On the Performance of Protocols for Collecting Responses over a Multiple-Access Channel

Mostafa H. Ammar, *Member, IEEE*, George N. Rouskas, *Member, IEEE*

***Abstract*— We consider a generalization of the multiple access problem where it is necessary to identify a subset of the ready users, not all. The problem is motivated by several "response collection" applications that arise in distributed computing and database systems. In these applications, a collector is interested in gathering a set of responses from a number of potential respondents. The collector and respondents communicate over a shared channel. We define three collection objectives and investigate a suite of protocols that can be used to achieve these objectives. The protocols are based on the use of Polling, TDMA, and Group Testing. Using a binomial respondent model we analyze and, where applicable, optimize the performance of the protocols. Our concern is with cost measures that reflect the computational load placed on the system, as well as the delay incurred for achieving a particular objective.**

I. Introduction

We investigate the problem of how to best collect a specified number of responses from a set of nodes over a multiple access channel. Several situations in distributed systems where such a problem arises are described later. We consider a system where nodes share a common communication channel. One node in the system is interested in collecting responses from the other nodes. Not all nodes can or will respond when requested and the node soliciting responses is interested in achieving a collection objective.

The problem we consider is actually a generalization of the multiple access communication problem where we are concerned with identifying a *subset* of ready users, not all. A response collection process will be aimed at achieving one of a set of *collection objectives* to be described later. We describe and analyze a suite of protocols that can be used for response collection. Our concern is with the cost of the collection process in terms of the amount of computation resources it consumes, as well as the amount of time expended to achieve a certain collection objective. The protocols we use are based on the use of Polling, time division multiple access (TDMA) and Group Testing.

Whereas Polling and TDMA are well known multiple access techniques, Group Testing warrants a short introduction. It is a technique that can be used to efficiently identify "defective" items in a set. It has been studied extensively in different contexts (see for example [1], [2], [3], [4]). The basic idea of the technique is the testing of items being inspected in groups. The composition of the group to be tested at any one point in time being dictated by the history of previous test outcomes. Each test is counted as a single step and the objective is to determine group composition rules to minimize the number of steps. In its original form, the problem assumes the outcome of each test would indicate one of two situations: "all items are not defective" or "there is at least one defective item." We are concerned here with the potential use of Group Testing as a technique for collision resolution over a multiple access channel. Such use has been described in [5], [6], [7], [8]. The additional feature when using Group Testing over a multiple-access channel is the ability to differentiate among three possible outcomes when a group is enabled: no transmission, one transmission, and more than one transmission (a collision).

This paper is organized as follows. Section II contains a model of our system. In Section III we discuss some applications that motivate our work. Section IV presents a description of the protocols we consider. In Section V we analyze and optimize the performance of some of our proposed protocols. Some numerical examples are presented in Section VI and Section VII contains some concluding remarks.

II. System Model

A. The Collector and Respondents

The system under consideration has a node (connected to a shared channel) which is attempting to collect responses. We call this node *the collector*. The collector actively solicits responses by transmitting messages on the channel.

All the nodes that can potentially respond to a collector's request are called *respondents*. We assume there are N such respondents. This collector-respondent classification may be permanent or it may be temporary. In the latter case, the collector will abandon its role once its response collection objective has been achieved. At that time another node may assume the collector's role. As several nodes may desire to become collectors at the same time, a fair "election" protocol needs to be available for use by the nodes. In this paper we only concern ourselves with the system behavior from the time a new collector is identified until the collector's objective is achieved.

Paper approved by I. Gopal, the Editor for Network Protocols of the IEEE Communications Society. Manuscript received September 28, 1990; revised August 12, 1992. This work was supported in part by NSF grant NCR-8604850. This paper was presented in part at the IEEE INFOCOM '91 conference, Miami, FL, April 9-11, 1991.

Mostafa H. Ammar is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA.

George N. Rouskas was with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. He is now with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA.

IEEE Log Number 9410073.

The collector is the (perhaps temporary) master in the system and actively solicits responses from the respondents. We distinguish between the *soliciting* and *enabling* of a respondent. A respondent is solicited once it receives a message from the collector making it aware that a collection process is underway and indicating the collection objective. A respondent is enabled if the protocol rules allow it to transmit a response on the channel if indeed it can respond. A respondent can be enabled only after or at the same time as it is solicited.

The collector is assumed to operate with some statistical knowledge of the state of the respondents. In our analysis we will assume the *binomial respondent* model. At the instant the collection process begins, each respondent will transmit a response when enabled with a probability q and with probability $p = 1 - q$ a respondent will not transmit a response when enabled.

B. Collection Objectives

With respect to a collector's request, a respondent is classified as *active* if it will respond when given a chance (i.e., solicited and enabled). A respondent is said to be *inactive* otherwise. The goal of the collector is to identify and collect "enough" responses from active respondents to satisfy its application. Note that in some instances, the collector's goal may be achieved if it determines (from the *lack* of responses) that the desired number of responses cannot be collected. We consider three distinct collection objectives.

1. *L or Nothing* (or "Exactly L"): Terminate successfully after collecting exactly L responses or abort when a determination is made that the number of active respondents is less than L.
2. *L or Maximum* (or "At Most L"): Terminate successfully after collecting L responses, or after all respondents have been given a chance to respond, whichever occurs first.
3. *L or More* (or "At Least L"): Terminate successfully if L or more responses have been collected *and* all respondents have been given a chance to respond. Abort when a determination is made that the number of active respondents is less than L.

For example, assume the number of respondents N is 20 and $L = 6$. A collector with the "6 or Nothing" objective will terminate if 6 responses have been received or it will abort the search if out of the respondents enabled a total of 15 did not transmit responses. With the "6 or More" objective, the collector will abort in the same situation above, it will, however, continue to gather responses after 6 responses have been received. In the case of a "6 or Maximum" objective the collector will terminate (before all respondents have been enabled) only if 6 responses have been collected.

We will use the superscript (y) to denote a collection objective. The superscript will be *L or no*, *L or max*, or *L or more* to denote the objectives in 1, 2, and 3 above, respectively. When only the number being targeted is relevant we will use that number to denote any of the three objectives, i.e., the superscript (ℓ) will denote the collection objectives ℓ or max, ℓ or no, or ℓ or more.

We note the following equivalences between the various objectives:

$$N \text{ or } Maximum \equiv 0 \text{ or } More \equiv \textit{Find All Active} \quad (1)$$

C. Network Environment

All communication takes place over an error-free shared channel with capabilities for single destination, multicast and broadcast addressing. Simultaneous transmissions over the shared channel result in a collision. All response packets are assumed to be of the same size and the network can operate in a slotted mode where each slot is long enough for the transmission of a response packet. Respondents are constrained to begin transmission at a slot boundary and thus all collisions are the result of the complete overlap of response packets. The channel is assumed to provide the so-called (0, 1, e) feedback where the nodes on the channel are informed whether the previous slot contained no transmissions (0), one transmission (1), or a collision (e).

D. Collection Costs

For a particular protocol z and a given collection objective y, we identify three types of costs incurred in the collection process:

1. Delay Cost, $D_z^{(y)}$: The average number of response slots needed until the collection objective is achieved or until a determination is made that the collection objective is not attainable.
2. Respondent Solicitation Cost, $S_z^{(y)}$: The average number of respondents that are solicited in the collection process. As each solicitation message received requires interpretation and perhaps the generation of a response this measures the computation cost incurred by the respondents.
3. Collector Solicitation Cost, $C_z^{(y)}$: The average number of solicitation messages sent by the collector during the collection process. This is a measure of the computation cost incurred by the collector, as well as the delay incurred each time the collector needs to send a solicitation message.

The *total cost* incurred by collection protocol z with objective y is given by:

$$A_z^{(y)} = \alpha D_z^{(y)} + \beta S_z^{(y)} + \gamma C_z^{(y)} \quad (2)$$

where α, β, and γ are weights assigned to the various costs. We will drop the subscript describing the protocol when it is clear from the context to which protocol the quantity refers.

III. Some Applications

The following are some applications that make use of response collection.

A database system with multiple query optimization: Here we have a shared channel LAN with the primary purpose of giving a set of attached users access to a database (also connected to the network). The users are moderately active and the database employs sophisticated query processing techniques. These include schemes to speedup query processing through the use of multiple query optimization (see e.g., [9]). Rather than processing each query individually, the database tries to process a number of queries (up to a maximum) at a time. In order to manage memory and processing resources efficiently, the database processor prefers to actively collect responses, rather than receiving responses asynchronously.

The database processor in this scenario is an example of a collector with the "L or Maximum" collection objective. The collector in this scenario is permanent.

A broadcast delivery information system: An information system using broadcast as a delivery mechanism has the potential for *shared response*, i.e., responding to several users with one transmission (see e.g., [10], [11], [12], [13], [14]). In order to maximize the benefit of the response sharing feature, the information source needs to be aware of the information needs of a representative set of users at any one time. Thus, before transmitting responses, the information source needs to collect a set of requests from the users.

Again this is an example of a permanent collector with the "L or Maximum" collection objective.

Quorum collection for synchronization in a distributed system: Quorum consensus is a general class of synchronization protocols for distributed systems [15], [16], [17], [18]. An operation may proceed to completion only if it is granted permission from a number of nodes. If mutually exclusive execution of operations is desired, e.g., as would be required when writing replicated data, then the node executing the operation needs to collect permission from a majority of nodes. Other applications, such as the reading of replicated data, may require permission from a certain number of nodes, not necessarily a majority [19]. When quorum consensus protocols are used, if a quorum cannot be collected, the operation requesting the quorum aborts. Nodes, in some situations, may not be able to grant permission when requested if they have already granted permission to another node. They may also not be able to grant permission because they have failed or are too busy.

In this scenario the collector is typically temporary. The CPU collecting a quorum is typically interested in the "L or Nothing" collection objective. When quorum consensus is used to update replicated data the collector might also be interested in updating a minimum number of replica, but more can be updated if possible; this is the "L or More" collection objective.

Finding multiple instances of a named resource: An application running in a distributed system often requires access to multiple instances of a resource. The application typically knows the name or property of such a resource, and may not be aware of where the resource is physically located in the network. The searching application needs to determine a set of addresses where the resource resides [20], [21]. Some examples of this are:

- a node is searching for four or more processors that are lightly loaded in order to run a parallel program; an example of the "L or More" collection objective,
- a node is searching for up to four disks with a given amount of available space to store a certain file; an example of the "L or Maximum" collection objective.

IV. Protocol Description

A. Polling and TDMA

One possible technique that can be used by the collector is to poll all respondents individually. Each polling message sent by the collector solicits and enables one respondent. This may require a significant amount of time to complete as each poll requires two messages to be sent if the outcome is positive (i.e., a response is generated by the respondent) or a message followed by a timeout period if the outcome is negative.

Another approach which would require less time is for the collector to declare its objective to the entire network via a broadcast message and have the active respondents send their responses. If the responses are transmitted using a random access scheme, a considerable amount of time and bandwidth may be wasted until the required number of respondents successfully transmit their responses. Alternatively, the protocol may operate by having the respondents ordered in some (perhaps random) way and allocate a time slot to each respondent. Active respondents transmit their responses in the allocated slot. Slots allocated to inactive respondents remain idle. As all respondents can hear channel activity, they all know when the collection objective (declared by the collector in its broadcast message) has been achieved and this collection phase is terminated. We call this technique, *the TDMA collection protocol.* Note that in this scheme the respondents are all solicited by the initial broadcast message. A respondent is enabled at the beginning of its allocated slot.

The TDMA protocol will achieve the collector's objective in less time than a Polling procedure. The TDMA protocol, on the other hand, will involve all the respondents (whether or not they are active) as they will receive the initial broadcast message which will have to be interpreted by all the receiving hosts.

B. Group Testing

The Group Testing response collection procedure is initiated by a broadcast solicitation message sent by the collector and received by all respondents. Once this message is received by all respondents, the channel operates in the slotted mode where a group of respondents is enabled at the beginning of each slot. The choice of group to enable is determined entirely by the respondents by observing the channel activity and does not require intervention by the collector. The protocol operates in a similar manner to the one described in [6], with the major difference being that the protocol will terminate whenever the collection objective is achieved.

Each respondent observes the channel activity during each slot and updates its knowledge of the state of the respondents accordingly. The state of the system is described by membership in four sets [6], [1]:

- A *classified set* is a set of respondents who are known to either be active or inactive. Each time a group is enabled and no transmission is heard during the slot, all the members are judged to be inactive. Active members are discovered when a group which contains exactly one active respondent is enabled, the active respondent's transmission is heard and the other members are known to be inactive.
- A *binomial set* is a set of respondents whose members each have probability q of transmitting a response when enabled independently of one another. When the protocol starts all N respondents belong to this set.
- A *defective set* is a set of respondents known to contain at least one active respondent.
- A *conflicted set* is a set of respondents known to contain at least two active respondents. Note that, typically, the conflicted set, if not empty, will be a superset of the defective set.

At the beginning of each slot, depending on the membership of the binomial, defective and conflicted sets we say that

- an *H-situation* has occurred if the defective set and the conflicted set are empty. In this case a subset of the respondents from the binomial set are enabled in the next slot.
- an *F-situation* has occurred if the defective set is not empty but the conflicted set is empty, in which case a subset of the of the defective set is enabled or the entire defective set and a subset of the binomial set are enabled.
- a *G-situation* has occurred if the conflicted set is not empty. In this situation a subset of the defective set is enabled or a subset of the conflicted set that contains the entire defective set is enabled.

More details on the operation of this protocol can be found in [6].

C. Staged Protocols

In both the TDMA and Group Testing protocols described above, the collector starts by sending a broadcast message that has to be received and interpreted by all respondents. In situations where it is necessary to identify all active respondents (e.g., the "traditional" multiple access problem), this computational burden is unavoidable. However, in the more general scenarios discussed here, it is possible to reduce the amount of computation at the respondents at the expense of requiring the collector to send more messages. This can be accomplished by ***staging*** the collection process.

In a staged protocol, respondents are subdivided into disjoint groups, say g_i, $i = 1, \dots, M$. Responses are gathered by having the collector *multicast* a message to each group, one at a time. All respondents in that group are solicited (i.e., made aware that a collection process is underway) by the multicast message. The respondents are then enabled (i.e., asked to send a response if they have one) according to the rules of the protocol as described below. (Similar ideas for the staging of a search can be found in [22], [23].)

In a staged protocol each multicast solicitation message sent by the collector contains the current collection objective. The objective declared in the $(i+1)$th solicitation message is, in general, a "reduced" version of the one declared in the ith message. The amount of reduction is determined by the number of responses collected in the ith stage. All respondents solicited in a stage operate with the knowledge of the number of the not-yet-solicited respondents. This allows the respondents to individually determine when the collection objective has been met or when it cannot be achieved because the number of unexplored respondents is not sufficient.

Two staged protocols can be derived from the TDMA and Group Testing collection techniques described previously:

1. *Staged TDMA*: After the ith multicast solicitation message is received, all respondents in the ith group are allocated slots in which to respond.
2. *Staged Group Testing*: After the ith multicast solicitation message is received, respondents in the ith group are enabled according to a Group Testing procedure that involves *only* the members of the group.

For both staged protocols we distinguish between ***fixed-group*** and ***adaptive-group*** staging. When fixed groups are used, a set of mutually exclusive and collectively exhaustive groups are determined ***a priori***. In an adaptive-group staged procedure, on the other hand, we decide on the constitution of a group after the result of exploring the previous groups is known. The use of optimal adaptive groups will intuitively incur less or equal cost than the use of optimal fixed groups. Adaptive groups, however, may require the use of multiple destination addresses in multicast messages, as single multicast addresses cannot be set up ahead of time.

The staged protocols have the advantage that they may achieve the collection objective without involving (i.e., soliciting) all the respondents. They may, however, require somewhat more time to complete when compared to the single-stage protocols because of the delay involved in sending solicitation messages. For a performance measure that incorporates the time to complete, as well as the number of involved respondents, the performance of the staged protocols can be optimized by selecting the groups appropriately. We emphasize, however, that if the collection objective is to identify all active respondents, as is the case in [6], [8], then no advantage is gained by staging the collection procedure.

V. Analysis and Optimization

Fig. 1 shows the relationship among the protocols discussed in Section IV. Both TDMA and Polling are "points" in the collection protocol space in the sense that there is only one way to define their operation. The operation of Group Testing, Staged TDMA and Staged Group Testing depends on various parameters that are needed to fully describe the protocols. For Group Testing, it is necessary to

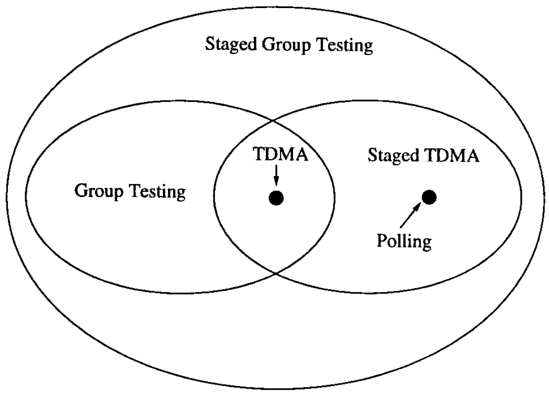

Fig. 1. Relationship among the various protocols described

determine how respondents are enabled in particular situations. In the staged protocols, the particular grouping of nodes (fixed or adaptive) needs to be defined.

In the following we focus on the analysis and optimization of the Group Testing and Staged Group Testing protocols, the latter being the most general. The same techniques can be applied in a straightforward manner to analyze the other protocols [24].

A. Group Testing

The size of the groups enabled can be found through the solution of a set of recursive equations shown below. In the following $H^{(y)}(n)$ denotes the average number of slots needed to satisfy the collection objective y when the binomial set is of size n, $F^{(y)}(m,n)$ denotes the average number of slots when the defective set is of size m and the binomial set is of size $n-m$ and $G^{(y)}(k,m,n)$ denotes the average number of slots when the defective set is of size k, the conflicted set is of size m and the binomial set is of size $n-m$. Then we can write for $\ell \geq 1$: [1] (These equations are similar to those shown in [6]. There are, however, small but critical differences that have to do with the fact that the collection objective is now an influencing parameter.)

$$H^{(\ell)}(n) = 1 + \min_{1\leq x\leq n}\Big\{P_0 H^{(\ell)}(n-x) + P_1 H^{(\ell-1)}(n-x) + (1-P_0-P_1)G^{(\ell)}(0,x,n)\Big\} \quad n\geq 1 \tag{3}$$

$$F^{(\ell)}(m,n) = 1+\min\{A_1,A_2\}, \quad n\geq 1 \tag{4}$$

$$A_1 = \min_{1\leq x\leq m}\Big\{P_2F^{(\ell)}(m-x,n-x) + P_3H^{(\ell-1)}(n-x) + (1-P_2-P_3)G^{(\ell)}(0,x,n)\Big\}$$

$$A_2 = \min_{m<x\leq n}\Big\{P_4H^{(\ell-1)}(n-x) + (1-P_4)G^{(\ell)}(m,x,n)\Big\}$$

[1] Recall that the superscript (ℓ) denotes any of the three collection objectives with parameter ℓ.

$$G^{(\ell)}(0,m,n) = 1+\min_{1\leq x<n}\Big\{P_5G^{(\ell)}(0,m-x,n-x) + P_6F^{(\ell-1)}(m-x,n-x) + (1-P_5-P_6)G^{(\ell)}(0,x,n)\Big\} \quad n\geq m\geq 2 \tag{5}$$

$$G^{(\ell)}(k,m,n) = 1+\min\{B_1,B_2\} \quad n\geq m\geq 2, \quad m-1\geq k\geq 1 \tag{6}$$

$$B_1 = \min_{1\leq x\leq k}\Big\{P_7G^{(\ell)}(k-x,m-x,n-x) + P_8F^{(\ell-1)}(m-x,n-x) + (1-P_7-P_8)G^{(\ell)}(0,x,n)\Big\}$$

$$B_2 = \min_{k<x\leq m}\Big\{P_9F^{(\ell-1)}(m-x,n-x) + (1-P_9)G^{(\ell)}(k,x,n)\Big\}$$

where P_0 = probability that no transmission occurs $= p^x$ and P_1 = probability that exactly one transmission occurs $= xp^{x-1}q$. The expressions for P_2 through P_9 are identical to those in [6, equations (5.1)-(5.4)] and are not repeated here.

The above recursive equations are applicable to all three collection objectives with the following boundary conditions:

- *L or Maximum*

$$H^{(0 \text{ or } max)}(n) = H^{(\ell \text{ or } max)}(0) = 0 \tag{7}$$

$$F^{(0 \text{ or } max)}(m,n) = 0 \tag{8}$$

$$F^{(\ell \text{ or } max)}(0,n) = H^{(\ell \text{ or } max)}(n) \tag{9}$$

$$F^{(\ell \text{ or } max)}(1,n) = 1 + H^{(\ell-1 \text{ or } max)}(n-1) \tag{10}$$

$$G^{(0 \text{ or } max)}(k,m,n) = G^{(\ell \text{ or } max)}(0,1,n) = 0 \tag{11}$$

$$G^{(\ell \text{ or } max)}(1,m,n) = 1 + F^{(\ell-1 \text{ or } max)}(m-1,n-1) \tag{12}$$

$$G^{(\ell \text{ or } max)}(k,2,n) = 2 + H^{(\ell-2 \text{ or } max)}(n-2) \tag{13}$$

- *L or Nothing*

  In addition to the boundary conditions for the *L or Maximum* objective we have the following.

$$H^{(\ell \text{ or } no)}(n) = F^{(\ell \text{ or } no)}(m,n) = G^{(\ell \text{ or } no)}(k,m,n) = 0, \quad n<\ell \tag{14}$$

- *L or More*

  The boundary conditions are the same as the ones for the *L or Nothing* objective with the exception that boundary conditions (7), (8) and (11) are replaced by the following.

$$H^{(0 \text{ or } more)}(n) = H^{(n \text{ or } max)}(n) \tag{15}$$

$$F^{(0 \text{ or } more)}(m,n) = F^{(n \text{ or } max)}(m,n) \tag{16}$$

$$G^{(0 \text{ or } more)}(k,m,n) = G^{(n \text{ or } max)}(k,m,n) \tag{17}$$

If the Group Testing collection protocol is used, a single solicitation message is sent which reaches all N respondents. Also, all respondents are initially in the binomial set. Thus, for all collection objectives y we have that:

$$C_{GT}^{(y)} = 1, \quad S_{GT}^{(y)} = N \quad \text{and} \quad D_{GT}^{(y)} = H^{(y)}(N) \tag{18}$$

B. Staged Group Testing

B.1 Fixed Groups

When fixed-group Staged Group Testing is used a set of M disjoint groups, g_i, for $i = 1, \ldots, M$ are given. The size of group g_i is given by n_i and $\sum_{i=1}^{M} n_i = N$. (We address the determination of the best such grouping shortly.) We let $\underline{n} = (n_1, n_2, \ldots, n_M)$. Both the collector and respondent solicitation costs will be a function of $\underline{n}$. We also define the integer $1 \leq J(k) \leq M$ as the smallest integer such that $\sum_{i=1}^{J(k)} n_i \geq k$ for $1 \leq k \leq N$. This represents the number of groups that need to be solicited if k respondents should be enabled. Using the definitions of the three collection objectives we obtain the following set of expressions:

- *L or Maximum*

$$C^{(L \text{ or } max)}(\underline{n}) = M \sum_{k=0}^{L-1} \binom{N}{k} p^{N-k} q^k + \sum_{k=L}^{N} J(k) \binom{k-1}{L-1} p^{k-L} q^L \qquad (19)$$

$$S^{(L \text{ or } max)}(\underline{n}) = N \sum_{k=0}^{L-1} \binom{N}{k} p^{N-k} q^k + \sum_{k=L}^{N} \sum_{i=1}^{J(k)} n_i \binom{k-1}{L-1} p^{k-L} q^L \qquad (20)$$

- *L or Nothing*

$$C^{(L \text{ or } no)}(\underline{n}) = \sum_{k=min(L,N-L+1)}^{N} J(k) \left\{ \binom{k-1}{N-L} p^{N-L+1} q^{k-1-N+L} + \binom{k-1}{L-1} p^{k-L} q^L \right\} \qquad (21)$$

$$S^{(L \text{ or } no)}(\underline{n}) = \sum_{k=min(L,N-L+1)}^{N} \sum_{i=1}^{J(k)} n_i \left\{ \binom{k-1}{N-L} p^{N-L+1} q^{k-1-N+L} + \binom{k-1}{L-1} p^{k-L} q^L \right\} \qquad (22)$$

- *L or More*

$$C^{(L \text{ or } more)}(\underline{n}) = M \sum_{k=L}^{N} \binom{N}{k} p^{N-k} q^k + \sum_{k=N-L+1}^{N} J(k) \binom{k-1}{N-L} p^{N-L+1} q^{k-1-N+L} \qquad (23)$$

$$S^{(L \text{ or } more)}(\underline{n}) = N \sum_{k=L}^{N} \binom{N}{k} p^{N-k} q^k + \sum_{k=N-L+1}^{N} \sum_{i=1}^{J(k)} n_i \binom{k-1}{N-L} p^{N-L+1} q^{k-1-N+L} \qquad (24)$$

It remains to determine the average delay cost (or the average number of response slots required for the collection objective to be satisfied). In order to capture the fact that the Group Testing procedure within any one stage operates with the knowledge of the number of the not-yet-solicited respondents, we define: $\mathcal{H}^{(y)}(n;t)$ as the average number of slots remaining in a stage when the binomial set is of size n, the number of not-yet-solicited respondents is t and the current collection objective is y. $\mathcal{F}^{(y)}(m,n;t)$ and $\mathcal{G}^{(y)}(k,m,n;t)$ are defined in a similar manner. The above three quantities are related in exactly the same way as the corresponding quantities in Section V.A. The boundary conditions for these quantities are essentially the same as those shown in Section V.A, except that (14) is replaced by the following:

$$\mathcal{H}^{(\ell \text{ or } no)}(n;t) = \mathcal{F}^{(\ell \text{ or } no)}(m,n;t) = \mathcal{G}^{(\ell \text{ or } no)}(k,m,n;t) = 0, \quad n+t < \ell \qquad (25)$$

We next investigate how the total cost of the fixed-group, Staged Group Testing may be optimized by selecting the appropriate fixed group sizes. The total cost satisfies the following recursive equation:

$$A^{(\ell)}(\underline{v}) = \alpha \mathcal{H}^{(\ell)}(v_1; \sum_{i=2}^{T} v_i) + \beta v_1 + \gamma + \sum_{x=0}^{v_1} A^{(\ell-x)}(\underline{v}^{-1}) \binom{v_1}{x} p^{v_1-x} q^x \qquad (26)$$

where $\underline{v} = (v_1, v_2, \ldots, v_T)$ and $\underline{v}^{-1} = (v_2, v_3, \ldots, v_T)$. The cost in equation (26) is derived as the sum of the delay cost for the first group (of size v_1) plus the cost of the collection protocol as it proceeds through the rest of the groups with a diminished collection objective. The above expression is applicable to all collection objectives with the following boundary conditions

$$A^{(j \text{ or } max)}(\underline{v}) = A^{(j \text{ or } no)}(\underline{v}) = 0 \qquad \text{for } \ j \leq 0 \qquad (27)$$

$$A^{(j \text{ or } more)}(\underline{v}) = A^{(j \text{ or } no)}(\underline{v}) = 0 \quad \text{for } \ j > \sum_{i=1}^{T} v_i \qquad (28)$$

$$A^{(\ell \text{ or } max)}(\underline{v}) = A^{(j \text{ or } more)}(\underline{v}) = A^{(0 \text{ or } more)}(\underline{v}) \ \text{ for } j \leq 0 \ \text{ and } \ell > \sum_{i=1}^{T} v_i \qquad (29)$$

$$A^{(\ell)}(v_1) = \alpha H^{(\ell)}(v_1) + \beta v_1 + \gamma \quad \text{for} \quad 0 < \ell \leq v_1 \qquad (30)$$

We now turn our attention to the determination of the best fixed grouping that will minimize the $A^{(\ell)}(\underline{n})$ for a given q, α, β and γ. One straightforward method is to enumerate all potential groupings of the N respondents and evaluate the cost of each using (26). The optimum grouping is the one with the minimum such cost.

The approach just described is obviously not feasible as it is prohibitively time consuming even for moderate values of N. We thus adopt a heuristic approach aimed at determining a near-optimal grouping. Our approach is based on the assumption that the optimal total cost for achieving objective y given the set of m respondents are subdivided into T groups, satisfies the following recursive relationship (based on (26)):

$$A_{opt}^{(\ell)}(\underline{v}_{T,m}) = \min_{0<v_1\le m-T+1}\Big\{\alpha\mathcal{H}^{(\ell)}(v_1;m-v_1)+\beta v_1 + \gamma + A_{opt}^{(rnd[\ell-qv_1])}(\underline{v}_{T-1,m-v_1})\Big\} \quad (31)$$

where $rnd[\bullet]$ rounds its argument to the nearest integer. Note that if a group of size v_1 is tested, an average of qv_1 responses are anticipated. Thus, the expression in (31) is based on the assumption that the optimal grouping for finding exactly ℓ is approximated by the optimum grouping for finding an average of ℓ.

The optimization procedure is thus as follows:

1. For each possible number of groups $M = 1,\ldots,N$ determine the grouping of the respondents into M groups using (31) and the following boundary conditions

$$A_{opt}^{(j\ or\ max)}(\underline{v}_{T,m}) = A_{opt}^{(j\ or\ no)}(\underline{v}_{T,m}) = 0,\ j\le 0 \quad (32)$$

$$A_{opt}^{(j\ or\ more)}(\underline{v}_{T,m}) = A_{opt}^{(j\ or\ no)}(\underline{v}_{T,m}) = 0,\ j> m \quad (33)$$

$$A_{opt}^{(\ell\ or\ max)}(\underline{v}_{T,m}) = A_{opt}^{(j\ or\ more)}(\underline{v}_{T,m}) = A_{opt}^{(0\ or\ more)}(\underline{v}_{T,m}),\ j\le 0,\ \ell>m \quad (34)$$

$$A_{opt}^{(\ell)}(\underline{v}_{1,m}) = \alpha H^{(\ell)}(m)+\beta m+\gamma \ \text{ for } 0<\ell\le m \quad (35)$$

Note that the values of the boundary conditions above are independent of the grouping used. Thus, whenever, while using (31), the value of A is evaluated using these boundary conditions, we assume that the T remaining groups are such that the last group contains $m-T+1$ respondents and the other $T-1$ groups contain one respondent each.

2. Choose the grouping (from among the N different ones produced in Step 1) that yields the lowest cost as evaluated by (26).

We can judge how near-optimal the grouping found using the heuristic above by comparing its cost to the cost of the best adaptive grouping (as determined in the next subsection). This latter cost is a lower bound on the best fixed-group cost.

B.2 Adaptive Groups

We define $A_{opt}^{(y)}(m)$ as the optimum total cost when the number of unsolicited respondents is m. Using the same arguments leading to (31) we can write:

$$A_{opt}^{(\ell)}(m) = \min_{0<n\le m}\Big\{\alpha\mathcal{H}^{(\ell)}(n;m-n)+\beta n+\gamma + \sum_{x=0}^{n} A_{opt}^{(\ell-x)}(m-n)\binom{n}{x}p^{n-x}q^{x}\Big\} \quad (36)$$

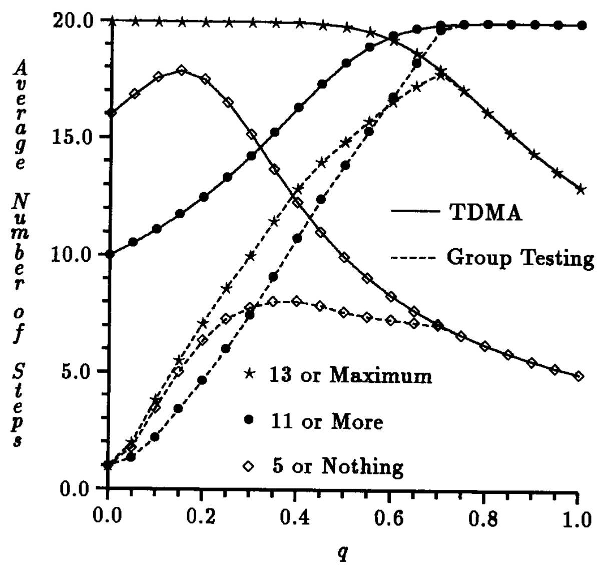

Fig. 2. Average Delay Cost for Various Collection Objectives (TDMA and Group Testing, $N = 20$)

where the following boundary conditions are satisfied:

$$A_{opt}^{(j\ or\ max)}(m) = A_{opt}^{(j\ or\ no)}(m) = 0 \quad \text{for } j\le 0 \quad (37)$$

$$A_{opt}^{(j\ or\ more)}(m) = A_{opt}^{(j\ or\ no)}(m) = 0 \quad \text{for } j> m \quad (38)$$

$$A_{opt}^{(j)}(0) = 0 \quad ; \quad A_{opt}^{(1)}(1) = \alpha+\beta+\gamma \quad (39)$$

$$A_{opt}^{(\ell\ or\ max)}(m) = A_{opt}^{(j\ or\ more)}(m) = \alpha H^{(0\ or\ more)}(m)+\beta m+\gamma,\ j\le 0,\ \ell>m \quad (40)$$

VI. Numerical Examples

A. Group Testing

Group Testing (in a single stage) incurs a respondent and collector solicitation costs of N and 1, respectively, regardless of the value of q. Fig. 2 shows, for $N = 20$, the average delay cost as a function of q for the three collection objectives. In the same figure we also plot the average delay cost of the single stage TDMA protocol (derived in [24]). Observe that for the entire range of q the use of Group Testing provides for a lower or equal delay cost than a single stage TDMA protocol. Also, the Group Testing collection procedure adapts to the (single stage) TDMA procedure (i.e., in each slot a group of size 1 is enabled) when $q \ge \frac{1}{\sqrt{2}}$. (This was found to be true in all the numerical experiments we conducted. No formal proof is available yet.)

B. Staged Group Testing

The respondent and collector solicitation costs of the Staged Group Testing procedure are given by expressions (19)-(24). For the L or Maximum collection objective ($N = 20, L = 13$), the variations of these costs with q

are shown in Fig. 3, where the 20 respondents are subdivided into four groups of sizes (6,6,4,4). The delay cost of the Staged Group Testing procedure for the same grouping of the respondents is also shown in Fig. 3. Comparing this delay to that incurred by a single stage Group Testing procedure (as shown in Fig. 2) we observe that:

- For values of $q \geq \frac{1}{\sqrt{2}}$ the delay cost for the two approaches is the same since in both the protocols adapt to a single stage TDMA procedure.
- For values of $q < \frac{1}{\sqrt{2}}$, the delay incurred is lower when single stage Group Testing is used. In particular, at the limit as q approaches zero, the single stage Group Testing procedure needs only one group test to determine that the respondents are not active, whereas the Staged Group Testing procedure needs a number of group tests equal to the number of groups of respondents.

Table I shows the grouping of 15 respondents in a fixed-group, Staged Group Testing procedure for various values of the cost weights and for the L or Maximum collection objective. The table also shows the cost when an optimal adaptive-group, Staged Group Testing procedure is used. The fixed grouping is determined using the heuristic in Section V.B.1. Observe that the near-optimal fixed groupings achieve the same or slightly higher cost than the optimal adaptive groupings.

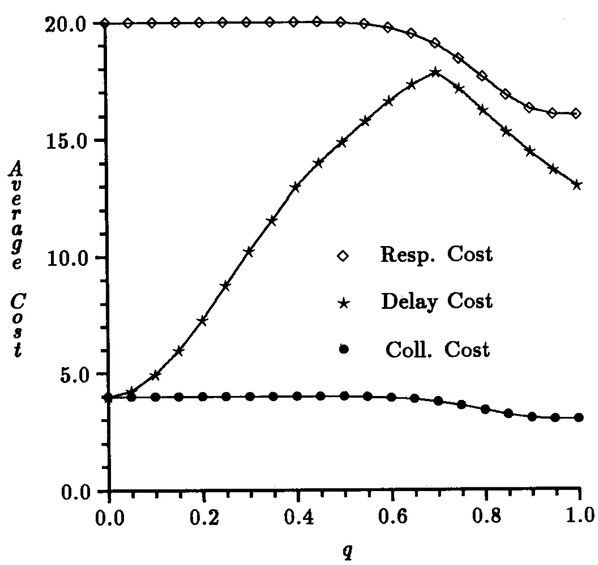

Fig. 3. Average Delay Cost, Respondent Solicitation Cost and Collector Solicitation Cost for the L or Maximum Collection Objectives (Fixed Group Staged Group Testing, $N = 20, L = 13$)

VII. Concluding Remarks

In this paper we have considered response collection strategies that can be used over a multiple access channel. We were motivated by some distributed computing applications to define a set of collection objectives. Five protocols that can be used to achieve these collection objectives were investigated: Polling, TDMA, Staged TDMA, Group Testing and Staged Group Testing. In analyzing the performance of these protocols, three cost components were taken into account: the number of steps required to complete the objective, the number of solicitations required by the collector, and the the number of respondents receiving solicitation messages. The idea of staging stems from the inclusion of the latter two cost components and from the fact that the response collection procedure will terminate once the collection objective has been achieved.

Our findings are summarized in Table II where we use the terms "low", "medium", and "high" to denote relative values of the costs. Our conclusion is that, in general, a suitably optimized adaptive-group, Staged Group Testing protocol can achieve the best performance. A near optimal fixed-group Staged Group Testing procedure can achieve almost similar performance but can be easier to implement as the groups are determined a priori.

Although we have assumed an error free environment for our analysis we do not expect the presence of errors to affect our conclusions regarding the relative merits of our proposed protocols. In general, errors might cause a collector to either (1) spend more time achieving its objective, or (2) declare that a collection objective is unreachable when in fact it is achievable. The most straightforward approach to dealing with this latter problem is to require that a collector retry collecting responses in a second round even if the first round fails. Whether or not this is desirable depends on the error rates expected. The specification and analysis of such protocols should be explored in future research. The work in [22] describes and analyzes a similar approach in a somewhat different context.

References

[1] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *The Bell Systems Technical Journal*, pages 1179–1252, September 1959.

[2] F. K. Hwang. On finding a single defective using binomial group testing. *Journal of the American Statistical Association*, 69:146–150, March 1974.

[3] M. R. Garey and F. K. Hwang. Isolating a single defective using group testing. *Journal of the American Statistical Association*, 69:151–153, March 1974.

[4] M. C. Hu, F. K. Hwang, and J. K. Wang. A boundary problem for group testing. *SIAM Journal of Algebra and Discrete Mathematics*, 2:81–87, June 1981.

[5] J. Wolf. Born again group testing: Multiaccess communications. *IEEE Transactions on Information Theory*, 31:185–191, March 1985.

[6] T. Berger, N. Mehravari, D. Towsley, and J. Wolf. Random multiple-access communication and group testing. *IEEE Transactions on Communications*, 32:769–779, July 1984.

[7] N. K. Garg and S. Mohan. Group testing with capture for random access communication. *IEEE Transactions on Communications*, 35:849–854, August 1987.

[8] D. Kurtz and M. Sidi. Multiple-access algorithms via group testing for heterogeneous popoulation of users. *IEEE Transactions on Communications*, 36:1316–1323, December 1988.

[9] T. K. Sellis. Multiple-query optimization. *ACM Transactions on Database Systems*, 13(1):23–52, March 1988.

[10] M. H. Ammar. Teletext-like information delivery using broadcast polling. *Computer Networks and ISDN Systems*, 12(2):107–115, March 1987.

TABLE I
NEAR-OPTIMAL FIXED GROUP AND OPTIMAL ADAPTIVE GROUP STAGED GROUP TESTING (L OR MAXIMUM)

L	q	α	β	γ	Opt. fixed groups	FG cost	AG cost
3	.85	1.0	10.0	1.0	3 1 ...1	40.35	40.29
1	.90	1.0	10.0	1.0	1 ...1	13.33	13.33
1	.10	1.0	10.0	1.0	2 2 2 1 ...1	93.35	92.09
1	.10	1.0	.10	1.0	15	4.33	4.33
8	.80	1.0	.10	0.0	10 3 1 1	12.45	12.28
8	.10	1.0	10.0	1.0	15	153.86	153.86

TABLE II
RELATIVE PERFORMANCE OF THE VARIOUS PROTOCOLS

Cost Measure	Polling	TDMA	Staged TDMA	Group Testing	Staged Group Testing
Delay Cost (no. of slots)	High	High (=Polling)	High (=Polling)	Low	Medium
Respondent Solicitation Cost	Low	High ($= N$)	Medium	High ($= N$)	Medium (= staged TDMA, for same grouping)
Collector Solicitation Cost	High	Low (=1)	Medium	Low (=1)	Medium (= staged TDMA, for same grouping)

[11] M. H. Ammar and H. J. Kim. Prototyping a broadcast delivery information system. Technical Report GIT-ICS-90/16, Georgia Institute of Technology, Atlanta, GA, 1990.

[12] D. Gifford. Polychannel systems for mass digital communication. *Communications of the ACM*, 33(2):141–151, February 1990.

[13] G. Herman, G. Gopal, K. Lee, and A. Weinrib. The datacycle architecture for very high throughput databases. In *Proceedings of SIGMOD*, pages 97–103. ACM, 1987.

[14] J. W. Wong and M. H. Ammar. Analysis of broadcast delivery in a videotex system. *IEEE Transactions on Computer*, 34(9):863–866, September 1985.

[15] L. Lamport. The implementation of reliable distributed multiprocess systems. *Computer Networks*, 2:95–114, 1978.

[16] D. Gifford. Weighted voting for replicated data. In *Proceedings of 7th Symposium on Operating Systems*, pages 150–162. ACM, 1979.

[17] M. Ahamad and M. H. Ammar. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–496, April 1989.

[18] D. Barbara and H. Garcia–Molina. Mutual exclusion in partitioned distributed systems. *Distributed Computing*, 1:119–132, 1986.

[19] S. Y. Cheung, M. Ahamad, and M. H. Ammar. Optimizing vote and quorum assignments for reading and writing replicated data. *IEEE Transactions on Knowledge and Data Engineering*, 1(3):387–397, September 1989.

[20] S. Mullender and Paul Vitanyi. Distributed match-making. *Algorithmica*, 3:367–391, 1988.

[21] R. Fowler. *Decentralized Object Finding Using Forwarding Addresses*. PhD thesis, University of Washington, 1985.

[22] J. Bernabeu, M. H. Ammar, and M. Ahamad. Optimal selection of multicast groups for resource location in a distributed system. In *Proceedings of INFOCOM '89*, pages 312–321. IEEE, 1989.

[23] M. Ahamad. An efficient algorithm for name resolution in a computer network. *Computer Networks and ISDN Systems*, 13(4):301–312, August 1987.

[24] M. H. Ammar and G. N. Rouskas. On the performance of protocols for collecting responses over a multiple-access channel. In *Proceedings of INFOCOM '91*, pages 1490–1499. IEEE, April 1991.

Mostafa H. Ammar (S'83-M'85) received his Ph.D. Degree in Electrical Engineering from the University of Waterloo in Ontario, Canada, 1985. His S.M.(1980), S.B.(1978) degrees were acquired in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, Cambridge, MA. Dr. Ammar is currently an Associate Professor in the College of Computing at Georgia Tech. He has been with Georgia Tech since 1985. For the years 1980-82 he worked at Bell-Northern Research (BNR), first as a Member of Technical Staff and then as Manager of Data Network Planning.

Dr. Ammar's research interests are in the areas of computer network architectures and protocols, distributed computing systems, and performance evaluation. He is the co-author of the textbook *Fundamentals of Telecommunication Networks*, published by John Wiley and Sons. Dr. Ammar is the holder of a 1990-1991 Lilly Teaching Fellowship and received the 1993 Outstanding Faculty Research Award from the College of Computing. He is a member of the editorial board of *Computer Networks and ISDN Systems*. He is a member of ACM and a member of the Association of Professional Engineers of the Province of Ontario, Canada.

George N. Rouskas (S'92-M'95) received the Diploma in Electrical Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, in 1991 and 1994, respectively. He was the recipient of the 1994 College of Computing Graduate Research Assistant Award.

He joined the Department of Computer Science, North Carolina State University, in 1994, where he is currently an Assistant Professor. His research interests include lightwave network architectures, high-speed networks, and performance evaluation. He is a member of ACM and of the Technical Chamber of Greece.