McGill University
Montreal

Access Protocols for the Multi-Access Packet Broadcast Channel with Long Propagation Delay

A dissertation submitted to the Faculty of Graduate Studies in partial fulfillment of the requirements for the degree of Master of Engineering.

Charles Meubus, B.Eng.

March 1979

Protocols for the Multi-Access Broadcast Channel

ABSTRACT

The delay-throughput performance of two types of distributed access protocols for the multi-access packet satellite channel is studied using simulation. The first category of protocols studied consists of tree-search-based schemes in both static and adaptive versions. These are variants of a scheme due to Capetenakis. Two main improvements on the basic scheme are identified: allowing newly arrived packets to participate in conflict resolution and using global queue management. The second type is an instantaneous throughput maximizing protocol. In each slot, a broadcast group is selected based on Bayesian estimates of terminal-occupancy probabilities. In its present form, this scheme does not perform as well as adaptive tree search schemes, but it does provide a novel framework for protocol design.

SOMMAIRE

La relation entre le délai et le débit pour deux types de protocoles d'accès décentralisés est étudié par simulation. Ces protocoles sont destinés à l'usage dans les systemes à entrée banalisée de transmission de paquets par satellite. La première classe de protocole est fondée sur des schémas de recherches arborescentes en mode statique et adaptatif. Ces schémas sont des améliorations de la méthode due à Capetenakis. Les deux principales ameliorations apportées à la méthode fondamentale consistent à permettre aux paquets récemment arrivés de participer à la résolution de conflits, et à gérer une file d'attente commune. Les protocoles de la seconde categorie visent a maximaliser le débit instantané. Durant chaque intervalle de temps, le sous-ensemble d'utilisateurs qui peuvent émettre est determiné à partir d'un estimé Bayesien de la probabilité d'occupation des terminaux. Dans son etat actuel de developpement, cette méthode produit des résultats inferieurs à ceux obtenus en utilisant des méthodes adaptatives de recherche arborescente. Toutefois, son intérêt réside dans le fait qu'elle offre un cadre nouveau pour la conception de protocoles d'accès.

ACKNOWLEDGMENT

The authour wishes to express his sincere appreciation to Professor Michael Kaplan, under whose supervision this work was accomplished, for his constant encouragement and inspiring guidance.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Problem Definition

The problem addressed in this thesis is that of devising efficient distributed control procedures enabling a large homogeneous population of geographically dispersed, possibly mobile, user terminals, each generating bursty fixed-length packet traffic, to share a single satellite channel. The results and concepts derived apply more generally to a wide class of contention systems.

The problem is of current interest because of the increasing attractiveness of random access satellite packet switching in the design of large, geographically scattered computer communication subnets. Because of the size of the population served by a single satellite, a high degree of stochastic load averaging is attained, permitting efficient utilisation of the resource. The satellite solution circumvents topological design and routing problems arising in wire networks of similar capacity, since, by its nature, the satellite provides complete connectivity to all users with distance-independent performance. Furthermore, due to the use of smaller receiving dishes, economies of scale in semiconductor technologies, and the predicted availability of cheap payload capability in space (e.g. with the advent of the space shuttle), the cost of the associated communications equipment (ground stations and satellite) is decreasing at a faster rate than cost reductions can be achieved in the wire plant [1].

The satellite link (by assumption) is the sole medium for exchange of information among the terminals. It is a broadcast channel with a long round trip propagation delay: all terminals receive the echo of a channel usage after a large, fixed delay, R. It is also multi-access: at any time any number of terminals may have access rights to the channel and decide to broadcast. However, any attempt by two or more terminals to use the channel concurrently leads to mutual, destructive interference and an undecodeable channel output (termed a 'collision' of packets).

The control strategy consists of a rule which allows each terminal to decide at each instant of time whether or not to transmit. The success of a particular protocol is evaluated by examining the trade-off between the average delay incurred by a packet and throughput (probability of a successful transmission).

The design of an access protocol depends critically on the kind of information available to those contending for the resource. In the present case, the decision to allocate access rights to a terminal at time t must be based on the history of channel states prior to time t-d. Furthermore, the state of the channel, as observed by the terminal users after the requisite delay, is ternary: the channel may be idle, contain a valid packet, or contain a collision. The state of the system, on the other hand, consists of the number and identity of busy terminals in the population. Therefore, the record of past terminal observations (or some statistic thereof), supplemented perhaps by knowledge of the terminal population size and the parameters of the packet generation process, provides only partial information about the state of the system. It is the incompleteness of the system state information available to the independent decision-makers (the terminals) that makes the access-assignment problem non-trivial.

1.2 History

The distinguishing characteristic of terminal-to-computer traffic is its large peak-to-average data rate ratio. This was established early by Jackson et al

[2,3]. Their statistics, together with those of Kleinrock [5] indicating that 96% of ARPA network traffic consists of single packet messages, lend credibility to the modelling of the terminal packet generation process as a Bernoulli point process, and the use of packet delay, rather than message delay, as the performance measures for small users.

In dealing with such a population of low duty cycle users, an efficient protocol allocates channel capacity to a terminal only during a burst of activity. TDMA and FDM do not satisfy this requirement, since they allocate a fixed proportion of the channel capacity to each user.

The classical (circa 1970) approach to achieving the above objective in the context of the satellite problem is embodied in the ALOHA system [4]. In the ALOHA system, a terminal with a packet transmits immediately, then listens to the echo returning from the satellite transponder. In the event of a collision, the terminal retransmits its packet at a randomly selected instant of time, usually chosen from a uniform or geometric distribution. ALOHA is said to be slotted if packet transmissions begin at integer times, where the unit of measurement of time is the time required to transmit a single packet, and all terminals are synchronized. The capacity (maximum attainable throughput) of slotted ALOHA with an infinite user population has been estimated to be 1/e by approximating the steady state input to the channel, consisting of both new and retransmitted packets, by a Poisson process.

The delay-throughput-stability performance of ALOHA depends critically on the choice of retransmission strategy. Metcalfe [6] was one of the first to analyze this dependence in the infinite population case, and to suggest channel control procedures based on varying the retransmission distribution parameters according to the backlog of packets to be transmitted. Kleinrock and Lam [7,8,9] refined the model and extended the analysis, exhibiting the bistable behaviour of the finite population model. In addition to retransmision controls, they introduced channel stabilizing input controls, so as to refuse admission to a growing proportion of new packets as potential instabilities develop. Assuming knowledge of the number of impeded terminals at any time, they derived optimal stationary dynamic threshold control limit policies (using one of two possible transmission/retransmission parameters) by couching the problem in terms of Markov decision theory and applying Howard's policy iteration algorithm. For practical implementation, they proposed a heuristic procedure for estimating the number of blocked terminals, which, in simulation, showed small degradation in performance from the theoretical optimum.

Other types of access protocols, called 'reservation' systems, attempt to marry the conflict free nature of TDMA and the random multi-access feature of ALOHA. As a class, reservation schemes are primarily useful when dealing with a population of terminals generating multi-packet messages or having sufficient buffering to smooth the terminal demand process. Examples of such schemes include those introduced by Crowther, Roberts, and Rubin.

In the implicit reservation scheme proposed by Crowther et al [10], frames of at least one round trip duration are defined, and any terminal which has transmitted successfully in one frame has exclusive ownership of the same slot in subsequent frames. When a slot is found idle, then all terminals with packets and without reservations contend for it, ALOHA fashion.

In Roberts' scheme [11] an ALOHA reservation channel consisting of mini-slots coexists with an information channel of packet size slots. Terminals make reservations, ALOHA style, for one or more slots in the information channel. When a reservation is acknowledged , the channel switches to reserved mode and services all reservations received. When the queue of reservations is exhausted, the channel reverts to reservation mode.

In both schemes, the delay incurred by a message is underbounded by two round trip delays. The superior performance observed in the case of heavy loading (throughputs in excess of .27) is at the expense of poor performance in the small traffic case.

Other reservation schemes, described and analyzed by Rubin [12,13], are in the same vein. His Fixed Reservation Access Control discipline (FRAC)[12] employs a fixed periodic pattern consisting of a random access reservation slot followed by N-1 (reserved) service slots. An adaptive variant (DFRAC) is obtained by varying the parameter N according to an estimate of the traffic intensity. Asynchronous Reservation Demand Assignment (ARDA) schemes generate reservation slots according to traffic characteristics rather than in a fixed sequence as in FRAC or DFRAC. In the simplest variant, a reservation slot is created whenever an idle slot is available; in an extension of that scheme to heavy traffic, a lower limit on the frequency of reservation slot creation is enforced.

In Integrated Random Access schemes (IRAR) [13], a random access sub-channel for information packets coexists with a reservation sub-channel operated in one of the ways described above. Collisions occurring in random access information slots are rescheduled for transmission in the reservation sub-channel. The reservation sub-channel (reservation and service slots) has precedence over the background random access channel: a reservation slot is created as soon as a collision is sensed, and service slots are created immediately in response to reservation requests, precluding their use as random access slots. This last family of schemes provides a means of exploiting the excellent throughput-delay performance of ALOHA at low traffic levels while retaining the advantages of pure reservation schemes.

An entirely non-ALOHA inspired system was offered by Binder [14]. In his round-robin dynamic assignment system, each terminal is a privileged user of one slot in a frame. When a slot is left idle, a distributed control mechanism is invoked which allocates slots on a round-robin basis to terminals which appear in a table of terminals known to possess a packet, until the privileged user reclaims his slot by deliberately generating a conflict.

Most reservation schemes imply the distributed management of a global queue of either occupied slots [10], access reservations [11], or active users [14]. Different scheduling disciplines can be used, according to design objectives: FIFO [11], round-robin [14], HOL, or others. In principle, however, this queue could be centrally managed, for example, by a satellite with processing capability.[15]

Computation of the waiting time distribution for entries in the global queue constitutes the key to the analysis of reservation schemes. The behaviour of the reserved slot portion of the channel can be segregated from that of the ALOHA component by making suitable assumptions. Then, by means of this artifice, the analysis of performance of a reservation scheme reduces to the analysis of the global queue. Lam [16], in his recent analysis of Crowther's scheme, uses a device due to Ferguson [17], namely the assumption that the optimally controlled ALOHA produces successful transmissions at a constant rate, to enable him to describe the global queue as a generalized M/G/1 queue in which the first customer of each busy period receives exceptional service. Rubin makes the bolder assumption that the reservation slots are conflict free, enabling an arbitrary number of terminals to reserve service slots successsfully in the same reservation slot. The global queue may be described as an Er/G/1 queue. This leads to closed form expressions for the delay-throughput characteristic valid in the limit where reservation to information packet size ratio becomes small.

In more recent work, a new perspective on the satellite access problem has developed, motivating the research reported in this thesis. Kleinrock and Yemini [18] and Capetenakis [19] have identified the root problem in access protocol design as that of choosing, in each slot, a subgroup of the population such that some measure related to delay or throughput is extremized. This observation leads both to formulate access strategies which, in spirit, resemble each other very closely, and are, in turn, directly related to 'probing' strategies introduced by Hayes [20], in a more general context.

1.3 Scope

The present work stems, in particular, from that of Capetenakis [19]. We begin by identifying certain improvements to his scheme which retain the spirit of his approach, then gradually enlarge the scope of the search for better access protocols. In order not to be restricted by analytical complexity, we resort to simulation. The search remains constrained, nevertheless, to a limited category of protocols.

The protocols which are considered in this thesis have the following properties:

(1) They are deterministic.

In ALOHA type schemes a randomization experiment is carried out independently by each terminal in deciding when to transmit a collided packet and/or accept a new packet arrival. Such randomizing behaviour is antithetical to the objective of coordinating transmissions; it increases the amount of uncertainty in the system. In fact, it has been proven (for the case where the total number of backlogged packets is known to all users) [18] that optimal strategies are pure strategies (probabilities are chosen to be 0 or 1). (See also [21]). This result is analogous to a result familiar from Bayesian decision theory. The removal of uncertainty is achieved at a cost, namely the need for each terminal to be cognizant of its identity. Consequently, deterministic systems are less flexible in admitting changes in the terminal population.

(2) Information is acquired directly via the channel state process.

Reservation schemes attempt to reduce uncertainty by allowing terminals to communicate side information via a reservation channel. Their performance at low traffic intensities is dominated by the delays incurred in booking reservations. The ideal protocol permits each terminal to derive 'reservation' information from the channel state process, so as to eliminate explicit reservation overhead.

(3) They are fundamental.

We seek to study the problem of conflict resolution in its simplest context. Therefore, the protocols presented in this thesis could be applied to mixed systems; for example, for making reservations in a reservation scheme.

(4) They are tailored to the satellite problem.

The essential feature of the satellite problem is the presence of a long round trip delay. This fact must guide the design of the protocol.

(6) They exhibit superior delay-throughput performance.

Superior throughput-delay performance is sought. The desired delay-throughput curve should ,at least over the normal operating range, lie within the shaded region of Fig.1.1.1. It is well known that pure random access (ALOHA), and self-scheduled time-division-multiplexing (TDMA) are optimal in the limits of low and high traffic, respectively. The optimal curve (σ^*(D)on the graph) is unknown, except under restrictive assumptions; the purpose of the present work is to approach it.

In addition to average performance measures, extrema of these measures will be of interest. For example, it is desirable that the protocol have the property that delay be bounded, i.e., that individual packet delay can be guaranteed to be inferior to a known limit. The performance of any scheme should be evaluated by comparison with that of schemes which are optimal with respect to a criterion of interest.

Attention is given also to the following factors:

(1) complexity and ease of implementation

(2) non-parametric performance

The operation of the protocol should not depend on exact knowledge of system parameters.

(3) fairness

Protocols vary, even at equal delay-throughput performance, in the degree to which they are equitable in dealing with terminals. An example of equitable behaviour is the delivery of messages to the transponder in first-come-first-served order. It will be assumed in this thesis that fairness is a desirable property of a protocol. Fairness is not always desirable, however, since discriminatory behaviour suggests a subjacent priority structure, which can be useful in some applications.

1.4 Summary

The aim and scope of the present study have already been outlined. The organization of the thesis is as follows: Chapter 2 presents the model and discusses its validity as a representation of the physical entity. Chapter 3 considers tree searching as a means of organising the allocation of access rights in a random access system. The tree search is presented in its simplest context, R=0. In Chapter 4 we introduce the complication of long propagation delay, which is an intrinsic feature of the satellite problem, and suggest tree-search-based access schemes which cope with the problem. We show the delay-throughput performance obtained in simulation. In chapter 5 we present a locally optimimum, instantaneous throughput maximizing algorithm. Though its overall performance is not uniformly superior to that of previously examined schemes, it is, nevertheless, of interest as a novel approach to the access protocol design problem. Moreover, the causes for its poorer-than-anticipated performance can be traced to an independence assumption, and this may provide the key to possible improvements. Suggestions for further research are made in Chapter 6.

The main results of this thesis are:

(1) The development, without analysis, of new random access protocols for the satellite channel.

(2) The systematic comparison of the delay-throughput performance of these schemes using a simple common model.

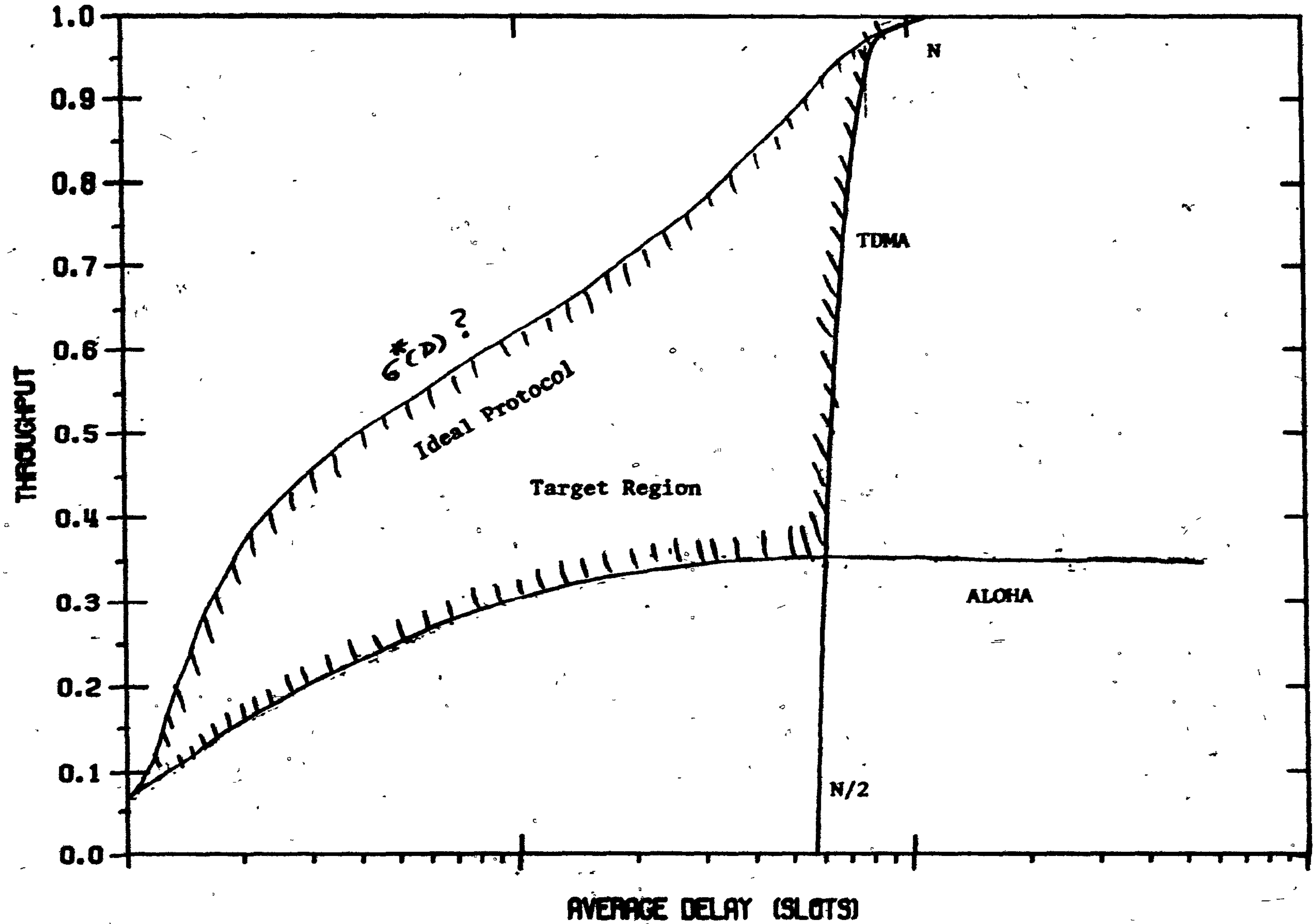

Fig.1.1.1 Region of interest in access protocol design.

CHAPTER 2
THE MODEL

2.1 Channel:

The channel operates in slotted time (with events occuring at integer times) and in perfect synchronization with the packet sources. These are themselves mutually synchronized and create unit length packets. All transmissions are received by the entire user community (including the originator), but subject to a fixed delay, R. An attempt by two or more users to transmit a packet in the same slot leads to a 'collision' and the destruction of the packets involved. Collided packets are rescheduled for transmission. The channel is error free.

There is evidence to support the adoption of this model as a representation of the physical satellite channel. The nominal bit-error rate for such a channel with a forward error correcting code has been estimated to be on the order of 10^{-9}, justifying the error-free assumption [22]. Similarly, synchronization of all sources to the satellite reference time to within very strict specifications has been found to be feasible [23]. Variations in the round trip delay, as a function of time and location of the user, due to drifting of the geosynchronous satellite from its nominal position and to differences in distance from user to satellite, can be assumed to be negligable from the point of view of protocol design and performance analysis.

On the other hand, the abstract model is a simplification, differing materially in some respects from the physical channel. For example, Metzner [24] and Roberts [25] have noted that, because of the capture effect in FM receivers, concurrent usages of the channel by stations with unequal power at the transmitter (resulting from differences in location or intentional inhomogeneity of terminal characteristics) need not lead to obliteration of all transmitted packets. Consequently, the results obtained using the above model are pessimistic.

2.2 Sources:

The user population is large (but finite), homogeneous (identical users), and of constant size. Each terminal generates packets acording to a Bernoulli process of parameter p, which is assumed time invariant. The amount of storage provided at each terminal varies according as the model is of the direct entry or buffered entry type. In the first case, when a packet is created, it fills a single transmit buffer and is held there until successfully transmitted; packets generated while the terminal buffer is occupied are lost. In the second case, there is, in addition to the single transmit buffer, a hold buffer, also of single packet capacity. Its purpose is to retain the first packet arriving at a terminal in the conflict resolution period following a collision (during which time all transmit buffers are serviced), until the end of that period, at which time its contents are transferred into the transmit buffer, forming the initial load for the next conflict resolution period.

The buffered entry model is introduced because of a useful property. Transfers from hold to transmit buffers are renewal points for the system state process. The sequence of time intervals between these events (sequence of cycle lengths) constitutes an imbedded Markov chain.

Either source model is a valid representation of some physical entities. An example is an interactive terminal with geometrically- distributed thinking time operating on a per-line basis (that is, the user does not begin to consider the next input until assured by receipt of a line feed of the successful entry of the current line). The time invariance of the model parameters is a reasonable assumption given the relative scale of slot time and the period over which fluctuations in user number or characteristics are significant.

2.3 Performance measures:

The performance measures of primary interest will be the average delay experienced by a packet from creation to successful transmission, and the throughput of the channel. The throughput of the channel is defined as the average number of successful transmissions per slot. The delay of a particular packet is computed as the difference between the index of the slot during which a packet is generated by a terminal, and the index of the slot during which it is succesfully transmitted. Note that this definition may differ by a constant amount R+1 from delay figures quoted by other authors, e.g. [7].

2.4 Numerical Constants:

The following numerical constants, relating to the ALOHA system, will be used for numerical examples and simulation throughout this thesis.

The channel round trip propogation delay will be assumed to be .270 sec, and the bit rate 50 KBPS. Assuming packet lengths (including source and destination address bits as well as error correcting bits) of 1125 bits, each time slot will be of 22.5 msec duration. Hence, expressed in slots, R=12.

CHAPTER 3

TREE SEARCH ACCESS SCHEMES

Yemini's urn scheme with window mechanism [18] and Capetenakis' algorithm [19] are examples of tree search access schemes. These schemes consist basically of a service mechanism for resolving conflicts in channel usage. Terminals in a population group to which the tree search scheme is applied broadcast all together until a conflict arises. A systematic procedure for resolving this conflict, the tree search, is then invoked.

3.1 Terminology:

A tree graph is shown in Fig 3.1.1. The tree consists of nodes and branches. From every node n in the tree, except terminal nodes, emanate dn branches, where dn is the degree of the node. These branches join the progenitor node n to its progeny. Each generation of nodes constitutes a level of the tree. The root node is at level 0; the terminal nodes are at ground level. Every node is the root node for the subtree consisting of its progeny and subsequent generations thereof, and is said to subtend or span that subtree. The depth of a subtree is the number of generations included. A terminal node subtends a subtree of depth 0. A symmetric tree is one of constant nodal degree. An n-ary tree, with no further qualification , is a tree in which all nodes have degree n.

3.2 Source Addressing:

The specification of a source address can be represented as the selection of a path through a tree. Imagine an M-ary tree with as many terminal nodes as there are users (N), with the terminal nodes assigned an address from 0 to N. In base M representation, the digits of the address of a particular terminal node specify a path from the root node to that terminal node. Any n-length prefix of the address corresponds to a unique path from the root node to a node at level n which subtends a subtree containing the terminal. Consecutive nodes in the path prescribed by the address digits thus specify subgroups of decreasing size (address groups), each including the addressed terminal node. In subsequent discussions, a statement to the effect that a node 'broadcasts', or is 'probed', should be interpreted as saying that the members of an address group defined by that node have broadcast.

When N does not bear the fortuitous relation to M permitting full occupancy of ground level nodes, or when N varies (terminals join or leave the terminal population), there arises the issue of how to allocate terminal addresses. Addresses should be assigned to new terminals (in a tree of sufficient depth to accommodate the maximum anticipated N, say N') in such a manner as to minimize the number of digits needed to specify a subtree which contains only that terminal. For the binary tree, for example, a rule [26] yielding an appropriate addressing sequence is the following:

Assign to n'th arrival the address comp(rev(n)).

Here rev is the operator which reverses the bits of the binary representation of n (for example, rev(1011)=1101), and comp is the complementation operator. The order of address assignment for the case N'=4 is shown in Fig.3.2.1. Generalization to non-binary trees is obvious.

3.3 General tree search:

A tree search access scheme can be described generally as follows. Each terminal possesses a copy of a tree graph representing a chosen subgroup of the population. The sequence of channel idle, busy, and collision states guides each terminal in following the excursions of a control pointer, initially positioned on the root node, through the tree. The position of the control pointer at each step defines a broadcast group, that is, a group of terminals given access rights to the next slot. This broadcast group consists of terminals associated with terminal nodes in the subtree spanned by the current node. Conflicts in channel usage resulting from subgroup broadcasts are handled according to the principle of 'divide and conquer'; that is, the control cursor moves down into the subtree. Absence of conflict at a node eliminates the subtended tree from further consideration (terminals in that subtree have no packets at this time), and the control pointer moves on to a node not yet explored in the current excursion. When all nodes have been processed, either through elimination or broadcast, the control pointer returns to the root node. The interval (in slots) between visits to the root node is called a cycle.

Tree search schemes differ in the path followed by the control pointer as it traverses the tree. They all share the feature that the movements of the cursor are deterministic, given the history of channel slot occupancy, so that all the terminals know, with certainty, the position of the cursor.

3.4 Classification:

A tree search can be static or dynamic. A static algorithm is one in which the degree of each node in the tree is fixed. A dynamic algorithm is one in which the degrees of the nodes are updated, in a manner known to all terminals, according to some statistic of the channel process so as to adapt the tree structure to the traffic.

There are two basic tree traversal patterns: ground seeking and cross level. In ground seeking, the control pointer proceeds towards ground level without respite; in cross level, previously unexplored nodes at the same level are examined before probing the progeny of a conflict nodes. The choice of one or the other method has no impact on the cycle time, but it does affect the delay distribution.

3.5 Example

The operation and implementation of a tree search scheme can be illustrated by a simple example. For the sake of clarity in presentation, it is assumed that packets generated during conflict resolution are held in hold buffers until the end of the cycle, at which time the hold buffer contents are transferred to the transmit buffers and become the initial load for the next cycle (buffered entry model).

Consider a binary tree capable of servicing 16 sources.

The course of one cycle, in the static case, is shown in Fig.3.5.1 in terms of displacements of the control cursor within the tree and the corresponding channel states at each step. The tree traversal pattern is left-oriented and ground seeking. The cycle under consideration begins (Step (1)) when the root node broadcasts (time to), and a conflict, due to the presence of packets at the square terminal nodes, is sensed. Control then passes (Step (2)) to the left offspring. That node broadcasts, the conflict persists, and control passes (in Step (3) to left offspring at the next generation. At this point, none of the terminals belonging to the broadcast group defined by the current node has a packet, and control passes (Step (4)) to the sibling of the current node. Again a conflict occurs, and the control cursor presses on, examining the left offspring. This action releases a packet in Step (5). Step (6) is a lateral move right (to the right sibling), and produces another successful broadcast. The last node whose right offspring has not been examined is the root node, and control passes to that offspring in Step (7). Conflict resolution proceeds in this fashion until Step (13), at which point all conflicts existing at time to are resolved.

A cross-level version of the algorithm is as illustrated in Fig.3.5.2. In Fig.3.5.3 the channel states in successive slots (R=0) for cross-level and ground seeking patterns are juxtaposed to illustrate the impact of the choice of tree traversal pattern on delay distribution. When R=0, ground seeking patterns are always preferable.

The ground seeking protocol may be described more succinctly in terms of operations on a push-down pop-up stack (STACK) containing node identifiers. Let RS(NODE) and LS(NODE) denote the right and left son of NODE, respectively. The following pseudocode describes the protocol:

Protocol A

```
1. If stack empty, place root node in stack.
2. NODE <-- STACK(TOP)
3. While conflict sensed at NODE, DO
   STACK <-- RS(NODE), NODE <-- LS(NODE)
   Else, go to 1
```

This sequence is executed in each slot by each terminal. Exactly one node is probed per slot (R=0).

A slight improvement of the algorithm is obtained by omitting steps that are uninformative [27]. These occur when a left probe following a conflict produces an idle slot. In that case, there is no need to probe the right sibling, since a conflict will ensue with certainty. Therefore, results of a left probe (as in Protocol A) ought to be handled differently from those of a right probe. This consideration leads to the folllowing look-ahead protocol:

Protocol B

```
1. NODE <-- ROOT_NODE
2. If conflict at NODE, Then DO
            STACK <-- RS(NODE)
            NODE  <-- LS(NODE)
            go to 2
   If stack empty, go to 1
   If idle at NODE, Then DO
            SKIP  <-- STACK(TOP)
            NODE  <-- LS(SKIP)
            STACK <-- RS(SKIP)
            go to 2
   IF success at NODE, Then NODE <-- STACK(TOP)
3. If conflict at NODE, Then DO
            NODE <-- LS(NODE)
            STACK <-- RS(NODE)
            go to 2
   Else, DO
```

```
If stack_empty,go to 1
NODE <-- STACK(TOP)
go to 3
```

In the above, step (2) treats left probes, while step (3) treats right probes. Both protocols A and B are strictly serial (ground-seeking) binary searches (SSBS).

The static procedure in the previous paragraph works well when the tree is lightly loaded; however, in the heavily loaded case excessive overhead is incurred visiting non-terminal nodes whose subtrees have a high probability of including more than one busy terminal. In the limit of full terminal node occupancy, 2N-1 nodes are examined to release N terminals, yielding a maximum throughput of N/(2N-1). A dynamic scheme can be used to combat this excessive overhead.

Given the number of busy terminals (b) in the population, the probability of a successful transmission at a node which subtends k terminals is

$$\frac{k\binom{N-k}{b-1}}{\binom{N}{b}}$$

This quantity is maximized by choosing $k = \lfloor N/b \rfloor$.

This observation can be applied to the design of dynamic variants of the algorithms in this example. These variants, which might be called "level skipping" algorithms, could consist, for example, in modifying step 1 so as to initialize the stack when empty with all nodes at level 1+ logsub2(b). The quantity b may be estimated by E(b|Lo), where Lo is the length of the previous cycle. Clearly,

$$E(D|Lo) = N\,(1 - (1-p)^{Lo})$$

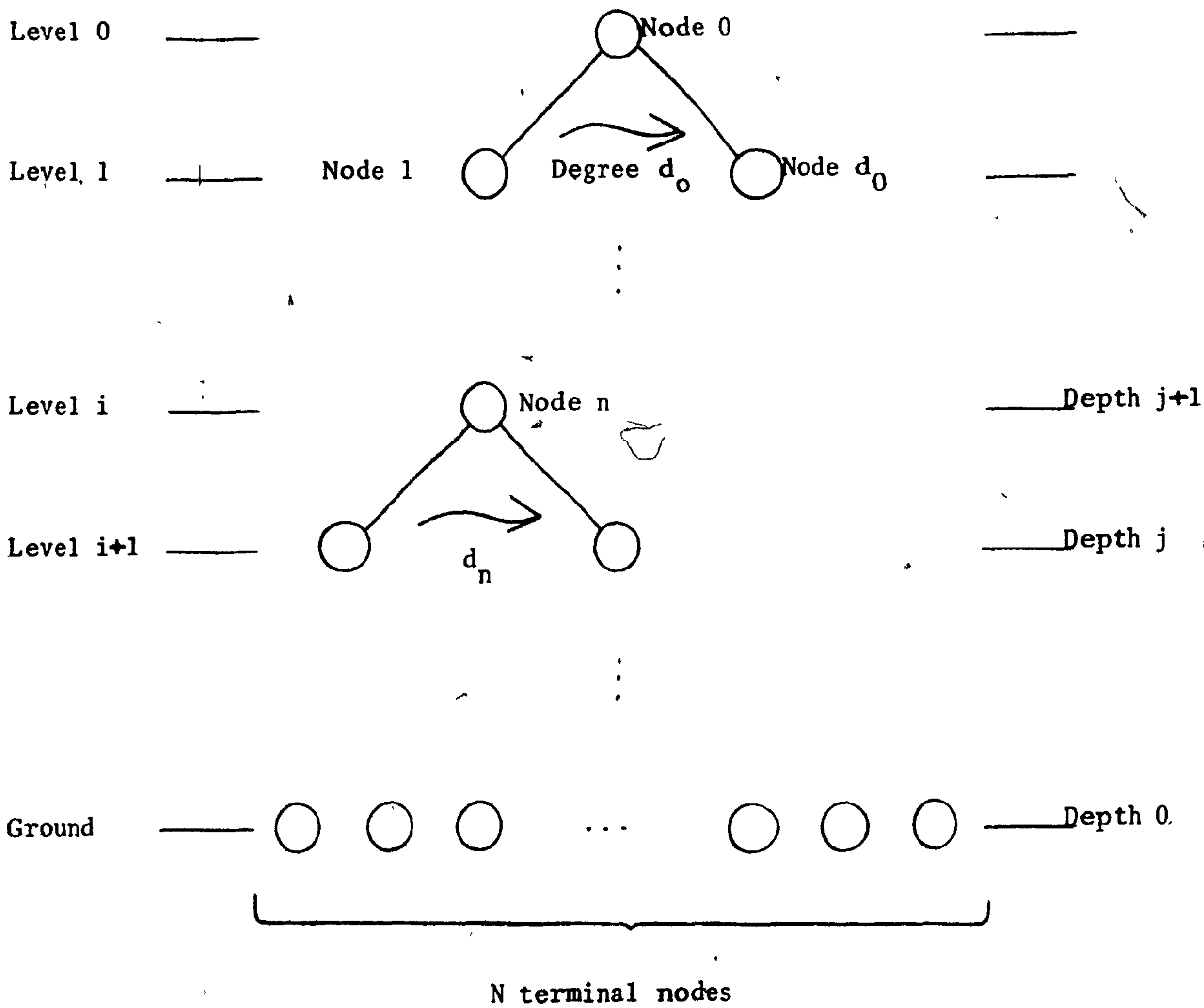
Level 0
Node 0
Level 1
Node 1
Degree d_0
Node d_0
Level i
Node n
Depth j+1
Level i+1
d_n
Depth j
Ground
Depth 0
N terminal nodes

Fig.3.1.1 Tree

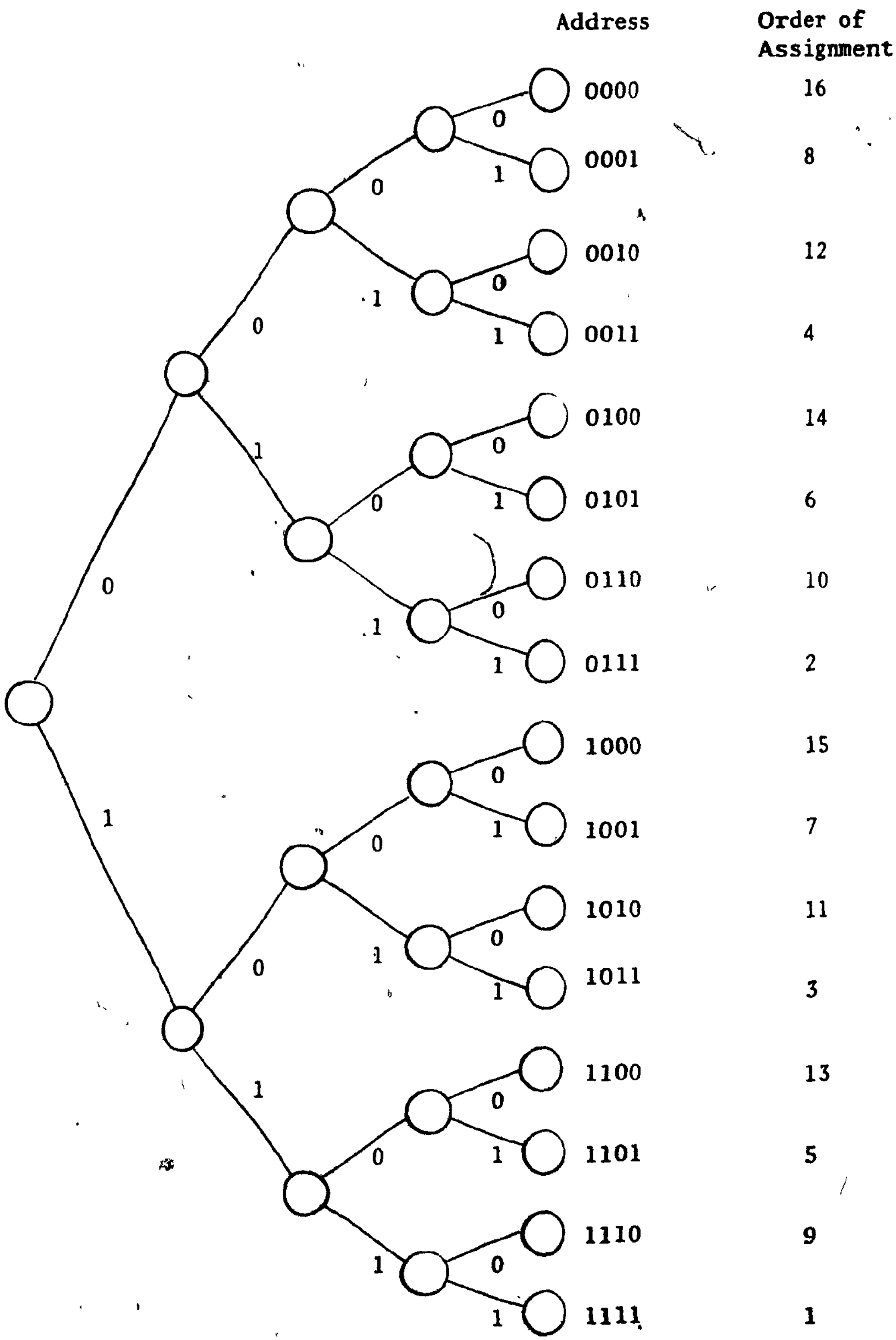

Fig.3.2.1 Source Addressing Using a Binary Tree

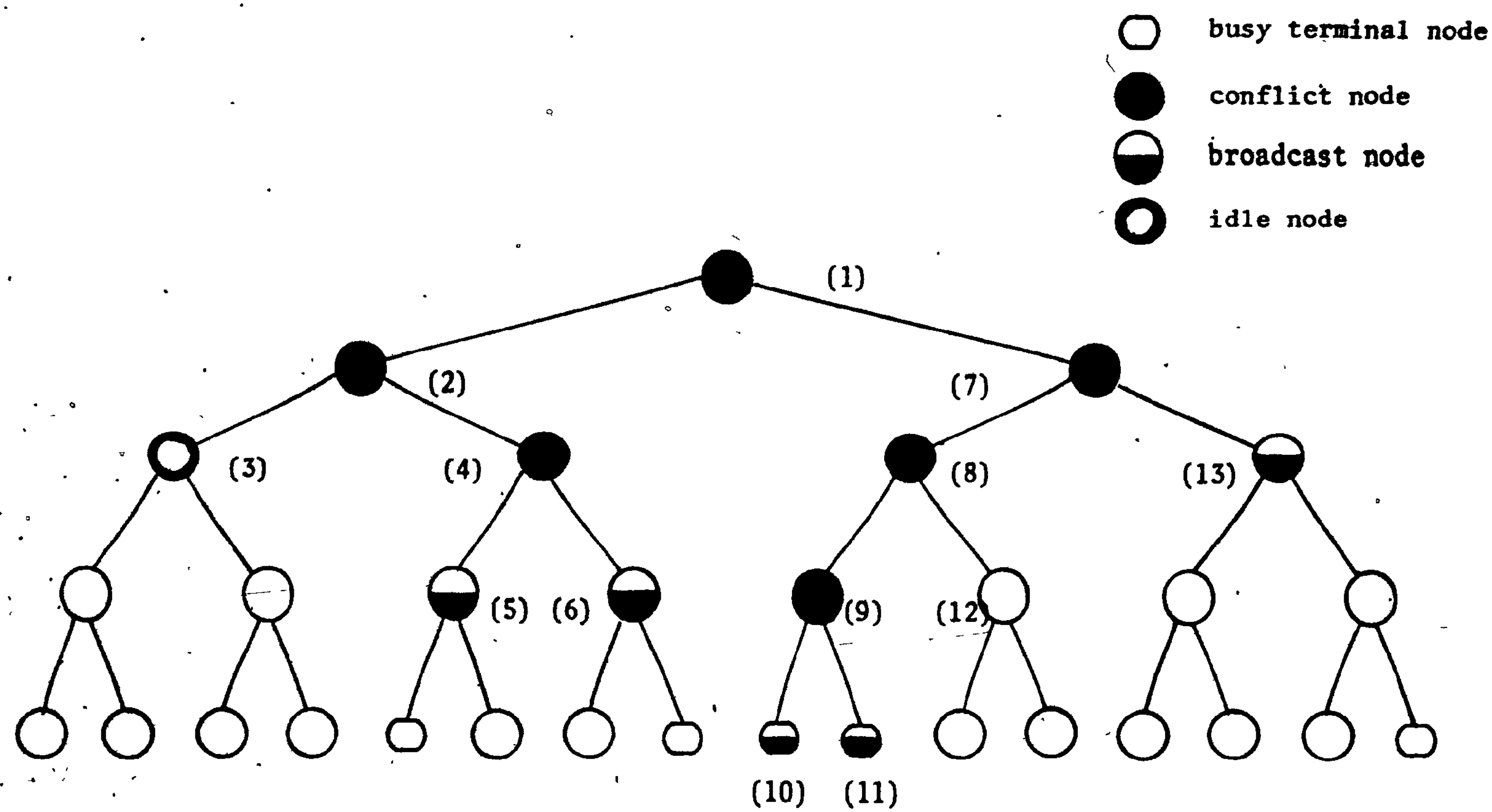

Fig.3.5.1 Ground-Seeking Binary Tree Search

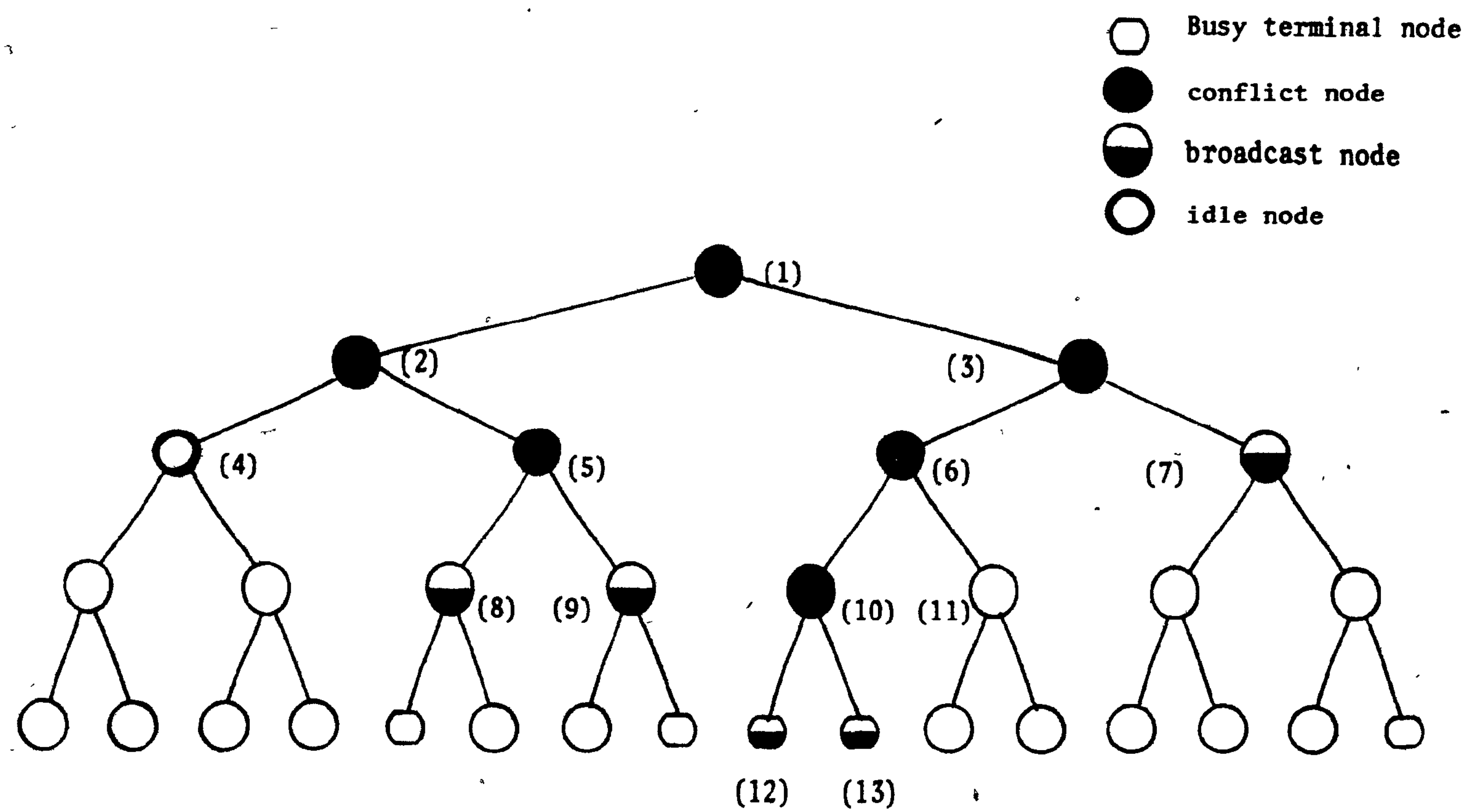

Fig. 3.5.2 Cross-Level Binary Tree Search

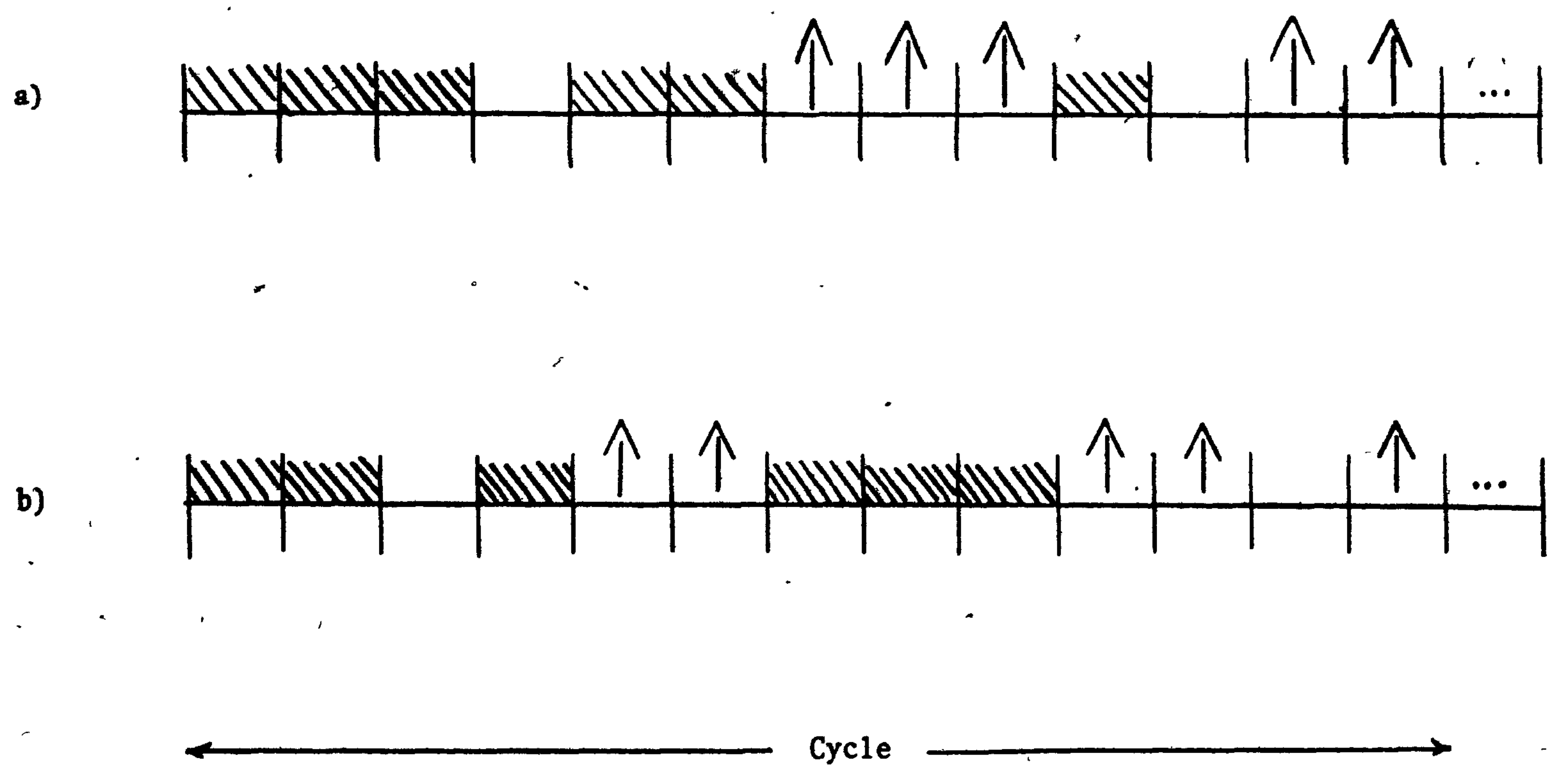

Fig.3.5.3 Channel States Resulting from Use of Cross-level (a) and Ground-Seeking (b) Probes.

CHAPTER 4

APPLICATION OF TREE SEARCH ACCESS SCHEMES

The tree search access schemes of chapter 3 are, in themselves, adequate as access protocols in the special case R=0. For application to the satellite problem, the elaboration of a tree search-access-based protocol requires an additional level to deal with the non-zero propagation delay. The purpose of this new level is to coordinate the concurrent conduct of several tree searches.

The point of departure for the work in this chapter is the tree-search-based protocol due to Capetenakis. His scheme is examined first. The remainder of the chapter deals with several simple extensions. A description of these variants is followed by a discussion of their relative merits, in the light of simulation results.

4.1 Capetenakis' Scheme

In Capetenakis' scheme (Fig.4.1.1), the time axis is divided into frames of R/2+1 di-slots (a pair of consecutive slots). To each di-slot is assigned an equal, fixed-size subgroup of the population (size Ng), and a binary tree search conflict resolution algorithm is applied in successive periodic occurrences of dislots belonging to that group. The execution of the tree search in each subgroup is independent of that in other groups.

Certain features of Capetenakis' scheme serve to simplify the analysis. Subdivision of the population into disjoint, non-communicating groups reduces the problem to the solution of the protocol design problem for the case R=0. The performance of the system is that of the subgroup, and analysis can focus exclusively on a particular subgroup. In addition, in both the static and dynamic versions, the buffered source model is assumed. As will be seen subsequently, this approach facilitates analysis at the expense of efficiency.

4.1.1 Static Case:

The pseudocode for the static tree search employed in every dislot is

Protocol C

```
1. If stack empty, STACK <-- ROOT_NODE
2. NODE <-- STACK(top)
3. DO for NXT=RS(NODE),LS(NODE);
     If conflict sensed at NXT, STACK <-- NXT;
```

Else;

4. Go to 1

An algorithmic step (the execution of the four step sequence above) requires R+2 slots in real time. The progeny of the current node are probed in the first two slots, and the resultant channel states are known R slots later. R/2+1 programs of this sort execute concurrently. Note that the stack contents, in this case, are different from those in protocols A or B (Section 3.5). Here, the stack contains only nodes which are certified to be conflict nodes, further action on them being held in abeyance until other conflicts are resolved. In protocols A and B, the stack contains nodes which are unresolved, being present merely for the purpose of pointing to unterminated search stubs.

Protocol C exploits the fact that, at any step in the tree search, independent information about two nodes (the progeny of NODE) can always be obtained in the same algorithmic step. (The root node is not explicitly probed.) When the length of the algorithmic step is dominated by the value of R, this is a crucial feature.

In spite of the simplicity of the model and protocol description, only relatively loose bounds on throughput-delay performance are available, exact results being obtained only in trivial limits of parameter values. It is easily shown (Appendix A.1) that the delay-throughput curve is bounded above on the ordinate (average delay) by (3Ng+log Ng-5)(R/2+1) slots, and on the abscissa (throughput) by Ng/(2Ng-2). The maximum delay which a packet can experience is (4Ng-5)(R/2+1). Other bounds, valid over the entire delay-throughput plane, are derived in [19].

4.1.2 Dynamic Case

One seeks to produce an algorithm which, on the basis of the history of the channel state process, varies the tree structure so as to minimize mean packet delay. The problem is unsolved in its general formulation. Certain assumptions can be made, however, to render the problem amenable to solution. If the choice of a tree is made at the beginning of each cycle, and that tree is maintained throughout the cycle, then it is reasonable to restrict the search for the optimal tree to the set of symmetric trees. Moreover, one can assume that a certain inference made in the infinite population case applies to the finite population case. Subject to these assumptions, Capetenakis has shown that the optimal tree (with respect to a criterion related to mean delay) is binary, except for the root node, which is of variable degree, depending on the traffic intensity. In the next few paragraphs, we outline the derivation of this result and interpret it.

In the infinite population case (N tends to infinity while the product Np tends to λ, a constant), the tree which minimizes the expected length of the current cycle (L), given the length of the previous cycle (Lo), $E(L|Lo)$, is binary except for the root node. (Recall that this development applies to the buffered entry model, so that Lo is a sufficient statistic.)

Note that the use of $E(L|Lo)$ (effectively the mean maximum delay) as an optimality criterion is arbitrary; its use stems from ease of computation. One could just as well attempt to maximize conditional throughput in the cycle, E(probability of a successful transmissions per node visited |Lo). The quantity of paramount interest is $E(D|Lo)$, but no useful expression for it has been found.

Assume that the binary nature of the optimal tree in the infinite population case carries over to the finite population case, and that a 'level skipping' algorithm (cf section 3.5) is desired for ease of implementation; that is, the degree of the root node (K) is constrained to be a power (k) of two. The problem is to choose k so as to minimize

$$E(L|Lo) = 2^{k} + \sum_{i=1}^{n-k} 2^{k+i} \Phi(n-k-i+1,q)$$

where,

$n = \log_2 N$

$q = Pr\{\text{terminal is occupied in current cycle}\}$

$\quad = 1 - (1-p)^{Lo}$

$\Phi(j,q) = Pr\{\text{conflict at root node of subtree depth } j|q\}$

$\quad = 1 - (1-q)^{2^{j}} - 2^{j} q (1-q)^{2^{j}-1}$

The expression for $E(L|Lo)$, given above, follows from elementary considerations. At level i in the tree, nodes subtend subtrees of depth $n-(k-1)-i$. The probability of conflict at a node at level i is accordingly $\Phi(n-k-i+1,q)$, and the average number of conflict nodes at level i is thus $2^{k+i-1} \Phi(n-k-i+1,q)$. Each one of these conflicts entails that two more nodes are to be visited, so that the average contribution to L at level $i>0$ is $2^{k+i} \Phi(n-k-i+1,q)$.

Since k takes on integer values, there exists a range of q for which a particular choice of k is optimal. Let k^* be the optimal k, for a given q. Assume that there exists a quantity $\hat{q}(k,n)$ defined by

$$k^* = k \qquad \hat{q}(k,n) < q < \hat{q}(k+1,n)$$

then $\hat{q}$ satisfies

$$E(L|\hat{q},n,k-1) = E(L|\hat{q},n,k)$$

This implies that

$$\Phi(n-k+1,\hat{q}) = 1/2 ,$$

which is easily solved for $\hat{q}$ as a function of n-k (Fig.4.1.2).

The algebra shows that the optimal dynamic algorithm (where optimality is with respect to $E(L|L0)$) proceeds directly to the level where the average information obtained by probing a node is maximized. From the point of view of the length of a cycle, the channel state information is binary: either a node is a conflict node and the search proceeds through it to its progeny, or it is not, and the search goes no further in that subtree. Thus, the optimal algorithm, in search of maximum average information, begins probing at level m, where $\Phi(m,q)=1/2$, since there

$h(\Phi(m,q))=1$ ($h(.)$ is the entropy function for a binary source).

If the significant channel states are taken to be presence or absence of a successful transmission in a slot, then one has the prescription for throughput maximization by level skipping,

$$\max_m \Psi(m,q)$$

where,

$$\Psi(m,q) = 2^m q (1-q)^{2^m -1}$$

the solution of which is $m=i$ for $q \in (2^{-i}, 2^{-(i+1)}]$, $i=0,\ldots,n$.

Such a strategy performs worse than the cycle minimizing strategy, as we shall show by simulation (cf section 4.5.2), and it is introduced here primarily as a means of highlighting an interesting feature of Capetenakis' dynamic algorithm. Superficially, it might appear that liquidating as many packets as possible in the first level visited is a means to achieving cycle length minimization. Such a strategy, however, is myopic. A better strategy proceeds by initially devoting a few slots to learning, at the expense of instantaneous throughput, in order to dispatch packets more efficiently in subsequent slots.

4.1.3 Non-parametric Variant

The dynamic algorithm presented in the previous section requires knowledge of p. There is a computationally efficient estimator $\tilde{q}$ for q given by

$$\tilde{q} = Min((bo/Ng)(Lo\ /Loo),1)$$

Here bo is the number of packets successfully transmitted in the last cycle, Lo is its length, and Loo is the length of the cycle preceding the last. Note that, because of the dependence existing between successive cycles (a long cycle will tend to engender another long cycle), this estimate is preferable to one based on a long term average. The effect of the use of a long term average would be to smooth out the fluctuations to which one wishes the adaptation algorithm to react.

4.2 Direct Entry Variant

If new arrivals in any slot are permitted to participate in conflict resolution (direct entry model), superior throughput-delay performance is observed under simulation. The success of this variant hinges on the reduction of delay, rather than on the increase of throughput (at given p). In the steady state, the throughput and cycle length statistics should be - and this intuition is corroborated by simulation - identical in both the direct and buffered entry models.

It is also noteworthy that the direct entry protocols are inherently more equitable than the buffered entry protocols. In the buffered case, the grade of service perceived by a terminal varies according to both address and p. For moderate to high p values, the delay experienced by a terminal is least if that terminal is leftmost, and greatest if rightmost. Of course, a randomised allocation of addresses (for example, at the beginning of every cycle) could alleviate this problem. No such readdressing is required in the direct entry model, since the packet delay has the same distribution for each terminal.

4.2.1 Static Case

The analysis is complicated by the fact that successive cycle lengths do not constitute a Markov chain. The bounds on throughput versus delay advanced by Capetenakis have no obvious analogs in the present situation. We can show

(Appendix A.2) that at p=1, D=(2Ng-3)(R/2+1) slots, and, as before, σ =.5Ng/(Ng-1).

4.2.2 Dynamic Case

The modification of Capetenakis' dynamic algorithm is non-trivial. Each terminal now has a different q, depending on when it was last serviced. One approach, admittedly a sub-optimal one, is to use the estimate q=bo/Ng to determine the extent of level skipping. Simulation results suggest that this is a reasonable way to proceed (cf.4.5.2). At p=1, D=N and σ=1, since one obtains TDMA.

4.3 SSBS Variant

One could conceive of forming R+1 subgroups of the population, instead of R/2+1 as in Capetenakis' scheme. A strictly serial binary search (SSBS) can then be applied to each group. Such a modification permits the use of the more efficient look-ahead search scheme (protocol B of section 3.5). The benefit obtained is offset by the fact that the search is conducted on a smaller tree. It turns out, under simulation, that for the model parameters of interest (N=448,R=12), there is no clear superiority of such a scheme over Capetenakis'.

4.4 Family of Global Queue Variants

A guiding principle in the design for minimum average delay of queueing systems is that the server should never be idle while there exists a backlog of unfinished work in the system. Applied to the problem at hand, this means that in any slot, all conflicts known to exist should be resolved before exploring the possible existence of others. One notes that Capetenakis' scheme violates this principle; the TDMA component (the segregation of the user population into non-interfering sub-groups) implies that exactly one algorithmic step (corresponding to the servicing of one node) may be performed per group in each frame, though though there may be several conflict nodes extant in that group, and none in another.

What is suggested by the principle enunciated above is a scheme like that illustrated in Fig.4.4.1. Here, a work hopper contains a record of all nodes in the system where conflicts are known to exist, but upon which immediate action could not be undertaken. Since a type C search (operating in dislots) has the property of stacking conflict nodes, it is a natural choice for this application. Thus, we allow R/2+1 servers, each providing R+2 units of service. Exactly one server is free in any given dislot. In the first unit of service, a node is read from the hopper by the available server, the labels of the progeny nodes are generated, and the terminals included in each node broadcast. R slots later, the two echoes are received and the server consults a scheduling algorithm to return those nodes where conflicts have been sensed to the work hopper. The scheduling algorithm also determines the reinitialization procedure. The choice of the scheduling algorithm, which governs the ordering of nodes in the queue of unfinished work, defines a member of a family which we have called the family of global queue variants.

A particular implementation of the general scheme proposed in the previous paragraph is that of Fig.4.4.2. Here, a multi-queue (LIFO) network is tended by a cyclical server who exhausts all work in a queue before proceeding to the next. The server has zero transit time and suffers no time penalty in visiting empty queues. The queues contain the conflict nodes generated by a binary tree search. When an individual tree is exhausted it is reinitialized. Note that Capetenakis' algorithm can be described in the same way, with the difference that the server, instead of providing exhaustive service at each step of its itinerary, services one node per step, cycling through all queues, busy or not, in one frame.

The question of the choice of scheduling algorithm naturally arises. It is appealing to conjecture that the performance of the system in terms of average delay is independent of this choice. Although we have been unable to prove this hypothesis, there is, nonetheless, evidence, obtained by simulation, to support it. Should the hypothesis be true, fairness considerations may motivate the choice of scheduling algorithm.

4.5 Simulation

In this section, the schemes described above are compared by simulation. The performance of optimally controlled ALOHA and TDMA are introduced as benchmarks by which these schemes can be judged.

4.5.1 Procedure

Simulation programs are coded in FORTRAN, and use a thoroughly tested, assembler language, random variate generator package, SUPER-DUPER [28]. Each slot in which the protocol under study operates is simulated individually. New arrivals to the system are computed at the beginning of a slot and labelled as to time of origin. The processing prescribed by the particular protocol is carried out during the slot. At the end of each slot, successfully transmitted packets are cleared, and statistics updated. The simulation, though performed on a per slot basis, keeps track of cycles, and always proceeds for an integral number of cycles.

The choice of a stopping rule poses a problem. Formal stopping rules consistent with our lack of knowledge of the underlying distributions of the simulation results (rules based on a Normality assumption or, better still, on the weaker assumption that the Chebyshev inequality holds) indicate long simulation times for moderately high confidence levels (80% or so). As a result it seemed reasonable to apply an empirical method of choosing the simulation length.

The procedure used was this: Short term statistics, compiled over consecutive segments containing integral numbers of cycles, were examined in order to ascertain the extent of transients due to choice of initial conditions and the degree of statistical stability (stationarity). The portion visibly contaminated by transient effects was discarded, and the remaining results were averaged. The simulation was repeated for different choices of random number generator seeds. If the cluster of points in (σ,D) space corresponding to a simulation (choice of N,p, and protocol) seemed too large the length of the simulation was increased. This process was repeated until the spread of values along each axis did not exceed 1% of the sample mean.

4.5.2 Results

Our simulation studies of the systems described above are summarized in figures 4.5.1-4.5.5.

Simulation results are displayed in (σ,D) space with p as a parameter. The points shown are the centroids of the clusters referred to in section 4.5.1; the curves are cubic splines with the centroids used as knots in the spline fitting. Where a knot is not explicitly labelled with the corresponding value of the parameter p, that value of p is to be inferred from the labelling of a curve with identical symbol markings. In that case, the sequence of p values on the unlabelled curve, in order of increasing p, is exactly that of the labelled curve.

Fig 4.5.1 exhibits the target region for random access protocol design. A desirable operating characteristic for a tree search protocol should lie within an area bounded above by TDMA, and to the left by ALOHA. Obviously, not every point in that region is achievable. The curve for TDMA follows from analysis (Appendix A.2), whereas that for optimally controlled ALOHA follows from simulation. The simulation length is 5000 slots/ run.

Fig.4.5.2 compares the simulated performance of Capetenakis' scheme to that of the theoretical bounds given in [19] (R=0,N=64). The performance of the same scheme using direct entry (curve A) and combining direct entry with look-ahead (SSBS variant - curve B) are also shown.

Fig.4.5.3 and Fig.4.5.4 examine the behaviour of tree search schemes, both static and dynamic, for the buffered and direct entry models respectively, under the more interesting condition R=12. The performance curve for the SSBS variant is essentially co-located with that for the direct entry variant of Capetenakis for N=448 (R+2 sub-divisions), and, for that reason, is not shown explicitly.

The delay-throughput performance of the global queue variants (GQV) are investigated in Fig.4.5.5. Two GQV's are considered, in both static and dynamic modes, and compared with the static version of Capetenakis' scheme. In the first, a single work queue is maintained with a FIFO discipline. If, in any slot, there is no work in the queue, all sub-groups not represented in the system have their search trees reinitialized. The second is the cyclical service system of Fig.4.4.2, with stacks reinitialized whenever one is found empty and no information is pending from that group. In terms of delay-throughput, the two schemes are indistinguishable, account being taken of simulation error. In both GQV's, adaptivity is introduced by reinitializing the search trees at the level prescribed by the cycle length minimization procedure using the estimate of section 4.2.2.

From these results one can conclude that the greatest gain in performance is derived from the use of direct entry instead of buffered entry. (The comparison is fair because the two exhibit the same blocking probability for given throughput). The improvement due to GQ management is marginal, although it does occur in an important region of the curve. For operation in this region, GQ management is an alternative to the use of a dynamic strategy. As expected, cycle minimisation is preferable to instantaneous throughput maximization in dynamic protocols. The inferior performance of the dynamic GQV relative to the dynamic direct entry protocol without GQ management is imputable to the inadequacy of the estimate $q=bo/N_s$ when applied to this more complex system where there are interations between groups.

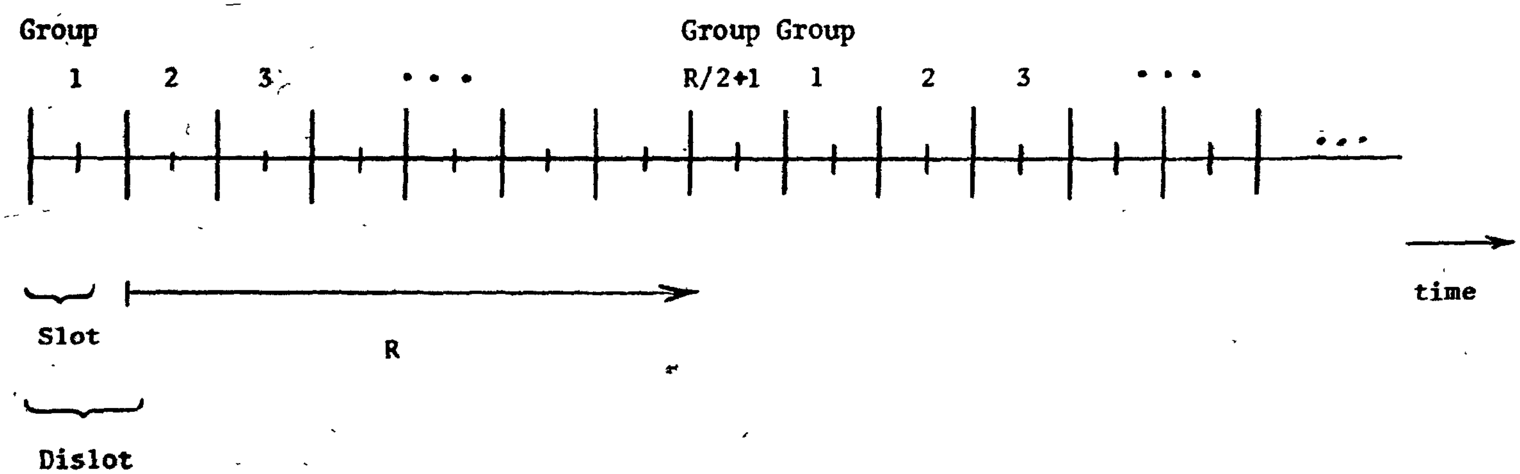

Fig.4.1.1 Partitioning of time in Capetenakis' scheme

q	m = n-k
.707	0
.38	1
$.8/2^m$	m = 2,3,...,n

Fig.4.1.2Solution of level skipping equation for mean cycle length minimization.

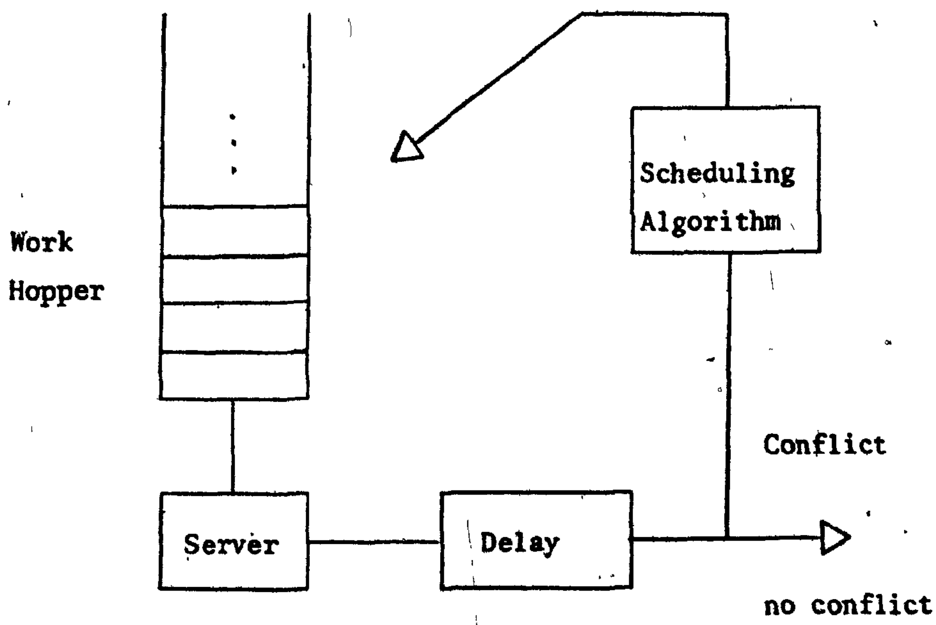

Fig.4.4.1 General Model for Global Queue Variant

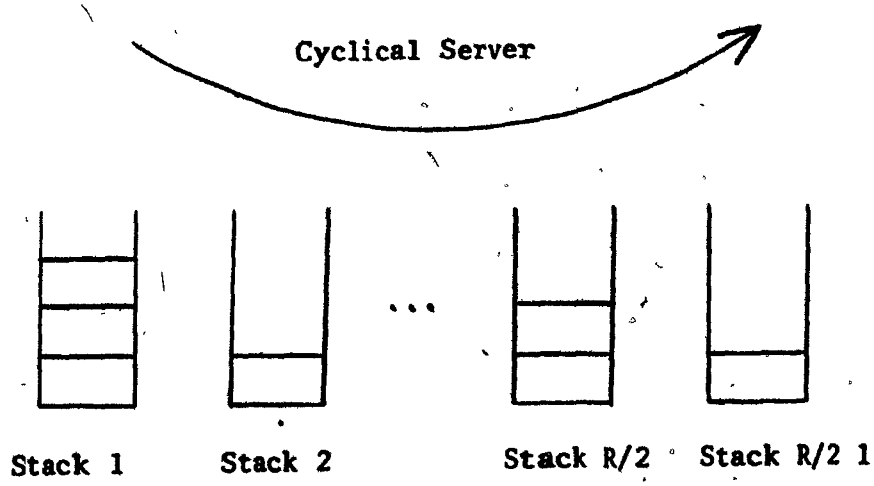

Fig.4.4.2 A Particular Implementation of a GQV

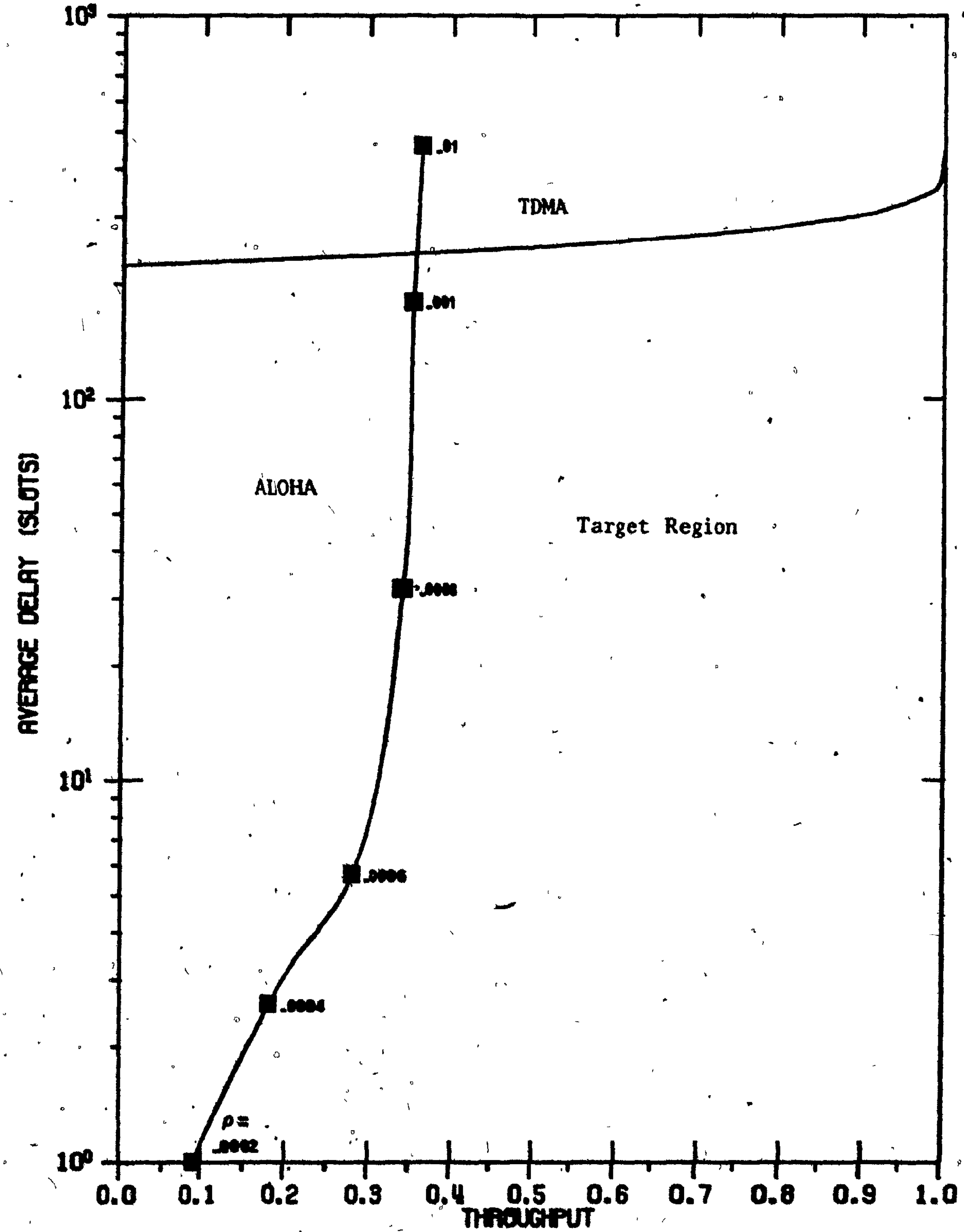

Fig.4.5.1 Target Region for Protocol Design

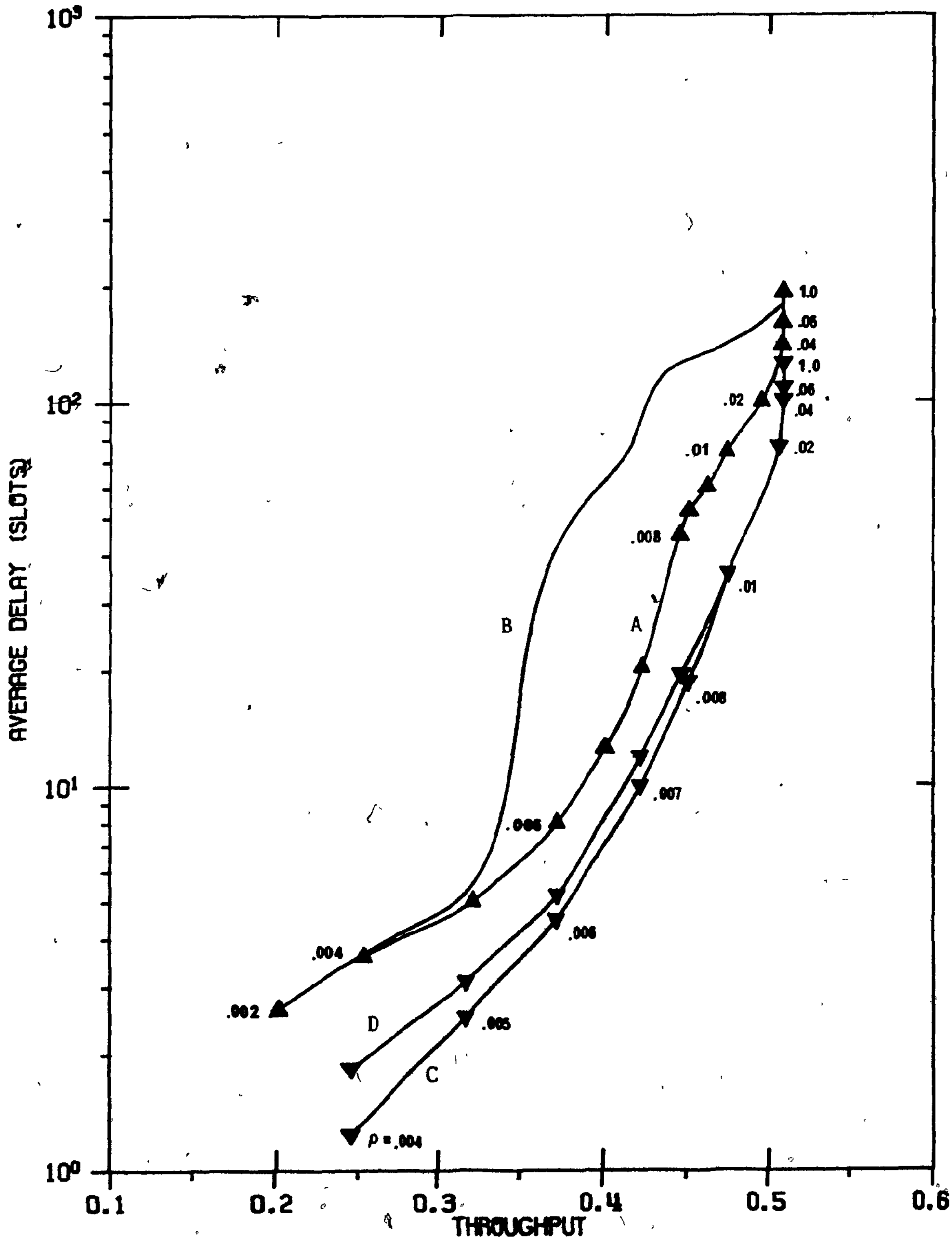

Fig.4.5.2 Simulated performance of Capetenakis' scheme (A) versus theoretical bounds (B) and direct entry and SSBS variants (D and C).

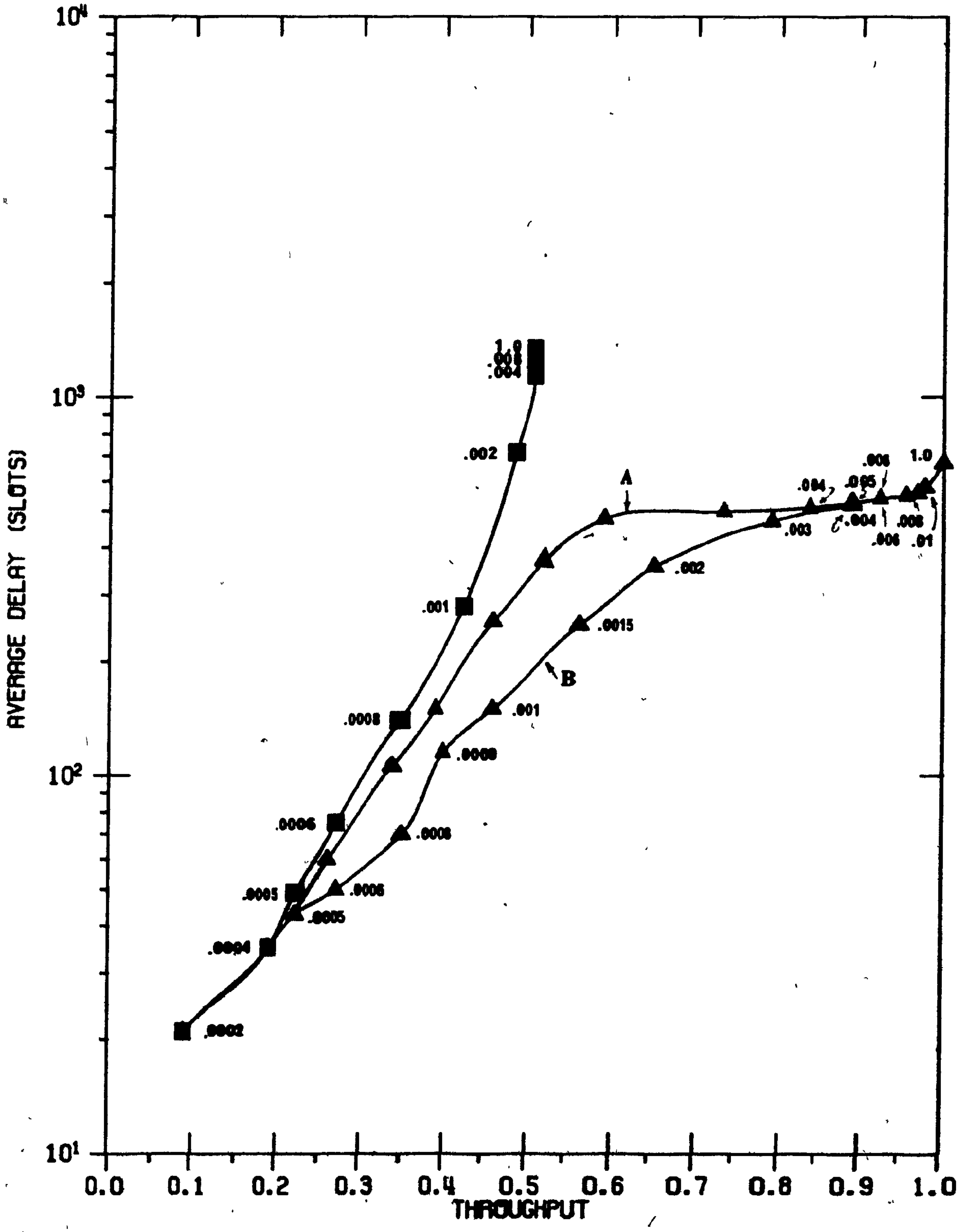

Fig.4.5.3 Static version of Capetenakis' scheme versus two dynamic strategies: instantaneous throughput maximizing (A) and cycle length minimizing (B) level skipping. (Buffered entry, N=448, R=12)

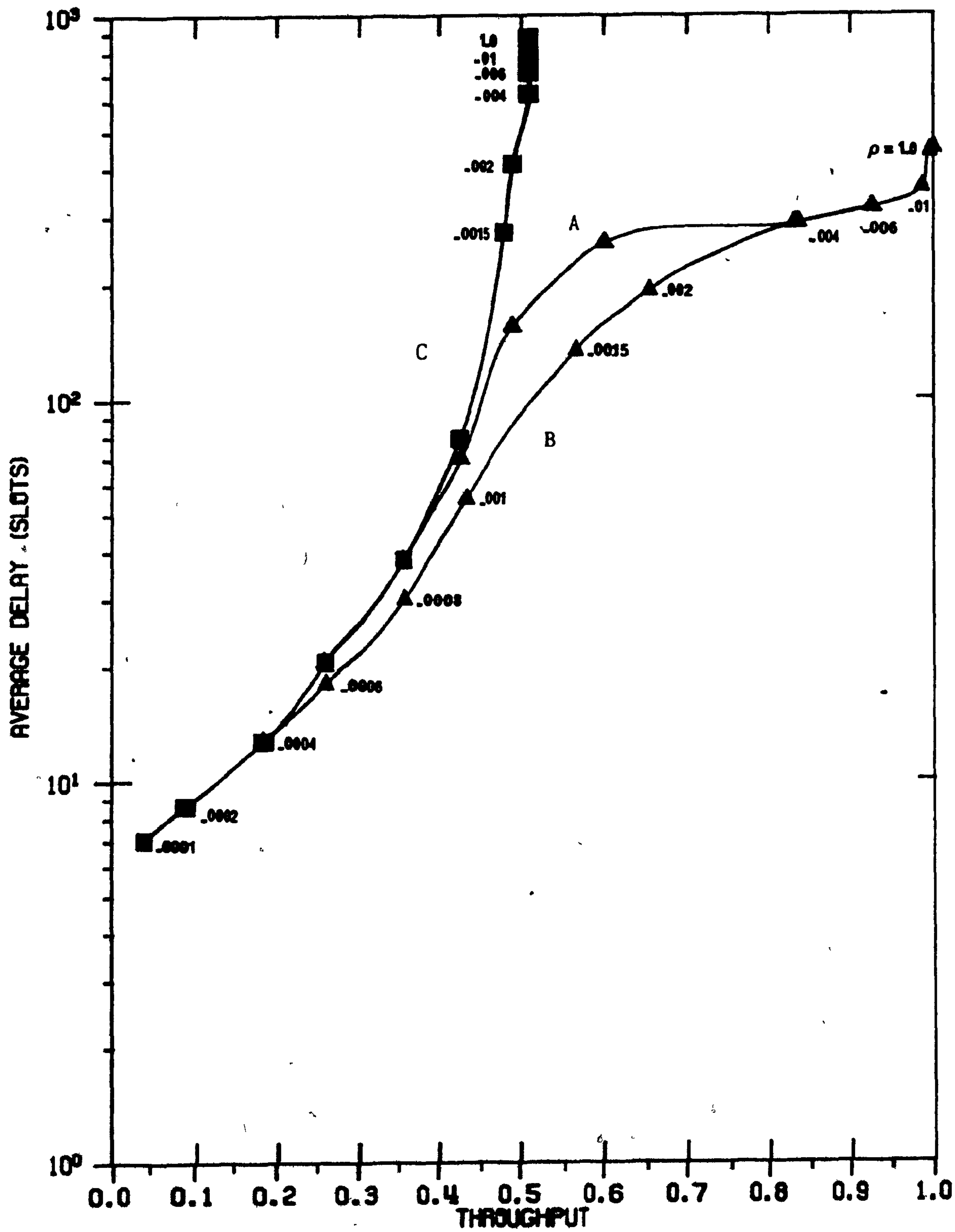

Fig.4.5.4 Static version of Capetenakis' scheme (C) versus two dynamic strategies: instantaneous throughput maximizing (A) and cycle length minimizing (B) level skipping. (Direct entry, N=448, R=12)

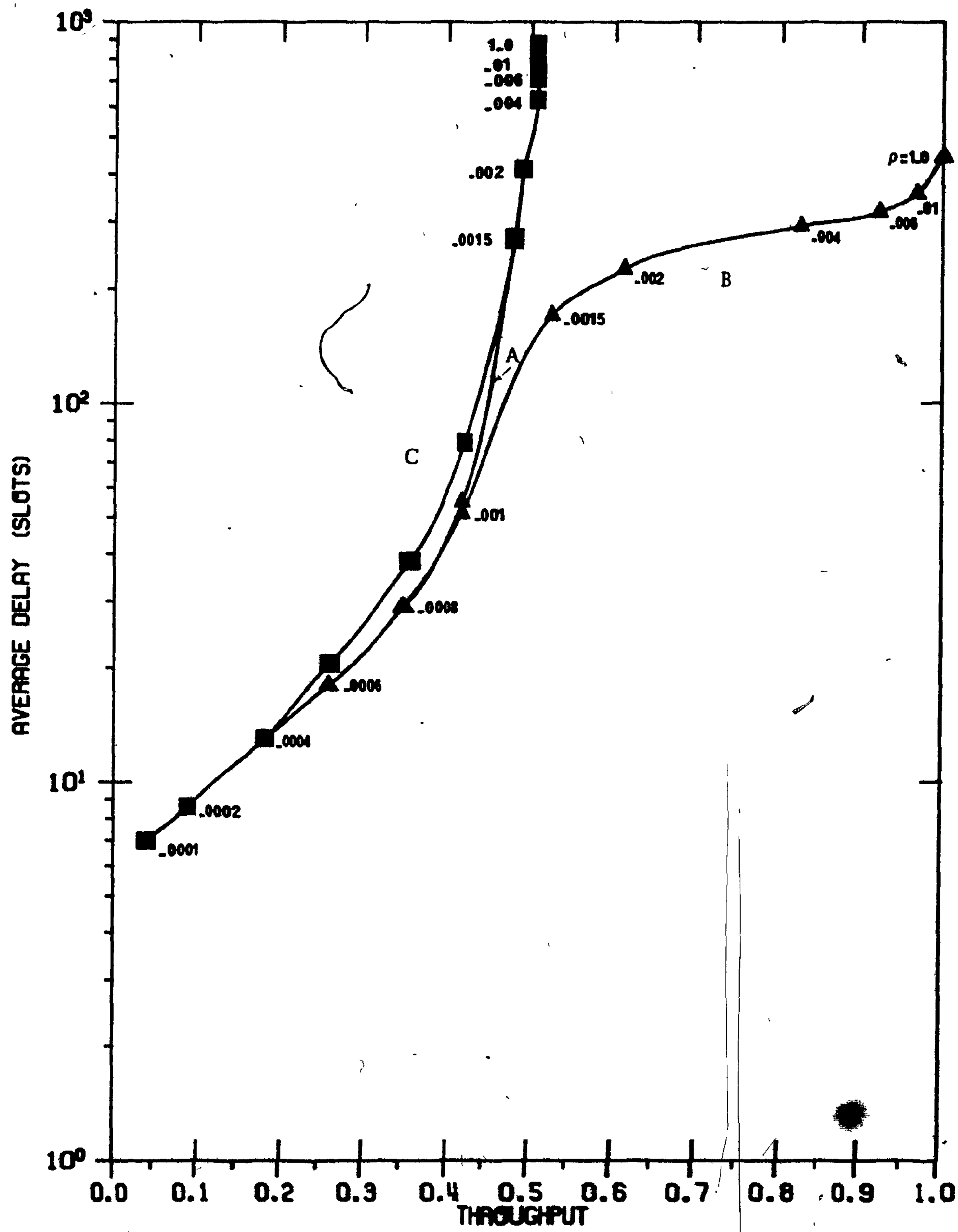

Fig. 4.5.5 Comparison of two global queue variants in static and dynamic versions (A and B resp.) with Capetenakis' scheme (static - curve C). (Direct entry, N = 448, R = 12)

CHAPTER 5

AMORPHOUS: A LOCALLY OPTIMUM THROUGHPUT MAXIMIZING ALGORITHM

In this chapter, the search for efficient access protocols is extended beyond procedures described in terms of tree searches. The structure inherent in a tree search suggests optimizing over a cycle. We now consider a scheme (AMORPHOUS), applied to the direct entry case, which is locally optimum in the sense that it attempts to maximize the throughput in each slot.

Local throughput maximization is accomplished by defining ,in each slot, a broadcast group of optimum size and composition. The broadcast group is drawn from the pool of users - designated the 'active' group - for whom channel status information resulting from the latest broadcast is available. The selection is based on estimates (one for each terminal) of the occupancy probabilities of the individual terminals in the current slot. The design problem consists accordingly of two subproblems: the partitioning of the active population into broadcast and quiescent groups (a bin packing or knapsack type of problem), and the updating of the probability q associated with each terminal (Bayesian learning).

5.1 The Partitioning Algorithm

The problem is to select a broadcast group from a (typically) non-homogeneous active population (size Na) so as to maximize the throughput (σ), given the terminal busy probabilities q_i , $i=1,...,Na$. The terminals are ranked in order of decreasing packet possession probability before the partition is computed. Any reasonable updating rule must have the property that the estimate of a terminal-occupancy probability increases as delay accrues for the packet held by that terminal. Since the terminals of higher q have packets which have sojourned longer in the system, these terminals are included in the partition with a priority commensurate with their rank. Let c_i be the variable which takes the value 1 if terminal i in the sorted list is to be included in the present broadcast group, and 0 otherwise. The vector $\underline{c}(\lambda)$ will have the form

$$c_i = \begin{cases} 0 & j > \lambda \\ 1 & j \leq \lambda \end{cases} ,$$

where λ is a threshold to be determined. Without loss of generality, we assume that λ is integral.

Problem

To choose λ^* such that

$$\sigma(\lambda^*) = \max_{(\lambda)} \sum_{i=1}^{\lambda} q_i \prod_{\substack{j \neq i \\ j=1}}^{\lambda} (1-q_j)$$

The solution is readily found in terms of a test quantity $S(\lambda)$. Let,

$$x_i = q_i / (1-q_i)$$

$$S(\lambda) = \sum_{i=1}^{\lambda} x_i$$

$$\underline{1} = (1,1,1,\ldots,1)$$

$$L = \{ c : \sigma \text{ attains its maximum } \}$$

In Appendix B, it is shown that,

Solution

(1) If $S(Na) \leq 1$, then $\underline{1} \in L$, and L contains no other point.

(2) If $S(Na) > 1$, and $1 < S(\lambda) \leq 1+x_{\lambda}$, then $\underline{c}(\lambda)$ is the unique point in L.

(3) If $S(\lambda) = 1$, then both $c(\lambda)$ and $c(\lambda+1)$ are elements

of L, and L contains no other point.

5.2 The Probability Updating Algorithm

The history of terminal (i) relevant to the problem of deciding whether that terminal is occupied in the present slot is summarized in the attribute q_i. Initially, q_i is equal to the a priori probability p of packet creation in a slot, given that the terminal was free in the preceeding slot. With the passage of time, and with successive attempts to transmit, the probability q_i is modified in order to reflect new knowledge of the probable state of the terminal. Let q_i' be the updated version of q_i. If the terminal is a member of the broadcast group, of size determined by application of the partitioning algorithm of section 5.1, then, as shown in Appendix B, the updating rule, applied R slots after broadcast, is,

(1) If no collision ensues,

$$q_i' = 1.-(1.-p)^{R+1} \triangleq \rho$$

(2) If a collision ensues,

$$q' = \frac{\dfrac{\rho + x_i}{1 + x_i} - x_i\,(1-\rho)\,\dfrac{\sigma(\lambda)}{S(\lambda)} - \rho\,\sigma(\lambda)\left[1 + \dfrac{1}{S(\lambda)}\right]}{1 - \sigma(\lambda)\left[1 + \dfrac{1}{S(\lambda)}\right]}$$

In obtaining this rule, we have assumed that the states of the terminals in a broadcast group are independent from terminal to terminal. For members of the quiescent group we let,

$$q'_i = q_i + p(1-q_i)$$

Again we make an independence assumption, namely that the states of the terminals in the quiescent group are independent of the states of all other terminals, including those in the broadcast group.

5.3 Operation

We have outlined the techniques used to create an optimal partition, given current values of the q_i's, and the manner in which the q_i's are updated. Fig.5.4.1 shows the information structures involved. The protocol may be expressed in terms of the following steps.

1. Start with all terminals in the active group.
2. new slot: update q's for quiescent group.
3. If any output from delay line, update q's for output group members and insert in active list according to q.

4. Partition to form broadcast group.
5. Place in delay line.
6. Go to 2.

It is easy to see that the algorithm degrades to TDMA at large traffic and to ALOHA at low traffic.

AMORPHOUS has another salutory feature, which becomes important in the multi-priority environment. There is no fixed TDMA component. AMORPHOUS is elastic in slot usage, using only as many slots in a round trip delay as required to accommodate optimum size partitions, and leaving the rest of the round trip delay for lower priority classes to use while running in background mode.

5.4 Simulation

The simulation program is written in FORTRAN, and makes extensive use of two and three dimensional singly linked lists [30]. For low to moderate traffic, groups of terminals will tend to share the same terminal occupancy probabilities. Thus, instead of processing individual terminals, groups of terminals are processed by manipulating the links. For example, sorting is accomplished by inserting (linking) each common-q group constituting a broadcast group retrieved from the delay line into the appropriate position in the active list. The active group is a list of common-q headers, and is thus treated as a two-dimensional list, while the delay line , being a list of partitions of active lists is (trivially) a three-dimensional list. This relatively complex structure was found to be an asset for Np<.3.

The simulation procedure, as well as the manner of presentation of results, is similar to that described in sections 4.5.2 and 4.5.3. The results are summarized in Fig.5.5.1.

5.6 Discussion

The delay-throughput performance of AMORPHOUS for the single class case (curve A in Fig.5.5.1) is, over much of the region of interest, inferior to that of the tree searches examined previously. However, the comparison with tree searches is mitigated to some extent by the observation that throughput in the context of AMORPHOUS has a different interpretation than it has for the schemes considered earlier. At low traffic values, AMORPHOUS tends not to use every slot in the round trip delay, thereby leaving vacant slots which can be used for other purposes (by other classes, for example) . If throughput is computed on the basis of the number of slots actually used by the algorithm, then one obtains curve (B) in Fig.5.5.1.

The effect of the independence assumption of section 5.2 is to cause the algorithm to be conservative in sizing broadcast groups. It is believed that a refinement of the algorithm aimed at circumventing the independence assumption will also lead to a more significant proprtion of idle slots, enhancing the opportunity to generate excess capacity by using these slots for other purposes.

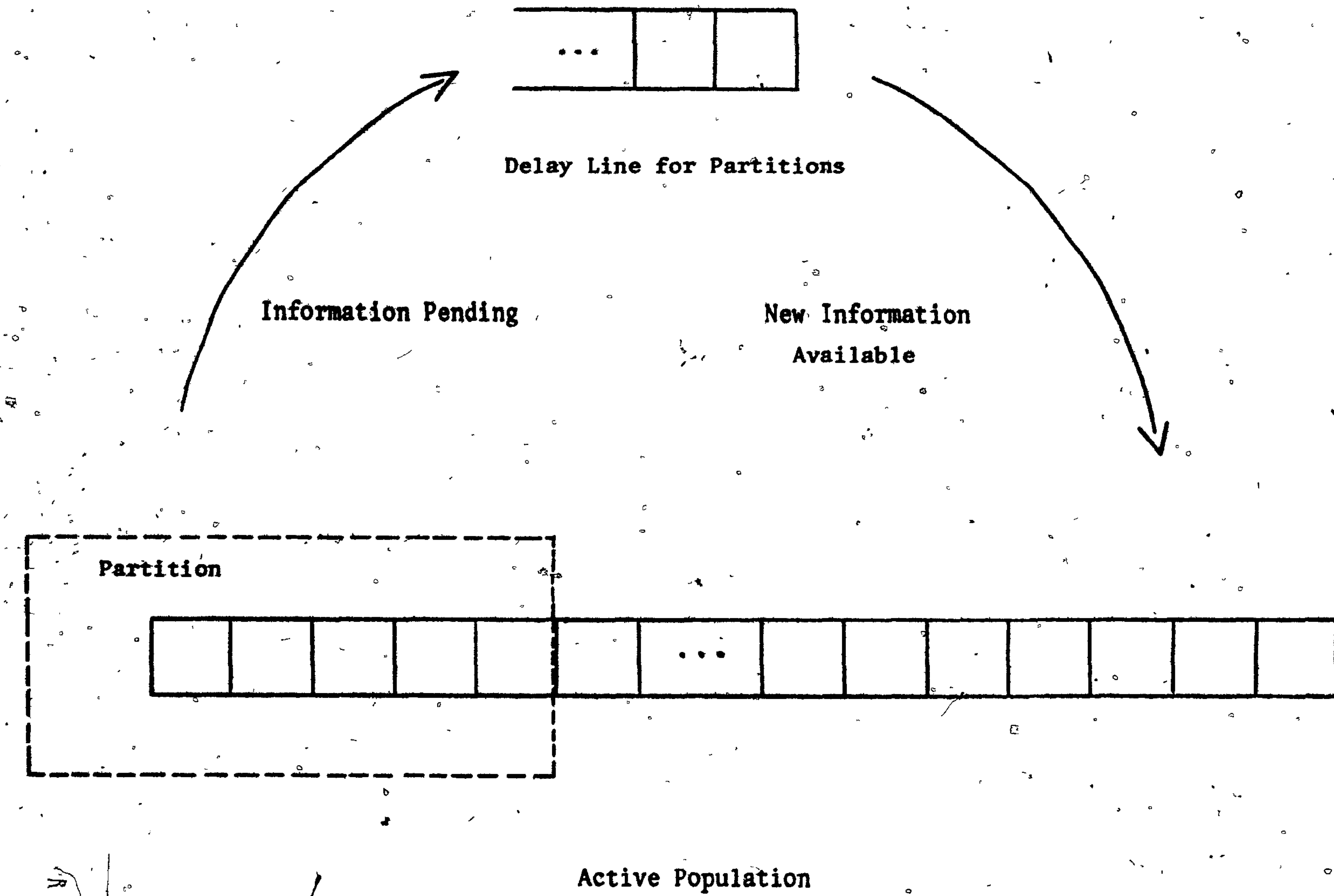

Fig.5.4.1 Illustrating Information Structures in AMORPHOUS

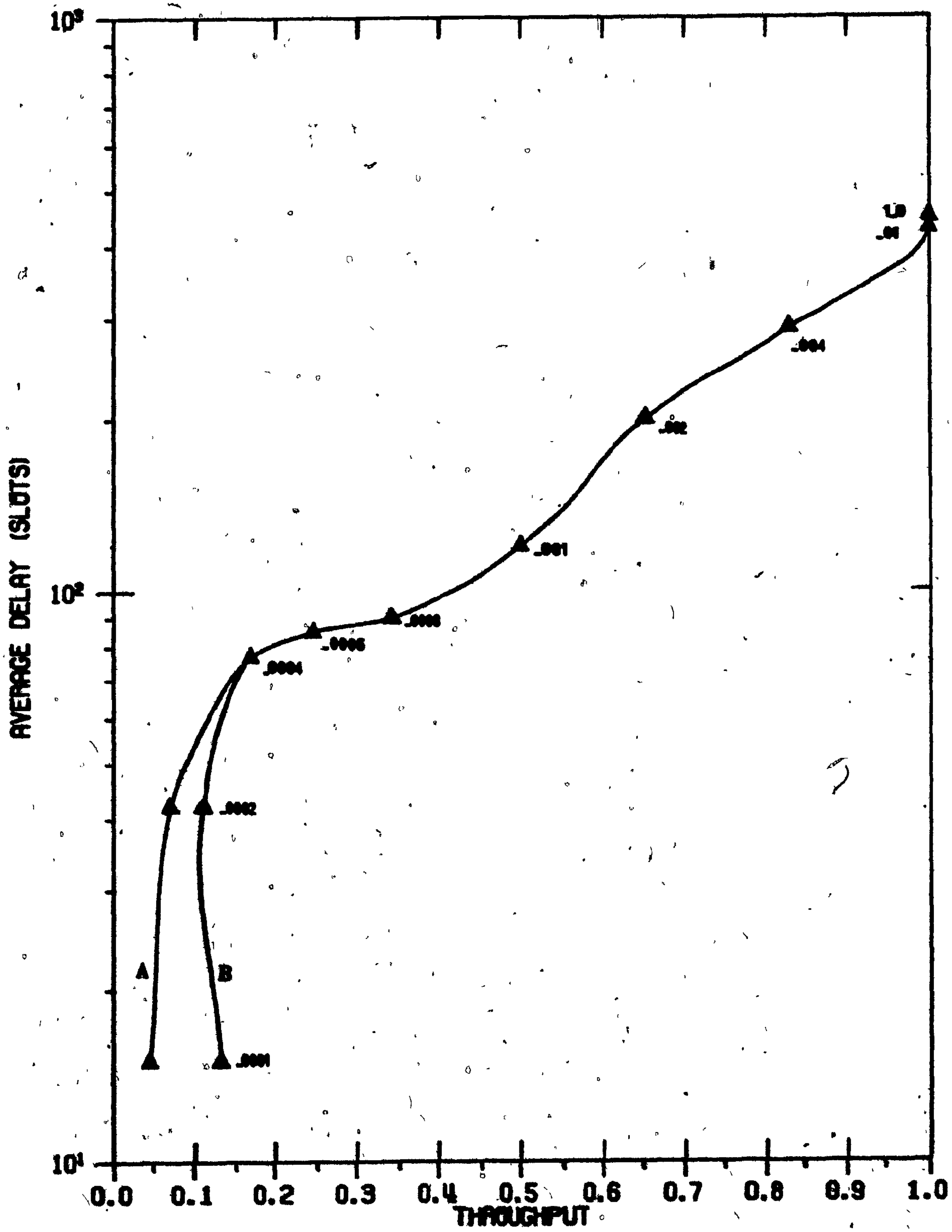

Fig.5.5.1 AMORPHOUS under simulation: conventional interpretation of throughput (A) and throughput computed on the basis of slots used by the algorithm (B).

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

The delay-throughput behaviour of tree-search-based access protocols has been examined with reference to the single-hop packet satellite problem. It has been shown, by simulation, that certain modifications to a scheme due to Capetenakis [19] yield substantial improvements. A novel protocol has been proposed which attempts to maximise instantaneous throughput by composing broadcast groups based on Bayesian estimates of terminal-occupancy probabilities. Its relatively poor performance has been ascribed to an independence assumption.

The conclusions to be drawn from this work are the following:

(1) Tree-search-based schemes exhibit excellent delay-throughput performance (better than predicted by Capetenakis).

(2) Direct entry variants are uniformly preferable (in terms of delay-throughput performance) to Capetenakis' buffered entry arrangement.

(3) Optimal level-skipping adaptive tree searches jump to nodes where the average information acquired in a probe is maximized. Jumping to the level where a probe is most likely to yield a successful transmission (instantaneous throughput maximizing level skipping) is shown to be a poorer policy.

(4) A global queue management approach provides a marginal benefit at moderate p values in the static case, but no improvement in the dynamic case - perhaps due to difficulty in formulating the estimator.

(5) Further work is required to formulate a true instantaneous throughput maximizing policy; in particular, the independence assumptions made in deriving the updating rule for the terminal-busy probabilitities (cf. section 5.2) must be circumvented. Nevertheless, AMORPHOUS provides a framework in which the access protocol design problem can be condensed to the sub-problem of designing an updating rule.

In spite of the simplicity of the schemes considered here, there is a dearth of analytical results. The aim of future work could be to supply some of these. Also, the details of a practical implementation of tree searches needs further examination. For example, in a real system, there may be occasional errors made in the conduct of a tree search. (Tree searches are particularly vulnerable to such impairments.) The effect of such a loss of synchronization and its remedy should be considered. The use of access protocols in the multi-priority environment might also be examined. At a fundamental level, the problem of access protocol design may be viewed as a problem in team decision theory [29]. Such a purview may answer such important unresolved questions as what constitutes an optimal policy for a given information structure (e.g. when information is acquired exclusively via the channel state process or when a limited amount of side information is available.).

There continues to be interest in access protocols for the multi-access broadcast channel. This is evinced by recent publications [26,28] extending Capetenakis' work and applying it to multi-hop packet radio networks. This new work points to other research topics.

APPENDIX A

For the special case p=1, the expected delay and the worst case delay for both the buffered and direct entry models can be computed.

A.1 Maximum mean delay (buffered entry):

The delay incurred by a packet consists of two contributions: one due to the time spent in hold buffers (d_h) and the other due to time spent in the transmit buffer (d_t).

The first is clearly,

$$E[d_h | p=1]=2N-3$$

We now direct our attention to the second contribution, $E[d_t | p=1]$.

Define, $n = \log_2 N$

s_j =no. alg. steps to release packet at term.(j)

$$S_\ell = \sum_{j=1}^{2^{\ell-1}} s_j$$

Then,

$$E[d_l \mid p=1] = S_n \;/N$$

Writing the recursion,

$$S_{l+1} = 2\,S_l + 2^{l-1}(2^l + 1) \qquad l=1,\ldots,n-1$$

$$S_0 = 0$$

and solving by z-transforms one obtains,

$$S_l = [2^{2l} + 2^l(l-1)]/2$$

Hence, in slots,

$$E[d \mid p=1] = N+n-1$$

and,

$$E[D \mid p=1] = 3N+n-5$$

A.2 Maximum Delay (p=1, buffered input):

The maximum delay is incurred by the rightmost terminal in the tree:

$$D_{max} = 2N-3 + 2N-2$$

$$= 4N - 5$$

A.3 Worst case and maximum mean delay for direct entry:

The cycle length is constant at (2N-2)(R/2+1). The maximum mean delay (excluding the last round trip delay, as usual) is then,

$$E[D|p=1]=(2N-3)(R/2+1)+1$$

This is also the worst case delay.

A.4 TDMA

Consider the single packet buffer, direct entry model, with N terminals and a per slot arrival rate of p. A given terminal is served every N slots. The throughput is, thus,

$$\sigma(p)=1.-(1.-p)^N$$

If a terminal is to contribute to the delay accumulated in a cycle, it must acquire a packet within N slots of a service. The probability that a the packet is acquired within i slots of the last service is,

$$a_i = p\,(1-p)^i$$

The average packet delay is,

$$\bar{D} = N - \sum_{i=1}^{N} i a_i + 1$$

The average delay is thus,

$$\bar{D} = N + 1 - \frac{\sum_{i=1}^{N} i\, p\, (1-p)^{i-1}}{\sigma}$$

$$= \frac{N}{\sigma} - \frac{1}{p} + 1$$

APPENDIX B

In this appendix, the selection and updating rules used in AMORPHOUS are derived.

B.1 Derivation of partitioning rule:

As in the body of the text, let

N = population size

N = no. terminals in the active group

q = occupancy probability i'th terminal active list sorted in order of decreasing q_i $(q_1 > q_2 > \ldots q_N > 0)$

$$x_i = q_i/(1 - q_i)$$

$$S(\lambda) = \sum_{i=1}^{\lambda} x_i$$

$$P(\lambda) = \prod_{j=1}^{\lambda} (1-q_j)$$

$$(\lambda) = \sum_{i=1}^{\lambda} q_i \prod_{\substack{j\neq i \\ j=1}}^{\lambda} (1-q_j)$$

$$= S(\lambda)\, P(\lambda)$$

$$L = \{ c : \sigma(c) = \max_{(\lambda)} \sigma(c) \}$$

Claim:

(1) $\sigma(\lambda) = \sigma(\lambda+1)$ iff $S(\lambda)=1$

(2) $\sigma(\lambda) > \sigma(\lambda+1)$ iff $S(\lambda)>1$

(3) $\sigma(\lambda) \geq \sigma(\lambda-1)$ iff $S(\ ) \leq 1+x_\lambda$

(4) $\sigma(\lambda+1) - \sigma(\lambda) < \sigma(\lambda) - \sigma(\lambda-1)$

all s.t. $S(\lambda) < 1+x_\lambda$

Proof:

The claim can be verified by direct computation.

The partitioning rule of section 5.1 follows directly from these claims. By (2), if $S(N_\alpha) < 1$, then $\underline{1} \in L$, and L contains no other point. By (2), (3), and (4), if $\underline{c}(\lambda) \in L$, then $S(\lambda) \leq 1+x_\lambda$. Hence, if $S(\lambda) > 1$ and $\underline{c}(\lambda) \in L$, then $1 \leq S(\lambda) \leq 1+x_\lambda$. If $s(\lambda)=1$, then $\underline{c}(\lambda)$ and $c(\lambda+1)$ are both elements of L.

Though the partitioning rule has been derived here under the intuitively reasonable assumption that terminals should be included in the broadcast group in order of their estimated probability of occupancy, this assumption is not essential in arriving at the results. It can be shown rigorously that optimization over all possible N_α length vectors of binary valued elements leads to the same selection rule.

B.2 Derivation of the updating rule;

In addition to the symbology of B.1, let

q_i = terminal (i)-occupancy probability in slot (1)

q_i' = terminal (i)-occupancy probability in slot (R+1) given channel state resulting from the broadcast in slot(1). (i=1,...,N_a)

$\rho = 1 - (1-p)^{R+1}$

ξ = probability of collision in slot (1)

Define the following events:

B = { terminal (i) busy in slot (1) }

C = { collision in slot (1) }

A = { terminal (i) acquires a packet during the next round trip delay (in slots 2,...,R+1) }

I = { terminal (i) is idle in slot (1) }

The complement of an event E is denoted by $\bar{E}$.

There are two groups of terminals to consider: the broadcast group and the quiescent group. Members of the broadcast group (i=1,...,λ) are updated in slot (R+1), those of the quiescent group can be updated immediately. The crucial assumption is made that the states of the terminals are independent from terminal to terminal.

1) The broadcast group:

Case (i): no collision in slot (1)

With probability 1, all members of the broadcast group have empty buffers in slot (1). The probability of having acquired a packet while in the delay line is ρ. Hence, $q=\rho$, $i=1,\ldots,\lambda$.

Case (ii): Conflict in slot (1) First, note that,

$$\xi = 1 - P(\lambda) - \sigma(\lambda)$$

By Bayes' rule:

$$\begin{aligned} q_i' &= \Pr\{\text{term}(i), \text{busy slot } (R+1), \text{collision slot } (1)\}/\xi \\ &= [\, P(B,C) + P(I,C,A)] \,/\xi \\ &= [\, P(C|B)P(B) + P(I,A) - P(\bar{C},I|A)P(A) \,] \,/\xi \\ &= [q_i(1-\prod_{j\neq i}(1-q_j)] + \rho[1-q_i - \prod_{j=1}^{\lambda}(1-q_j) - \sum_{j=1}^{\lambda} q_j \prod_{k\neq j}(1-q_k)]/\xi \\ &= [q_i - x_i P(\lambda) + \rho\,(1-q_i - P(\lambda) + x_i P(\lambda) - \sigma(\lambda))]\,/\xi \\ &= [(\rho + x_i)/(1+x_i) - (1-\rho)x_i P(\lambda) - \rho(1-\xi)]/\xi \end{aligned}$$

QED

2) The quiescent group

By virtue of the independence assumption, the updating rule for members of the quiescent group need only consider the effect of the passage of time:

$$q'_i = \Pr\{\text{terminal } (i) \text{ busy in slot } (2)\}$$
$$= \Pr\{\text{busy slot}(1)\} + \Pr\{\text{not busy slot}(1), \text{creates packet}\}$$
$$= (1-q_i) + (1-q_i)p$$

REFERENCES

[1] L.Roberts, 'Data by Packet', IEEE Spectrum, Feb 1974.

[2] N.Abramson, 'ALOHA: Another Alternative for Computer Communication', AFIPS Conf. Proc., Vol.37, 1970.

[3] E.Fuchs and P.Jackson, 'Estimates of Distributions of Random Variables for Certain Computer Communication Models', ACM Sym. in Opt. of Comm. Sys., Pine Mt., 1969.

[4] P.Jackson and C.Stubbs, 'A study of Multi-Access Computer Communications', Spring Joint Comp. Conf. Rec., Vol.34, 1969.

[5] L.Kleinrock and W.Naylor, 'On Measured Behaviour in the ARPA Network', AFIPS Conf. Proc., Vol.43, 1974.

[6] R. Metcalfe, 'Steady State Analysis of a Slotted and Controlled ALOHA System with Blocking', Proc.Sixth Hawaii Int. Conf. on Sys. Sc., Univ. of Hawaii, Jan. 1973.

[7] L.Kleinrock and S.Lam, 'Packet Switching in a Multi-Access Broadcast Channel: Performance Evaluation', IEEE Trans. Comm. Vol. COM-23, April 1975.

[8] S.Lam and L.Kleinrock, 'Packet Switching in a

Multi-Access Broadcast Channel: Dynamic Control Procedures', IEEE Trans. Comm., Vol. COM-23, Sept. 1975.

[9] S.Lam, 'Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network', PhD dissertation, UCLA, 1974.

[10] W.Crowther, R.Rettberg, D.Walden, S.Ornstein, and F.Heart, 'A System for Broadcast Communication: Reservation ALOHA', ARPA Network Info. Center, SRI, Menlo Park, Dec. 1972.

[11] L.Roberts, 'Dynamic Allocation of Satellite Capacity through Packet Reservation', AFIPS Conf. Proc., Vol. 42, 1973.

[12] I.Rubin, 'Reservation Schemes for Dynamic Packet Access Control of Multi-Access Communication Channels', Technical Report UCLA-ENG-7712, Jan. 1977.

[13] I.Rubin, 'Integrated Random Access Schemes for Multi-Access Communication Channels', Technical Report UCLA-ENG-7752, July 1977.

[14] R. Binder, 'A Dynamic Packet Switching System for Satellite Broadcast Channel', IEEE Int. Conf. Com., 1975.

[15] S.Ng and J.Mark, 'A Multi-Access Model for Packet Switching with a Satellite Having Some Processing Capability', IEEE Trans. Comm., Vol. COM-25, Jan. 1977.

[16] S.Lam, 'An Analysis of the Reservation ALOHA Protocol for Satellite Packet Switching', IEEE Int. Conf. Comm., 1978

[17] M.Ferguson, 'On the Control, Stability, and Waiting Time in a Slotted ALOHA Random Access System', IEEE Trans. Comm., Vol. Comm-23, Nov.1975.

[18] L.Kleinrock and Y.Yemini, 'An Optimal Adaptive Scheme for Multiple Access Broadcast Communication', IEEE Int. Conf. Comm., 1978.

[19] J. Capetenakis, 'The Multiple Access Broadcast Channel: Protocol and Capacity Considerations', Report ESL-R-806, Elect. Sys. Lab., MIT, March 1978. (Also PhD dissertation with same title).

[20] J.Hayes, 'An Adaptive Technique for Local Distribution', IEEE Trans. Comm., COM-26, Aug. 1978.

[21] F.Schoute, 'Decentralized Control in Packet Switching Satellite Communication',IEEE Trans. Aut. Control., Vol AC-23, April 1978.

[22] E.Cacciamani, 'Data Services - American Satellite Corp.', Seventh Hawaii Int. Conf. Sys. Sc., Univ.Hawaii, 1974.

[23] R.Rettberg, 'Slotting', ARPA Network Info. Center, SRI, Menlo Park, Cal. ASS Note 32, May 1973.

[24] J.Metzner, 'On Improving the Utilization in ALOHA Networks', IEEE Trans. Comm., Vol.COM-24, April 1976.

[25] L.Roberts, 'ALOHA Packet Systems with and Without

Slots and Capture', Comp. Comm. Rev., Vol. 5, April 1975.

[26] L.Siegel, 'Multiple-Coupled Random Access Techniques for Packet Radio Networks,' Technical Report, Electronic Systems Lab., MIT, May 1978.

[27] R.Gallager, 'Conflict resolution in Random Access Broadcast Networks', 1978 AFOSR Workshop in Comm. Theory and Applications', Provincetown, Mass. Sept. 1978. (IEEE cat. no. EHO 139-6)

[28] G.Marsaglia et al, 'SUPER-DUPER', Internal Report, McGill University,1975.

[29] F.Scnoute,' Symmetric Team Problem and Multi-Access wire Communications',Automatica, Vol 16, May 1978.

[30] D.Knuth, The Art of Computer Programming Vol.1: Fundamental Algorithms, Addison Wesley, 1976.