SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*

DYNAMIC CONTROL OF SESSION INPUT RATES IN COMMUNICATION NETWORKS

5. TYPE OF REPORT & PERIOD COVERED

Technical Report

6. PERFORMING ORG. REPORT NUMBER

LIDS-P-1228

7. AUTHOR(s)

Eli Gafni and Dimitri P. Bertsekas

8. CONTRACT OR GRANT NUMBER(s)

ARPA Contract
ONR-N00014-75-C-1183

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Massachusetts Institute of Technology
Laboraotry for Information and Decision Systems
Cambridge, Massachusetts 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

Program Code No. 5T10
ONR Identifying No. 049-383

11. CONTROLLING OFFICE NAME AND ADDRESS

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

12. REPORT DATE

August 1982

13. NUMBER OF PAGES

5

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

Office of Naval Research
Information Systems Program
Code 437
Arlington, VA 22217

15. SECURITY CLASS. *(of this report)*

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

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

We consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. The algorithm receives periodically information regarding the level of congestion along each session path and iteratively corrects the session input rate. In this paper we place emphasis on voice networks but the ideas involved can also be used for dynamic routing and flow control in data networks.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Proc. of MILCOM '82, Oct. 82, Boston, Mass.

August 1982 LIDS-P-1228

DYNAMIC CONTROL OF SESSION INPUT RATES IN COMMUNICATION NETWORKS*

Eli Gafni and Dimitri Bertsekas

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
Cambridge, Massachusetts 02139

ABSTRACT

We consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. The algorithm receives periodically information regarding the level of congestion along each session path and iteratively corrects the session input rate. In this paper we place emphasis on voice networks but the ideas involved can also be used for dynamic routing and flow control in data networks.

1. INTRODUCTION

Recently a group at Lincoln Laboratory (ref. 1) has introduced an interesting Voice Coder (Vocoder) scheme. The proposed Vocoder uses a digitization method, dubbed "embedded coding". Essentially, a segment of talkspurt is coded into packets of different priority levels. The higher priority packets contain the "core" of the speech while the lower priority packets contain the information that "fine tunes" it. This coding schemes allows for the implementation of a sophisticated flow control mechanism.

While traditional voice flow control mechanisms use blocking either by preventing the initiation of a call or by discarding small segments of it, when the call is already in progress, the embedded coding scheme allows for the alleviation or prevention of congestion by dynamically trading off between voice quality and congestion, by discarding the lower "priority" packets either at the point of congestion or the point of entry. The level of congestion at which the gaps between the segments, delivered by the traditional schemes, render the speech unintelligible is much lower than the one at which the embedded coding scheme delivers unintelligible information. This flexibility in exercising flow control makes the embedded coding scheme attractive.

Alleviation and prevention of congestion by discarding lower priority packets at the point of entry seems to be superior to discarding them at the point of congestion. The later amounts to a waste of network resources. But, it would not be advisable to forgo the capability of discarding lower priority packets at the point of congestion, because of the time delay involved in making the entry points aware of the congestion build-up situation. As a result, we advocate the use of the two capabilities in complementary roles, in analogy to the complementary roles of quasi-static routing and flow control of data. The rates at the entry points will be determined upon longer time averages of congestion levels while the capability of discarding packets at the point of congestion will serve to alleviate intolerable momentary congestion. The rates at the entry point will be adjusted so that the capability of discarding packets at the point of congestion will not be exercised too often.

In this paper we discuss a method of determining the input rates at the entry point. To this end we will ignore the capability of discarding packets within the network in order to simplify the analysis.

We are interested in an algorithm that will adapt the input rate to the changing flows in the network, resulting from the initiation and termination of sessions. As in quasi-static routing we employ an "on-line" iterative algorithm that will solve a static problem. The hope is that the algorithm converges fast enough relative to the sessions initiation and termination process, and as a result will be able to "track" its variation keeping the rates in the ballpark of the optimal rates at all times.

The criterion used to determine input rates is based on the notion of "fair allocation" introduced in Section 2. In Section 3 we introduce the algorithm and describe its convergence properties. Proofs of all the results stated may be found in (ref. 2). The same reference describes distributed implementation issues as well as a method for adjusting the input rates through the use of windows for the case where the algorithm is applied to data networks.

2. PROBLEM FORMULATION

Let N denote a network with nodes 1,2,...,N and let L be a set of directed links connecting the nodes. With each link $a \in L$ we associate a number C_a, called the capacity of link a. Let S denote a set of sessions taking place between nodes. Each session $s \in S$ has an origin node, a destination node and a simple path p_s leading from the origin node to the destination node. Define

*Work supported by Defense Advanced Research Projects Agency under Contract No. ONR/N00014-75-C-1183.

$$1_{p_s}(a) = \begin{cases} 1 \text{ if a belongs to } p_s \\ 0 \qquad \text{otherwise.} \end{cases} \tag{2.1}$$

If γ_s is the input rate of session s, then the total flow of link a will be

$$F_a = \sum_{s\varepsilon S} \gamma_s 1_{p_s}(a).$$

The problem broadly stated is to choose a vector of session input rates $\gamma = (\ldots,\gamma_s,\ldots)$ which results in a set of "satisfactory" link flows F_a, $a\varepsilon L$ (for example less than some fraction of the link capacity C_a), and at the same time maintains a certain degree of "fairness" for all sessions.

It is customary to consider as one of the characteristics of a fair allocation of resources in a network, the feature that it is indifferent to the geographical separation of the session's origin and destination. Although there might be different priorities assigned to sessions, these priorities are not assigned on the basis of geographical separation. Moreover, two sessions of the same priority should obtain the same rate, if the rate of one can be traded for the rate of the other. This is in the spirit of making the network "transparent" to the user. The user should have no idea of the length of the path assigned to him through the rate allocated to him.

To capture the notions of fairness and priority as presented above we define the notion of "fair allocation":

Let Q be a totally ordered set and let X be a given subset of Q^n. A vector $b = (b^1,\ldots,b^n)$ is said to be lexicographically less than or equal to the vector $d = (d^1,\ldots,d^n)$ if $b^i > d^i$ implies the existence of $j < i$ such that $b^j < d^j$. The vector $x = (x^1,\ldots x^n) \varepsilon X$ is called a <u>fair allocation over X</u> if for each $y\varepsilon X$ there exists a permutation $\tilde{x}$ of x which is lexicographically greater than or equal than all permutations $\tilde{y}$ of y.

If we consider the set X as a "feasible" set, a fair allocation vector x over X solves an hierarchy of nested problems. The first one maximizes the minimal entry of vectors in X. The second maximizes the second minimal entry of all vectors which solve the first problem, etc.

The usual difficulty with such a problem is that in order to solve the jth subproblem in the hierarchy, the solutions to the preceding subproblems must be available. This was the case in (ref. 5) where the solution to a fair allocation problem was obtained by solving a nested sequence of linear problems. Our iterative algorithm has the advantage that it solves all the subproblems in the hierarchy simultaneously.

Hayden (ref. 4) proposed a quasi-static distributed algorithm which results in a vector $\gamma = (\ldots,\gamma^s,\ldots)$ which is a fair allocation over the set defined by

$$F_a \le \rho \cdot C_a \qquad \forall\, s\varepsilon S,\ \forall\, a\varepsilon L \tag{2.2}$$

where $o<\rho<1$ is a certain constant, usually taken to be 0.8. Jaffe (ref. 3) proposed a distributed nonquasi-static algorithm resulting in a vector such that the vector $(\ldots,\gamma^s/\beta^s,\ldots)$ is a fair allocation over the set defined by

$$\gamma^s/\beta^s \cdot 1_{p_s}(a) \le C_a - F_a \quad \forall\, s\varepsilon S,\ \forall\, a\varepsilon L \tag{2.3}$$

where β^s is some constant associated with session $s\varepsilon S$.

The rationale behind (2.2) is quite simple: we do not allow the total flow of each link to be more than some fraction of the total capacity. The rationale behind (2.3) is more sophisticated. Primarily, it allows us to accomodate fluctuations of a session rate which are a function of the rate, and in addition, it enables us to establish preferences among sessions.

While Jaffe's algorithm is not iterative and suitable for distributed operation, Hayden's may result in transient flows that are much larger than the capacity available to accommodate them (for an example see (ref. 2)). We generalize the set defined by (2.3) in the following way:

Let $g_a: R^+ \to R^+$ be a function associated with link $a\varepsilon L$. Let $f_s: R^+ \to R^+$ be a function associated with session $s\varepsilon S$ and which possesses an inverse f_s^{-1}. We are interested in a quasi-static algorithm which will result in a vector γ such that the vector $(\ldots,f_s^{-1}(\gamma^s),\ldots)$ is a fair allocation over the set defined by:

$$f_s^{-1}(\gamma^s)\cdot 1_{p_s}(a) \le g_a(C_a - F_a)\cdot 1_{p_s}(a) \quad \forall\, s\varepsilon S,\ \forall\, a\varepsilon L, \tag{2.4}$$

$$F_a \le C_a \qquad \forall\, a\varepsilon L, \tag{2.5}$$

$$\gamma^s \ge 0 \qquad \forall\, s\varepsilon S. \tag{2.6}$$

The introduction of the function $f_s(\cdot)$ allows us to assign different priorities to different sessions--compare with (2.3) and the scalar β_s. To see the role of g_a, assume that f_s is the identity function. Depending on the length of time over which the rate of a session is measured, we can have two interpretations of the role of our algorithm. Both interpretations suggest the same type of form for the function g_a.

In our first interpretation, the length of time over which the rate is averaged is relatively short with respect to the "time constant" of the counting process of the number of off-hook speakers which are currently at the talkspurt mode. Since about 30% of a talkspurt is silence and some segment of the talkspurt needs more encoding than others, we view the bit rate generated by the Vocoder for session $s\varepsilon S$ as a stochastic process with mean γ^s--the rate assigned to user $s\varepsilon S$. This amounts to the assumption that the Vocoder has the means of dynamically reconfiguring to the demands of the voice to achieve the desired average rate.

Suppose that we want to reserve excess capacity on each link so as to be able to accommodate at least a variation as large as the standard deviation of the flow on the link. Assume that the standard deviation of the rate of session $s \in S$ which was allocated an average rate γ^s is $\beta \cdot \gamma^s$ for some $0<\beta<1$. Let $s' \in S$ be such that

$$s' = \arg \max_{t \in S} \gamma^t \cdot 1_{p_t}(a) \quad , \tag{2.7}$$

then, by the independence of rates of different sessions we get (abusing notations)

$$\begin{aligned} \sigma(F_a) &\leq (C_a/\gamma^{s'})^{1/2} \, \sigma(\gamma^{s'}) \\ &\leq (C_a/\gamma^{s'})^{1/2} \beta \cdot \gamma^{s'} \leq (C_a)^{1/2} \beta \cdot (\gamma^{s'})^{1/2} \\ &\leq \beta \cdot (C_a)^{1/2} [g_a(C_a - F_a)]^{1/2} \end{aligned} \tag{2.8}$$

where the last inequality follows from (2.4). Thus if we take

$$g_a(C_a - F_a) = \frac{1}{\beta^2 C_a} (C_a - F_a)^2 \quad \forall \, a \in L \tag{2.9}$$

we obtain $\sigma(F_a) \leq C_a - F_a$ and we are guaranteed to accommodate the standard deviation of the flow resulting from the fair allocation.

In the second interpretation, the length of time over which the rate is averaged is relatively long with respect to the "time constant" of the counting process of the number of off-hook speakers in talkspurt mode. In this case we deal concurrently with all the off-hook sessions and want to be able to accommodate the standard deviation around the mean of the process (i.e. the instantaneous effect of the number of speakers at the talkspurt mode is washed out by the long time average). Let q be the fraction of time a speaker is in the talkspurt mode and assume his rate while in the talkspurt mode is constant. Then using notations as before

$$\begin{aligned} \sigma(F_a) &\leq \sum_{t \in S} [(\gamma^t/q)^2 \cdot q(1-q) \cdot 1_{p_t}(a)]^{1/2} \\ &\leq (C_a/\gamma^{s'})^{1/2} \gamma^{s'} \frac{[q(1-q)]^{1/2}}{q} \\ &< \left(\frac{1-q}{q} C_a\right)^{1/2} [g_a(C_a - F_a)]^{1/2}. \end{aligned} \tag{2.10}$$

Again by choosing g_a as in (2.9) with $\beta = \left(\frac{q}{1-q}\right)^{1/2}$ we obtain $\sigma(F_a) \leq C_a - F_a$.

The point we want to make by the above arguments is the need to allow g_a to be a nonlinear function, which may depend on C_a, rather than only on the excess capacity as (2.3) implies. The exact role of g_a is up to the network designer to decide, and our formulation allows him a great deal of flexibility in this regard.

As will be explained in the final section, in order to carry out the algorithm of the next section we have to store in link $a \in L$, the functions g_a, and f_s for all s traversing a. This is not too difficult if there are few "priority" classes of sessions and correspondingly few possibilities for f_s. All that a link has to know in this case is merely the class number of each session traversing it.

3. THE ALGORITHM

We will state the algorithm in a centralized context.

Assume that γ_k^s is given for all $s \in S$ and that

$$0 \leq F_a < C_a \qquad \forall \, a \in L \tag{3.1}$$

then γ_{k+1}^s is determined by

$$\gamma_{k+1}^s = \min_{a \in A_s} [\gamma_k^s + \alpha_k^a [f_s g_a (C_a - F_a^k) - \gamma_k^s]] \quad \forall \, s \in S \tag{3.2}$$

where A_s is the set of links traversed by session s, α_k^a is a scalar to be specified later and $f_s g_a(\cdot)$ denotes $f_s(g_a(\cdot))$.

We make the following assumptions concerning g_a and f_s:

Assumption (A): $g_a(\cdot)$ and $f_s(\cdot)$ for all $a \in L$ and $s \in S$ are monotonically non-decreasing.

Assumption (B): $f_s \cdot g_a(\cdot)$ is convex (concave is possible too, but will not be pursued) differentiable with

$$f_s g_a(0) = 0$$

$$f_s g_a(C_a) \triangleq m_{sa} < \infty$$

for all $s \in S$ and $a \in L$.

There are two options for choosing α_k^a. The first

$$\alpha_k^a = \frac{1}{1 + \dfrac{\sum_{t \in S} [m_{ta} - f_t g_a (C_a - F_a^k)] \cdot 1_{p_t}(a)}{F_a^k}} \qquad \forall \, a \in L, \; k=1,2,\ldots \tag{3.3}$$

and the second

$$\alpha_k^a = \frac{1}{1 + \sum_{t \in S} f_t g_a (C_a - F_a) \cdot 1_{p_t}(a)} \; . \tag{3.4}$$

The step size (3.3) is not defined for

$$F_a^k = 0.$$

In such a case we use the step size (3.4).

For the two options of step sizes (3.3), (3.4) we have the following lemma whose proof may be found in (ref. 2).

<u>Lemma 1</u>: Let γ_k satisfy (3.1) and let γ_{k+1} be determined from γ_k by (3.2) with α_k^a as in (3.3) or (3.4). Then under Assumptions (A) - (B)

$$0 \leq \gamma_{k+1}^t,\ F_a^{k+1} < C_a \qquad \forall\ t\epsilon S,\ \forall\ a\epsilon L \qquad (3.5)$$

and

$$\limsup_{k\to\infty} F_a^k \leq \sum_{t\epsilon S} f_t g_a\ (C_a - \limsup_{k\to\infty} F_a^k)\cdot 1_{P_t}(a) \qquad \forall\ a\epsilon L. \qquad (3.6)$$

The idea behind the choice of the expressions (3.3) and (3.4) as well as the simple intuition behind Lemma 1 can be best explained by the use of Figures 1 and 2. Let the function $G_a(\cdot)$: $R^+ \to R^+$ be given by

$$G_a(F_a) = \sum_{s\epsilon S} f_s g_a (C_a - F_a)\cdot 1_{P_s}(a)$$

The two figures depict the relations between F_a^k and F_a^{k+1}, as if the network consisted of the single link a. When this is not the case, $(3.2)_{k+1}$ implies that we have at most overestimated F^{k+1}. In Figure 1 that corresponds to α_k^a as in (3.3), F_a^{k+1} is determined by intersection of the line connecting the point $(0, G_a(0))$ with the point $(F_a^k, G_a(F_a^k))$, with the line $y = F_a$. In Figure 2 that corresponds to α_k^a as in (3.4), F_a^{k+1} is determined by intersecting the tangent to the graph of $G_a(F_a)$ at the point $(F_a^k, G_a(F_a^k))$, with the line $y=F^a$. The reader can easily convince himself that $\limsup_{k\to\infty} F_a^k$ must, in both cases, lie in the area where

$$F_a \leq G_a(F_a)$$

which gives rise to the lemma.

As for the possible functions for f_s and g_a, Assumption (B) is not too restrictive since it will be satisfied for instance when both f_s and g_a are convex increasing on $(0,\infty)$. Figure 3 shows why just monotonicity of $G_a(\cdot)$ is not sufficient for the lemma to hold

We can now state the main result of this paper. The proof is given in [2].

<u>Proposition</u>: Under Assumptions (A), (B), with α_k^a as in (3.3) or (3.4), the sequence $\{\gamma_k\}$, generated by (3.2) with γ_0 satisfying (3.1), converges to a vector $\bar{\gamma}$. Moreover the vector $(\ldots, f_s^{-1}(\bar{\gamma}^s), \ldots)$ is a fair allocation over the set defined by (2.4)-(2.6).

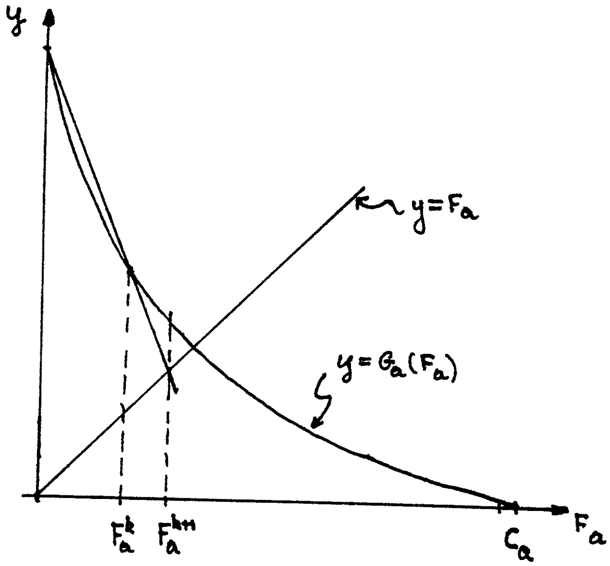

Figure 1

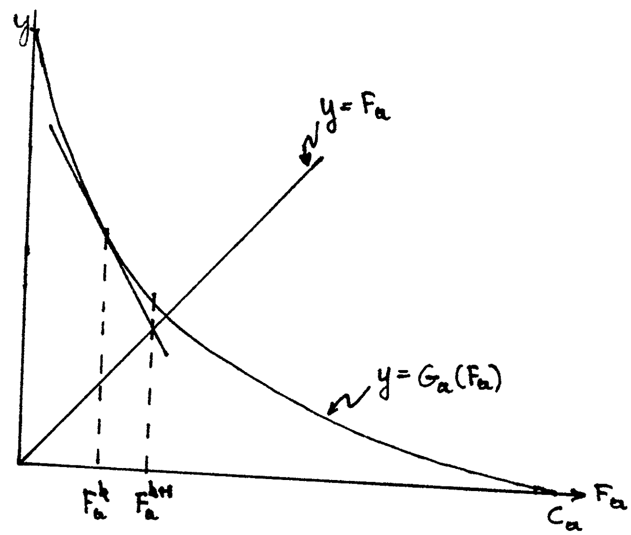

Figure 2

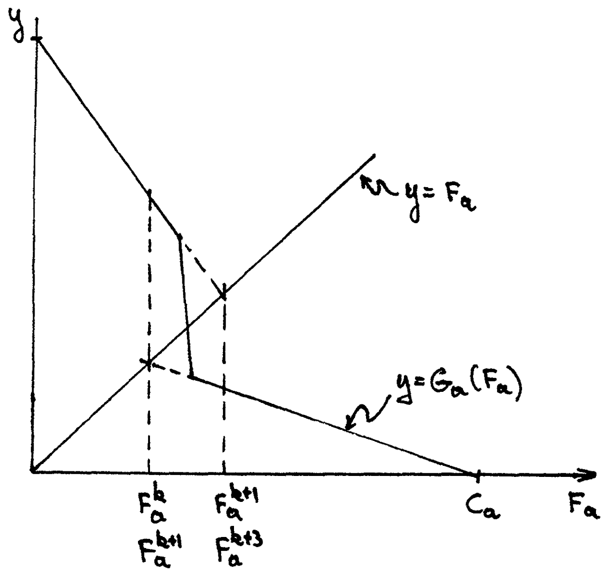

Figure 3

REFERENCES

[1] T. Bially, B. Gold, and S. Seneff, "A Technique for Adaptive Voice Flow Control in Integrated Packet Networks", IEEE Trans. Comm., Vol. COM-28, March 1980.

[2] E. M. Gafni, "The Integration of Routing and Flow Control for Voice and Data in Computer Communication Network", Ph.D. Dissertation, Dept. of Elect. Eng. and Comp. Science, Mass. Inst. of Tech., Cambridge, Mass., Aug. 1982.

[3] J. M. Jaffe, "A Decentralized Optimal, Multiple-User Flow and Control Algorithm", Proceeding of Fifth Conference on Computer Communication (ICCC-80), pp. 839-844, Atlanta, GA, Nov. 1980.

[4] H. Hayden, "Voice Flow Control in Integrated Network", M.S. Thesis, Dept. of Elec. Eng. and Comp. Science, Mass. Inst. of Tech., Cambridge, Mass., June 1981.

[5] F. Ros-Peran, "Routing to Minimize the Maximum Congestion on a Computer Network", Ph.D. Dissertation, Department of Elec. Eng. and Comp. Science, Mass. Inst. of Tech., Cambridge, Mass., Feb. 1979.

Distribution List

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314 — 12 Copies

Assistant Chief for Technology
Office of Naval Research, Code 200
Arlington, Virginia 22217 — 1 Copy

Office of Naval Research
Information Systems Program
Code 437
Arlington, Virginia 22217 — 2 Copies

Office of Naval Research
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210 — 1 Copy

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605 — 1 Copy

Office of Naval Research
Branch Office, Pasadena
1030 East Greet Street
Pasadena, California 91106 — 1 Copy

Naval Research Laboratory
Technical Information Division, Code 2627
Washington, D.C. 20375 — 6 Copies

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380 — 1 Copy

Office of Naval Research
Code 455
Arlington, Virginia 22217 — 1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217 — 1 Copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152 — 1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084 — 1 Copy

Captain Grace M. Hopper
Naval Data Automation Command
Code OOH
Washington Navy Yard
Washington, DC 20374 — 1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209 — 1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217 — 1 Copy

Prof. Fouad A. Tobagi
Computer Systems Laboratory
Stanford Electronics Laboratories
Department of Electrical Engineering
Stanford University
Stanford, CA 94305