t_2, t]$ is called the "lag at t", and the arrival interval $(t_2, t_3]$ is called the "examined interval." If $(t_2, t_3]$ contains zero or one packets, then it is resolved at slot t. If $(t_2, t_3]$ contains, instead, at least two packets, then a collision occurs at t, whose resolution begins at slot t + 1. Until the collision at t is resolved, no packets that have arrived after t_3 are allowed transmission. The time period required for the resolution of the latter collision is called the "collision resolution interval" (CRI). Let $d=t-t_2$ be the length of the lag at t. The length of the collision resolution interval is not allowed to exceed $A=T-d$. Thus, packets in $(t_2, t_3]$, that are not successfully transmitted by time $t+A$, are rejected. This last rule assures that all the successfully transmitted packets have delay less than T. Note that under this rule it is possible that a packet (e.g., packet 2 in Fig. II.1) with delay $T-\Delta$, and not T, be rejected. However, as we shall see in section 5, the optimum value of Δ is close to 2 slots and, therefore, this rule does not represent any severe restriction in practice.

Let x_t denote the feedback corresponding to slot t, where either $x_t = c$, for collision slot, or $x_t = nc$, for noncollision slot. During some CRI, each involved user acts independently, utilizing a counter, whose value at time t is denoted by r_t. When a user transmits a packet for the first time he sets $r_t = 1$. The value of r_t is updated and used as follows:

1. The user transmits each time t that $r_t = 1$. The corresponding packet is successfully transmitted at t if and only if,

$$r_t = 1 \text{ and } x_t = nc$$

2. The transitions in time of the counter value r_t are as follows:

   (a) If $x_{t-1} = nc$ and $r_{t-1} = 2$, then $r_t = 1$.

   (b) If $x_{t-1} = c$ and $r_{t-1} = 2$, then $r_t = 2$.

   (c) If $x_{t-1} = c$ and $r_{t-1} = 1$, then

$$r_t = \begin{cases} 1 & ;\ w.p.\ 0.5 \\ 2 & ;\ w.p.\ 0.5 \end{cases}$$

The algorithmic operation can be depicted by a two cell stack. At each slot t, cell 1 contains the transmitting users (those with $r_t = 1$), and cell 2 contains the withholding users (those with $r_t = 2$). In contrast Capetanakis's algorithm distributes the unsuccessful users across the cells of an infinite cell stack. The algorithm allows every user in the system to determine accurately the time instant t' when a CRI has ended; a CRI ends either when two consecutive noncollision slots occur, or when its length equals T-d where d is the length of the lag at the beginning of the CRI. Then, the algorithm reinitializes with a new examined interval. We note that the algorithmic parameter Δ is subject to optimal selection, for performance enhancement.

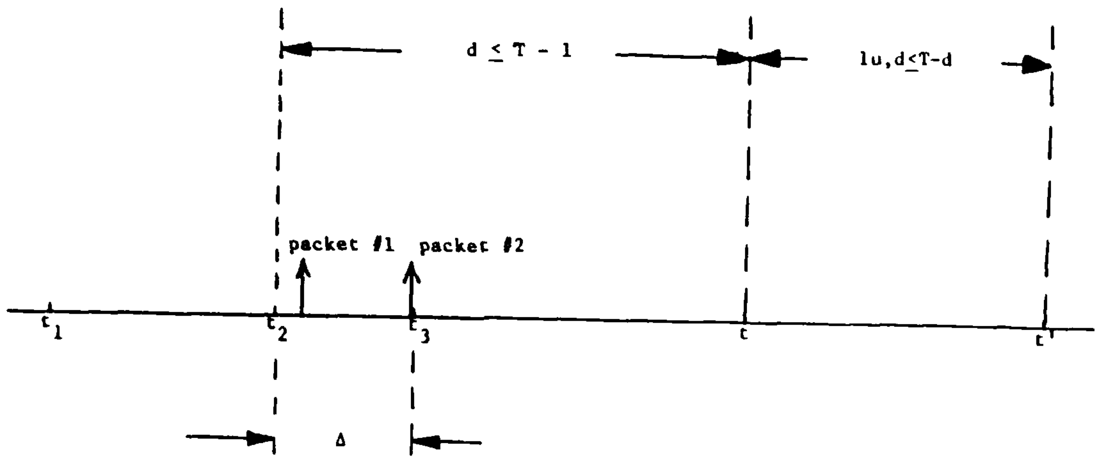


Figure II.1

Illustration of the relationships among certain random variables related to the operation of the algorithm.

II.4 ANALYSIS

Consider the algorithm in section II.3. Let the system start operating at time zero, and let us consider the sequence in time of lags that are induced by the algorithm. Let X_i denote the length of the i-th lag, where $i \geq 1$. Then, the first lag corresponds to the empty slot zero; thus, $X_1 = 1$. In addition, the sequence X_i; $i \geq 1$ is a Markov chain whose state space is at most countable. Let D_n denote the delay experienced by the n-th successfully transmitted packet arrival, as induced by the algorithm; that is, the time between the arrival of the packet and its successful transmission. Let the sequence T_i, $i \geq 1$ be defined as follows: Let $T_1 = 1$, and define T_{i+1} as the first CRP after T_i at which the lag has length one.

Let R_i, $i \geq 1$ and F_i, $i \geq 1$ denote respectively the number of successfully transmitted packets and the number of rejected packets in the time interval $(0, T_i]$. Then, $Q_i = R_{i+1} - R_i$, $i \geq 1$ and $G_i = F_{i+1} - F_i$, $i \geq 1$ denote respectively the number of successfully transmitted and the number of rejected packets in the interval $(T_i\ , T_{i+1}]$. The sequences Q_i, $i \geq 1$ and G_i, $i \geq 1$ are sequences of i.i.d. random variables; thus R_i, $i \geq 1$ and F_i, $i \geq 1$ are renewal processes. In addition, the delay process D_n, $n \geq 1$ induced by the algorithm is regenerative with respect to the process R_i, $i \geq 1$ and the distribution of Q_i is nonperiodic, since $P\ (Q_i = 1) > 0$.

Let us define,

$$Z = E\{Q_1\}\ , \quad W = E\{\sum_{i=1}^{Q_1} D_i\}\ , \quad H = E\{T_2 - T_1\} \tag{II.1}$$

From the regenerative arguments in [20], it follows that the fraction, ρ, of successfully transmitted packets, and the expected steady-state delay, **D**, of the successfully transmitted packet are respectively given by the following expressions:

$$\rho = Z(\lambda H)^{-1} \tag{II.2}$$

$$\mathbf{D} = WZ^{-1} \tag{II.3}$$

where λ is the intensity of the Poisson packet generating process.

Towards the computation of the expected values Z, H, and W, let us define the following quantities (see Figure II.1), where t_2, t_3, and t are as in section II.3:

$n_{u,d}$: Number of packet arrivals in $[t_2 , t_3)$ that are successfully transmitted during the collision resolution process, given that $t_3 - t_2 = u$., and $t - t_2 = d$

$z_{u,d}$: Sum of the delays of the $n_{u,d}$ packets, after time t.

$\psi_{u,d}$: Sum of the delays of the $n_{u,d}$ packets, until the instant t_3 .

$l_{u,d}$: The number of slots needed to examine an interval of length u given that $t - t_2 = d$. Note that $l_{u,d} \le T - d$.

$E\{X \mid u,d\}$: Conditional expectation of the random variable X, given that the length of the initially transmitted interval is u, and $t - t_2 = d$.

h_d : The number of slots needed to reach a CRP with lag equal to one,when starting from a CRP with lag equal to d .

w_d : The cumulative delay experienced by all the packets that were successfully transmitted during the h_d slots.

α_d : The number of packets that are successfully transmitted within the interval

$P(l \mid u,d)$: Given that the interval to be examined has length u and $t-t_2=d$, the probability that the corresponding collision resolution interval has length l.

$$H_d = E\{h_d\}$$

$$W_d = E\{w_d\} \tag{II.4}$$

$$A_d = E\{\alpha_d\}$$

We note that the quantities in (2), (3), and (4) are such that, $Z = A_1$, $H = H_1$, and $W = W_1$. Denoting by x_d either one of the random variables h_d, w_d, α_d, the operations of the algorithm in section II.3 induce the following relationships.

$$x_d = \begin{cases} \theta_d + x_{l_{d,d}} & ; \lfloor T-d \rfloor \geq l_{d,d} > 1 \\ \theta_d & ; l_{d,d}=1 \end{cases} \quad ; 1 \leq d \leq \Delta \tag{II.5}$$

$$x_d = \theta_d + x_{d-\Delta+l_{\Delta,d}} ; T-1 \geq d > \Delta$$

; where $\lfloor a \rfloor$ denotes integer part of a, and

$$\theta_d = \begin{cases} l_{\min(\Delta,d),d} & ; \text{ for the r.v. } h_d \\ \psi_{\min(\Delta,d),d} + z_{\min(\Delta,d),d} + \max(d-\Delta,0) n_{\min(\Delta,d),d} & ; \text{ for the r.v. } w_d \\ n_{\min(\Delta,d),d} & ; \text{ for the r.v. } \alpha_d \end{cases} \tag{II.6}$$

Taking expectations in (5), and denoting $X_d=E\{x_d\}$, we obtain:

$$X_d = E\{\theta_d\} + \sum_{m=2}^{\lfloor T-d \rfloor} X_m \, P(m \mid d,d) \quad ; \quad 1 < d \leq \Delta$$

(II.7)

$$X_d = E\{\theta_d\} + \sum_{m=1}^{\lfloor T-d \rfloor} X_{d-\Delta+m} \, P(m \mid \Delta,d) \quad ; \quad \Delta < d \leq T-1$$

; where d takes at most denumerable values in $[1,T-1]$. In (7), the index m takes values up to $\lfloor T-d \rfloor$ because $l_{u,d} \leq T-d$.

Note that the system in (7) is a finite system. Moreover, if Δ is an integer, then d can only take integer values in [1, T-1]. In either case, the system is of the following form:

$$\bar{x} = \bar{b} + \Theta\bar{x} \quad or \quad (I-\Theta)\bar{x} = \bar{b}$$

where Θ is a strictly substochastic irreducible matrix. It follows [34] that if $\bar{b}\geq 0$, then the system in (7) has a unique nonnegative solution. In Appendix A, we include the recursions pertinent to the computation of $E\{\theta_d\}$ and $P(l \mid u,d)$. The latter recursions, in conjunction with the system in (7), provide the fraction of the successfully transmitted traffic and the expected delay of the successfully transmitted packets, for given T, Δ and λ . Given the latter parameters, the quantities in (2) and (3) are functions of them; we will denote them by, $\rho_T(\Delta,\lambda)$ and $D_T(\Delta,\lambda)$, respectively.

II.5 PERFORMANCE EVALUATION

The value of Δ is a basic design parameter for window Random Access Algorithms. If no delay constraints are imposed, Δ is chosen so that the stability region of the algorithm is maximized. For systems with strict delay constraints, an upper bound T on the transmission delay of each successfully transmitted packet is given. Since a fraction of the generated packets is then rejected, an important quality measure in this case is a lower bound, e_1, on the fraction of the successfully transmitted traffic. Given T and e_1 as above, the value of Δ is determined from the following optimization problem : Determine Δ, so that the input traffic rate is maximized while the fraction of the successfully transmitted traffic remains greater than e_1. That is, the quantity λ^*_{T,e_1} below is sought.

$$\lambda^*_{T,e_1} = \sup\ \{\lambda : \rho_T(\Delta,\lambda) \geq e_1\}$$

Due to the complexity of the expression that provides the system parameter $\rho_T(\Delta, \lambda)$, it is hard to solve the optimization problem in (8) in an analytical fashion. For fixed Δ, however, it is simple to compute the quantity $\lambda^*_{T,\, e_1}(\Delta)$ defined below.

$$\lambda^*_{T,\, e_1}(\Delta) \;=\; sup\;\; (\lambda : \rho_T(\Delta,\lambda) \geq e_1) \tag{II.9}$$

In table II.1 we exhibit the $\lambda^*_{T,\, e_1}(\Delta)$ values, for $\Delta = 1.5$, 2, 2.5, 3, and various values of the parameters T and e_1. From the above table, we observe that the algorithm is relatively insensitive to the selection of the parameter Δ, for the $\Delta \in [2, 3]$.

If in addition to e_1, it is also desirable to impose an upper bound e_2 on the expected delay of the successfully transmitted packet, then the value of Δ is determined from the following optimization problem: Select Δ so that the quantity $\lambda^*_{T,\, e_1, e_2}$, defined below, is attained.

$$\lambda^*_{T,\, e_1,\, e_2} = \sup_{0 \leq \Delta \leq T-1} (\lambda : \rho_T(\Delta,\, \lambda) \geq e_1 \;,\; \mathbf{D}_T(\Delta,\, \lambda) \leq e_2) \tag{II.10}$$

The analytic determination of $\lambda^*_{T,\, e_1,\, e_2}$, is again not feasible. We thus compute instead the rates $\lambda^*_{T,\, e_1,\, e_2}(\Delta)$ defined below.

$$\lambda^*_{T,\, e_1,\, e_2}(\Delta) = \;sup\;\; (\lambda : \rho_T\,(\Delta,\lambda) \geq e_1,\; \mathbf{D}_T(\Delta,\lambda) \leq e_2) \tag{II.11}$$

In table II.2 we exhibit $\lambda^*_{T,\, e_1,\, e_2}(\Delta)$ values, for $\Delta = 1.5$, 2, 2.5, and 3, and for various values of the parameters T, e_1, and e_2. For $\Delta \in [2,3]$, we observe again insensitivity of the algorithm to the selection of the parameter Δ.

From table II.1 we observe that for fixed e_1 and Δ, $\lambda^*_{T,\, e_1}(\Delta)$ is an increasing function of T. In contrast, from table II.2 we observe that for fixed e_1, e_2, and Δ, there exists some $T_{max}(e_1,\, e_2,\, \Delta)$, such that $\lambda^*_{T, e_1, e_2}(\Delta)$ increases for $1 \leq T \leq T_{max}(e_1,\, e_2,\, \Delta)$ and decreases for $T \geq T_{max}(e_1,\, e_2,\, \Delta)$. The latter observation can be explained as follows:

cessfully transmitted traffic. For large T values, on the other hand, the determining factor is the constraint on the expected delay of the successfully transmitted packet.

It is interesting to compare the performance of the algorithm under strict delay constraints, to its performance when no such constraints are imposed. In the latter case, for input traffic rates within the stability region of the algorithm, the fraction ρ of the successfully transmitted traffic equals one, and the throughput represents then the maximum maintainable rate of the input traffic. In Appendix A, we compute relevant useful bounds. Using those bounds and the methodology in [20], we compute the throughput μ^*, the optimal parameter Δ^* that attains μ^*, and the expected delays $\mathbf{D}_\infty(\Delta^*, \lambda)$ for $\lambda \varepsilon (0, \mu^*)$. We found:

$$\mu^* = \mu^*(\Delta^*) = 0.429 \quad , \Delta^* = 2.33 \tag{II.12}$$

Figure II.2, illustrates expected delays for various values of T and for Δ=2.. In the same Figure we illustrate the delays induced when no delay constraints are imposed. In Figures II.3 and II.4 we illustrate the fraction of the successfully transmitted traffic and the successfully transmitted traffic respectively, for various values of T and for Δ=2. We observe that for input traffic rates above 0.30, imposing strict constraints on the delays of the successfully transmitted traffic results in significant improvement on the expected delay of the successfully transmitted packet; the price paid then is loss of some of the generated traffic. For example, for $\lambda = 0.4$, $\Delta = 2$, and T=20, the algorithm attains $\rho_{20}(2, 0.4) \approx 0.95$, and $\mathbf{D}_{20}(2,0.4) \approx 7.6$, versus $\mathbf{D}_\infty(\Delta^*, 0.4) \approx 19$ in the absence of an upper bound T, while for T = 20 the delay of each successfully transmitted packet is simultaneously not exceeding 20 slots. The tradeoff between acceptable delays and acceptable loss in traffic is determined by each particular application.

feedback [11]. Compared to Capatanakis's algorithm, the proposed algorithm has the property that both the unconstrained and the constrained version can be easily modified to operate under limited feedback sensing. This is because during a collision resolution process, no more than two consecutive noncollision slots can be encountered. This property can be used to provide a limited feedback sensing version of the algorithm presented in this chapter (see also [17], [18]). It is also important to note that due to the very simple rules of the collision resolution process, the algorithm is less sensitive to feedback channel errors than the Capetanakis's dynamic algorithm. A complete analysis of the unconstrained version of the algorithm is presented in Chapter III. The following argument can be used, however, to justify our claim: Let δ be the probability that an empty slot is erroneously interpreted as a collision slot, and assume that this is the only form of error on the feedback channel. Let $B_k^{(\delta)}$, $E_k^{(\delta)}$, $S_k^{(\delta)}$ be respectively the average number of collision, empty and successful slots during a multiplicity k collision resolution process. Let $L_k^{(\delta)}$ be the average number of slots needed for a multiplicity k collision to be resolved. It is easy to see that

$$B_k^{(\delta)} = B_k^{(0)}, \quad S_k^{(\delta)} = S_k^{(0)}, \quad E_k^{(\delta)} = E_k^{(0)} \frac{1}{(1-\delta)} \quad ; 0 \le \delta < 1, \ k \ge 2 \tag{II.13}$$

It follows that

$$L_k^{(\delta)} = B_k^{(0)} + S_k^{(0)} + E_k^{(0)} \frac{1}{(1-\delta)} = L_k^{(0)} + E_k^{(0)} \frac{\delta}{(1-\delta)} \le L_k^{(0)} \frac{1}{(1-\delta)} \quad ; 0 \le \delta < 1, \ k \ge 2 \tag{II.14}$$

It can also be seen that

$$L_0^{(\delta)} = \frac{1}{(1-\delta)^2} \quad ; 0 \le \delta < 1 \tag{II.15}$$

If $f_\delta(x) = \sum_{k=0}^{\infty} L_k^{(\delta)} e^{x} x^k / k!$, the throughput λ_δ^* of the algorithm is

$$\lambda_\delta^* \geq \lambda_0^*(1-\delta)^2 \tag{II.16}$$

Therefore, the throughput of the algorithm remains positive for $0 \leq \delta < 1$. It is well known [11], however, that the throughput of Capetanakis's algorithm reduces to zero for $\delta \geq 0.5$.

Gallager's algorithm [11], can also be modified to operate under delay constraints. The method developed in this chapter can be applied to the analysis of the latter algorithm, but the computations become much more involved and greater care is needed since the developed systems of linear equations become infinite. For systems where the Poisson user model is valid, it is expected that the modification of Gallager's algorithm will provide improved performance. One advantage of the algorithm presented here, is that it can operate in systems where the Poisson user model is not valid (e.g. when more than one packets are generated at a specific instance due to synchronous transmission), in contrast to the algorithm in [11], and that analytical evaluation of its performance and optimal choice of the design parameter Δ is then feasible.

II.6 CONCLUSIONS

We considered the case where strict constraints on the delays of successfully transmitted packets are imposed. For this case, we presented, analyzed, and evaluated a simple random access algorithm. The algorithm requires continuous feedback sensing. We analyzed the algorithm under the Poisson user model assumption. The analysis of the algorithm is based on its regenerative characteristics. For various bounds on the transmission delays and for various input rates, we computed the percentage of the successfully transmitted traffic and the expected delay of the successfully transmitted the successfully transmitted packet is attained. The possibility of employing other window Random-Access algorithms, and their advantages and disadvantages as compared to the algorithm presented here is also discussed.

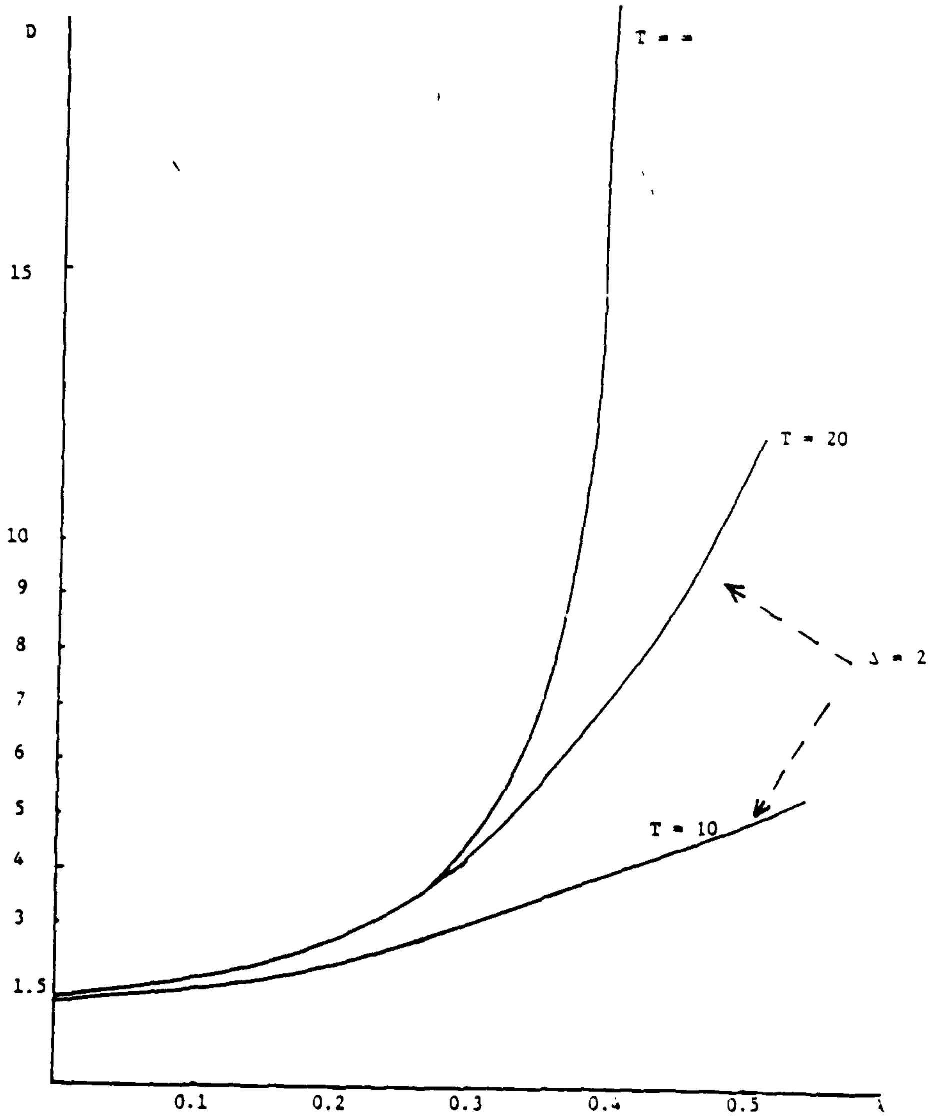


Figure II.2

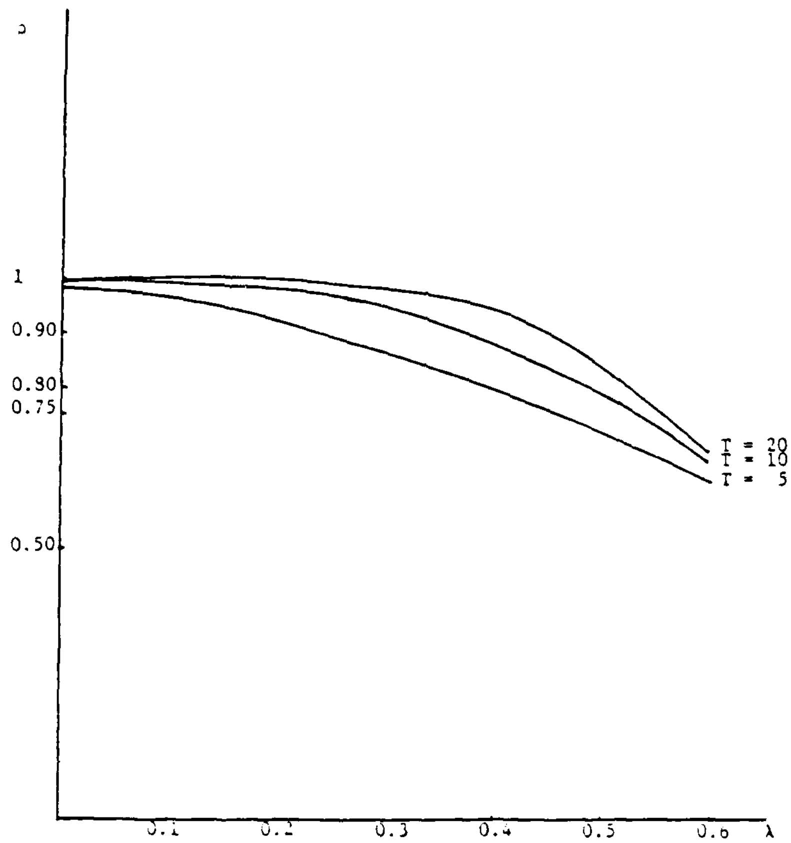


Figure II.3

Fraction of the transmitted traffic versus input rate,

$\Delta = 2$.

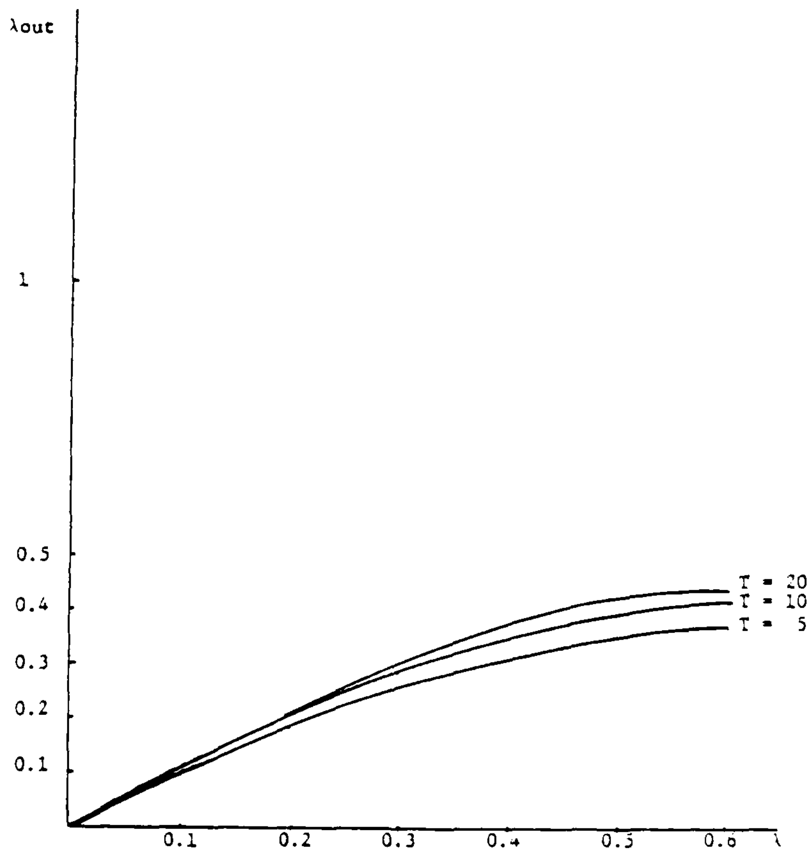


Figure II.4

Output versus input rate, $\Delta = 2$, $\lambda_{out} = \rho * \lambda$.

		$\lambda_{T,e_1}(\Delta)$			
T	e_1	Δ=1.5	Δ=2	Δ=2.5	Δ=3
5	.99	0.038	0.038	0.038	0.037
	.95	0.142	0.144	0.140	0.140
	.90	0.243	0.257	0.240	0.238
10	.99	0.150	0.160	0.150	0.143
	.95	0.290	0.300	0.285	0.280
	.90	0.372	0.380	0.364	0.360
15	.99	0.242	0.247	0.240	0.230
	.95	0.348	0.361	0.352	0.343
	.90	0.408	0.425	0.420	0.405
20	.99	0.289	0.306	0.294	0.284
	.95	0.373	0.392	0.383	0.374
	.90	0.423	0.445	0.435	0.425
25	.99	0.314	0.333	0.319	0.309
	.95	0.406	0.426	0.416	0.407
	.90	0.439	0.462	0.452	0.442
30	.99	0.330	0.350	0.335	0.325
	.95	0.427	0.448	0.437	0.428
	.90	0.450	0.474	0.464	0.452

Table II.1

Maximum throughput as a function of T and e_1

			$\lambda_{T,e_1,e_2}(\Delta)$			
T	e_1	e_2	Δ=1.5	Δ=2	Δ=2.5	Δ=3
10	.99	3	0.150	0.160	0.150	0.143
		5	0.150	0.160	0.150	0.143
		7	0.150	0.160	0.150	0.143
	.95	3	0.260	0.297	0.285	0.280
		5	0.290	0.300	0.285	0.280
		7	0.290	0.300	0.285	0.280
15	.99	3	0.220	0.240	0.240	0.230
		5	0.242	0.247	0.240	0.230
		7	0.242	0.247	0.240	0.230
	.95	3	0.220	0.240	0.320	0.260
		5	0.322	0.361	0.352	0.343
		7	0.348	0.361	0.352	0.343
20	.99	3	0.209	0.230	0.240	0.240
		5	0.289	0.306	0.319	0.284
		7	0.289	0.306	0.319	0.284
	.95	3	0.209	0.230	0.240	0.240
		5	0.289	0.322	0.340	0.340
		7	0.347	0.383	0.384	0.374
30	.99	3	0.181	0.198	0.198	0.196
		5	0.313	0.329	0.356	0.317
		7	0.313	0.329	0.356	0.322
	.95	3	0.181	0.198	0.198	0.198
		5	0.313	0.345	0.380	0.380
		7	0.376	0.407	0.408	0.393

Table II.2

Maximum throughput as a function of T, e_1, and e_2

CHAPTER III

A WINDOW RANDOM ACCESS ALGORITHM FOR MULTI-USER PACKET RADIO SYSTEMS

III.1 INTRODUCTION

In data networks with bursty users, the most appropriate multiple-access techniques are those belonging to the class of Random Access Algorithms (RAAs). This is due to the fact that RAAs are distributed, and they induce low delays for low input rates.

The important performance characteristics of a RAA are: throughput, induced delays, and sensitivity to feedback channel errors. Insensitivity to feedback errors is important when operating under noisy conditions. Different sources of noise include: (i) interference between closely located cells, which use the same frequency band (multi hop networks), or, (ii) multi-path fading. The effects of noise can be modeled as erroneously observed feedback, see [35].

In this chapter, we consider the RAA presented in chapter II without imposing any delay constraints; i.e., $T = \infty$. We then perform throughput, delay, and feedback error sensitivity analyses. We also evaluate the output traffic interdeparture distribution induced by the algorithm.

Under the Poisson user model assumption, the algorithm attains the same throughput with that attained by the Capetanakis's dynamic algorithm, [11], while it induces lower delays for arrival rates above 0.30. It also exhibits better resistance to feedback errors. The evaluation of the output traffic interdeparture distribution is important, in the case of interconnected networks each of which employs the RAA for internal transmissions. The evaluation of the latter distribution when either the Capetanakis's, [11], or, the Gallager's, [11], algorithms are used remains an open problem.

The organization of this chapter is as follows: In section III.2, we present throughput and delay analysis. In section III.3, we present feedback error sensitivity analysis, and discuss the operation of the algorithm under limited feedback sensing. In section III.4, the output traffic interdeparture distribution, induced by the algorithm, is evaluated. In section III.5, some conclusions are drawn.

III.2 THROUGHPUT AND DELAY ANALYSES

In this section, we present throughput and delay analyses of the algorithm. We assume continuous feedback sensing and the Poisson user model.

Consider the algorithm in Section II.3. Since we do not impose any delay constraints, we have that $t_1 = t_2$. Let the system start operating at time zero. Let t_i ; $i \geq 1$ be the sequence of successive CRPs, and X_i be the lag at t_i. The sequence X_i ; $i \geq 1$ is a Markov Chain with state space **F**, **F** is an at most denumerable subset of the interval [1, ∞). It can be seen that any state can be reached from any other; therefore X_i ; $i \geq 1$ is an irreducible Markov Chain. Since $P(X_{i+1} = 1 \mid X_i = 1) > 0$, we conclude that X_i; $i \geq 1$ is also aperiodic. Therefore Pake's Lemma [30] applies, and gives that the following condition is sufficient for the ergodicity of the Markov Chain, (stability of the system):

$$E(l \mid \Delta, d) < \Delta \qquad \text{(III.1)}$$

where $E(l \mid \Delta, d)$ denotes the expected length of a CRI, given that it starts with an examined interval of length Δ and with a lag of length d.

Since the Markov Chain is uniformly downward bounded (there exists a constant m such that the transition probabilities p_{kj} satisfy $p_{kj} = 0$ for $j < k - m$. Here $m = \Delta - 1$), Kaplan's Theorem [31] applies and gives that:

If

$$E(l \mid \Delta, d) > \Delta$$

then the Markov Chain is not ergodic and the system is unstable.

Let L_k denote the expected length of a CRI given that it starts with a collision of multiplicity k. We can then write:

$$E(l \mid \Delta, d) = \sum_{k=0}^{\infty} E(l \mid \Delta, d, k)\, e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} = \sum_{k=0}^{\infty} L_k e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \tag{III.2}$$

since

$$E(l \mid \Delta, d, k) = L_k \tag{III.2a}$$

depends only on k.

In Appendix B we show that:

(i) L_k ; $k \geq 0$ can be computed recursively, and

(ii) L_k are quadratically upper bounded, $L_k \leq L_k^u = \alpha k^2 + \beta k + \gamma$; $k \geq 2$.

Expression (1) together with (i) and (ii) are used in the computation of the algorithmic throughput. The details of the analysis are presented in Appendix B.

We define the delay D_n, experienced by the n-th packet as the time difference between its arrival and the end of its successful transmission. We are interested in evaluating the first moment of the steady state delay process, when it exists. Let $T_1 = 1$, $X_1 = 1$, and define T_{i+1} as the first CRP after T_i at which the lag has length one. From the description of the algorithm it can be seen that the induced delay process probabilistically restarts itself at the beginning of each slot T_i , $i = 1, 2, \ldots$ The interval $[T_i, T_{i+1})$ will be referred to as the i-th session. Note that the sessions have lengths that are i.i.d random variables.

Let R_i ; $i = 1, 2, \ldots$, denote the number of packets successfully transmitted in the interval $(0, T_i]$; (note that R_i also represents the number of packets arrived during the interval $[0, T_i - 1)$, since T_i is a CRP at which the lag has length one). Then, $Q_i = R_{i+1} - R_i$; $i \geq 1$, is the number of packets successfully transmitted in the interval $(T_i, T_{i+1}]$, these are the packets that arrived during the interval $[T_i - 1, T_{i+1} - 1)$. The sequence R_i ; $i \geq 1$, is a renewal process since Q_i ; $i \geq 1$, is a sequence of nonnegative i.i.d random variables. Furthermore, the delay process D_n ; $n \geq 1$, is regenerative with respect to the renewal process R_i ; $i \geq 1$, with regeneration cycle Q_1. From the regenerative theorem [20], we conclude that if $Q = E(Q_1) < \infty$ and $W = E\{\sum_{i=1}^{Q_1} D_i\} < \infty$, then there exists a real number **D** such that,

$$\mathbf{D} = \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} D_i = \lim_{n\to\infty} n^{-1} E(\sum_{i=1}^{n} D_i) = \frac{W}{Q} \tag{III.3}$$

The first and second equalities in (3) above, are $w.p.\ 1$. In addition, since $P(Q_1 = 1) > 0$, the distribution of Q_1 is aperiodic and there exists a random variable D_∞ such that the sequence D_n ; $n \geq 1$ converges in distribution to D_∞. D_∞ represents the steady state delay induced by the algorithm and its mean satisfies the equality

$$E(D_\infty) = \frac{W}{Q} \tag{III.4}$$

The quantity **D** will be referred to as the mean packet delay. From (4) we observe that the mean packet delay can be determined by computing the quantities of the right hand side of the equality. In Appendix B we develop two systems of linear equations whose solution may be used to compute the mean cycle length Q and the mean cumulative delay W.

In Table III.1, we include the computed upper and lower bounds, $\mathbf{D}^u$ and $\mathbf{D}^l$ respecproposed and the Capetanakis's dynamic algorithms.

	Proposed algorithm		Capetanakis dynamic algorithm	
λ	D^l	D^u	D^l	D^u
0.02	1.562	1.563	1.563	1.564
0.06	1.708	1.716	1.713	1.719
0.10	1.888	1.917	1.903	1.921
0.16	2.257	2.363	2.308	2.362
0.20	2.607	2.812	2.712	2.809
0.24	3.103	3.467	3.308	3.476
0.30	4.412	5.197	4.976	5.365
0.32	5.162	6.170	5.973	6.501
0.36	7.941	9.665	9.798	10.883
0.38	11.008	13.398	14.121	15.855
0.40	18.262	22.024	24.427	27.736
0.42	57.354	67.665	78.530	90.212

Table III.1

Upper and Lower Bounds on Steady-State Expected Delays

Regarding the throughput λ^* and the optimal window size Δ^*, the following results have been found.

Proposed Algorithm:	$\lambda^* = 0.4295$	$\Delta^* = 2.33$
Capetanakis' Dynamic Algorithm:	$\lambda^* = 0.4295$	$\Delta^* = 2.677$

Table III.2

Throughputs and Optimal Window Sizes

From Table III.2, we observe that, under the Poisson user model, the algorithm in algorithm, but uses a smaller window size. From Table III.1, we observe that the two algorithms induce practically identical delays for Poisson rates in (0, 0.30), while for Poisson rates in (0.30, 0.42], the proposed algorithm induces lower delays.

Remarks It may seem surprising that the algorithm in this chapter attains the same throughput with that of the Capetanakis's dynamic algorithm, since the expected lengths L_k in (2a) are bounded by a quadratic function of k, while the corresponding lengths for the Capetanakis's algorithm, L_k' , are bounded by linear a function of k. An intuitive explanation is as follows: The proposed algorithm is on the average faster than the Capetanakis's algorithm, when the collision multiplicity is 2, since $L_2 = 4.5$ and $L_2' = 5$. For collision multiplicities higher than 2, the proposed algorithm is on the average slower than Capetanakis's algorithm since $L_k > L_k'$, $k \geq 3$. Furthermore, the probability of a higher than two collision multiplicity for the proposed algorithm, is smaller as compared to the corresponding probability for the Capetanakis's algorithm (this is because the proposed algorithm uses a smaller optimal window). The advantage of the proposed algorithm for collision multiplicities equal to 2, is balanced by the advantage of the Capetanakis's algorithm for collision multiplicities greater than 2.

III.3 PERFORMANCE UNDER FEEDBACK ERRORS AND OPERATION UNDER LIMITED FEEDBACK SENSING

In this section, we study two important characteristics of the algorithm. Its performance under feedback channel errors, and its operation and under limited feedback sensing.

III.3.1 Performance Under Feedback Errors

Let us assume that due to noisy conditions, the following types of feedback errors may occur: With probability ε an empty slot may be perceived by the users as a collision slot. Also, with probability δ a slot occupied by a single transmission may be perceived by the users as a collision slot. We assume that a collision slot is always recognized correctly by the users. We consider the case where the probabilities ε and δ, are known a priori. Then given ε and δ, the window size Δ is optimized for throughput maximization. We performed throughput analysis, (the details are included in Appendix B), for both the proposed algorithm and the Capetanakis's dynamic algorithm, [11]. We exhibit the results in Table III.3. From Table III.3, we conclude that the proposed algorithm is very insensitive to feedback errors. Even for the practically extreme case $\varepsilon = \delta = 0.1$, the throughput is almost 90% of its value in the error free case. We also conclude that the proposed algorithm allows operation (positive throughput) as long as $\varepsilon < 1$ and $\delta < 1$, while if $\varepsilon \geq 0.5$ the throughput for the Capetanakis's algorithm is zero. We notice for example that for $\varepsilon = 0.5$ and $\delta = 0$, the proposed algorithm attains throughput equal to $\lambda^* = 0.325$.

ε	δ	λ^* proposed alg.	λ^* Cap.
0.00	0.00	0.4295	0.4295
0.00	0.01	0.4248	0.4258
0.00	0.10	0.3873	0.3920
0.00	0.20	0.3463	0.3535
0.00	0.40	0.2655	0.2731
0.00	0.50	0.2251	0.2310
0.01	0.00	0.4272	0.4272
0.10	0.00	0.4117	0.4043
0.20	0.00	0.3930	0.3706
0.40	0.00	0.3503	0.2329
0.45	0.00	0.3382	0.1524
0.50	0.00	0.3250	0.0000
0.60	0.00	0.3050	0.0000
0.70	0.00	0.2750	0.0000
0.80	0.00	0.2280	0.0000
0.90	0.00	0.1700	0.0000
0.10	0.10	0.3706	0.3672
0.20	0.20	0.3139	0.2972
0.30	0.30	0.2589	0.2166
0.40	0.40	0.2064	0.1205
0.30	0.50	0.1885	0.1511
0.30	0.70	0.1183	0.[illegible]86
0.90	0.90	0.0105	0.0000

<u>Table III.3</u>

Throughputs as a function of ε and δ. Window sizes optimized for every pair (ε , δ)

In some systems, the probabilities ε and δ may not be known a priori. In this case, we designed the algorithm subject to the assumption of error free feedback. The corresponding results are shown in Table III.4.

ε	δ	λ^* prop. alg.	λ^* Cap.
0.00	0.00	0.429	0.429
0.00	0.01	0.424	0.410
0.00	0.10	0.386	0.391
0.00	0.40	0.255	0.262
0.01	0.00	0.427	0.427
0.10	0.00	0.408	0.401
0.20	0.00	0.377	0.355
0.3	0.00	0.320	0.248
0.3305	0.00	0.289	0.000
0.37	0.00	0.214	0.000
0.3787	0.00	0.000	0.000
0.10	0.10	0.363	0.361
0.10	0.20	0.318	0.318
0.10	0.30	0.273	0.272
0.20	0.10	0.328	0.311
0.20	0.20	0.279	0.266
0.20	0.30	0.232	0.221
0.30	0.10	0.261	0.192
0.30	0.20	0.205	0.138
0.30	0.30	0.153	0.088

Table III.4

Throughputs as a function of ε and δ. Window sizes unchanged for every pair (ε , δ)

From Table III.4, we observe that the proposed algorithm is better than the Capetanakis's algorithm. For example the maximum ε value for which the proposed algorithm is stable is 0.378, while the Capetanakis's algorithm becomes unstable for $\varepsilon \geq 0.33$.

III.3.2 Operations Under Limited Feedback Sensing

Under limited sensing, it is required that each user monitors the channel feedback from the time he generates a packet, until the time his packet is successfully transmitted.

our objective is to prevent transmission of new arrivals until the current collision resolution process has been completed. The latter can be accomplished, provided that each user with a new arrival can decide, in a finite number of slots, whether a collision resolution process is in progress. A user with a new arrival, who observes a C slot, decides to wait because he deduces that collision resolution is in progress. Two consecutive NC slots correspond to either two consecutive CRIs of length one, or, to the end of a CRI which started with a collision slot. Therefore, if two consecutive NC slots are observed, the user decides that the system is empty. Summarizing, under limited feedback sensing the algorithm can be modified as follows:

The window size remains the same as in the case of continuous feedback sensing. The window slides through the unexamined interval from the current time to the past (since a user with a new arrival has no information regarding the events that happened prior to the arrival time). Its edge is maintained one slot before the current time, (see also Figure III.1). Within a CRI, the algorithm operates exactly as under continuous feedback sensing.

For very light input rate a user with a new arrival will observe an empty slot (arrival slot) with high probability. He then waits for the next slot (observation slot) which will be also empty with high probability. The observation slot being empty, the user transmits in the subsequent slot (transmission slot). Based on the light input rate condition, we silently assume that no collision occurs over the transmission slot. Because Poisson arrivals over a time interval are uniformly distributed, we may infer that the new packet will show up in the middle of the arrival slot; i.e., delay over the arrival slot equals 0.5. Adding up the delays over the arrival, observation, and transmission slots, we

As the input rate increases, a user with a new arrival observes a collision slot with high probability, and waits for its resolution (occurrence of two consecutive NC feedbacks). Therefore, under heavy traffic conditions, a new user can determine the status of the channel immediately upon arrival (without the waste of any observation slots). Consequently, the total delay equals that under continuous feedback sensing. The throughput of the algorithm remains identical to that under continuous feedback sensing (since the window size, and the collision resolution algorithm are the same in both cases).

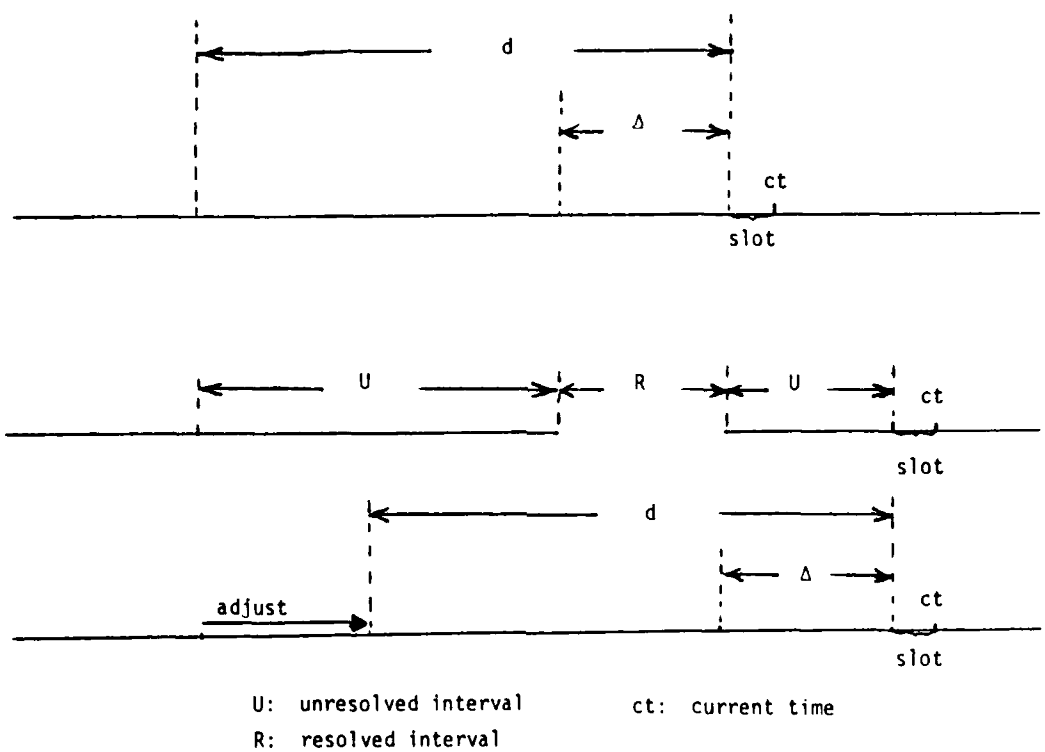


Figure III.1

Window selection in the limited sensing case.

III.4 THE OUTPUT TRAFFIC INTERDEPARTURE DISTRIBUTION

In this section, we analytically evaluate the steady-state distribution of the distance between two consecutive, successful transmissions, when the proposed algorithm is employed. The algorithm generates an output traffic process with memory. Consequently, our computations correspond to the first order distribution from this process. The first order distribution together with a memoryless assumption, may be used as an approximation of the actual output traffic process, when one studies interconnected systems which employ the algorithm for their internal transmissions. Typical example is the multi-hop problem, where part of the output (internally successfully transmitted packets) of a random access system has to be transmitted through another random access system.

The methodology utilizes the regenerative character of the output traffic process. We define the sequence $\{P_i\}_{i\geq 1}$ as follows: Each P_i is a collision resolution point, (CRP), which follows a slot containing a successful transmission and at which the lag equals one. P_1 is the first after zero such CRP, and for every $i\geq 1$, P_{i+1} is the first after P_i such CRP. Let S_i, $i\geq 1$ denote the number of successful transmissions in $(0, P_i]$, and let d_n denote the distance between the (n-1)-th and the n-th successful transmission. Then, S_i, $i\geq 1$ is a renewal process, and the process d_n, $n\geq 1$ is regenerative with respect to it. Let us define, $C_i = S_{i+1} - S_i$, $i\geq 1$. Then C_i denotes the number of successful transmissions in the interval $(P_i, P_{i+1}]$, where this interval will be called the <u>i-th cycle.</u> Let us define,

$$I_n(s) = \begin{cases} 1 & \text{, if } d_n = s \\ 0 & \text{, } otherwise \end{cases} \tag{III.5}$$

$$H = E\{P_{i+1} - P_i\} \tag{III.6}$$

From the regenerative theorem [20], we conclude that if $C = E\{C_1\} < \infty$, then,

$$\lim_{N \to \infty} N^{-1} \sum_{n=1}^{N} I_n(s) = \lim_{N \to \infty} N^{-1} E\{\sum_{n=1}^{N} I_n(s)\} = C^{-1} E\{\sum_{n=1}^{C_1} I_n(s)\} \qquad \text{(III.7)}$$

;where the first equality in (7) is *w.p.* 1. We also have that,

$$C = \lambda H \qquad \text{(III.8)}$$

; where λ denotes the intensity of the Poisson input traffic. In addition, since $P(C_1 = 1) > 0$, the distribution of C_1 is aperiodic and there exists a random variable d_∞, such that the sequence d_n, n=1,2,... converges in distribution to d_∞. Then, d_∞ represents the steady state interdeparture distance induced by the algorithm, and its distribution satisfies the equality,

$$P(d_\infty = s) = C^{-1} E\{\sum_{n=1}^{C_1} I_n(s)\} \qquad \text{(III.9)}$$

The finiteness and the computation of the quantities C and $E\{\sum_{n=1}^{C_1} I_n(s)\}$ in (9) are related to the existence and computation of appropriate solutions to infinite-dimensionality linear systems. The details of the analysis are presented in Appendix B. In Table III.5, we include the computed upper and lower bounds on the probability $P(d_\infty = s)$. In Figure III.2, we plot the lower bounds against s, for various Poisson input rates λ.

	λ =0.1		λ =0.4	
s	P^l_s	P^u_s	P^l_s	P^u_s
1	0.1420	0.1427	0.4702	0.4728
2	0.0816	0.0832	0.2000	0.2048
3	0.0704	0.0739	0.0998	0.10806
4	0.0641	0.0696	0.0612	0.0787
5	0.0537	0.0603	0.0401	0.0603
6	0.0502	0.0591	0.0280	0.0397
7	0.0420	0.0503	0.0196	0.0264
8	0.0364	0.0452	0.0099	0.0173
9	0.0332	0.0431	0.0057	0.0094
10	0.0265	0.0393	0.0013	0.0062

Table III.5

Upper and lower bounds on the interdeparture distribution for Poisson rates

λ =0.1 and λ =0.4

From Table III.5 and Figure III.2, we conclude the following:

(1) For small λ values ($\lambda \leq 0.1$), the interdeparture distribution is close to the Bernoulli distribution with parameter $p = \lambda e^{-\lambda}$. In particular, denoting by $P_s \triangleq P(d_\infty = s)$, we have $P_s \approx p\ (1-p)^{s-1}$, for $s \geq 2$. The probability P_1 however, is larger than the Bernoulli parameter p. An intuitive explanation of the latter is as follows: For small λ values, single arrivals in two consecutive slots occur with probability $p^2 = (\lambda e^{-\lambda})^2 \approx \lambda^2$. The probability of a collision slot is approximately equal to $2^{-1}\ \lambda^2\ e^{-\lambda}\ \approx\ 2^{-1}\ \lambda^2$. Therefore, under light traffic single arrivals in two consecutive slots contribute two thirds of P_1, while the remaining one third is due to consecutive departures at the end of a collision resolution interval.

(2) As the rate of the Poisson input traffic increases, the interdeparture distribution induced by the algorithm deviates further from the Bernoulli distribution. In that case the mass of the distribution accumulates at relatively small s values (see Table III.5, for

$\lambda = 0.4$, we have $P_1 = 0.471$ and $\sum_{s=1}^{10} P_s \approx 1$.

Remark Our results show that it is generally wrong to conjecture Bernoulli interdeparture distribution.

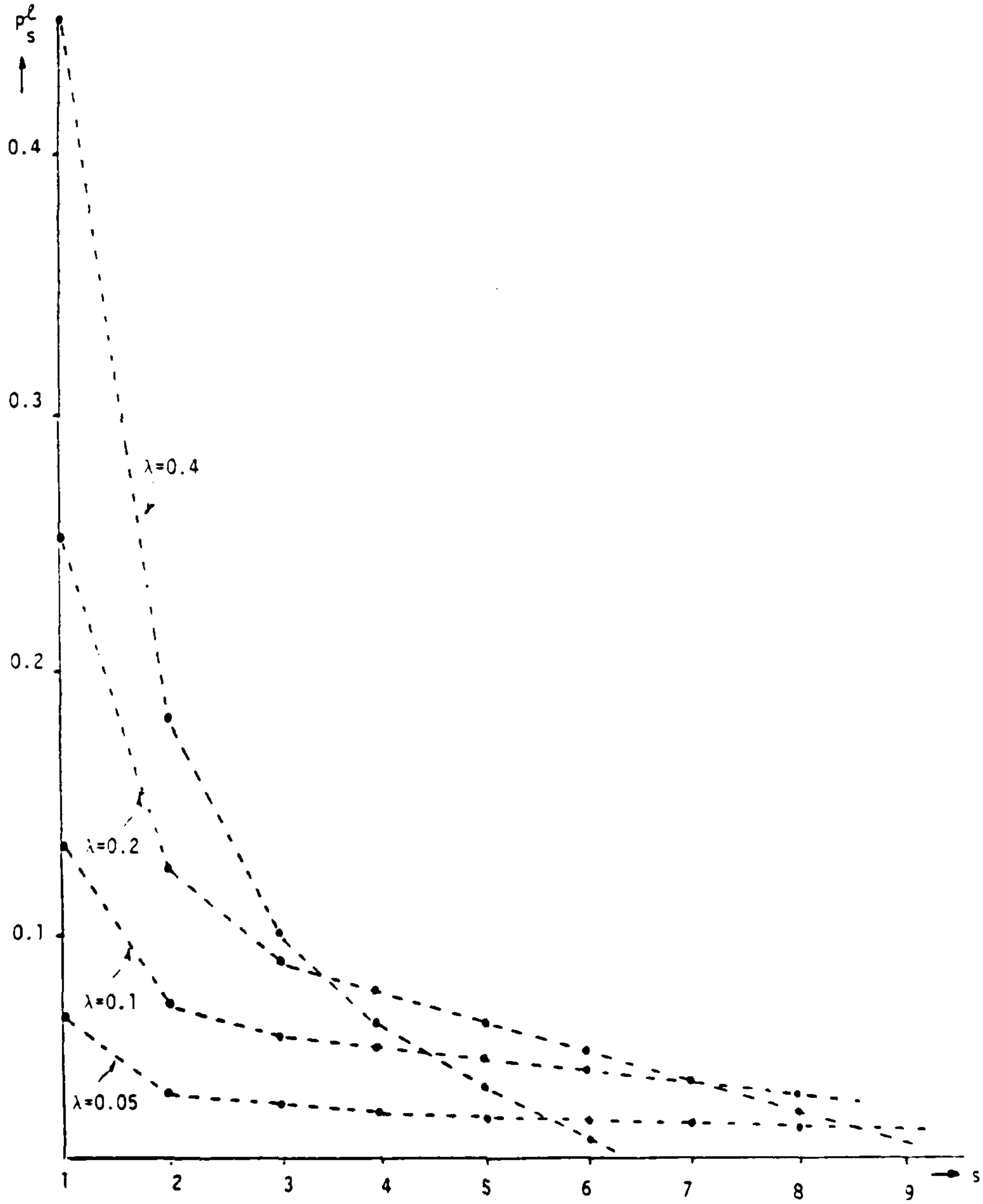


Figure III.2

Lower bounds on the interdeparture distribution.

III.5 CONCLUSIONS

We presented a simple window random access algorithm for systems with binary, collision versus noncollision, feedback. We analyzed the algorithm assuming the Poisson user model, and continuous feedback sensing. In addition to the throughput and delay analyses, we studied the effect of feedback errors on the throughput of the algorithm. We also evaluated the output traffic interdeparture distribution. Both the proposed and the Capetanakis's algorithm attain the same throughput. In terms of induced delays and feedback error insensitivity, the proposed algorithm is better than the Capetanakis's dynamic algorithm. Furthermore, the proposed algorithm can be easily modified to operate under limited feedback sensing, and its output traffic interdeparture distribution can be analytically evaluated.

CHAPTER IV

DELAY DISTRIBUTION ANALYSIS OF WINDOW RANDOM-ACCESS ALGORITHMS

IV.1 INTRODUCTION

Window Random-Access algorithms constitute an important class of Multiple-Access algorithms; they are distributive and attain high throughputs and low delays by controlling the number of simultaneously transmitting users. The throughput analysis of algorithms in this class, is a relatively easy task. The delay analysis, however, presents difficulties, due mainly to the variable window sizes and the complicated state space that some of these algorithms create. These restrictions prohibit the application of results from standard queueing theory in the delay analysis.

Many attempts for the delay analysis have been made. In [23], the class of algorithms with constant window size was considered, and upper bounds on the expected delays were developed. In [24], [20], methods for the computation of bounds on the moments of the delays were presented. A method for the computation of delay distribution for constant window size algorithms appears in [25]. The method in [25] relies on a clever decomposition of the delay process, which allows the application of results from standard queueing theory. The computation of the delay distribution for variable window size algorithms, however, remains an open problem. One possible approach is to compute bounds on the moments of the delays as in [24] or [20], which can then be used for an approximate evaluation of the delay distribution. This approach, however, is not computationally practical.

In this chapter, we show that the methodology employed in [20], can be extended to provide bounds on the distribution of the delays. The quantities of interest are related to the solution of a denumerable system of linear equations. Methods for the computation of the constant terms and the coefficients of the unknowns of the system are developed. The methodology is applied to the delay distribution analysis of both the Capetanakis Window-Access algorithm with binary feedback and the Part-and-try algorithm with binary feedback. It can also be applied directly to other Window Random-Access algorithms with different feedback. An interesting result of the analysis is that as the arrival rate increases, the tails of the distribution become longer, but the median grows much slower than the expected delay.

IV.2 MODEL SPECIFICATION

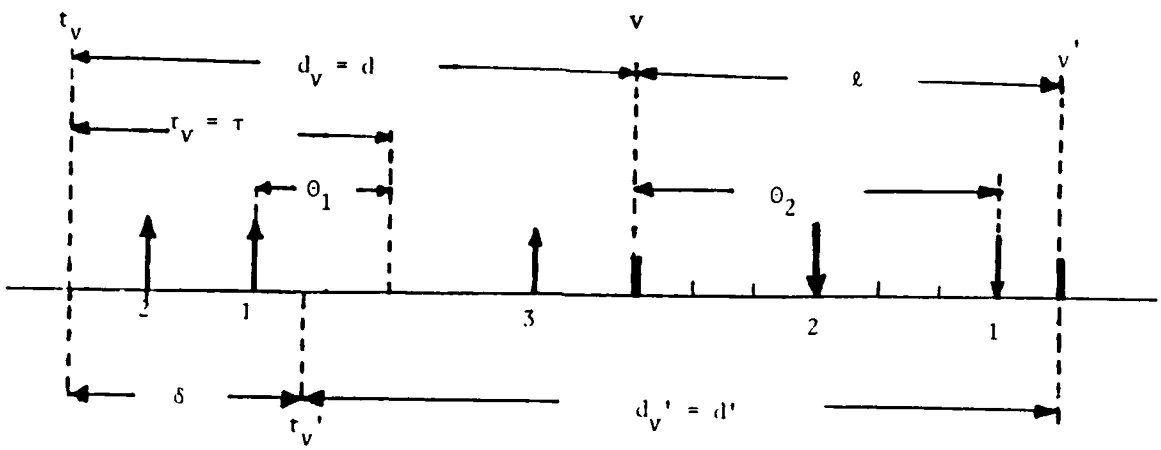
We consider a single slotted channel that is being accessed by a number of independent packet transmitting users. The length of a packet is equal to the length of a slot, and packet transmission may start only at the beginning of a slot. Simultaneous transmission of more than one packets in the same slot, results in complete loss of the information included in the involved packets. The latter event is referred to as a "collision" event. At the end of each slot, all users receive a feedback that provides some information about the channel activity in that slot. Common types of feedback are the binary C-NC (collision versus noncollision) feedback, and the ternary 0-1-C (empty versus success versus collision) feedback. To resolve the collision, the users follow the rules of a Random-Access algorithm. The algorithm is implemented by each user in a distributed fashion, using only the available feedback. The cumulative packet generating process is assumed to be Poisson with rate λ packets per slot.

We assume that a Window Random-Access algorithm is employed, whose basic operating characteristics are the following (see Figure IV.1): Suppose that at the beginning of slot v all packet that arrived before time $t_v < v$ have been successfuliy transmitted, and there is no information concerning the packets that may have arrived in the interval $[t_v, v)$, (i.e., the distribution of the interarrival times of the packets in $[t_v, v)$ is the same as the one assumed originally). The beginning of such a slot v is called a *Conflict Resolution Point (CRP)*. The time difference $d_v = v - t_v$ is referred to as the *lag at v*. In slot v, the users that generated packets in the interval $[t_v,\ t_v + \tau_v)$, where $\tau_v = min(d_v, \Delta)$, are allowed to transmit; Δ is a parameter to be properly chosen for throughput maximization. After a random number of slots and following the rules of the algorithm, another CRP, v', is reached, with a corresponding $t_{v'} > t_v$. All the packets that have been generated in the interval $[t_v,\ t_{v'})$, have been successfully transmitted in the interval $[v, v')$. The intervals $[v, v')$, $[t_v,\ t_v + \tau_v)$, $[t_v,\ t_{v'})$ are called *conflict resolution interval, transmitted interval,* and *resolved interval,* respectively. The length of τ_v, is called the *window size at time* v. Clearly, the window size varies with time, and its maximum size is Δ. Note also, that the length of the conflict resolution interval is one, if and only if there are at most one packets in the transmitted interval.

Algorithms that operate as described above, are the Capetanakis Window Random-Access algorithm [11] and the Part-and-Try algorithm, under either binary C-NC feedback [29], or, ternary 0-1-C feedback [11].

IV.3 STEADY STATE DELAY DISTRIBUTION ANALYSIS

Let packets be labeled 1,2,3,... according to the order of their arrival instants. The delay D_n experienced by the n-th packet is defined as the time difference between its



↑ j : generation instant of the j-th packet

↓ j : successful transmission of the j-th packet

Figure IV.1

Illustration of relationships among certain random variables related to the operation of the algorithm.

steady state distribution of D_n, when it exists.

Let v_i ; $i \geq 1$ be the sequence of successive CRPs and let d_i be the lag at v_i. The sequence d_i ; $i \geq 1$ is a Markov chain with state space F. For most of the existing Window Random-Access algorithms, F is a denumerable subset of the interval $[1,\infty)$. Let $T_1=1$, $d_1=1$, and define T_{i+1}, as the first CRP after T_i, at which the length of the lag equals one. From the description of the algorithm it can be seen that the induced delay process probabilistically restarts itself at the beginning of each slot T_i, $i=1,2,\ldots$ The interval $[T_i,T_{i+1})$ will be referred to as the *i-th session*. Note that the sessions have lengths that are i.i.d. random variables.

Let R_i, $i=1,2,\ldots$, denote the number of packets successfully transmitted in the interval $(0,T_i]$; (note that R_i also represents the number of packets arrived during the interval $[0, T_i-1)$, since T_i is a CRP at which the lag is one). Then, $C_i=R_{i+1}-R_i$; $i \geq 1$, is the number of packets successfully transmitted in the interval $(T_i, T_{i+1}]$ - these are the packets that arrived during the interval $[T_i-1, T_{i+1}-1)$. The sequence R_i; $i \geq 1$, is a renewal process, since C_i, $i \geq 1$, is a sequence of nonnegative i.i.d. random variables. Furthermore, the delay process D_n, $n \geq 1$, is regenerative with respect to the renewal process R_i, $i \geq 1$, with regeneration cycle C_1.

Let

$$I_n(s)=\begin{cases} 1 & \text{if } D_n \leq s \\ 0 & \textit{otherwise} \end{cases}$$

From the regenerative theorem [20], we conclude that if $C=E(C_1) < \infty$, then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} I_n(s) = \lim_{N\to\infty} \frac{1}{N} E\left(\sum_{n=1}^{N} I_n(s)\right) = \frac{E\left(\sum_{n=1}^{C_1} I_n(s)\right)}{C} \qquad \text{(IV.1)}$$

distribution of C_1 is aperiodic and there exists a proper random variable D_∞, such that the sequence D_n; n=1,2,... converges in distribution to D_∞. D_∞ represents the steady state delay induced by the algorithm and its distribution satisfies the equality

$$P(D_\infty \le s) = \frac{E(\sum_{n=1}^{C_1} I_n(s))}{C} \tag{IV.2}$$

From (2) we observe that the steady state distribution of the delays can be determined by computing the quantities of the right hand side of the equality. In [20] it was shown that the finiteness and the computation of C is related to the existence and the computation of an appropriate solution to an infinite system of linear equations. In this section we will show that the same is true for the quantity $E(\sum_{n=1}^{C_1} I_n(s))$.

The following definitions will be used in the sequel.

l	: Length of a conflict resolution interval
δ	: Length of a resolved interval
τ	: Window size.
$E(X/\tau)$	: Expected value of the random variable X, given that the window size is τ.
$p(x,r/\tau)$	: The probability that the conflict resolution interval has length x and the resolved interval has length r, given that the window size is τ.
$p(x/\tau)$	: The probability that the conflict resolution interval has length x, given that the window length is τ.
h_d	: Number of slots needed to reach a CRP with lag 1 given that we start

$k_d(s)$: Number of successfully transmitted packets with delay less than s, in the interval h_d.

$m_{\tau,d}(s)$: Number of successfully transmitted packets with delay less than s during a conflict resolution interval, given that the window size is τ and the lag is d.

$n_\tau(s)$: Number of successfully transmitted packets with delay less than s, during a conflict resolution interval, given that the window size is τ *and the lag is* τ. That is, $n_\tau(s)=m_{\tau,\tau}(s)$.

Let us also define,

$$K_d(s) = E(k_d(s))$$

$$M_{\tau,d}(s) = E(m_{\tau,d}(s))$$

$$N_\tau(s) = E(n_\tau(s))$$

$$H_d = E(h_d)$$

Note that by definition,

$$K_1(s) = E(\sum_{n=1}^{C_1} I_n(s))$$

Also,

$$C=\lambda H_1$$

Therefore, the determination of $K_1(s)$ and H_1, will permit the computation of the steady-state distribution of the delays.

Consider the arrangement of Fig. IV.1. The delay D of the successfully transmitted packet 1, can be decomposed as follows:

$$D = \theta_1+d-\tau+\theta_2$$

Therefore,

$$D \leq s \;\; \textit{iff} \;\; \theta_1 + \theta_2 \leq s - d + \tau \tag{IV.3}$$

But $\theta_1 + \theta_2$ is statistically identical to the delay that the successfully transmitted packet experiences if the transmitted interval is τ *and the lag is* τ. The last observation shows that $m_{\tau,d}(s)$ is identically distributed with $n_\tau(s-d+\tau)$. Observe now that

$$k_d(s) = \begin{cases} m_{\tau,d}(s) & \text{if } d_v' = 1 \\ m_{\tau,d}(s) + k_{d'}(s) & \text{if } d_v' = d' \neq 1 \end{cases} \tag{IV.4}$$

and that

$$d_v' = d_v - \delta + l, \quad \tau = \begin{cases} d_v & \text{if } d_v \leq \Delta \\ \Delta & \text{if } d_v > \Delta \end{cases} \tag{IV.5}$$

From (4) and (5) we conclude that

$$K_d(s) = M_{d,d}(s) + \sum_{\substack{r,x \\ x \neq 1}} K_{d-r+x}(s) p(x,r/d) \quad \text{if } 1 \leq d \leq \Delta, \;\; d \in F \tag{IV.6a}$$

$$K_d(s) = M_{\Delta,d}(s) + \sum_{r,x} K_{d-r+x}(s) p(x,r/\Delta) \quad \text{if } d > \Delta, \; d \in F \tag{IV.6b}$$

Since $m_{\tau,d}(s)$ is identically distributed with $n_\tau(s-d+\tau)$, equations (6a) and (6b) become,

$$K_d(s) = N_d(s) + \sum_{\substack{r,x \\ x \neq 1}} K_{d-r+x}(s) p(x,r/d) \quad \text{if } 1 \leq d \leq \Delta, \;\; d \in F \tag{IV.7a}$$

$$K_d(s) = N_\Delta(s-d+\Delta) + \sum_{r,x} K_{d-r+x}(s) p(x,r/\Delta) \quad \text{if } d > \Delta, \;\; d \in F \tag{IV.7b}$$

Equations (7a), (7b) comprise a denumerable system of linear equations. Of interest to us is the element $K_1(s)$ of a particular solution of this system. The methodology developed in [20] can be used for the study of system (7). Note that the coefficients of the unknowns are independent of s. This observation represents a computational advantage when the solution to (7) is approximated by the solution of appropriate finite linear system of equations, [20]. In this case, the approximate solution can be represented in the form, be computed once, and then used for the computation of the approximate solution for various values of s.

We now proceed in the development of an initial upper bound on the solution of system (7). Following the methodology in [20], such a bound will be the sequence $K_d^0(s) = \gamma_u(s)d + \zeta_u(s)$, if $\gamma_u(s)$, $\zeta_u(s)$, can be determined so that the following inequalities are satisfied

$$K_d^0(s) \geq N_d(s) + \sum_{\substack{r,x \\ x \neq 1}} K_{d-r+x}^0(s) p(x, r/d) = K_d^1(s) \quad \text{if } 1 \leq d \leq \Delta, \quad d \in F \tag{IV.8a}$$

$$K_d^0(s) \geq N_\Delta(s-d+\Delta) + \sum_{r,x} K_{d-r+x}^0(s) p(x, r/\Delta) = K_d^1(s) \quad \text{if } d > \Delta, \quad d \in F \tag{IV.8b}$$

Substituting $K_d^0(s)$ in the right hand side of inequalities (8), it can be seen that if $d \in F$,

$$K_d^1(s) = K_d^0(s) + N_d(s) + \gamma_u(s)(E(l/d) - E(\delta/d) - (1+\lambda d)e^{-\lambda d}) - \zeta_u(s)(1+\lambda d)e^{-\lambda d}$$
$$\text{if } 1 \leq d \leq \Delta \tag{IV.9a}$$

$$K_d^1(s) = K_d^0(s) + N_\Delta(s-d+\Delta) - \gamma_u(s)(E(\delta/\Delta) - E(l/\Delta)) \quad \text{if } d > \Delta \tag{IV.9b}$$

Observe now that $N_\Delta(s)$ is an increasing function of s. Therefore, from (9b) we conclude that,

$$K_d^1(s) \leq K_d^0(s) + N_\Delta(s) - \gamma_u(s)(E(\delta/\Delta) - E(l/\Delta)) \quad \text{if } d > \Delta \tag{IV.10}$$

From (10) we conclude that if $E(l/\Delta) < E(\delta/\Delta)$, the condition for stability of the system, inequalities (9b) are satisfied if

$$\gamma_u(s) = \frac{N_\Delta(s)}{E(\delta/\Delta) - E(l/\Delta)} \tag{IV.11}$$

With this value of $\gamma_u(s)$, it can be seen that inequalities (9a) are satisfied if

$$\psi(d) = \frac{N_d(s) + \gamma_u(s)[E(l/d) - E(\delta/d) - (1+\lambda d)e^{-\lambda d}]}{(1+\lambda d)e^{-\lambda d}} \tag{IV.13}$$

From the above discussion we conclude that the solution to system (7) satisfies the inequalities

$$K_d(s) \le \gamma_u(s)d + \zeta_u(s), \ \ d \in F \tag{IV.14a}$$

where $\gamma_u(s)$, $\zeta_u(s)$, are given by equations (11), (12) respectively. The uniqueness of the solution is guaranteed by the same techniques as in [20]. If we use a similar method for the development of a lower bound, we find that

$$\gamma_l(s)d + \zeta_l(s) = K_d(s), \ d \in F \tag{IV.14b}$$

where

$$\gamma_l(s)=0 \ \text{ and } \ \zeta_l(s) = \inf_{1 \le d \le \Delta} \{(N_d(s)/((1+\lambda d)e^{-\lambda d})\}$$

Bounds on H_1 are given in [20], formulas (18), (19).

As is explained in [20], the bounds (14) can be used to further improve the bounds on $K_1(s)$. To proceed further, however, the computation of the quantities $E(l/d)$, $E(\delta/d)$, $p(x,r/d)$, and $N_d(s)$ is necessary. In section 4, we present a method for the computation of $N_d(s)$ for the Capetanakis dynamic algorithm, and then proceed in the computation of tight upper and lower bounds on the quantities of interest. In section 5, we present a method for the computation of the quantities $N_d(s)$ and $p(x,r/d)$ for the Part-and-Try algorithm, and then we develop bounds on the distribution of the delays. Due to the complicated state space of the latter algorithm, the development of tight bounds for high input rates becomes computationally cumbersome.

IV.4 THE CAPETANAKIS ALGORITHM WITH C-NC FEEDBACK

A complete description of the algorithmic rules can be found in [11]. In this algorithm, the resolved interval is always equal to the window size i.e., $\tau \equiv \delta$. This results in a significant simplification of the state space F, when the maximum window size Δ is a rational number. The restriction of Δ to the rational numbers facilitates the development of tight bounds for the delay distribution and does not represent any disadvantage in practice. Moreover, if m packets are involved in a conflict, the conflict resolution process depends only on m and not on the window length or the generation time of each of the packets. The last property facilitates the development of efficient methods for the computation of the quantities of interest.

Since $\tau \equiv \delta$, we have that, $E(\delta/d) = d$ and

$$p(x,r/d) = \begin{cases} p(x/d) & \text{if } r=d \\ 0 & \textit{otherwise} \end{cases}$$

Formulas for the computation of $E(l/d)$, $p(x/d)$, can be developed by following the reasoning in [11]. In the next section we develop a method by which the quantities $N_d(s)$ can be computed.

IV.4.1 The Computation of $N_d(s)$.

Let M be the number of packets in the window d (see Fig. IV.2). Then,

$$N_d(s) = E(n_d(s)) = E(E(n_d(s)/M)) = \sum_{m=1}^{\infty} E(n_d(s)/M=m)e^{-\lambda d}\frac{(\lambda d)^m}{m!} \quad \text{(IV 15)}$$

Let

$$J_{i,m}(s) = \begin{cases} 1 & \text{if } \textit{the delay of the } i\textit{-th packet in } d \textit{ is at most } s \\ 0 & \textit{otherwise} \end{cases} \quad \text{(IV.16)}$$

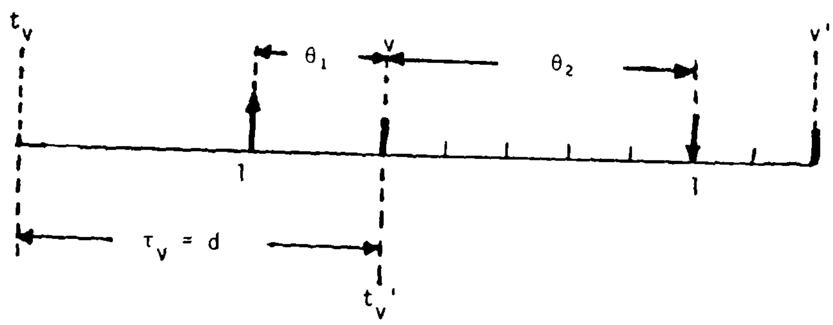


Figure IV.2

Illustration of the delay's decomposition for

the Capetanakis's algorithm.

The "i-th packet" in (16), is the i-th packet in a random enumeration of the m packets (randomly chosen packet). Then

$$n_d(s) = \sum_{i=1}^{m} J_{i,m}(s) \qquad \text{(IV.17)}$$

Given M, the generation time of each of the M packets is uniformly distributed in the window d and independent of the generation time of the rest of the packets. From this observation and the definition in (16) we conclude that the random variables $J_{i,m}(s)$, $1 \le i \le m$, are identically distributed (although not independent). Therefore, from (17) we conclude that

$$E(n_d(s) \,/\, M = m) = mE(J_{1,m}(s) \,/\, M = m) \qquad \text{(IV.18)}$$

Let θ_1, (θ_2), be the delay of the first randomly chosen packet before (after) the initialization of the collision resolution process i.e. until (after) time v (Fig. IV.2). Since θ_1 is uniformly distributed in [0, d), we have that

$$E(J_{1,m}(s) \,/\, M = m) = \frac{1}{d}\int_0^d E(J_{1,m}(s) \,/\, M = m, \theta_1 = \theta)\, d\theta \qquad \text{(IV.19)}$$

Since the conflict resolution process is independent of the packet generation time in a window, we conclude that given m, the random variables θ_1, θ_2, are independent. Therefore.

$$E(J_1(s) \,/\, M = m,\ \theta_1 = \theta) = p(\theta_1 + \theta_2 \le s \,/\, M = m, \theta_1 = \theta) = p(\theta_2 \le s - \theta \,/\, M = m) \quad \text{(IV.20)}$$

and

$$E(J_{1,m}(s) \,/\, M = m) = \frac{1}{d}\int_0^d p(\theta_2 \le s - \theta \,/\, M = m)\, d\theta \qquad \text{(IV.21)}$$

Observe now, that θ_2 takes only positive integer values. Let

$$P_q^{(m)} = p(\theta_2 = q \,/\, M = m)\,;\ q = 1, 2, 3, \ldots$$

then,

$$p(\theta_2 \le s-\theta / M=m) = \sum_{q=1}^{[s-\theta]} P_q^{(m)} \tag{IV.22}$$

where, [a] denotes the integer part of a. Substituting (22) in (21), we conclude that

$$E(J_{1,m}(s) / M=m) = \frac{1}{d} \int_0^d \sum_{q=1}^{[s-\theta]} P_q^{(m)}\, d\theta \tag{IV.23}$$

Let $e=d-s+[s]$. Then, since $P_q^{(m)}$ is independent of θ, (23) can be written as follows:

$$E(J_{1,m}(s)/M=m) = \frac{1}{d}\left((s-[s])\sum_{q=1}^{[s]} P_q^{(m)} + \sum_{q=1}^{[s]-1} P_q^{(m)} + \ldots + \sum_{q=1}^{[s]-[e]} P_q^{(m)} + (e-[e])\sum_{q=1}^{[s]-[e]-1} P_q^{(m)}\right)$$

From (15), (18) and (24), we observe that for the computation of $N_d(s)$, only the quantities $P_q^{(m)}$ are needed. In Appendix C, we provide recursive formulas for the computation of the quantities $P_q^{(m)}$. It is worth noticing that formulas (15), (18) and (24), are valid for any window Random-Access algorithm that has the properties described in the first paragraph of section 3.

IV.4.2 <u>Development of bounds on $K_1(s)$, H_1</u>

For the development of bounds on the delays we chose Δ=2.5 As a result, the state space F becomes,

$$F = \{1,\ 1.5,\ 2.,\ 2.5, \ldots.\}$$

For the algorithm considered in this section, the maximum throughput is achieved for Δ=2.67 [11]. The reduction in throughput due to the choice Δ=2.5 is not significant (less than .1%).

Since $\delta \equiv \tau$, equations (7a), (7b), become:

$$K_d(s) = N_d(s) + \sum_{\substack{x \\ x \ne 1}} K_x(s) p(x/d) \quad \text{if } d=1,\ 1.5,\ 2,\ 2.5 \tag{IV.25a}$$

$$K_d(s) = N_\Delta(s-d+\Delta) + \sum_x K_{d-\Delta+x}(s) p(x/\Delta) \quad \text{if } d=3,\ 3.5,\ 4, \ldots \tag{IV.25b}$$

For the development of bounds on $K_1(s)$ we followed the method of truncation of the infinite system (25), see also [20]. Specifically, in system (25), we replaced the unknowns $K_d(s)$, for $d>40$ with the upper bounds in (14a). The substitution results in a finite system of equations whose solution is an upper bound to the solution of (25) for $d=1$, 1.5, 2,...,40. For the development of lower bounds, we replaced the unknowns $K_d(s)$, $d>40$ with the lower bound in (14b). The same methodology is employed for the development of bounds on H_1. The resulting upper and lower bounds on the distribution of the delays differ by .01 in the worst case. In Figure IV.3, we provide the distribution of the delays for various values of the arrival rate. An important observation is that as the arrival rate increases, the tails of the distribution become longer, but the median grows much slower than the expected delays.

IV.5 THE PART-AND-TRY ALGORITHM WITH C-NC FEEDBACK

A detailed description of the algorithm can be found in [29]. The algorithm has throughput .45, and its basic difference from the Part-and-Try algorithm with ternary feedback in [11] is that if a collision is followed by an empty slot, the packets that are involved in the collision retransmit in the next slot (no splitting takes place). The techniques used in this section, can be easily applied to the analysis of the algorithm under ternary feedback.

IV.5.1 The computation of $N_d(s)$.

Let M be the number of packets in the window d (see Fig IV.4). As in section 4.1, formula (15) holds. Since not all the packets in a window are successfully transmitted, however, we need to modify the definition of $J_{i,m}(s)$ in (16). Let

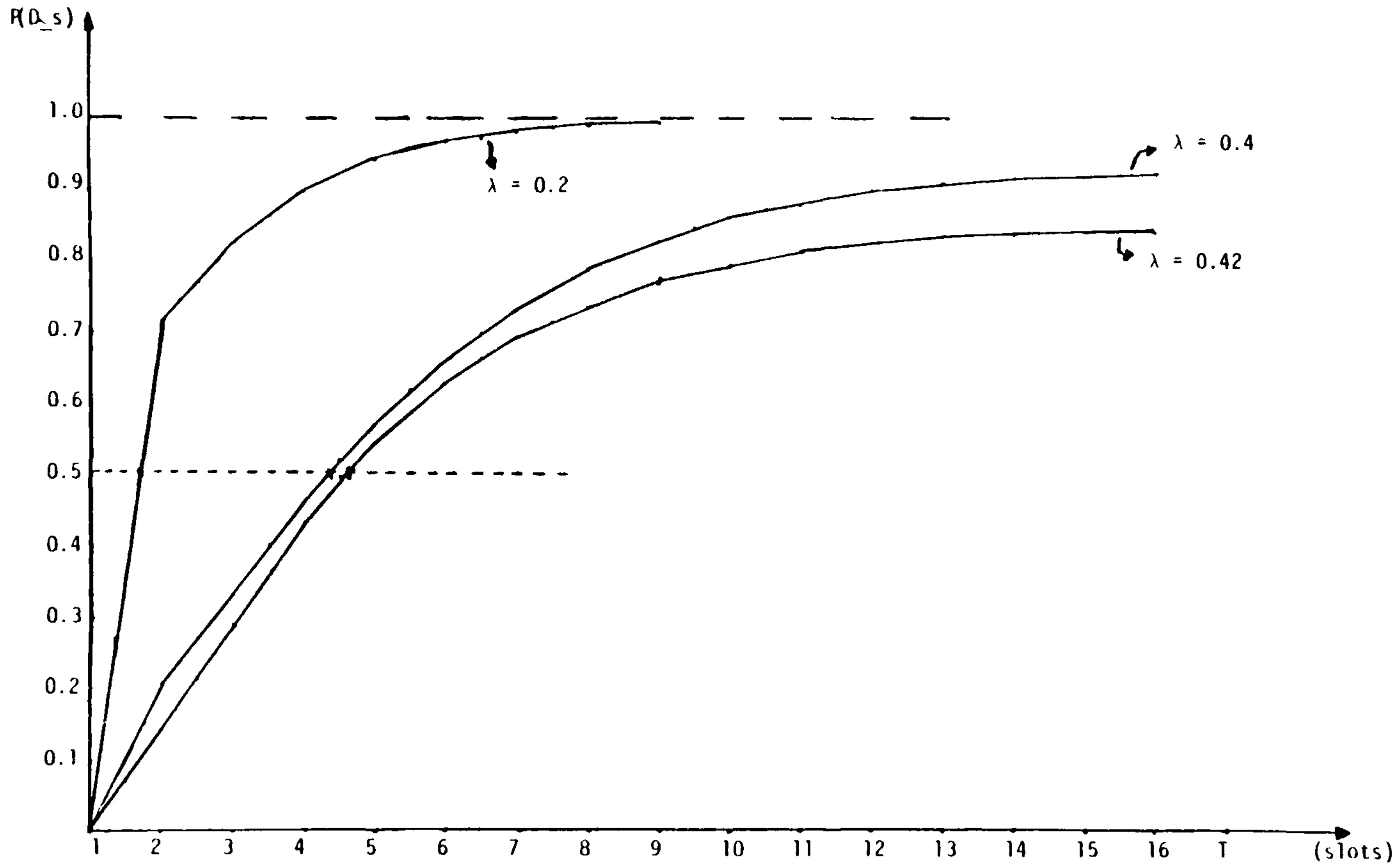


Figure IV.3

Delay distribution for the Capetanakis's algorithm

$$J_{i,m}(s) = \begin{cases} 1 & \text{if } \textit{the } i\textit{-th randomly chosen packet in } d \textit{ is sucessfully} \\ & \textit{transmitted} \text{ and } \textit{its delay is at most } s. \\ 0 & \textit{otherwise} \end{cases} \quad \text{(IV.26)}$$

then, as in (18),

$$E(n_d(s) / M{=}m) = mE(J_{1,m}(s) / M{=}m) \quad \text{(IV.27)}$$

and

$$E(J_{1,m}(s) / M{=}m) = \frac{1}{d}\int_0^d E(J_{1,m}(s) / M{=}m,\ \theta_1{=}\theta)d\theta \quad \text{(IV.28)}$$

Again, θ_2 takes integer values. However, θ_1 and θ_2 are not independent. To proceed further, we define the events

$A_{i,m}(s)$={ *the* i*-th randomly chosen packet is successfully transmitted* and $\theta_2 \le s$.}

$a_{i,m}(q)$ = {*the* i*-th randomly chosen packet is successfully transmitted* and $\theta_2{=}q,\ q{=}1,2,\ldots$}

Since θ takes only integer values, we have that

$$A_{i,m}(s) = \bigcup_{q=1}^{[s]} a_{i,m}(q) \quad \text{(IV.29)}$$

From (28) and (29) we conclude that

$$E(J_{1,m}(s) / M{=}m) = \frac{1}{d}\int_0^d p(A_{1,m}(s-\theta) / M{=}m,\ \theta_1{=}\theta)d\theta \quad \text{(IV.30)}$$

$$= \frac{1}{d}\int_0^d \sum_{q=1}^{[s-\theta]} p(a_{i,m}(q) / M{=}m,\ \theta_1{=}\theta)d\theta$$

Observe now that $p(a_{1,m}(q) / M{=}m,\ \theta_1{=}\theta)$ depends on θ through the ratio $\phi{=}\theta/d$. Let us define

$$f_1(q, m, \phi) = p(a_{i,m}(q) / M{=}m,\ \theta_1{=}\phi d)\ ;\ \phi{=}\frac{\theta}{d} \in [0,1) \quad \text{(IV.31)}$$

then, the following equations hold:

$$f_1(0, m, \phi) = 0,\ \text{ for } m{=}1,2,\ldots,\ \text{and } \phi \in [0,1) \quad \text{(IV.32a)}$$

$$f_1(1,m,\phi)=0,\ \text{ for } m=2,3,\ldots,\ \text{and } \phi\in[0,1) \tag{IV.32b}$$

$$f_1(q,1,\phi)=\begin{cases}1 & \text{if } q=1\\ 0 & \textit{otherwise}\end{cases};\quad \phi\in[0,1) \tag{IV.32c}$$

$$\underset{m\geq 2}{f_1(q,m,\phi)}=\begin{cases}\sum_{j=0}^{m-1} f_1(q-1,j+1,2\phi-1)\begin{bmatrix}m-1\\ j\end{bmatrix}2^{(-(m-1))} & .5\leq\phi<1\\ f_1(q-2,m,2\phi)2^{-(m-1)}+f_1(q-2,m-1,2\phi)(m-1)2^{(-(m-1))} & 0\leq\phi<.5\end{cases} \tag{IV.32d}$$

Formulas (32a), (32b), (32c), are clear. Let us explain formula (32d).

Assume that $.5\leq\phi<1$. Then, the packet under consideration (packet 1 in Figure IV.4), lies in the left hand half (l.h.h.) of the window. Let θ_2' be the delay of packet 1, after the first collision. Then,

$$\theta_2=\theta_2'+1 \tag{IV.33}$$

The location of packet 1 in the l.h.h. of the window, is,

$$\theta_1'=\theta_1-d/2 \tag{IV.34}$$

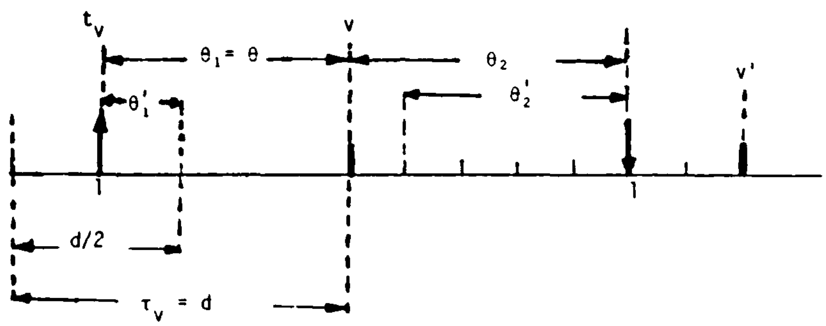
Therefore, the new ratio becomes

$$\phi'=\frac{\theta_1'}{d/2}=2\phi-1 \tag{IV.35}$$

Let $G_{j,m-1}$ be the set

$G_{j,m-1}$=*{j of the* $m-1$ *packets (other than packet* 1*), are located in the l.h.h. of the window.}*

Note that $p(G_{j,m-1})=\begin{bmatrix}m-1\\ j\end{bmatrix}2^{-(m-1)}$. From (33), (34) and (35), we derive formula (32d) for $.5\leq\phi\leq 1$ by conditioning on $G_{j,m-1}$. If $0\leq\phi<.5$, packet 1 is located in the right hand half (r.h.h.) of the window. In this case, we note that if in the l.h.h. of the window there are more than one packets, then packet 1 *is not transmitted* during the collision resolution process. By conditioning again on the $G_{i,m-1}$, and taking into account the last observa-



$$\phi = \theta_1/d$$

$$\phi' = \theta_1' \;/\; (d/2) = 2\phi - 1$$

Figure IV.4

Illustration of delay's decomposition for the Part-and-Try algorithm.

From equations (32) we conclude by induction the following property for the function $f_1(q,m,\phi)$:

Property 1. For fixed q and m, the function $f_1(q,m,\phi)$ is simple (i.e., it takes a finite number of values), and left continuous. The jumps of the function occur at the points $0, 2^{-(q-1)}, \ldots, k2^{-(q-1)}, \ldots, 1-2^{-(q-1)}$.

Taking into account Property 1, we can compute $E(J_{1,m}(s))$, using formula (30).

IV.5.2 The Computation of $p(x,r/d)$.

The method of conditioning on the number of packets in a window (applied in sections 4.1 and 5.1), does not seem to lead to easily computable recursive formulas. In this section, we present an alternative methodology that results in simple recursive formulas for the computation of p(x,r/d), and provides insight into the structure of these probabilities.

Observe first that p(x,r/d) depends on r through the ratio s=r/d. Let

$$f_2(x,s,d) = p(l{=}x,\ \delta{=}sd,\ M\geq 2 \,/\, d)$$

Conditioning on the events {M=0}, {M=1} and $\{M\geq 2\}$, and observing that if M=0,1 then l=1 and $\delta=d$, we conclude that

$$p(x,r/d) = \delta_K(x-1)\delta_K((s/d)-1)e^{-(\lambda d)}(1+\lambda d) + f_2(x,s,d) \qquad \text{(IV.36)}$$

where

$$\delta_K(y) = \begin{cases} 1 & \text{if } y=0 \\ 0 & \textit{otherwise} \end{cases}$$

The function $f_2(x,s,d)$ satisfies the following recursive formulas

$$f_2(x,s,d)=\begin{cases} f_2(x-1,2s,d/2) & 0\le s\le .5 \\ f_2(x-2,2s-1,d/2)e^{-(\lambda d/2)}(1+\lambda d/2) & .5<s\le 1 \\ +\ \delta_K(x-3)\delta_K(s-1)e^{-(\lambda d)}(\lambda d/2)^2 & \end{cases} \tag{IV.37b}$$

We derive equation (37a) for $0\le s\le .5$ The case $.5<s\le 1$ can be derived by a similar reasoning. Let M' be the number of packets in the l.h.h of the window. Let l' be the number of slots during a collision resolution process *after* the first collision. Then,

$$p(l{=}x,\ \delta{=}sd,\ M\ge 2\ /\ d)=p(l'{+}1{=}x,\ \delta{=}sd,\ M\ge 2,\ M'\ge 2\ /\ d)+p(l{=}x,\ \delta{=}sd,\ M\ge 2,\ M'\le 1\ /\ d)$$

But if $M'\le 1$, then s>.5. Therefore $p(l{=}x,\ \delta{=}sd,\ M\ge 2,\ M'\le 1\ /\ d)=0$ if $0.\le s\le .5$. It remains to observe that

$$p(l'{+}1{=}x,\ \delta{=}sd,\ M\ge 2,\ M'\ge 2\ /\ d)=p(l'{=}x-1,\ \delta{=}2(sd/2),\ M'\ge 2\ /\ d)$$

and that if $M'\ge 2$, the collision resolution process after the first collision is statistically identical to a collision resolution process that started with a window of size d/2.

From (37a), (37b) we conclude by induction that the function $f_2(x,s,d)$ has the following property:

Property 2 For fixed $x\ge 3$, and for any d, s takes a finite number r_x of values. The sequence r_x satisfies the following recursions: $r_3=r_4=1$, $r_x=r_{x-1}+r_{x-2}$; $x\ge 5$. Let A_x be the set of values of s for given x. Then,[1]

$$A_3=\{1\},\ \ A_4=\{1/2\},\qquad A_x=(.5A_{x-1})\cup(.5A_{x-2}+.5)\ ;\ x\ge 5$$

The values of $f_2(x,s,d)$ for various x and s, can be computed from (37b). The probabilities p(x,r/d), are easily computed from (36).

IV.5.3. The Development of bounds on $K_1(s), H_1$.

The state space F for the Part-and-Try algorithm is a dense subset of $[1, \infty)$. This property complicates the development of finite systems of linear equations whose solution will provide upper or lower bounds to the quantities $K_1(s)$, H_1. Taking advantage of the structure of the probabilities p(x,r/d) (see section 5.2), however, we can proceed as follows: Let B be a finite subset of F, that includes the state d=1. We develop the following finite system of linear equations:

$$Y_d = N_d(s) + \sum_{(d-r+x)\in B^c} (\gamma_u(s)(d-r+x) + \zeta_u(s))p(x,r/d) + \sum_{\substack{(d-r+x)\in B \\ x\neq 1}} Y_{d-r+x}p(x,r/d)$$

$$\text{if } 1 \le d \le \Delta,\ d\in B \qquad \text{(IV.38a)}$$

$$Y_d = N_\Delta(s-d+\Delta) + \sum_{(d-r+x)\in B^c} (\gamma_u(s)(d-r+x) + \zeta_u(s))p(x,r/d) + \sum_{(d-r+x)\in B} Y_{d-r+x}p(x,r/\Delta)$$

$$\text{if } d > \Delta,\ d\in B \qquad \text{(IV.38b)}$$

$\gamma_u(s)$ and $\zeta_u(s)$ are determined from (11), (12). Due to property 2 of section 4.2, it is simple to find for a given $d\in B$, the values of x and r such that $(d-r+x) \in B$. The summation over the infinite set B^c, can be computed in terms of $E(l/d)$, $E(\delta/d)$, $\gamma_u(s)$, $\zeta_u(s)$, and the probabilities $p(x,r/d)$; $(d-r+x) \in B$. The solution Y_d; $d\in B$ of system (38), is an upper bound to the solution $K_d(s)$; $d\in B$ of system (7), see also [20]. Since $1\in B$, we can determine a bound on $K_1(s)$. Similarly, lower bounds on $K_1(s)$, and upper and lower bounds on H_1 can be developed. For the computations we used the set $B = \{1, 1.125, \ldots, 1+(k/8), \ldots, 9\}$. The resulting bounds on the delay distribution are presented in Figure IV.5 for $\lambda = .1, .2, .3$. For higher arrival rates, the bounds are not tight and although they can be improved by enlarging the set B, the computations become cumbersome.

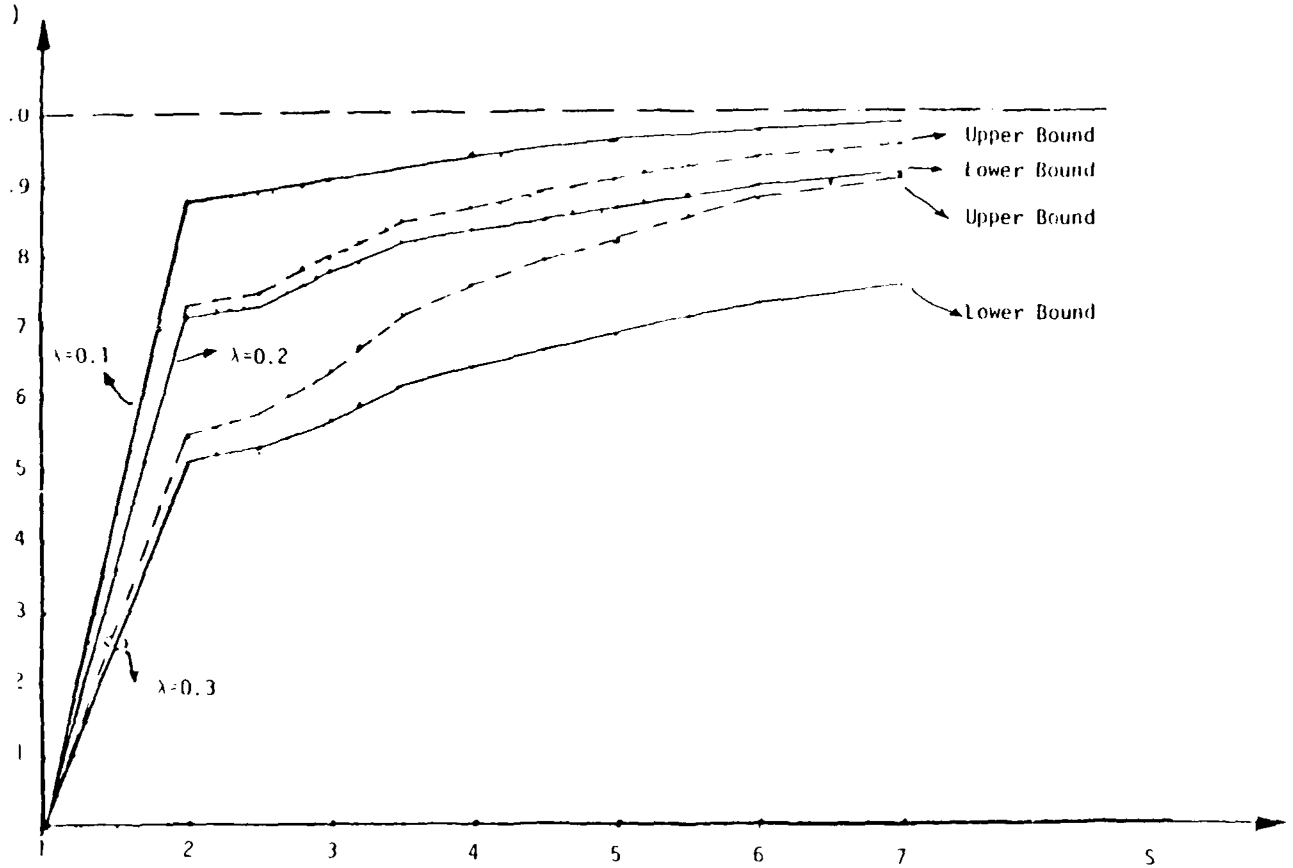


Figure IV.5

Delay distribution for the Part-and-Try algorithm.

IV.6 CONCLUSIONS

We developed a method for the computation of bounds on the delay distribution of Window Random-Access algorithms. The method has been applied to the delay distribution analysis of the Capetanakis Window Random-Access algorithm and the Part-and-Try algorithm both under binary C-NC feedback. The bounds developed for the Capetanakis algorithm, are tight for all arrival rates within the stability region of the algorithm. For the Part-and-Try algorithm, however, the bounds are satisfactory for relatively low arrival rates. The computational difficulty in obtaining tight bounds for the latter algorithm, is due to its complicated state space. The techniques can be easily applied to other Window Random-Access algorithms whose operating characteristics are as described in Section 2.

CHAPTER V

RANDOM ACCESS ALGORITHMS FOR TWO-CLUSTER PACKET RADIO SYSTEMS

V.1 INTRODUCTION

In this chapter we consider a two-cluster packet radio network. In each cluster a single common channel is used for packet transmissions by the users in it. We assume that the users are not necessarily static. Consequently, limited-sensing random access algorithms are the most appropriate since they require that a user observes the feedback channel from the time he generates a packet to the time this packet is successfully transmitted.

In a multi-cluster packet radio network neighboring clusters may overlap. The users located in the overlapping regions are then exposed to transmissions and feedbacks from more than one clusters. The latter phenomenon can be exploited to improve the delays for the users located in the overlapping regions, (see [36] for a similar problem in mobile telephone systems). Consider for example clusters 1 and 2, and call the users located in their overlapping region, *marginal users*, let us also call the users in cluster i , i = 1 ,2, which are not located in the overlapping region, *local users*. The local users in each cluster transmit their packets over the common channel dedicated to that cluster. To resolve possible collisions they employ a random access algorithm, called RAAi (i stands for cluster i). Due to the double exposure the marginal users have a choice: they can join RAA1 or RAA2 for the transmission of their packets. This choice can be implemented in a static or dynamic way. The static implementation consists of a priori assigned probabilities. Upon generation of a new packet, each marginal user decides to join RAAi with probability p_i, where $p_1 + p_2 = 1$, and remains there until his packet is successfully transmitted. The policy described above is simple. However, it requires that the marginal users know the a priori assigned probabilities at all times. The latter is not considered a serious disadvantage, unless the a priori assignment changes with time due to changes in system dynamics. This is the case of multi-cluster systems with mobile users, where users enter and leave the system freely. In such a dynamically changing topology it is hard to update all the marginal users regarding the optimal values of the probabilities p_i. Consequently a dynamic implementation should be adopted. Such an implementation may in addition give a delay advantage to the marginal users. The latter may be desirable in systems where the marginal users play a special role, i.e. they transmit priority or control messages. In this chapter, we consider two overlapping clusters, each employing the limited sensing RAA presented in chapter III. Next, we propose a dynamic protocol via which the marginal users decide which cluster to join. The organization of this chapter is as follows: In section V.2 we present the system model. In section V.3 we describe the protocol used by the marginal users. In sections V.4 and V.5 we present the stability analysis. Section V.6 contains our conclusions.

V.2 SYSTEM MODEL

Consider a two-cluster packet-radio system. We assume that in each of the two clusters, the synchronous limited sensing random access algorithm presented in chapter III is employed. Time is divided into slots of length equal to the packet duration, and the starting instants of the slots are identical in both clusters. By the end of each slot all the users in each cluster receive feedback information that informs them about the transmission activity in the cluster. This feedback is either ternary, (Collision (C) versus Success (S) versus Empty (E)), or binary, (Collision (C) versus Non-Collision (NC)). In each cluster, a local user is required to monitor the feedback channel that corresponds to its cluster. We assume negligible propagation delays and error-free forward and feedback channels.

Each marginal user is able to monitor both the feedback channels. At the time when a marginal user generates a new packet, he starts monitoring the feedback channels from both clusters until he decides to join the operations of one of the two RAAs, for the transmission of his packet. Upon this decision, he maintains the continuous monitoring of the feedback channel that corresponds to the RAA he chose, until his packet is successfully transmitted.

The mobility of the users is assumed low so that, each user remains within the same geographical region (local or marginal) from the time he generates a packet to the time that this packet is successfully transmitted. Assuming that each of the RAAs operates away from the saturation point (throughput), then with high probability this time period is relatively small.

It is assumed that the local traffic generated in cluster i, i=1, 2, is Poisson distributed with intensity λ_i, i=1,2, and that the traffic generated by the marginal users is also Poisson distributed with intensity λ_3.

V.3 THE ALGORITHM

We assume that the two clusters employ the same random access algorithm. We decided to adopt the algorithm presented in Chapter III since it has simple operational characteristics and is very insensitive to feedback channel errors. Recall that for the algorithm in Chapter III, a user with a new packet waits for two consecutive noncollision (NC) slots before he transmits his packet for the first time.

Upon generation of a new packet, a marginal user monitors both the feedback channels until the first time that he is allowed to transmit his packet over one of the forward channels, (i.e., occurrence of two consecutive NC slots in the corresponding feedback channel. If the user is allowed to transmit his packet over both the forward channels at the same time, he then selects one of them with probability 0.5). If the received feedback is NC then, the packet has been successfully transmitted and the user stops observing both the feedback channels. However, if the received feedback is C then, the user only observes the feedback channel that corresponds to the forward channel over which he made his first transmission attempt and follows the steps of the RAA. Upon the successful transmission of his packet the user stops monitoring the aforementioned feedback channel.

Upon generation of a new packet, a local user monitors the feedback channel that corresponds to its cluster until the successful transmission of his packet. From the description of the first time transmission rule for both the local and marginal users, it is clear that the marginal users have an advantage over the local. In particular, their waiting delay (time interval from the packet's generation time until the first transmission attempt) is smaller than (sometimes equal to) that of the local users. Consequently, their total delays are smaller than those for the local users.

V.4 ALGORITHMIC ANALYSIS

For notational convenience we will refer to the local users in cluster 1, the local users in cluster 2, and the marginal users, as subsystem 1, subsystem 2, and subsystem 3, respectively. We assume that the three subsystem traffics are generated by independent processes, and that the number of packet arrivals per slot in subsystem j, j=1,2,3, is Poisson distributed with intensity λ_j. From the throughput analysis of the adopted RAA (see Chapter III), we know that the algorithm attains a maximum throughput 0.429, and that the optimal window size is $\Delta^* = 2.33$. Therefore, the following conditions are necessary for stability:

$$\lambda_1 < 0.429 \text{ , and } \lambda_2 < 0.429 \text{ , and } \lambda_1 + \lambda_2 + \lambda_3 < 2\,(0.429) = 0.858 \qquad \text{(V.1)}$$

The necessary conditions in (V.1) determine a $\{\lambda_j, \text{ j=1,2,3}\}$ hyperplane, which contains the $\{\lambda_j, \text{ j=1,2,3}\}$ region that determines the system throughput. Bounds on the $\{\lambda_j, \text{ j=1,2,3}\}$ space which provides the system throughput, will be obtained via the stability analysis in Section V.5 below.

V.5 SYSTEM STABILITY

Assume that the algorithms in both clusters start operating at time zero. Consider the sequence in time of the collision resolution intervals (CRIs), induced by the two RAAs. We define the sequence $\{T_n\}_{n\geq 0}$ as follows: (1) T_n corresponds to the starting slot of some CRI, (note that at T_n, two CRIs may simultaneously begin; one for each of the two RAAs). (2) T_0=2, and at T_0 two CRIs begin; one for each RAA.

Let $\{T_n^{(s)}\}_{n\geq 0}$ be the subsequence of the sequence $\{T_n\}_{n\geq 0}$, which consists of those time instants when two CRIs begin simultaneously. Clearly, $T_0^{(s)}=T_0=2$. Let $D_{n,s}^{(j)}$; j=1,2,3, denote the total length of the unresolved arrival intervals in subsystem j, at the time instant $T_n^{(s)}$. $D_{n,s}^{(j)}$ is then called "the lag of subsystem j at time $T_n^{(s)}$." From the operations of the algorithm we conclude that: (1) $D_{n,s}^{(j)} \geq 2$ and that the possible values of $D_{n,s}^{(j)}$, j =1,2 are countably many. (2) $D_{0,s}^{(j)}=2$, j=1,2,3. (3) At time $T_n^{(s)}$, the RAA in cluster k, k=1,2, examines two arrival intervals: one from subsystem k which has length $\min(D_{n,s}^{(k)}, \Delta)$ and contains arrivals generated by a Poisson process with intensity λ_k arrivals per slot, and one from subsystem 3 which has length $\min(D_{n,s}^{(3)}, \Delta)$ and contains arrivals generated by a Poisson process with intensity 0.5 λ_3 arrivals per slot. (4) The triple ($D_{n,s}^{(j)}$, j=1,2,3) describes the state of the system at time $T_n^{(s)}$, and the sequence $\{S_n\}_{n\geq 0} \triangleq \{D_{n,s}^{(j)}, j=1,2,3\}_{n\geq 0}$ is a three-dimensional irreducible and aperiodic Markov Chain.

The stability of the system is determined by the ergodicity of the three-dimensional Markov Chain $\{S_n\}_{n\geq 0}$. In addition, at time $T_n^{(s)}$, the backlogs of each of the three subsystems are represented by the three lags, $D_{n,s}^{(j)}$, j=1,2,3. Given $D_{n,s}^{(j)} = d_n^{(j)}$, j=1,2,3, we define the expected system backlog at time $T_n^{(s)}$ to be the $\sum_{j=1}^{3} \lambda_j d_n^{(j)}$, where $V(\bar{d}_n) = V(\{d_n^{(j)}\}_{1\leq j\leq 3}) \triangleq \sum_{j=1}^{3} \lambda_j\, d_n^{(j)}$ is a Lyapunov function of the three subsystem lags $\{d_n^{(j)}\}_{1\leq j\leq 3}$.

Let C denote the state space of the Markov Chain $\{S_n\}_{n\geq 0}$, and define the operator $AV(\bar{d}) = AV(\{d^{(j)}\}_{1\leq j\leq 3})$, called a generalized drift, as follows:

$$AV(\bar{d}) = AV(\{d^{(j)}\}_{1\leq j\leq 3}) \triangleq E\{V(\{D_{n+1,s}^{(j)}\}_{1\leq j\leq 3}) - V(\{D_{n,s}^{(j)}\}_{1\leq j\leq 3}) \mid D_{n,s}^{(j)}=d^{(j)}, j=1,2,3\}$$

$$= E\{\sum_{j=1}^{3}\lambda_j D_{n+1,s}^{(j)} \mid D_{n,s}^{(j)}=d^{(j)}, j=1,2,3\} - \sum_{j=1}^{3}\lambda_j d^{(j)};\ \bar{d}=\{d^{(j)}\}_{1\leq j\leq 3} \in C \qquad (V.2)$$

Since we are interested in determining the system's throughput we assume that the system operates at the saturation point therefore, all the lags are assumed to be sufficiently long, so that in the time interval $[T_n^{(s)}, T_{n+1}^{(s)}]$, the examined interval is always of length Δ. Consequently the number of packets that are successfully transmitted during the first CRI that belongs to RAAi, $i = 1,2$ in the above interval, is Poisson distributed with parameter $(\lambda_i + 0.5\,\lambda_3)\,\Delta$. Furthermore, the number of packets that are successfully transmitted during every subsequent CRI that belong to RAAi, $i = 1,2$ in the above interval, is also Poisson distributed with parameter $(\lambda_1 + \lambda_3)\,\Delta$. Under the above conditions and given the subsystem Poisson rates $\{\lambda_j\}_{1\le j\le 3}$, let us define:

$E_\lambda\{l \mid u\}$: Given that the length of the interval to be examined equals to u and, that the Poisson input rate equals λ, the expected number of slots needed for the successful transmission of all the packet arrivals within the examined interval.

$i=1,2$; $N_i(\{\lambda_j\},\Delta)$: The number of CRIs in $[T_n^{(s)}, T_{n+1}^{(s)}]$ that belong to RAAi, $i = 1,2$.

$AV(\bar{d}_\Delta, \{\lambda_j\})$: The generalized drift in (V.2). We next define a generalized drift for the RAA in the i-th cluster:

$$i=1,2;\ A^{(i)}V(\bar{d}_\Delta, \{\lambda_j\}) \stackrel{\Delta}{=} E_{\lambda_i+\lambda_3/2}\{l \mid \Delta\} - \Delta + \left[E\{N_i(\{\lambda_j\},\Delta)\} - 1\right]\left[E_{\lambda_i+\lambda_3}\{l \mid \Delta\} - \Delta\right] \tag{V.3}$$

Then, from (V.2) we derive the following expression:

$$AV(\bar{d}_\Delta, \{\lambda_j\}) = (\lambda_1 + \frac{\lambda_3}{2})A^{(1)}V(\bar{d}_\Delta, \{\lambda_j\}) + (\lambda_2 + \frac{\lambda_3}{2})A^{(2)}V(\bar{d}_\Delta, \{\lambda_j\}) + - \frac{\lambda_3\Delta}{2}\left[E\{N_1(\{\lambda_j\},\Delta)\} + E\{N_2(\{\lambda_j\},\Delta)\} - 2\right] \tag{V.4}$$

where,

$$E_{\lambda_1+\lambda_3/2}\{l \mid \Delta\} + [E\{N_1(\{\lambda_j\},\Delta)\} - 1]E_{\lambda_1+\lambda_3}\{l \mid \Delta\} = = E_{\lambda_2+\lambda_3/2}\{l \mid \Delta\} + [E\{N_2(\{\lambda_j\},\Delta)\} - 1]E_{\lambda_2+\lambda_3}\{l \mid \Delta\} \tag{V.5}$$

We now state a Lemma whose proof is similar to the one presented in [37].

Lemma 1

(i) Let there exist some $\varepsilon>0$, such that the two conditions below are satisfied:

$$A^{(1)}V(\bar{d}_\Delta, \{\lambda_j\}) < -\varepsilon$$
$$A^{(2)}V(\bar{d}_\Delta, \{\lambda_j\}) < -\varepsilon \qquad \text{(V.6)}$$

Then, the Markov chain $\{S_n\}_{n\geq 0}$ is ergodic at the Poisson rates $\{\lambda_j\}_{1\leq j\leq 3}$.

(ii) Let at least one of the two generalized drifts, $A^{(i)}V(\bar{d}_\Delta, \{\lambda_j\}), i=1,2$, be nonnegative. Then, the Markov Chain $\{S_n\}_{n\geq 0}$ is nonergodic at the Poisson rates $\{\lambda_j\}_{1\leq j\leq 3}$. □

The conditions in (V.6) define lower bounds on the $\{\lambda_j\}$ regions for which the Markov chain $\{S_n\}_{n\geq 0}$ is ergodic.

In Appendix D, we describe the methodology used in the evaluation of the maximum $\{\lambda_j\}$ values for which the chain $\{S_n\}_{n\geq 0}$ is ergodic. In Figure V.1, we plot the boundaries of the (λ_1, λ_2) stable regions, parametrized by various λ_3 values. Those boundaries are symmetric around the 45° straight line. In Figure V.2, we plot λ_3^* against $\lambda = \lambda_1 = \lambda_2$, (symmetrically-loaded system).

From Figures V.1 and V.2, we observe that the maximum value of the sum $\lambda_1+\lambda_2+\lambda_3$ is always strictly less than 0.858 (i.e., strictly less than twice the throughput of each local RAA). This is so because under the conditions described above the RAAi, $i = 1,2$, operates with an input rate equal to $(\lambda_i + \lambda_3)\Delta$ during most of the time. Consequently, if the rate of the local traffic is close to the algorithmic throughput for one of the RAAs in the system, the introduction of marginal users results in instability.

V.6 CONCLUSIONS

In this chapter, we performed stability analysis for a two-cluster packet radio network. Each cluster contains local and marginal users. In each cluster a single common channel is available for packet transmission. Packets that are generated by the local users in a cluster, can be only transmitted over the forward channel that corresponds to that cluster. In contrast, packets that are generated by the marginal users can be transmitted over either one of the two available forward channels. We propose a protocol according to which the marginal users are able to dynamically select the forward channel over which they will eventually transmit their packet. According to the protocol rules, each marginal user with a packet for transmission is required to observe both the feedback channels from the packet's generation time, in order to decide over which channel he will transmit his packet. No a priori knowledge of the input traffic rates, or the subsystem's states is required.

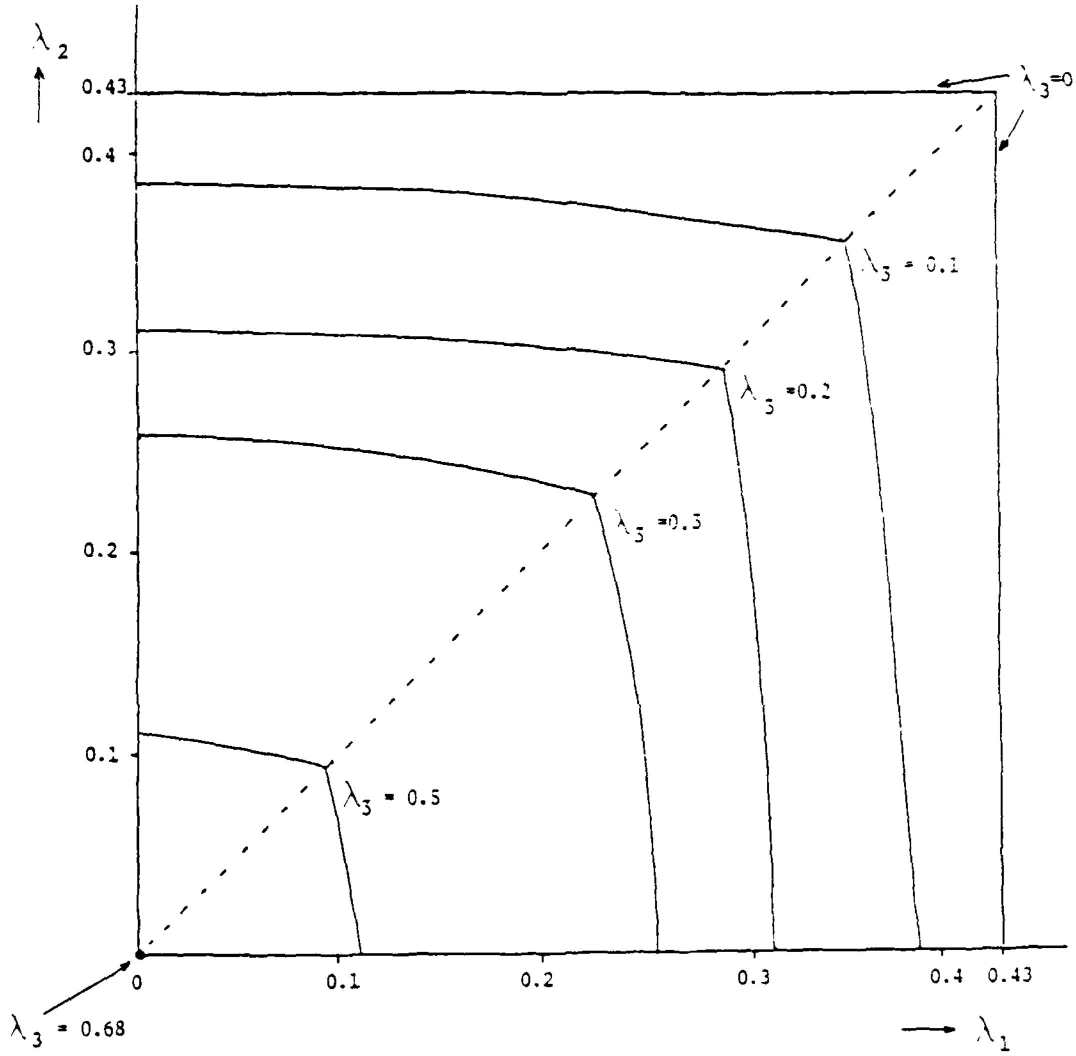


Figure V.1

Boundaries of the (λ_1 , λ_2) stable regions

parametrized by λ_3.

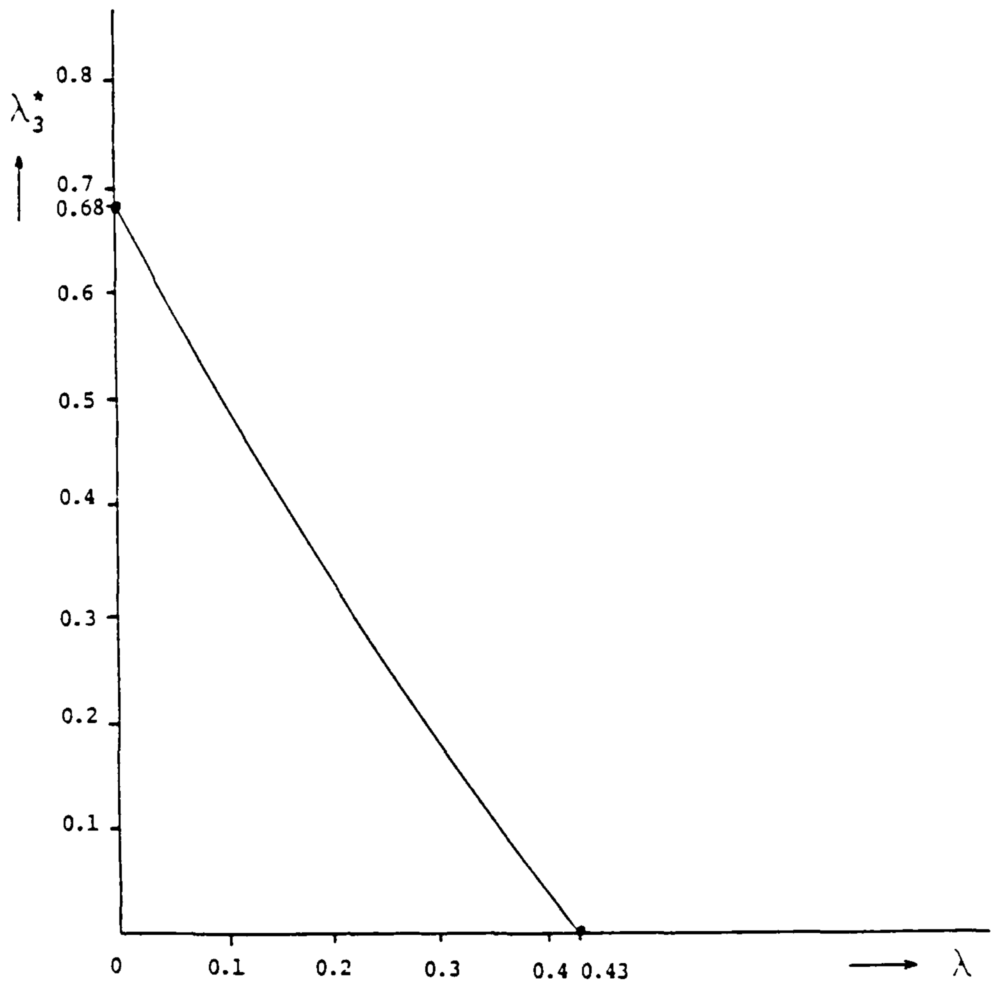


Figure V.2

Maximum acceptable rate λ_3^*, versus $\lambda = \lambda_1 = \lambda_2$

REFERENCES

[1] D. Bertsekas, R. Gallager, Data Networks, Prentice Hall, 1987.

[2] P. E. Jackson and C. D. Stubbs, " A Study of Multiaccess Computer Communication," in 1969 Spring Joint Comp. Conf. AFIPS Conf. Proc., Vol. 34, Montvale N.J, AFIPS Press, pp. 491-504.

[3] N. Abramson, " Packet Switching With Satellites," in 1973 Nat. Comp. Conf., AFIPS Conf. Proc., Vol. 42, New York, AFIPS Press, 1973, pp. 695-702.

[4] N. Abramson, " The ALOHA System - Another Alternative for Computer Communications," 1970 Fall Joint Comp. Conf., AFIPS Conf. Proc., Vol. 37, pp. 281-285.

[5] G. Fayolle and E. Gelenbe," Stability and Optimal Control of the Packet Switching Broadcast Channel," Journal of the Association for Computing Machinery, Vol. 24, No. 3, July 1977, pp. 375-386.

[6] B. S. Tsybakov and V. A. Mikhailov, " Ergodicity of a Slotted ALOHA System," Problemy Peredachi Informatsii, Oct.-Dec. 1979, Vol. 15, pp.73-87.

[7] J. I. Capetanakis, "Generalized TDMA: The Multiple Accessing Tree Protocol," IEEE Trans. on Communications, Oct. 1979, COM-27, pp.1476-1484.

[8] B. S. Tsybakov and V. A. Mikhailov, " Free Synchronous Packet Access in a Broadcast Channel With Feedback," Problemy Peredachi Informatsii, Oct.-Dec. 1978, Vol. 14, No. 4, pp. 34-59.

[9] R. G. Gallager, " A Perspective of Multiaccess Channels," IEEE Trans. on Inform. Theory, Special Issue on Random Access Communications, March 1985, Vol. IT-31,

[10] B. S. Tsybakov and V. A. Mikhailov, " Random Multiple Access of Packets, Part-and-Try Algorithm," Problemy Peredachi Informatsii, Oct.-Dec. 1980, Vol. 16, No. 3, pp. 65-79.

[11] J. L. Massey, Collision Resolution Algorithms and Random Access Communications, in Multi-User Communication Systems, Ed. G. Longo, CISM Courses and Lectures, No. 265, pp. 73-137, Springer-Verlag, New York, 1981.

[12] B. S. Tsybakov and N. D. Vvedenskaya, " Random Multiple Access Stack Algorithm," Problemy Peredachi Informatsii, July-Sept. 1980, Vol. 16, No. 3, pp. 80-94.

[13] N. D. Vvedenskaya and B. S. Tsybakov, " Random Multiple Access of Packets to a Channel With Errors," Problemy Peredachi Informatsii. 1983, Vol. 19, No. 2, pp. 52-68.

[14] P. Mathys, Analysis of Random Access Protocols, Ph.D Thesis, Swiss Federal Institute of Technology, Zurich, 1984.

[15] L. Merakos and C. Bisdikian, " Delay Analysis of the n-ary Stack Algorithm for a Random Access Broadcast Channel," Proceedings of the 22nd Allerton Conf. on Comm. Comp., Univ. of Illinois at Urbana-Champaign, Illinois, October 1984, pp. 385-394.

[16] L. Georgiadis, Limited Sensing Random Access Algorithms and Unified Methods for their Analysis, Ph.D Dissertation, Univ. of Connecticut, Storrs, 1986.

[17] L. Georgiadis and P. Papantoni - Kazakos, " A 0.487 Throughput Limited Sensing Algorithm," IEEE Trans. on Inform. Theory, Vol. IT-33, No. 2, March 1987.

[18] P. A. Humblet, " On the Throughput of Channel Access Algorithms With Limited Sensing," IEEE Trans. on Communications, Vol. COM-34, No. 4, April 1986.

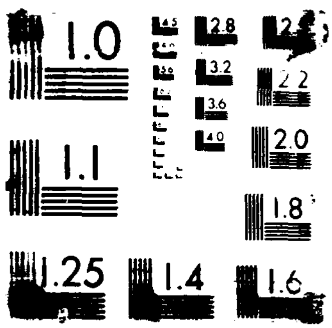
[19] B. S. Tsybakov, " Survey of USSR Contributions in Random Multiple Access Communications," IEEE Trans. on Info. Theory, Vol. IT-31, No. 2, March 1985.

[20] L. Georgiadis, L. Merakos, and P. Papantoni - Kazakos, " A Method for Delay

AD-A198 090 RANDOM ACCESS ALGORITHMS FOR MULTIUSER COMPUTER COMMUNICATION NETWORKS..(U) VIRGINIA UNIV CHARLOTTESVILLE DEPT OF ELECTRICAL ENGINEERING.. 2/2

UNCLASSIFIED P PAPANTONI-KAZAKOS ET AL. JUL 88 F/G 12/7 NL

END
FILMED
88


1.0
2.8
3.2
3.6
4.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

IEEE Journal on Selected Areas in Communications, July 1987.

[21] J. Kurose, M. Schwartz, and Y. Yemini, " Multiple Access Protocols and Time Constrained Communication," ACM Computing Surveys, Vol. 16, No. 1, March 1984.

[22] J. Kurose, M. Schwartz, and Y. Yemini, " Controlling Time Window Protocols for Time Constrained Communication in a Multiple Access Environment," in Proceedings of the 8th International Data Communications Symposium, Oct. 1983, pp. 75-84.

[23] B. S. Tsybakov, N. B. Likhanov, " Upper Bound for the Delay in a Random Access System With a Splitting Algorithm," Problemy Peredachi Informatsii, Vol. 18, No. 4, pp. 76-84, Oct.-Dec. 1982.

[24] J. C. Huang and T. Berger, " Delay Analysis of Interval Searching Contention Resolution Algorithms," IEEE Trans. on Inform. Theory, Vol. IT-31, No. 2, March 1985.

[25] G. Polyzos, M. Molle, and A. Venetsanopoulos, " Performance Analysis of Finite Non-Homogeneous Population Tree Conflict Resolution Algorithms Using Constant Size Window Access," IEEE Trans. on Comm., Vol. COM-35, No. 1, pp. 1124-1138, November 1987.

[26] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channel," IEEE Trans. Inform. Theory, Vol. IT-25, pp. 505-515, Sept. 1979.

[27] R. G. Gallager, "Conflict Resolution in Random Access Broadcast Networks", in Proc. AFOSR Workshop Communication Theory and Applications, Provincetown, MA, Sept. 1978, pp. 74-76.

[28] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, " On the Relation Between the Finite and the Infinite Population Models for a Class of RAAs", IEEE Trans. on Comm., Vol. COM-35, pp. 1239-1240, Nov. 1987.

[29] P. Studer, and H. Pletscher, "Q-ary Part-and-Try Algorithm for Packet Conflict Resolution", Inst. Telecommun. Swiss Fed. Inst. Technol., Zurich, Switzerland, Oct. 11,

1984.

[30] A. G. Pakes, "Some Conditions for Ergodicity and Recurrence of Markov Chains", Oper. Res., Vol. 21, pp. 617-622, 1973.

[31] M. Kaplan, "A Sufficient Condition for Nonergodicity of a Markov Chain", IEEE Trans. Info. Theory, Vol. IT-25, pp. 470-471, 1979.

[32] T. Gonsalves, "Packet-Voice Communication on an Ethernet Local Computer Network: An Experimental Study," In Proc. of the SIGCOM Communications Architectures and Protocols Symposium, (Austin, TX, March 1983), ACM, New York, pp. 178-185.

[33] G. Nutt, and D. Bayer, "Performance of CSMA/CD Networks Under Combined Voice and Data Loads," IEEE Trans. on Comm., Vol. COM-30, Jan. 1982, pp. 6-11.

[34] E. Seneta, Non-negative Matrices, John Wiley&Sons, New York, 1973.

[35] Ayal Bar-David, and Moshe-Sidi, "Collision Resolution Algorithms in Multistation Packet Radio Networks," submitted for publication.

[36] B. Eklundh, "Channel Utilization and Blocking Probability in a Cellular Mobile Telephone System with Directed Retry," IEEE Trans. Comm., Vol. COM-34, pp. 329-337, April 1986.

[37] W. Szpankowski, "Stability Conditions for Multidimensional Queueing Systems with Applications," J. of Operations Research, to appear.

[38] W. Szpankowski, "Some Theorems on Instability with Applications to Multi-Access Protocols," J. of Operations Research, to appear.

[39] R. L. Tweedie, "Criteria for Classifying General Markov Chains," Adv. Appl. Prob., Vol. 8, pp. 737-771, 1976.

APPENDIX A

A.1 Recursions

Towards the computation of $E\{\theta_d\}$ and $P(l \mid u,d)$, let us denote by $E\{\cdot \mid k_1,k_2,B\}$ the expected value, given that k_i packets have counter value equal to i, i=1,2, and that the maximum length of the collision resolution interval is B. Let $P(\cdot \mid k_1,k_2,B)$ denotes conditional probability, given k_i, $i=0,1$, and B as above. Then, for $\zeta_{u,d}$ denoting either one of the quantities $l_{u,d}$, $z_{u,d}$, and $n_{u,d}$, we clearly have:

$$E\{\zeta_{u,d}\} = \sum_{k=0}^{\infty} E\{\zeta_{u,d} \mid k, 0, T-d\} e^{-\lambda u} \frac{(\lambda u)^k}{k!}$$

$$P(l \mid u,d) = \sum_{k=0}^{\infty} P(l \mid k, 0, T-d) e^{-\lambda u} \frac{(\lambda u)^k}{k!}$$

It is also easy to prove that,

$$E\{\psi_{u,d}\} = 2^{-1} u \, E\{n_{u,d}\}$$

Note that $\sup_k E\{\zeta_{u,d} \mid k, 0, T-d\} < \infty$. Therefore, the computation of a finite number of terms is sufficient for developing tight bounds on the quantities of interest. Also, note that given k_1, k_2, B, the operation of the algorithm is independent of u and d. Let us denote:

$$L_{k_1,k_2,B} = E\{l_{u,d}/k_1,k_2,B\},\ N_{k_1,k_2,B} = E\{n_{u,d}/k_1,k_2,B\},\ Z_{k_1,k_2,B} = E\{z_{u,d}/k_1,k_2,B\}$$

From the operation of the algorithm we derive the following recursions:

$$L_{k_1,k_2,B} = \begin{cases} 1 + L_{k_2,0,B-1}, & 0 \le k_1 \le 1 \\ 1 + L_{i,k_2+k_1-i,B-1}, & w.p. \binom{k_1}{i} 2^{-k_1},\ k_1 > 1,\ k_1 \ge i \ge 0 \end{cases}$$

$$N_{k_1,k_2,B} = \begin{cases} N_{k_2,0,B-1}, & k_1 = 0 \\ N_{k_2,0,B-1} + 1, & k_1 = 1 \\ N_{i,k_1+k_2-i,B-1}, & w.p. \binom{k_1}{i} 2^{-k_1},\ k_1 > 1,\ k_1 \ge i \ge 0 \end{cases}$$

$$Z_{k_1,k_2,B} = \begin{cases} 1 + N_{k_2,0,B-1} + Z_{k_2,0,B-1}, & k_1 = 1 \\ N_{k_2,0,B-1} + Z_{k_2,0,B-1}, & k_1 = 0 \\ N_{i,k_1+k_2-i,B-1} + Z_{i,k_1+k_2-i,B-1}, & w.p. \binom{k_1}{i} 2^{-k_1}, k_1 > 1, k_1 \ge i \ge 0 \end{cases}$$

$$P(l \mid k_1,k_2,B) = \begin{cases} P(l-1 \mid k_2,0,B-1), & k_1 = 0 \text{ or } 1 \\ P(l-1 \mid i,k_1+k_2-i,B-1), & w.p. \binom{k_1}{i} 2^{-k_1},\ k_1 > 1,\ k_1 \ge i \ge 0 \end{cases}$$

The initial conditions in the above recursions are the following:

$$L_{0,0,B} = L_{1,0,B} = L_{2,0,1} = L_{0,1,1} = 1,\ B \ge 1$$

$$L_{2,0,2} = L_{0,1,B} = 2,\ B \ge 2$$

$$N_{0,0,B} = N_{0,1,1} = N_{2,0,1} = 0,\ B \ge 1$$

$$N_{1,0,B} = 1,\ B \ge 1,\ \ N_{2,0,2} = 0.5,\ B \ge 2$$

$$N_{0,1,B} = 1,\ B \ge 2$$

$$Z_{1,1,1} = Z_{1,0,B} = 1,\ B \ge 1$$

$$Z_{k_1,k_2,1} = 0,\ k_1 \ge 2,\ k_2 \ge 0,\ \ Z_{1,1,B} = 3,\ B \ge 2$$

$$Z_{k_1,k_2,2} = 2k_1 2^{-k_1},\ k_1 \ge 2$$

$$P(l \mid k_1,k_2,B) = 0,\ l > B$$

$$P(2 \mid 1,0,B) = P(l \mid 1,0,B),\ B \ge 1,\ l \ge 2$$

$$P(1 \mid k,0,B) = 0,\ B \ge 2,\ k \ge 2$$

$$P(1\mid 1,0,B)=P(1\mid k,0,1)=P(2\mid k,0,2)=1,\ B\geq 1,\ k\geq 2$$

$$P(2\mid k_1,k_2,2)=1,\ k_1\geq 2$$

A.2 Computations in the Absence of Bounds on Delays

Let $T=\infty$, and let us then define:

$L_{n,k-n}$: The expected number of slots needed by the algorithm, for the successful transmission of k packets, given that n of the packets have counter values equal to one, and $k-n$ of the packets have counter values equal to two.

From the operation of the algorithm, we then obtain:

$$L_{n,k-n} = 1 + \sum_{i=0}^{n} \binom{n}{i} 2^{-n} L_{i,k-i} \tag{A.1.a}$$

$$L_{0,0} = L_{1,0} = 1,\quad L_{0,i} = 1+L_{i,0};\quad i\geq 1 \tag{A.1.b}$$

It can be shown by induction that $L_{n,k-n}$ has the following form:

$$L_{n,k-n} = A_n^{(1)} L_{k,0} + A_n^{(2)} L_{k-1,0} + A_n^{(3)};\ k\geq 2,\quad 2\leq n\leq k \tag{A.2}$$

; where $A_n^{(i)}$, $i=1,2,3$ are independent of k and can be computed recursively as follows:

$$A_n^{(1)}=[1-2^{-n}]^{-1}\, 2^{-n}\left\{1+\sum_{i=2}^{n-1} A_i^{(1)}\binom{n}{i}\right\},\quad n\geq 3 \tag{A.3.a}$$

$$A_n^{(2)}=[1-2^{-n}]^{-1}\, 2^{-n}\left\{n+\sum_{i=2}^{n-1} A_i^{(2)}\binom{n}{i}\right\},\quad n\geq 3 \tag{A.3.b}$$

$$A_n^{(3)}=[1-2^{-n}]\left\{1+2^{-n}(n+1)+2^{-n}\sum_{(i=2)}^{n-1} A_i^{(3)}\binom{n}{i}\right\},\quad n\geq 3 \tag{A.3.c}$$

$$A_2^{(1)}=1/3,\ A_2^{(2)}=2/3,\ A_2^{(3)}=7/3 \tag{A.3.d}$$

For n=k, expression (A.2) gives:

$$\mathrm{L}_k = \mathrm{L}_{k,0} = \frac{A_k^{(2)}}{1-A_k^{(1)}}\mathrm{L}_{k-1,0} + \frac{A_k^{(3)}}{1-A_k^{(1)}} \tag{A.4}$$

where it can be found by induction, that:

$$A_k^{(3)} \le k+1 \ , \ k\ge 1 \tag{A.5}$$

$$A_k^{(1)} \le \frac{1}{3} \ , \ k\ge 1$$

$$\frac{A_k^{(2)}}{1-A_k^{(1)}} \le 1 \ , \ k\ge 1$$

From (A.4) and (A.5), we finally find:

$$0 \le \mathrm{L}_k = \mathrm{L}_{k,0} \le \frac{3}{4}k^2 + \frac{9}{4}k - 2 \ , \quad k \ge 1 \tag{A.6}$$

The bounds in (A.6) are used in the derivation of tight upper and lower bounds on the expected value in (A.7) below, where λ is the intensity of the input Poisson traffic.

$$E\{l_{u,d}\} = \sum_{k=0}^{\infty} \mathrm{L}_k e^{-\lambda u} \frac{(\lambda u)^k}{k!} \tag{A.7}$$

In the same way, we can develop recursive formulas and bounds for the quantities associated with the cumulative delays of the packets during a CRI. These quantities can then be used for the computation of the delays in the unconstrained case, by following the methodology presented in [20].

APPENDIX B

B.1. Stability Analysis

We present the stability analysis in the case feedback errors may occur. By setting $\varepsilon = \delta = 0$ in our results, we get the corresponding quantities in the error free case.

The stability region of the algorithm is provided by inequality (III.1), where the expected value, is given by expression (III.2). We start with the computation of the expected values L_k ; $k \geq 0$ in (III.2a).

Computation of L_k

We define:

$G_{n,k-n}$: The expected number of slots needed by the algorithm, for the successful transmission of k packets, given that n of those packets have counter values equal to one, and k-n of the packets have counter values equal to two.

Notice that $L_k = G_{k,0}$, for $k \geq 2$, while $L_k \neq G_{k,0}$, for $k < 2$. We first show how to compute L_0 and L_1.

Computation of L_0:

From the operation of the algorithm we have

$$L_0 = \begin{cases} 1 \; ; w.p. \; (1-\varepsilon) \\ 1 + G_{0,0} \; ; w.p. \; \varepsilon \end{cases} \tag{B.1.1}$$

where

$$G_{0,0} = \begin{cases} 1 + L_0 \ ; \ w.p. \ (1-\varepsilon) \\ 1 + G_{0,0} \ ; \ w.p. \ \ \varepsilon \end{cases} \tag{B.1.2}$$

From (B.1.1) and (B.1.2) we find that

$$L_0 = \frac{1}{(1-\varepsilon)^2} \tag{B.1.3}$$

<u>Computation of L_1</u>

We find that L_1 satisfies the following

$$L_1 = \begin{cases} 1 \ ; \ w.p. \ (1-\delta) \\ 1+G_{1,0} \ ; \ w.p. \ 0.5\delta \\ 1+G_{0,1} \ ; \ w.p. \ 0.5\delta \end{cases} \tag{B.1.4}$$

where $G_{1,0}$ and $G_{0,1}$ satisfy the following

$$G_{1,0} = \begin{cases} 1+L_0 \ ; \ w.p. \ (1-\delta) \\ 1+G_{1,0} \ ; \ w.p. \ 0.5\delta \\ 1+G_{0,1} \ ; \ w.p. \ 0.5\delta \end{cases} \tag{B.1.5}$$

$$G_{0,1} = \begin{cases} 1+L_1 \ ; \ w.p. \ (1-\varepsilon) \\ 1+G_{0,1} \ ; \ w.p. \ \varepsilon \end{cases} \tag{B.1.6}$$

From (B.1.4), (B.1.5), and (B.1.6) we find that

$$L_1 = \left[1 - \frac{\delta}{2-\delta} \right]^{-1} \left[1 + \frac{\delta}{2(1-\varepsilon)} + \frac{\delta}{(2-\delta)} \left[1 + \frac{(1-\delta)}{(1-\varepsilon)^2} + \frac{\delta}{2(1-\varepsilon)} \right] \right] \tag{B.1.7}$$

<u>Computation of L_k, for $k \geq 2$.</u>

From the operation of the algorithm we obtain:

$$G_{n,k-n} = 1 + \sum_{i=0}^{n} \binom{n}{i} 2^{-n} \, G_{i,k-i} \ ; \ n \geq 2 \, , \, k \geq n \tag{B.1.8}$$

where

$$G_{0,k} = \frac{1}{(1-\varepsilon)} + L_k \tag{B.1.9}$$

$$G_{1,k} = \frac{1+L_k(1-\delta) + 0.5\delta\left[\dfrac{1}{(1-\varepsilon)} + L_{k+1}\right]}{1-0.5\delta} \qquad \textbf{(B.1.10)}$$

It can be shown by induction that $G_{n,k-n}$ has the following form

$$G_{n,k-n} = A_n^{(1)}\, G_{k,0} + A_n^{(2)}\, G_{k-1,0} + A_n^{(3)} \ ; \ 2\le n\le k \qquad \text{(B.1.11)}$$

; where $A_n^{(i)}$, i=1,2,3 are independent of k and can be computed recursively as follows:

$$A_2^{(1)} = \frac{2+\delta}{6-3\delta}\ ,\ A_2^{(2)} = \frac{4(1-\delta)}{6-3\delta}\ ,\ A_2^{(3)} = \frac{14-3\delta-12\varepsilon+4\varepsilon\delta}{3(1-\varepsilon)\,(2-\delta)} \qquad \text{(B.1.12)}$$

$$A_n^{(1)} = [1-2^{-n}]^{-1}\, 2^{-n}\left\{1+\frac{\delta n}{(2-\delta)} + \sum_{i=2}^{n-1}\begin{bmatrix} n \\ i \end{bmatrix} A_i^{(1)}\right\} \ ,\ n\ge 3 \qquad \text{(B.1.13)}$$

$$A_n^{(2)} = [1-2^{-n}]^{-1}\, 2^{-n}\left\{\frac{n\,(1-\delta)}{(1-0.5\delta)} + \sum_{i=2}^{n-1}\begin{bmatrix} n \\ i \end{bmatrix} A_i^{(2)}\right\} ,\ n\ge 3 \qquad \text{(B.1.14)}$$

$$A_n^{(3)} = [1-2^{-n}]^{-1}\left\{1+\frac{n\,2^{-n}}{(1-0.5\delta)} + \frac{2^{-n}}{(1-\varepsilon)} + \frac{\delta n\, 2^{-n}}{(2-\delta)(1-\varepsilon)} + 2^{-n}\sum_{i=2}^{n-1}\begin{bmatrix} n \\ i \end{bmatrix} A_i^{(3)}\right\} ,\ n\ge 3 \qquad \text{(B.1.15)}$$

For n=k, expression (B.1.11) gives:

$$L_k = G_{k,0} = \frac{A_k^{(2)}}{1-A_k^{(1)}}\, L_{k-1} + \frac{A_k^{(3)}}{1-A_k^{(1)}} \ ,\ k\ge 2 \qquad \text{(B.1.16)}$$

Expression (B.1.16) together with the recursions in (B.1.13)-(B.1.15) provide a mean for the computation of L_k, $k\ge 2$.

Development of an upper bound on L_k

It can be seen by induction that:

$$A_k^{(1)} \le \frac{1}{3} + \frac{\delta}{2-\delta} \ ;\ k\ge 2 \qquad \text{(B.1.17)}$$

$$\frac{A_k^{(2)}}{1-A_k^{(1)}} = 1 \ ;\ k\ge 2 \qquad \text{(B.1.18)}$$

$$A_k^{(3)} \le \frac{(2-2\varepsilon+\delta)}{(2-\delta)(1-\varepsilon)}\, k + \frac{1}{(1-\varepsilon)} \qquad \text{(B.1.19)}$$

From (B.1.16)-(B.1.19) we find:

$$L_k \le L_{k-1} + \frac{3(2-2\varepsilon+\delta)}{(1-\varepsilon)(4-5\delta)}\, k + \frac{3(2-\delta)}{(1-\varepsilon)(4-5\delta)}\ , \ k\ge 2 \tag{B.1.20}$$

from which we can show that

$$L_k \le \frac{3(2-2\varepsilon+\delta)}{2(1-\varepsilon)(4-5\delta)}\, k^2 + \frac{3(6-2\varepsilon-\delta)}{2(1-\varepsilon)(4-5\delta)}\, k + \left[L_1 - \frac{6(2-\varepsilon)}{(1-\varepsilon)(4-5\delta)} \right] \stackrel{\Delta}{=} L_k^u = \alpha k^2 + \beta k + \gamma \tag{B.1.21}$$

Due to the upper bound on L_k, we conclude then that the following condition is sufficient for stability:

$$\sum_{k=0}^{30} L_k\, p(k \mid \Delta) + \sum_{k=31}^{\infty} L_k^u\, p(k \mid \Delta) < \Delta \tag{B.1.22}$$

where

$$p(k \mid \Delta) = e^{-\lambda\Delta}\, \frac{(\lambda\Delta)^k}{k!}$$

After some manipulations, we conclude that (B.1.22) is equivalent to:

$$f(\lambda\Delta) = \sum_{k=0}^{30} L_k\, p(k \mid \Delta) + \alpha \left\{ (\lambda\Delta)^2 + \lambda\Delta - \sum_{k=0}^{30} k^2\, p(k \mid \Delta) \right\} +$$

$$+ \beta \left[\lambda\Delta - \sum_{k=0}^{30} k\ \ p(k \mid \Delta) \right] + \gamma \left[1 - \sum_{k=0}^{30} p(k \mid \Delta) \right] < \Delta \tag{B.1.23}$$

Let us now define:

$$x \stackrel{\Delta}{=} \lambda\Delta \tag{B.1.24}$$

Then, from (B.1.23)-(B.1.24) we conclude that, for the stability of the algorithm, it is sufficient that the input rate λ satisfies the following inequality.

$$\lambda < \sup_{x\ge 0}\ \frac{x}{f(x)} \tag{B.1.25}$$

The following condition specifies a region of λ values for which the algorithm is unstable.

$$\lambda > \sup_{x\ge 0}\ \frac{x}{g(x)} \tag{B.1.26}$$

where $g(x) = \sum_{k=0}^{30} L_k\, e^{-x}\, x^k / k!$

The maximization of expessions in (B.1.25)-(B.1.26) has been done numerically, and provides the throughput, as well as the optimal window size Δ^*. In all the cases the order of the difference between the two suprema in (B.1.25)-(B.1.26) is less than 10^{-3}. The optimal window size is found as $x^* (\lambda^*)^{-1}$, where x^* is the value that attains the suprema in (B.1.25).

Computation of L_k', for the Capetanakis's Dynamic Algorithm

Here, the quantitites L_k', $k \geq 2$ can be computed recursively as follows:

$$L_k' = [1-2\,2^{-k}]^{-1} \left\{ 1+2\,L_0'\;2^{-k} + 2 \sum_{i=1}^{k-1} L_i' \binom{k}{i}\, 2^{-k} \right\}, \quad k \geq 2 \tag{B.1.27}$$

where

$$L_0' = \frac{1}{(1-2\varepsilon)} \text{ and } L_1' = \frac{1-2\varepsilon+\delta}{(1-2\varepsilon)(1-\delta)} \tag{B.1.28}$$

Moreover the following upper bound on L_k',$k \geq 4$, has been found.

$$L_k' \leq \left[\frac{3(1-\varepsilon)}{(1-2\varepsilon)} + \frac{2(\delta-\varepsilon)}{(1-2\varepsilon)(1-\delta)} \right] k - 1 \tag{B.1.29}$$

Using the bounds in (B.1.29), we computed upper and lower bounds on the throughput for the Capetanakis's dynamic algorithm. The computed upper and lower bounds were identical to each other up to the fourth decimal point, and are included in Table III.3.

B.2. Delay Analysis

The following definitions will be used in the sequel.

l : Length of a collision resolution interval (CRI).

τ : Window size.

$E(X/\tau)$: Expected value of the random variable X, given that the window size is τ.

$p(x/\tau)$: The probability that the CRI has length x given that the window size is τ.

g_d : Number of slots needed to reach a CRP with lag 1 given that the current lag is equal to d, $d \in \mathbf{F}$.

w_d : Cumulative delay experienced by all the packets that were successfully transmitted during g_d slots.

N : Number of packets transmitted in the CRI, that starts at time t.

z : Cumulative delay of the N packets, after the CRP t.

ψ : Cumulative delay of the N packets, until the instant t_2.

Let us also define,

$$G_d = E(g_d) \quad W_d = E(w_d)$$

Note that by definition,

$$W_1 = E\left(\sum_{n=1}^{Q_1} D_n\right) = W$$

Also, the mean session length $G = E(T_{i+1} - T_i)$; $i \geq 1$, equals to G_1. If $G_1 < \infty$, then by Wald's identity we have that,

$$Q = \lambda G_1$$

Therefore, the determination of W_1 and G_1 will permit the computation of the mean packet delay.

The operation of the algorithm yields the following relations for the g_d's, $d \in \mathbf{F}$.

$$1 \leq d \leq \Delta \quad g_d = l \qquad \text{if } l = 1 \tag{B.2.1}$$

$$1 \le d \le \Delta \quad g_d = l + g_l \quad \text{if } l > 1 \tag{B.2.1a}$$

and that

$$\Delta < d \quad g_d = l + g_{d-\Delta+l} \tag{B.2.2}$$

Taking expectations in (B1)-(B2) yields:

$$G_d = E(l \mid d) + \sum_{s \ne 1, s \in \mathbf{F}} G_s p(s/d) \quad \text{if } 1 \le d \le \Delta,\ d \in \mathbf{F} \tag{B.2.3}$$

$$G_d = E(l \mid \Delta) + \sum_{s \in \mathbf{F}} G_{d-\Delta+s} p(s/\Delta) \quad \text{if } d > \Delta,\ d \in \mathbf{F} \tag{B.2.4}$$

Equations (B.2.3) and (B.2.4) comprise a denumerable system of linear equations. Of interest to us is the element G_1 of a particular solution of this system. We now proceed in the development of an initial upper bound on the solution of the system in (B.2.3)-(B.2.4). Following the methodology in [20], such a bound will be the sequence $G^0_{d,u} = \gamma_u d + \zeta_u$, if γ_u, ζ_u, can be determined so that the following inequalities are satisfied

$$G^0_{d,u} \ge E(l \mid d) + \sum_{s \ne 1, s \in \mathbf{F}} G^0_{s,u} p(s/d) = G^1_{d,u} \quad 1 \le d \le \Delta,\ d \in \mathbf{F} \tag{B.2.5}$$

$$G^0_{d,u} \ge E(l \mid \Delta) + \sum_{s \in \mathbf{F}} G^0_{d-\Delta+s} p(s/\Delta) = G^1_{d,u} \quad \text{if } d > \Delta,\ d \in \mathbf{F} \tag{B.2.6}$$

Substituting G^0_d in the right hand side of inequalities (B.2.5) and (B.2.6), it can be seen that if $d \in \mathbf{F}$,

$$G^1_{d,u} = G^0_{d,u} + \gamma_u(E(l/d) - d - (1+\lambda d)e^{-\lambda d}) - \zeta_u(1+\lambda d)e^{-\lambda d} \quad \text{if } 1 \le d \le \Delta \tag{B.2.7}$$

$$G^1_{d,u} = G^0_{d,u} + E(l/\Delta) - \gamma_u(\Delta - E(l/\Delta)) \quad \text{if } d > \Delta \tag{B.2.8}$$

From (B.2.8) we conclude that if $E(l/\Delta) < \Delta$, the condition for stability of the system, inequalities (B.2.6) are satisfied if,

$$\gamma_u = \frac{E(l \mid \Delta)}{\Delta - E(l/\Delta)} \tag{B.2.9}$$

With this value of γ_u, it can be seen that inequalities (B.2.5) are satisfied if

$$\zeta_u = \max\{-\gamma_u,\ \sup_{1 \le d \le \Delta} (\theta(d))\} \tag{B.2.10}$$

where

$$\theta(d) = \frac{E(l/\,d) + \gamma_u[E(l \mid d) - d - (1+\lambda d)e^{-\lambda d}]}{(1+\lambda d)e^{-\lambda d}} \tag{B.2.11}$$

Therefore,the solution to system (B.2.3)-(B.2.4) satisfies the inequalities

$$G_d \le \gamma_u d + \zeta_u,\ d \in \mathbf{F} \tag{B.2.12}$$

The uniqueness of the solution is guaranteed by the same techniques as in [20]. If we use a similar method for the development of a lower bound, we find that

$$\gamma_l d + \zeta_l = G_d,\ d \in \mathbf{F} \tag{B.2.13}$$

where

$$\gamma_l = \gamma_u, \text{ and } \zeta_l = \inf_{1 \le d \le \Delta} (\theta(d)) \tag{B.2.14}$$

From the operation of the algorithm we also have that W_d ; $d \in \mathbf{F}$ satisfy the following system of linear equations,

$$W_d = E(z \mid d) + E(\psi \mid d) + \sum_{s \in \mathbf{F}, s \ne 1} W_s\, p(s \mid d),\quad 1 \le d \le \Delta,\ d \in \mathbf{F} \tag{B.2.15}$$

$$W_d = E(z \mid \Delta) + E(\psi \mid \Delta) + (d - \Delta)\,\lambda\,\Delta + \sum_{s \in \mathbf{F}} W_{d-\Delta+s}\, p(s \mid \Delta),\ \Delta < d,\ d \in \mathbf{F} \tag{B.2.16}$$

Following the methodology in [20] we can show that

$$W^0_{d,l} = \mu_l d^2 + \nu_l d + \xi_l \le W_d \le \mu_u d^2 + \nu_u d + \xi_u = W^0_{d,u} \tag{B.2.17}$$

where,

$$\mu_u = \mu_l = \frac{\lambda\Delta}{2(\Delta - E(l \mid \Delta))} \tag{B.2.18}$$

$$\nu_u = \nu_l = \frac{E(z \mid \Delta) + E(\psi \mid \Delta) - \lambda\Delta^2 + \mu_u E\{(\Delta - l)^2 \mid \Delta\}}{\Delta - E(l \mid \Delta)} \tag{B.2.19}$$

$$\xi_u = \sup_{1 \le d \le \Delta} (\phi(d)),\quad \xi_l = \inf_{1 \le d \le \Delta} (\phi(d)) \tag{B.2.20}$$

where,

$$\phi(d) = \frac{E(z \mid d) + E(\psi \mid d) + \mu_u(E(l^2 \mid d) - d^2 - (1+\lambda d)e^{-\lambda d}) - \nu_u(d - E(l \mid d) - (1+\lambda d)e^{-\lambda d})}{(1+\lambda d)e^{-\lambda d}}$$

Substituting $W^0_{d,\cdot}$ in the right hand side of (B.2.15)-(B.2.16), we obtain $W^1_{d,\cdot}$. We find,

$$W^1_{1,u} = E(z \mid 1) + E(\psi \mid 1) + \mu_u(E(l^2 \mid 1) - (1+\lambda)e^{-\lambda}) + \nu_u(E(l \mid 1) - (1+\lambda)e^{-\lambda}) + \xi_u(1 - (1+\lambda)e^{-\lambda})$$

$$W^1_{1,l} = W^1_{1,u} - (\xi_u - \xi_l)\,(1 - (1+\lambda)e^{-\lambda}) \tag{B.2.23}$$

Also,

$$G^1_{1,u} = E(l \mid 1) + \gamma_u(E(l \mid 1) - (1+\lambda)e^{-\lambda}) + \zeta_u(1 - (1+\lambda)e^{-\lambda}) \tag{B.2.24}$$

$$G^1_{1,l} = G^1_{1,u} - (\zeta_u - \zeta_l)(1 - (1+\lambda)e^{-\lambda}) \tag{B.2.25}$$

From the regenerative theorem [20], we have that

$$\mathbf{D}^l = \frac{W^1_{1,l}}{\lambda G^1_{1,u}} \le \mathbf{D} \le \frac{W^1_{1,u}}{\lambda G^1_{1,l}} = \mathbf{D}^u \tag{B.2.26}$$

In this Appendix we also show that the conditional expectations of the form $E(X \mid d)$ can be computed with high accuracy. Let us define,

$E(X \mid d,k)$: The conditional expectation of the random variable X, given that the arrival interval contains k packets, and has length d. Then,

$$E(X \mid d) = \sum_{k=0}^{\infty} E(X \mid d,k)\, e^{\ \lambda d} \frac{\lambda d^k}{k!} \tag{B.2.27}$$

The quantities $E(X \mid d,k)$ depend only on k. In Section B.1 we show that the quantities $L_k = E(l \mid d,k)$ can be computed recursively:

$$L_0 = L_1 = 1 \tag{B.2.28}$$

$$L_k = \frac{A^{(2)}_k}{1 - A^{(1)}_k} L_{k-1} + \frac{A^{(3)}_k}{1 - A^{(1)}_K}\ ;\ k \ge 2 \tag{B.2.29}$$

where $A^{(i)}_n$, i=1,2,3 can be computed recursively as follows:

$$A_n^{(1)}=[1-2^{-n}]^{-1}\, 2^{-n}\left\{ 1+\sum_{i=2}^{n-1} A_i^{(1)} \begin{bmatrix} n \\ i \end{bmatrix} \right\},\quad n\geq 3 \tag{B.2.30}$$

$$A_n^{(2)}=[1-2^{-n}]^{-1}\, 2^{-n}\left\{ n+\sum_{i=2}^{n-1} A_i^{(2)} \begin{bmatrix} n \\ i \end{bmatrix} \right\},\quad n\geq 3 \tag{B.2.31}$$

$$A_n^{(3)}=[1-2^{-n}]\left\{ 1+2^{-n}(n+1)+2^{-n}\sum_{(i=2)}^{n-1} A_i^{(3)} \begin{bmatrix} n \\ i \end{bmatrix} \right\},\quad n\geq 3 \tag{B.2.32}$$

$$A_2^{(1)}=1/3,\ A_2^{(2)}=2/3,\ A_2^{(3)}=7/3 \tag{B.2.33}$$

The quantities $Z_k = E(z \mid d,k)$, can be also computed recursively:

$$Z_0 = 0,\quad Z_1 = 1 \tag{B.2.34}$$

$$Z_k = \frac{A_k^{(5)}}{1-A_k^{(4)}} Z_{k-1} + \frac{A_k^{(6)}}{1-A_k^{(4)}}\ ;\ k \geq 2 \tag{B.2.35}$$

and the quantities $A_k^{(i)}$, $i = 4,5,6$ can be computed recursively as follows:

$A_n^{(4)}$; $n \geq 3$ satisfies the recursion in (B.2.30)

$A_n^{(5)}$; $n \geq 3$ satisfies the recursion in (B.2.31)

$$A_n^{(6)} = [1-2^{-n}]^{-1}\{1 + n2^{-n} + n^2 2^{-n} + 2^{-n}\sum_{i=2}^{n-1} \begin{bmatrix} n \\ i \end{bmatrix} A_i^{(6)}\}\ ,\quad n \geq 3 \tag{B.2.36}$$

$$A_2^{(4)} = \frac{1}{3},\ A_2^{(5)} = \frac{2}{3},\ A_2^{(6)} = \frac{14}{3} \tag{B.2.37}$$

The quantities $Y_k = E(l^2 \mid d,k)$ can be computed as follows:

$$Y_0 = 0,\quad Y_1 = 1 \tag{B.2.38}$$

$$Y_k = \frac{A_k^{(8)}}{1-A_k^{(7)}} Y_{k-1} + \frac{A_k^{(9)}}{1-A_k^{(7)}}\ ;\ k \geq 2 \tag{B.2.39}$$

and the quantities $A_k^{(i)}$, $i = 7, 8, 9$ can be computed recursively as follows:

$A_n^{(7)}$; $n \geq 3$ satisfies the recursion (B.2.30)

$A_n^{(8)}$; $n \geq 3$ satisfies the recursion (B.2.31)

$$A_n^{(9)} = [1-2^{-n}]^{-1}\{2L_n - 1 + 2^{-n}(1+2L_n) + n2^{-n}(1+L_{n-1}) + 2^{-n}\sum_{i=2}^{n-1} \begin{bmatrix} n \\ i \end{bmatrix} A_i^{(9)}\}\quad n \geq 3 \tag{B.2.40}$$

$$A_2^{(7)} = \frac{1}{3},\ A_2^{(8)} = \frac{2}{3},\ A_2^{(9)} = 16 \tag{B.2.41}$$

From the above formulas we see that a finite number, M, of terms from the infinite series in (B.2.27), can be computed. Also, for large k values, and based on the recursive expressions, simple upper and lower bounds on $E(X \mid d,k)$ can be developed. Those bounds can be used to tightly bound the sum

$$\sum_{k=M+1}^{\infty} E(X \mid d,k) e^{-\lambda d} \frac{(\lambda d)^k}{k!}$$

Remark It can be also proved that $E(\psi \mid d) = \dfrac{\lambda d^2}{2}$.

B.3 Interdeparture Distribution Analysis

We first give some definitions.

$l_{k,m}$: Given k packets with counter values equal to 1 and m packets with counter values equal to 2, the number of slots needed by the algorithm until the end of the first successful transmission, after the k-multiplicity collision has been observed.

$n_{k,s}$: The number of length s interdeparture intervals within a CRI wich starts with a k-multiplicity collision. If the length from the first slot of the CRI until the first successful transmission is equal to s, then this interval is included in the counting.

h_d: Starting with a CRP at which the lag equals d, $d \geq 1$, and which follows a successful transmission, the number of slots needed by the algorithm to reach the first CRP at which the lag is one, and which follows a slot containing a single transmission.

$m_{d,s}$: Starting with a CRP at which the lag equals d, $d \geq 1$, and which follows a successful transmission, the number of length s interdeparture intervals until the first CRP at which the lag is one, and which follows a slot containing a single transmission. If the length from the first slot of the CRI until the first successful transmission is equal to s, then this interval is included in the counting.

$P(k,l,\delta \mid d)$: Given an arrival interval of length d, the probability that there are k arrivals in it, that $l_{k,0}=\delta$, and that it takes l slots for its resolution.

$P_k(l)$: Given a k-multiplicity collision, the probability that it takes l slots for its resolution.

We also have:

$$H = E\{h_1\} \ , \ C = \lambda H$$

$$E\{\sum_{n=1}^{C_1} I_n(s)\} = E\{m_{1,s}\}$$

Recursions

From the operation of the algorithm we have the following recursions:

(I)

$$l_{1,m} = 0; \forall m \ , \ P(l_{k,m}=0)=0 \ ; \ \forall \ k \geq 2, \ \forall m$$

$$l_{0,m} = 1 + l_{m,0} \ , \ P(l_{0,m} = 1) = \begin{cases} 1 \ , \text{ if } m=1 \\ 0 \ , \text{ if } m \neq 1 \end{cases}$$

$$l_{k,m} = 1 + l_{i,m+k-i} \ ; \ w.p \ \binom{k}{i} 2^{-k}, \ k \geq 2$$

$$P(l_{k,m}=s)=\begin{cases}1 \text{ , if } k=1 \text{ and } s=0\\ P(l_{m,0}=s-1) \text{ , if } k=0,\ s\geq 1\\ 2^{-k}\sum_{i=0}^{k}\binom{k}{i} P\ (l_{i,\ m+k-i}=s-1), \text{ if } k\geq 2,\ s\geq 1\end{cases}$$

(II)

$$n_{1,s}=\begin{cases}1 \text{ , if } s=1\\ 0 \text{ , if } s\neq 1\end{cases}$$

$$k\geq 2\ ;\ n_{k,s}=\begin{cases}n_{k-1,\ s} & ,\ w.p.\ \ P(l_{k,0}\neq s-1)\\ 1+n_{k-1,s} & ,\ w.p.\ \ P(l_{k,0}=s-1)\end{cases}$$

$$N_{k,s}\overset{\Delta}{=}E\ \{n_{k,s}\}=\begin{cases}\sum_{i=2}^{k} P(l_{i,0}=s-1)\ \text{ , if } s>1\\ 1+\sum_{i=2}^{k} P(l_{i,0}=0)=1, \text{ if } s=1\end{cases}$$

(III)

$$k=0,1\ ;\ P_k(l)=\begin{cases}1 \text{ , if } l=1\\ 0 \text{ , } otherwise\end{cases},\ P_2(l)=P(l_{2,0}=l-2), \text{ for } l\geq 3$$

$$\left.\begin{array}{l}k\geq 2\\ l\geq k+2\end{array}\right\};\ P_k(l)=\sum_{s=1}^{l-k-1} P(l_{k,0}=s)\ P_{k-1}\ (l-s-1)$$

Therefore for λ denoting the Poisson intensity,

$$P(k,l,\rho\,|\,d)=e^{-\lambda d}\ \frac{(\lambda d)^k}{k!}\ P(l_{k,0}=\rho)P_{k-1}\ (l-\rho-1)$$

Recursions for h_d

The operation of the algorithm yields the following relationships:

$$d \leq \Delta \,;\; h_d = \begin{cases} 1 \;, \; w.p. \;\; \lambda d e^{-\lambda d} \\ 1+h_1 \;, \; w.p. \;\; e^{-\lambda d} \\ l+h_l \;, \; w.p. \;\; \sum_{k=2}^{\infty} e^{-\lambda d} \dfrac{(\lambda d)^k}{k!} P_k(l), \; l \geq 2 \end{cases}$$

$$d > \Delta \;;\; h_d = l + h_{d-\Delta+l} \;, \; w.p. \;\; \sum_{k=0}^{\infty} e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} P_k(l)$$

After taking expected values we obtain:

$$H_d \stackrel{\Delta}{=} E\{h_d\} = e^{-\lambda d} + E\{l \mid d\} + e^{-\lambda d} H_1 + \sum_{k \geq 2} \sum_{l \geq 2} e^{-\lambda d} \frac{(\lambda d)^k}{k!} P_k(l) H_l \;; \;\; d \leq \Delta$$

$$H_d = E\{l \mid \Delta\} + \sum_{k \geq 0} \sum_{l \geq 1} e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} P_k(l) H_{d-\Delta+l} \;;\; d > \Delta$$

where,

$$E\{l \mid d\} = \sum_{k \geq 0} \sum_{l \geq 1} e^{-\lambda d} \frac{(\lambda d)^k}{k!} P_k(l).l$$

Recursions for $m_{d,s}$

The operation of the algorithm yields the following relationships, where $\lfloor \; \rfloor$ denotes integer part, :

$$d \le \Delta;\ m_{d,s} = \begin{cases} n_{1,s}\ ;\ w.p.\ e^{-\lambda d}\lambda d \\ n_{k,s} + m_{l,s}\ ;\ w.p.\ e^{-\lambda d}\dfrac{(\lambda d)^k}{k!}P_k(l)\ ;\ k \ge 2 \\ n_{k-1,s} + m_{l,s}\ ;\ w.p.\ P(0|d) \displaystyle\sum_{\substack{n \ge 0 \\ n+\rho+2 \ne s}} P^n(0|1)P(k,l,\rho|1)\ ;\ k \ge 2 \\ 1 + n_{k-1,s} + m_{l,s}\ ;\ w.p.\ e^{-\lambda(d+1)}\dfrac{\lambda^k}{k!} \displaystyle\sum_{\substack{n \ge 0 \\ n+\rho+2 = s}} e^{-\lambda n}P(l_{k,0}=\rho)P_{k-1}(l-\rho-1); k \ge 2 \\ 1\ ;\ w.p.\ \lambda e^{-\lambda(d+s-1)}\ ;\ s \ge 2 \end{cases}$$

For $d > \Delta$:

$$m_{d,s} = n_{k,s} + m_{d-\Delta+l,s}\ ;\ w.p. e^{-\lambda\Delta}\frac{(\lambda\Delta)}{k!^k}P_k(l)\ ;\ k \ge 1$$

$$= n_{k-1,s} + m_{d-n(\Delta-1)+l,s}\ ;\ w.p. \sum_{\substack{\rho \ne s-n-1 \\ 1 \le n \le \lfloor \frac{d-\Delta}{\Delta-1} \rfloor}} e^{-\lambda\Delta(n+1)}\frac{(\lambda\Delta)^k}{k!}P(l_{k,0}=\rho)P_{k-1}(l-\rho-1),\ k \ge 1,\ \text{if } \left\lfloor \frac{d-\Delta}{\Delta-1} \right\rfloor \ge 1$$

$$= 1 + n_{k-1,s} + m_{d-n(\Delta-1)+l,s}\ ;\ w.p. \sum_{1 \le n \le \lfloor \frac{d-\Delta}{\Delta-1} \rfloor} e^{-\lambda\Delta(n+1)}\frac{(\lambda\Delta)^k}{k!}P(l_{k,0} = s-n-1)P_{k-1}(l-s+n),\ k \ge 1,$$

$$;if \left\lfloor \frac{d-\Delta}{\Delta-1} \right\rfloor \ge 1$$

$$n_{k-1,s} + m_{l,s}\ ;\ w.p.\ e^{-\lambda\left[d+\lfloor \frac{d-\Delta}{\Delta-1} \rfloor\right]}\frac{\left[\lambda(d-\lfloor \frac{d-\Delta}{\Delta-1} \rfloor(\Delta-1))\right]^k}{k!} \sum_{\lfloor \frac{d-\Delta}{\Delta-1} \rfloor+\rho+1 \ne s} P(l_{k,0}=\rho)P_{k-1}(l-\rho-1)\ ;\ k \ge 2$$

$$= 1 + n_{k-1,s} + m_{l,s}\ ;\ w.p.\ e^{-\lambda\left[d+\lfloor \frac{d-\Delta}{\Delta-1} \rfloor\right]}\frac{\left[\lambda\left[d-\lfloor \frac{d-\Delta}{\Delta-1} \rfloor(\Delta-1)\right]\right]^k}{k!}P\left[l_{k,0} = s-1-\left\lfloor \frac{d-\Delta}{\Delta-1} \right\rfloor\right].$$

$$.P_{k-1}\left[l-s+\left\lfloor \frac{d-\Delta}{\Delta-1} \right\rfloor\right]\ ;\ k \ge 2$$

$$n_{k-1,s} + m_{l,s}\ ;\ w.p. e^{-\lambda(d+\lfloor \frac{d-\Delta}{\Delta-1} \rfloor+1)} \sum_{\substack{n \ge 0 \\ n+2+\rho+\lfloor \frac{d-\Delta}{\Delta-1} \rfloor \ne s}} e^{-\lambda n}\frac{\lambda^k}{k!}P(l_{k,0}=\rho)P_{k-1}(l-\rho-1)\ ;\ k \ge 2$$

$$= 1 + n_{k-1,s} + m_{l,s}\ ;\ w.p. e^{-\lambda\left[d+\lfloor \frac{d-\Delta}{\Delta-1} \rfloor+1\right]} \sum_{n \ge 0} e^{-\lambda n}\frac{\lambda^k}{k!}P\left[l_{k,0} = s-n-2-\left\lfloor \frac{d-\Delta}{\Delta-1} \right\rfloor\right].$$

$$.P_{k-1}\left[l+n-s+\left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor +1\right] \quad ; k\geq 2$$

$$= 1 \ ; w.p. \lambda e^{-\lambda(d+s-1)} \ ; \text{if } s-2-\left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor \geq 0$$

$$= 1 \ ; w.p. \lambda\left[d-\left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor (\Delta-1)\right] e^{-\lambda\left[d+\left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor\right]} \ ; \text{if } s = 1+\left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor$$

Let us define,

$$U(x) \stackrel{\Delta}{=} \begin{cases} 1 \ , \ x \geq 0 \\ 0 \ , \ x < 0 \end{cases}$$

$$P_\delta(l) \stackrel{\Delta}{=} \sum_{k\geq 1} e^{-\delta} \frac{\delta^k}{k!} P_k(l)$$

$$N_{\delta,s} \stackrel{\Delta}{=} \sum_{k\geq 1} N_{k,s}\, e^{-\delta} \frac{\delta^k}{k!} \ , \ N_{\delta,1} = 1-e^{-\delta}$$

$$P_{\delta,\rho} \stackrel{\Delta}{=} \sum_{k\geq 1} e^{-\delta} \frac{\delta^k}{k!} P(l_{k,0} = \delta)$$

Using the above defined quantities, and the recursions on $m_{d,s}$, we find:

For $d \leq \Delta$:

$$M_{d,s} \stackrel{\Delta}{=} E\{m_{d,s}\} = N_{\lambda d,s} + \sum_{l\geq 3} M_{l,s}\left[P_{\lambda d}(l) + \frac{e^{-\lambda d}}{1-e^{-\lambda}} P_\lambda(l)\right] +$$

$$+ \frac{e^{-\lambda d}}{1-e^{-\lambda}}\left[N_{\lambda,s} - P_{\lambda,s-1}\right] + U(s-2)e^{-\lambda d}\left\{ e^{-\lambda(s-2)} \sum_{m=0}^{s-2} e^{\lambda m} P_{\lambda,m} + \right.$$

$$\left. + \lambda e^{-\lambda(s-1)} - \lambda \frac{e^{-\lambda}}{1-e^{-\lambda}}\right\}$$

For $d>\Delta$ and $p \stackrel{\Delta}{=} \left\lfloor \frac{d-\Delta}{\Delta-1}\right\rfloor$, $p = 0, 1, \ldots$:

$$M_{d,s} = N_{\lambda\Delta,s} + \sum_{l\geq 1} M_{d-\Delta+l,s}\, P_{\lambda\Delta}(l)$$

$$+ e^{-\lambda\Delta p}\left\{N_{\lambda(d+p-p\Delta),s} - P_{\lambda(d+p-p\Delta),s-1} + \sum_{l\geq 3} M_{l,s}\, P_{\lambda(d+p-p\Delta)}(l)\right\}$$

$$+ \frac{e^{-\lambda(d+p)}}{1-e^{-\lambda}}\left\{N_{\lambda,s} \quad - P_{\lambda,s-1} + \sum_{l\geq 3} M_{l,s}\, P_{\lambda}\,(l)\right\}$$

$$+ U(p-1)\left\{\frac{e^{-\lambda\Delta}(1-e^{-\lambda\Delta p})}{1-e^{-\lambda\Delta}}\left[N_{\lambda\Delta,s} - P_{\lambda\Delta,s-1}\right] + \sum_{1\leq n\leq p} e^{-\lambda\Delta n} \sum_{l\geq 1} M_{d-n(\Delta-1)+l,s}\, P_{\lambda\Delta}(l)\right\}$$

$$+ U(p-1)U(s-2)e^{-\lambda(s-1)} \sum_{m=s-1-min(p,\ s-1)}^{s-2} e^{\lambda m}\, P_{\lambda\Delta,\,m}$$

$$- U(s-2)\lambda e^{-\lambda(d+p)}\left[(d+p-p\Delta)e^{\lambda p\Delta} + \frac{e^{-\lambda}}{1-e^{-\lambda}}\right]$$

$$+ U(s-1-p)\, e^{-\lambda\Delta p}\, P_{\lambda(d+p-p\Delta),s-1-p}$$

$$+ U(s-2-p)\, e^{-\lambda(d+s-2)} \sum_{m=0}^{s-2-p} e^{\lambda m}\, P_{\lambda,m}$$

Bounds

For the numbers $N_{k,s}$, we used the following bounds:

$$0 \leq N_{k,s} \leq k-1 \ ; \ Vs$$

For the numbers H_d, we can prove that,

$$\alpha_l d + \beta_l \leq H_d \leq \alpha_u\, d + \beta_u \ , \ d \geq 1$$

where,

$$\alpha_l = \alpha_u = [\Delta - E\{l \mid \Delta\}]^{-1}\ E\{l \mid \Delta\}$$

$$\beta_l = \inf_{1\leq d\leq\Delta} Q(d) \ , \ \beta_u = \max\left[-\alpha_u\ ,\ \sup_{1\leq d\leq\Delta} Q(d)\right]$$

where:

$$Q(d) = \left[\lambda d e^{-\lambda d}\right]^{-1} \left\{ E\{l \mid d\} + \alpha_u \left[E\{l \mid d\} - d - \lambda d e^{-\lambda d} \right] \right\}$$

The following simple bound on $M_{d,s}$ has been used :

$$0 \le M_{d,s} \le H_d^u = \alpha_u \, d + \beta_u$$

APPENDIX C

Here, we provide recursive formulas for the computation of $P_q^{(m)}$. Clearly,

$$P_q^{(m)} = 0;\ q \le 0, \qquad P_q^{(1)} = \begin{cases} 1 & \text{if } q=1 \\ 0 & \text{otherwise} \end{cases} \tag{C.1}$$

Let l_n be the number of slots needed for the resolution of multiplicity $n \ge 0$ conflict. Then, from the operation of the algorithm we conclude that

$$\underset{m\ge 2}{P_q^{(m)}} = \begin{cases} P_{q-1}^{(n+1)} & w.p.\ \frac{1}{2}\binom{m-1}{n}\frac{1}{2^{m-1}} \\ P_{q-1-l}^{(m-n)} & w.p.\ \frac{1}{2}\binom{m-1}{n}\frac{1}{2^{m-1}}\, p\,(l_n = l) \end{cases} \tag{C.2}$$

The upper part of (C.2) is derived by considering the event that the packet under consideration, together with n of the rest m-1 packets retransmit immediately after the initial collision. The lower part of (C.2) is derived by considering the event that the packet under consideration does not transmit immediately, while the n of the rest m-1 packets retransmit immediately after the initial collision and it takes l number of slots to resolve a collision of multiplicity n. The probabilities $p\,(l_n = l)$ can be computed by similar reasoning. Averaging in (C.2), we finally have the following recursive formulas for $P_q^{(m)}$ for $m \ge 2$.

$$P_q^{(m)} = 2^{-m}\sum_{n=0}^{m-1} P_{q-1}^{(n+1)}\binom{m-1}{n} + 2^{-m}\sum_{n=0}^{m-1}\sum_{l=1}^{q-2}\binom{m-1}{n} P_{q-1-l}^{(m-n)}\, p\,(l_n = l) \tag{C.3}$$

APPENDIX D

We introduce a simpler notation. Given $\{\lambda_j\}_{1\le j\le 3}$, we define:

$$x_i \stackrel{\Delta}{=} (\lambda_i+\lambda_3)\Delta \;,\; i=1,2$$

$$y \stackrel{\Delta}{=} (\lambda_3\Delta)/2 \tag{D.1}$$

$$\mu_i \stackrel{\Delta}{=} \lambda_i+\lambda_3 \;,\; i=1,2$$

i=1,2, ; $P_i(e,m)$: The probability that $T_n^{(s)}+e$ is the m-th collision resolution point after $T_n^{(s)}$, for the RAA in cluster i. Note that m=1 refers to the end of the first CRI, which starts at $T_n^{(s)}$.

i=1,2 ; $P^{(i)}(e,m)$: The probability that $T_{n+1}^{(s)} - T_n^{(s)}$ = e, and there are m CRIs for the algorithm in cluster i, in $[T_n^{(s)}, T_{n+1}^{(s)}]$.

i=1,2 ; $P^{(i)}(m)$: The probability that there are m CRIs for the algorithm in cluster i, in $[T_n^{(s)}, T_{n+1}^{(s)}]$.

l_x: The length of a CRI, when it resolves all the packets that have arrived during an interval of length d. The number of arrivals per slot is Poisson distributed with parameter λ, and $\lambda d=x$.

$0\le n\le k$; $l_{n,k-n}$: The number of slots needed by either one of the two RAAs in the system to transmit k packets, when n of them have counter values equal to one, and the remaining k-n packets have counter values equal to two.

$\left.\begin{matrix} 0 \le l \le e-1 \\ i=1,2 \end{matrix}\right\}$; $P_i(e,m;l,n)$:

The probability that: $T_n^{(s)}+e$ is the m-th collision resolution point after $T_n^{(s)}$, for the RAA in cluster i, $T_{n+1}^{(s)} - T_n^{(s)} > e$, and the last before $T_n^{(s)}+e$ collision resolution point for the RAA in the other cluster occurs at $T_n^{(s)}+l$, and is the n-th such point after $T_n^{(s)}$. Observe that $T_n^{(s)}+e$ cannot be a collision resolution point for both RAAs since we then should have $T_{n+1}^{(s)} = T_n^{(s)} + e$.

$i, = 1,2$; $P_{i,i_c}(e,m;e,n)$:

$i_c = \begin{matrix} 1, \text{ if } i=2 \\ 2, \text{ if } i=1 \end{matrix}$

The probability that: $T_{n+1}^{(s)} = T_n^{(s)}+e$, $T_{n+1}^{(s)}$ is the m-th collision resolution point after $T_n^{(s)}$ for the algorithm in cluster i, and it is the n-th such point after $T_n^{(s)}$, for the algorithm in cluster i_c.

From the operation of the adopted RAA we conclude the following:

$$P(l_{1,k-1} = m) = P(l_{k-1,0} = m-1)$$

$$P(l_{0,k}=m) = P(l_{k,0} = m-1)$$

$$P(l_{1,0} = 1) = P(l_{0,0} = 1) = P(l_{0,1} = 2) = P(l_{1,1} = 2) = 1$$

$$\left.\begin{matrix} k \ge 2 \\ m \ge k+1 \end{matrix}\right\} ; P(l_{2,k-2}=m) = 2^{-2}\left\{ P(l_{k,0}=m-2)+P(l_{2,k-2}=m-1)+2P(l_{k-1,0}) = m-2) \right\}$$

$$\left.\begin{array}{l} k\geq n \\ n\geq 3 \\ m\geq 2k-1 \end{array}\right\} ; P(l_{n,k-n}=m)=2^{-n}\left\{ P(l_{k,0}=m-2)+P(l_{n,k-n}=m-1)+nP(l_{k-1,0}=m-2)+ \right.$$

$$\left. +\sum_{i=2}^{n-1}\binom{n}{i} P(l_{i,k-i}=m-1)\right\} \tag{D.2}$$

Taking expectations with respect to the number of packets in the examined interval we obtain, where $\lfloor\,\rfloor$ denotes integer part,:

$$P(l_x=m)=\sum_{0\leq k\leq\lfloor\frac{m+1}{2}\rfloor} e^{-x}\frac{x^k}{k!}P(l_{k,0}=m) \tag{D.3}$$

Furthermore, it can be proved that the following recursive expressions are true:

$$\left.\begin{array}{r} i=1,2 \\ e\geq m \end{array}\right\} ; P_i(e,m)=\begin{cases} P(l_{x_i-y}=e) \;;\; \text{if } m=1 \\ \sum\limits_{k=m-1}^{e-1} P_i(k,m-1)P(l_{x_i}=e-k) \;;\; \text{if } m\geq 2 \end{cases} \tag{D.4}$$

For,

$$i_c \overset{\Delta}{=} \begin{cases} 1\,,\ \text{if } i=2 \\ 2\,,\ \text{if } i=1 \end{cases} \quad U(x)\overset{\Delta}{=}\begin{cases} 1,\ \text{if } x\geq 0 \\ 0,\ \text{if } x<0 \end{cases} \tag{D.5}$$

we can also prove the following:

$$i=1,2,\ m\geq 1,\ e\geq m;\ P_i(e,m;0,0)=P_i(e,m)P(l_{x_{i_c}-y}>e) \tag{D.6}$$

$$i=1,2, m\geq 1, k\geq 1, e\geq m; P_i(e+k,1;e,m)=P_{i_c}(e,m;0,0)P(l_{x_{i_c}}>k)P(l_{x_i-y}=e+k)P^{-1}(l_{x_{i_c}-y}>e) \tag{D.7}$$

$$\left.\begin{array}{l} m\geq 2,\ e\geq m \\ i=1,2 \end{array}\right\} ; P_{i,i_c}(e,m;e,1)=P_i(e,m)P(l_{x_{i_c}-y}=e) \tag{D.8}$$

$$\left.\begin{array}{l} m\geq 2 \\ n\geq 1 \\ e\geq m \\ l\geq n \\ i=1,2 \end{array}\right\} ; P_i(e,m;l,n)=\sum_{k=1}^{\min(e-l-1,e-m+1)} P_i(e-k,m-1;l,n)P(l_{x_i}=k)P(l_{x_{i_c}}>e-l)P^{-1}(l_{x_{i_c}}>e-k-l)+$$

$$+U(l-m)\sum_{k=e-l+1}^{e+1-m} P_{i_c}(l,n;e-k,m-1)P(l_{x_i}=k)P(l_{x_{i_c}}>e-l)P^{-1}(l_{x_i}>l+k-e) \tag{D.9}$$

$$\left.\begin{matrix} m\geq 2 \\ n\geq 2 \end{matrix}\right\} ; P_{i,i_c}(e,m;e,n) = \sum_{l=n+1}^{e-1} \sum_{k=1}^{min(e-l-1,e+1-m)} P_i(e-k,m-1;l,n-1) P(l_{x_i}=k).$$

$$.P(l_{x_{i_c}}=e-l) P^{-1}(l_{x_{i_c}} > e-k-l) +$$

$$+ \sum_{l=m-1}^{e-1} \sum_{k=1}^{\min(e-l-1,e+1-n)} P_{i_c}(e-k,n-1;l,m-1) P(l_{x_{i_c}}=k) P(l_{x_i}=e-l) P^{-1}(l_{x_i} > e-k-l) \quad \text{(D.10)}$$

We therefore obtain:

$$P^{(i)}(e,m) = \sum_n P_{i,i_c}(e,m;e,n) \; ; \; i=1,2 \quad \text{(D.11)}$$

$$P^{(i)}(m) = \sum_{e\geq m} P^{(i)}(e,m) \; ; \; i=1,2 \quad \text{(D.12)}$$

The mean of the distribution in (D.12) is the quantity $E\{N_i(\{\lambda_j\}, \Delta)\}$ in (V.3).

DISTRIBUTION LIST

Copy No.	
1 - 6	Director Naval Research Laboratory Washington, D.C. 20375 Attention: Code 2627
7	Dr. R. N. Madan Electronics Division, Code 1114SE Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000
8	Mr. James G. Smith Office of Naval Research Code 1241 800 N. Quincy Street Arlington, VA 22217-5000
9	Professor Mike Athans MIT, Building 35 Cambridge, MA 02139
10	Professor A. Ephremides Electrical Engineering Dept. University of Maryland College Park, MD 20742
11	Dr. Dave Castanon ALPHATECH, Inc. 2 Burlington Executive Center 111 Middlesex Turnpike Burlington, MA 01803-4901
12	Dr. John P. Lehoczky Department of Statistics Carnegie Mellon University Schenley Park Pittsburgh, PA 15213-3890
13	Professor Alex Levis MIT, Building 35 Cambridge, MA 02139
14	Dr. Harold L. Hawkins Manager, Perceptual Sciences Code 1142 Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000

(Continued on Next Page)

15 Dr. G. Malecki
Office of Naval Research
Code 1142
800 N. Quincy Street
Arlington, VA 22217-5000

16 Dr. David Kleinman
ESE Department, U-157
The University of Connecticut
STORRS, CT 06268

17 - 28 Defense Technical Information Center, S47031
Building 5, Cameron Station
Alexandria, VA 22314

29 - 30 P. Papantoni-Kazakos, EE

31 M. Paterakis, EE

32 - 33 E. H. Pancake, Clark Hall

34 SEAS Publications Files

* Office of Naval Research Resident Representative
818 Connecticut Avenue, N.W.
Eighth Floor
Washington, D.C. 20006

Attention: Mr. Michael McCracken
Administrative Contracting Officer

* Send Cover Letter Only

JO#1747:jlb

UNIVERSITY OF VIRGINIA
School of Engineering and Applied Science

The University of Virginia's School of Engineering and Applied Science has an undergraduate enrollment of approximately 1,500 students with a graduate enrollment of approximately 560. There are 150 faculty members, a majority of whom conduct research in addition to teaching.

Research is a vital part of the educational program and interests parallel academic specialties. These range from the classical engineering disciplines of Chemical, Civil, Electrical, and Mechanical and Aerospace to newer, more specialized fields of Biomedical Engineering, Systems Engineering, Materials Science, Nuclear Engineering and Engineering Physics, Applied Mathematics and Computer Science. Within these disciplines there are well equipped laboratories for conducting highly specialized research. All departments offer the doctorate; Biomedical and Materials Science grant only graduate degrees. In addition, courses in the humanities are offered within the School.

The University of Virginia (which includes approximately 2,000 faculty and a total of full-time student enrollment of about 16,400), also offers professional degrees under the schools of Architecture, Law, Medicine, Nursing, Commerce, Business Administration, and Education. In addition, the College of Arts and Sciences houses departments of Mathematics, Physics, Chemistry and others relevant to the engineering research program. The School of Engineering and Applied Science is an integral part of this University community which provides opportunities for interdisciplinary work in pursuit of the basic goals of education, research, and public service.

DATE
ILMED
8