AD-769 706

COMPUTER NETWORK RESEARCH

Leonard Kleinrock, et al

California University

Prepared for:

Advanced Research Projects Agency

30 June 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED
Security Classification

AD 769 706

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Computer Science Department School of Engineering and Applied Science 90024 405 Hilgard, University of California, Los Angeles	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Computer Network Research

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Leonard Kleinrock

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 30, 1973	67 ~~64~~	72

8a. CONTRACT OR GRANT NO. DAHC15-73-C-0368 b. PROJECT NO. 2496 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

ARPA Semiannual Technical Report January 1, 1973 through June 30, 1973.

ia

Sponsored by

ADVANCED RESEARCH PROJECTS AGENCY

COMPUTER NETWORK RESEARCH

SEMIANNUAL TECHNICAL REPORT

June 30, 1973

ARPA Contract DAHC-15-73-C-0368*

ARPA Order No. 2496*
Program Code No. 3P10*

(*Note: Until 15 June 1973, Contract DAHC-15-69-C-0285
ARPA Order No. 1380, Program Code No. 9D30)

Principal Investigator: Leonard Kleinrock

Co-Principal Investigators: Gerald Estrin
Michel Melkanoff
Richard R. Muntz
Gerald Popek

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

ib

COMPUTER NETWORK RESEARCH

Advanced Research Projects Agency
Semiannual Technical Report

June 30, 1973

1. INTRODUCTION

This Semiannual Technical Report covers the period from January 1 through June 30, 1973. Our efforts have been in four major areas: advanced packet-switching systems, including multiaccess satellite and packet radio systems; computer communication network design; multiple resource multiple access computer systems models; and measurements on the ARPANET itself. In addition, we have been involved with some network protocol studies and some controlled access and security questions. The results of that research have been documented and are listed in Section 2 following.

In this report we have attached three of our publications which have appeared in the professional literature; we do not include material from other areas of activity, mentioned above, in this document, and the reader is referred to the referenced publications themselves.

The first paper we include in Section 3 below has to do with "Packet Switching in a Slotted Satellite Channel," by L. Kleinrock and S. S. Lam (AFIPS Conference Proceedings, 1973 National Computer Conference and Exposition, June 4-8, 1973, New York, N.Y., pp. 703-710). In this paper the basic behavior of throughput and delay were studied for some multiaccess schemes for satellite communications in a packet switching network. These schemes permit a number of earth stations to simultaneously access the capacity of a shared satellite channel, thereby extending the multiplexing principles of packet switching to satellites. Two related papers presented by others at the NCC session on satellites were "Dynamic Allocation of Satellite Capacity through Packet Reservation," by L. G. Roberts, and "Packet Switching with Satellites," by N. Abramson.

A second paper included below and entitled "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," by L. Fratta, M. Gerla, and L. Kleinrock (Networks, 3:97-133, 1973), summarizes some of the major concepts of the flow deviation method for computer network design; this method was discussed in the previous Semiannual Technical Report (December 31, 1972), but the current paper delves into the foundations more deeply. The flow deviation method leads to an efficient design procedure for networks. This paper is included as Section 4.

The fifth section contains the paper "On Non-Blocking Switching Networks," by D. G. Cantor (Networks, 1:367-377, 1972). The problem discussed is that of finding switching networks which are guaranteed to be non-blocking, in that any idle input terminal may always be connected to any idle output terminal. This is a basic problem in circuit switching and is the starting point for some of our studies comparing circuit switching to message switching.

2. LIST OF PUBLICATIONS

1. Baskett, F., and R. R. Muntz, "Networks of Queues," Proc. Seventh Annual Princeton Conference on Information Sciences and Systems, Princeton University, March 22-23, 1973.

2. Cantor, D. G., "On Non-Blocking Switching Networks," Networks, 1:367-377, 1972.

3. Cerf, V. G., "The Current Flow-Control Scheme for IMPSYS," ARPA Network Request for Comments #442, January 1973.

4. Crocker, D. C., "TELNET Extended Options List Option," ARPA Network Information Center Catalog #16239, May 1973.

5. Crocker, D. C. "TELNET Status Option," ARPA Network Information Center Catalog #16237, May 1973.

6. Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," Networks, 3:97-133, 1973.

7. Iseli, J., and D. C. Crocker, "Responding to User Needs," ARPA Network Request for Comments #462, February 1973.

8. Kleinrock, L., "Computer Networks," in Computer Science, A. F. Cárdenas, L. Presser, and M. A. Marin (eds.), Wiley Interscience, New York, 1972, pp. 241-284.

9. Kleinrock, L., "Survey of Analytical Methods in Queueing Networks," in Computer Networks, R. Rustin (ed.), Courant Institute Computer Science Symposium III, Prentice-Hall, Englewood Cliffs, N.J., 1972.

10. Kleinrock, L., "A Selected Menu of Analytical Results for Time-Shared Computer Systems," Proc. of Computer Science Seminar conducted by IBM Germany in September 1970, in Systemprogrammierung, Oldenbourg-Verlag, Munich, Germany, 1972.

11. Kleinrock, L., "Computer Network Design Principles Derived from Experience and Measurements on the ARPA Network," Summary in Proc. of the International Telemetering Conference, Oct. 11, 1972, Los Angeles, Calif., p. 440.

12. Kleinrock, L., and S. S. Lam, "Analytic Results with the Addition of One Large User," ARPANET Satellite System Note No. 27, UCLA ARPA Network Measurement Center, Computer Science Dept., Oct. 30, 1972.

13. Kleinrock, L., and F. Tobagi, "Routing in Packet Radio Systems Controlled Flooding Using Hanover Numbers," Packet Radio Temporary Note No. 11, UCLA ARPA Network Measurement Center, Computer Science Dept., Jan. 15, 1973.

14. Kleinrock, L., and F. Tobagi, "Throughput in Carrier-Sense (Autoslot) Packet Radio Systems," Packet Radio Temporary Note No. 24, UCLA ARPA Network Measurement Center, Computer Science Dept., Feb. 12, 1973.

15. Kleinrock, L., and F. Tobagi, "Carrier-Sense with Initial Random Transmission Delay," Packet Radio Temporary Note No. 37, UCLA ARPA Network Measurement Center, Computer Science Dept., Mar., 1973.

16. Kleinrock, L., and F. Tobagi, "Effect of Acknowledgment Traffic on Channel Throughput in Packet Radio Systems," Packet Radio Temporary Note No. 57, UCLA ARPA Network Measurement Center, Computer Science Dept., May 1973.

17. Kleinrock, L., and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel, AFIPS Conference Proc., 1973 National Computer Conference and Exposition, June 4-8, 1973, New York, N.Y. pp. 703-710.

18. Kleinrock, L., and J. Hsu, "A Continuum of Computer Processor-Sharing Queueing Models," Proc. of the Seventh International Teletraffic Congress, June 13-20, 1973, Stockholm, Sweden, pp. 341/1-341/6.

19. Kleinrock, L., and F. Tobagi, "Simulation of Various Channel Access Schemes," Packet Radio Temporary Note No. 67, UCLA ARPA Network Measurement Center, Computer Science Dept., June 29, 1973.

20. Kleinrock, L., and S. S. Lam, "Packet-Switching in a Slotted Satellite Channel," ARPANET Satellite System Note No. 39, UCLA ARPA Network Measurement Center, Computer Science Dept., July 1973.

21. Kleinrock, L., and F. Tobagi, "Throughput-Delay Tradeoffs for Reservation Access Modes in Packet-Radio Systems," Packet Radio Temporary Note No. 68, UCLA ARPA Network Measurement Center, Computer Science Dept., July 2, 1973.

22. Kleinrock, L., "Challenging Problems in the Design of Computer-Communication Networks," Proc. of the XX International Meeting of The Institute of Management Sciences (TIMS XX), June 23-July 8, 1973, Tel Aviv, Israel.

23. Kleinrock, L., and S. S. Lam, "Dynamics of the ALOHA Channel," ARPA Satellite System Note No. 50, UCLA ARPA Network Measurement Center, Computer Science Dept., Aug. 22, 1973.

24. Kline, C. S., "NCP Survey," ARPA Network Request for Comments #460, February 1973.

25. Muntz, R. R., and J. Wong," Efficient Computational Procedures for Closed Queueing Networks with the Product Form Solution," Modeling and Measurement Note #17, Computer Science Dept., UCLA, June 1973.

26. Neigus, N. J., and J. B. Postel, "Socket Number List," ARPA Network Request for Comments #503, April 1973.

27. Postel, J. B., "Lost Message Detection, ARPA Network Request for Comments #516, May 1973.

3. PACKET-SWITCHING IN A SLOTTED SATELLITE CHANNEL

by Leonard Kleinrock and Simon S. Lam

Packet-switching in a slotted satellite channel*

by LEONARD KLEINROCK and SIMON S. LAM

University of California
Los Angeles, California

INTRODUCTION

Imagine that two users require the use of a communication channel. The classical approach to satisfying this requirement is to provide a channel for their use so long as that need continues (and to charge them for the full cost of this channel). It has long been recognized that such allocation of scarce communication resources is extremely wasteful as witnessed by their low utilization (see for example the measurements of Jackson & Stubbs).[1] Rather than provide channels on a user-pair basis, we much prefer to provide a single high-speed channel to a large number of users which can be shared in some fashion; this then allows us to take advantage of the powerful "large number laws" which state that with very high probability, the demand at any instant will be approximately equal to the sum of the average demands of that population. In this way the required channel capacity to support the user traffic may be considerably less than in the unshared case of dedicated channels. This approach has been used to great effect for many years now in a number of different contexts: for example, the use of graded channels in the telephone industry,[2] the introduction of asynchronous time division multiplexing,[3] and the packet-switching concepts introduced by Baran et al.,[4] Davies,[5] and finally implemented in the ARPA network.[6] The essential observation is that the full-time allocation of a fraction of the channel to each user is highly inefficient compared to the part-time use of the full capacity of the channel (this is precisely the notion of time-sharing). We gain this efficient sharing when the traffic consists of rapid, but short bursts of data. The classical schemes of synchronous time division multiplexing and frequency division multiplexing are examples of the inefficient partitioning of channels.

As soon as we introduce the notion of a shared channel in a packet-switching mode then we must be prepared to resolve conflicts which arise when more than one demand is simultaneously placed upon the channel. There are two obvious solutions to this problem: the first is to "throw out" or "lose" any demands which are made while the channel is in use; and the second is to form a queue of conflicting demands and serve them in some order as the channel becomes free. The latter approach is that taken in the ARPA network since storage may be provided economically at the point of conflict. The former approach is taken in the ALOHA system[7] which uses packet-switching with radio channels; in this system, in fact, *all* simultaneous demands made on the channel are lost.

Of interest to this paper is the consideration of satellite channels for packet-switching. The definition of a packet is merely a package of data which has been prepared by a user for transmission to some other user in the system. The satellite is characterized as a high capacity channel with a fixed propagation delay which is large compared to the packet transmission time (see the next section). The (stationary) satellite acts as a pure transponder repeating whatever it receives and beaming this transmission back down to earth; this broadcasted transmission can be heard by every user of the system and in particular a user can listen to his own transmission on its way back down. Since the satellite is merely transponding, then whenever a portion of one user's transmission reaches the satellite while another user's transmission is being transponded, the two collide and "destroy" each other. The problem we are then faced with is how to control the allocation of time at the satellite in a fashion which produces an acceptable level of performance.

The ideal situation would be for the users to agree collectively when each could transmit. The difficulty is that the means for communication available to these geographically distributed users is the satellite channel itself and we are faced with attempting to control a channel which must carry its own control information. There are essentially three approaches to the solution of this problem. The first has come to be known as a pure "ALOHA" system[7] in which users transmit any time they desire. If, after one propagation delay, they hear their successful transmission then they assume that no conflict occurred at the satellite; otherwise they know a collision occurred and they must retransmit. If users retransmit immediately upon hearing a conflict, then they are likely to conflict again, and so some scheme must be devised for introducing a random retransmission delay to spread these conflicting packets over time.

The second method for using the satellite channel is to "slot" time into segments whose duration is exactly equal to the transmission time of a single packet (we assume constant length packets). If we now require all packets to begin their transmission only at the beginning of a slot, then we

* This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC-15-69-C-0285

enjoy a gain in efficiency since collisions are now restricted to a single slot duration; such a scheme is referred to as a "slotted ALOHA" system and is the principal subject of this paper. We consider two models: the first is that of a large population of users, each of which makes a small demand on the channel; the second model consists of this background of users with the addition of one large user acting in a special way to provide an increased utilization of the channel. We concern ourselves with retransmission strategies, delays, and throughput. Abramson[8] also considers slotted systems and is concerned mainly with the ultimate capacity of these channels with various user mixes. Our results and his have a common meeting point at some limits which will be described below.

The third method for using these channels is to attempt to schedule their use in some direct fashion; this introduces the notion of a reservation system in which time slots are reserved for specific users' transmissions and the manner in which these reservations are made is discussed in the paper by Roberts.[9] He gives an analysis for the delay and throughput, comparing the performance of slotted and reservation systems.

Thus we are faced with a finite-capacity communication channel subject to unpredictable and conflicting demands. When these demands collide, we "lose" some of the effective capacity of the channel and in this paper we characterize the effect of that conflict. Note that it is possible to use the channel up to its full rated capacity when only a single user is demanding service, this is true since a user will never conflict with himself (he has the capability to schedule his own use). This effect is important in studying the non-uniform traffic case as we show below.

SLOTTED ALOHA CHANNEL MODELS

Model I. Traffic from many small users

In this model we assume:

(A1) an infinite number of users* who collectively form an independent source

This source generates M packets per slot from the distribution $v_i = \text{Prob}[M = i]$ with a mean of S_0 packets/slot.

We assume that each packet is of constant length requiring T seconds for transmission; in the numerical studies presented below we assume that the capacity of the channel is 50 kilobits per second and that the packets are each 1125 bits in length yielding $T=22.5$ msec. Note that $S_0' = S_0/T$ is the average number of packets arriving per second from the source. Let d be the maximum roundtrip propagation delay which we assume each user experiences and let $R=d/T$ be the number of slots which can fit into one roundtrip propagation time; for our numerical results we assume $d=270$ msec. and so $R=12$ slots. R slots after a transmission, a user will either hear that it was successful or know that it was destroyed. In the latter case if he now retransmits during the next slot interval and if all other users behave likewise, then for sure they will collide again; consequently we shall assume that each user transmits a previously collided packet at random during one of the next K slots, (each such slot being chosen with probability $1/K$). Thus, retransmission will take place either $R+1$, $R+2$, ... or $R+K$ slots after the initial transmission. As a result traffic introduced to the channel from our collection of users will now consist of new packets and previously blocked packets, the total number adding up to N packets transmitted per slot where $p_i = \text{Prob}[N=i]$ with a mean traffic of G packets per slot. We assume that each user in the infinite population will have at most one packet requiring transmission at any time (including any previously blocked packets). Of interest to us is a description of the maximum throughput* rate S as a function of the channel traffic G. It is clear that S/G is merely the probability of a successful transmission and G/S is the average number of times a packet must be transmitted until success; assuming

(A2) the traffic entering the channel is an independent process

We then have,

$$S = Gp_0 \tag{1}$$

If in addition we assume,

(A3) the channel traffic is Poisson

then $p_0 = e^{-G}$, and so,

$$S = Ge^{-G} \tag{2}$$

Eq. (2) was first obtained by Roberts[11] who extended a similar result due to Abramson[7] in studying the radio ALOHA system. It represents the ultimate throughput in a Model I slotted ALOHA channel without regard to the delay packets experience; we deal extensively with the delay in the next section.

For Model I we adopt assumption A1. We shall also accept a less restrictive form of assumption A2 (namely assumption A4 below) which, as we show, lends validity to assumption A3 which we also require in this model. Assume,

(A4) the channel traffic is independent over any K consecutive slots

We have conducted simulation experiments which show that this is an excellent assumption so long as $K<R$.

Let,

$$P(z) = \sum_{i=0}^{\infty} p_i z^i \tag{3}$$

$$V(z) = \sum_{i=0}^{\infty} v_i z^i \tag{4}$$

* These will be referred to as the "small" users.

* Note that $S=S_0$ under stable system operation which we assume unless stated otherwise (see below).

Using only assumption A4 and the assumption that M is independent of $N-M$, we find [10] that $P(z)$ may be expressed as

$$\left[\frac{p_1}{K}(1-z)+P\left(1-\frac{1-z}{K}\right)\right]^K V(z)$$

If, further, the source is an independent process (i.e., assumption A1) and is Poisson distributed then $V(z)=e^{-S(1-z)}$, and then we see immediately that,

$$\lim_{K\to\infty} P(z)=e^{-G(1-z)}.$$

This shows that assumption A3 follows from assumptions A1 and A4 in the limit of large K, under the reasonable condition that the source is Poisson distributed.

We have so far defined the following critical system parameters: S_0, S, G, K and R. In the ensuing analysis we shall distinguish packets transmitting in a given slot as being either newly generated or ones which have in the past collided with other packets. This leads to an approximation since we do not distinguish how many times a packet has met with a collision. We have examined the validity of this approximation by simulation, and have found that the correlation of traffic in different slots is negligible, except at shifts of $R+1$, $R+2, \ldots, R+K$; this exactly supports our approximation since we concern ourselves with the most recent collision. We require the following two additional definitions:

$q=$ Prob[newly generated packet is successfully transmitted]

$q_t=$ Prob[previously blocked packet is successfully transmitted]

We also introduce the expected packet delay D:

$D=$ average time (in slots) until a packet is successfully received

Our principal concern in this paper is to investigate the trade-off between the average delay D and the throughput S.

Model II. Background traffic with one large user

In this second model, we refer to the source described above as the "background" source but we also assume that there is an additional single user who constitutes a second independent source and we refer to this source as the "large" user. The background source is the same as that in Model I and for the second source, we assume that the packet arrivals to the large user transmitter are Poisson and independent of other packets over $R+K$ consecutive slots. In order to distinguish variables for these two sources, we let S_1 and G_1 refer to the S and G parameters for the background source and let S_2 and G_2 refer to the S and G parameters for the single large user. We point out that the identity of this large user may change as time progresses but insist that there be only one such at any given time. We introduce the new variables

$$S=S_1+S_2 \tag{5}$$

$$G=G_1+G_2 \tag{6}$$

S represents the total throughput of the system and G represents the traffic which the channel must support (including retransmissions). We have assumed that the small users may have at most one packet outstanding for transmission in the channel; however the single large user may have many packets awaiting transmission. We assume that this large user has storage for queueing his requests and of course it is his responsibility to see that he does not attempt the simultaneous transmission of two packets. We may interpret G_2 as the probability that the single large user is transmitting a packet in a channel slot and so we require $G_2\leq 1$; no such restriction is placed on G_1 (or on G in Model I).

We now introduce a means by which the large user can control his channel usage enabling him to absorb some of the slack channel capacity; this permits an increase in the total throughput S. The set of packets awaiting transmission by the large user compete among each other for the attention of his local transmitter as follows. Each waiting packet will be scheduled for transmission in some future slot. When a newly generated packet arrives, it immediately attempts transmission in the current slot and will succeed in capturing the transmitter unless some other packet has also been scheduled for this slot; in the case of such a scheduling conflict, the new packet is randomly rescheduled in one of the next L slots, each such slot being chosen equally likely with probability $1/L$. Due to the background traffic, a large user packet may meet with a transmission conflict at the satellite (which is discovered R slots after transmission) in which case, as in Model I, it incurs a random delay (uniformly distributed over K slots) plus the fixed delay of R slots. More than one packet may be scheduled for a future slot and we assume that these scheduling conflicts are resolved by admitting that packet with the longest delay since its previous blocking (due to conflict in transmission or conflict in scheduling) and uniformly rescheduling the others over the next L slots; ties are broken by random selection. We see, therefore, that new packets have the lowest priority in case of a scheduling conflict; however, they seize the channel if it is free upon their arrival. The variable L permits us a certain control of channel use by the large user but does not limit his throughput. We also assume $K, L<R$. Corresponding to q and q_t in Model I, we introduce the success probabilities q_i and q_{it} $(i=1, 2)$ for new and previously blocked packets respectively and where $i=1$ denotes the background source and $i=2$ denotes the single large source. Finally, we choose to distinguish between D_1 and D_2 which are the average number of slots until a packet is successfully transmitted from the background and large user sources respectively.

RESULTS OF ANALYSIS

In this section we present the results of our analysis without proof. The details of proof may be found in Reference 10.

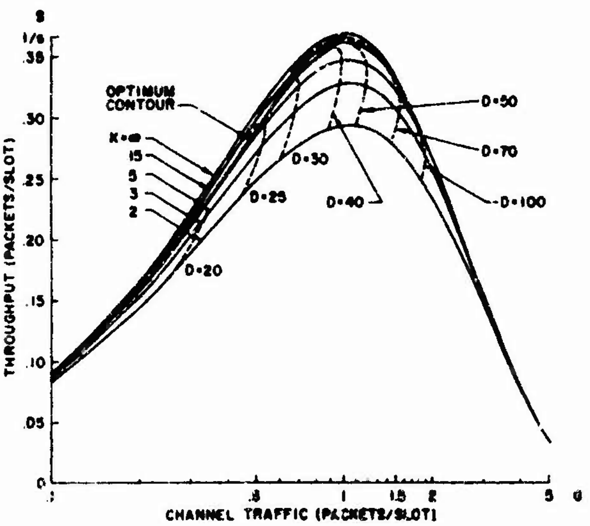


Figure 1—Throughput as a function of channel traffic

Model I. Traffic from many small users

We wish to refine Eq. (2) by accounting for the effect of the random retransmission delay parameter K. Our principal result in this case is

$$S=G\frac{q_t}{q_t+1-q} \tag{7}$$

where

$$q=\left[e^{-G/K}+\frac{G}{K}e^{-G}\right]^K e^{-S} \tag{8}$$

and

$$q_t=\left[\frac{1}{1-e^{-G}}\right]\left[e^{-G/K}-e^{-G}\right]\left[e^{-G/K}+\frac{G}{K}e^{-G}\right]^{K-1}e^{-S} \tag{9}$$

The considerations which led to Eq. (7) were inspired by Roberts[11] in which he developed an approximation for Eq. (9) of the form

$$q_t\cong\frac{K-1}{K}e^{-G} \tag{10}$$

We shall see below that this is a reasonably good approximation. Equations (7-9) form a set of non-linear simultaneous equations for S, q and q_t which must be solved to obtain an explicit expression for S in terms of the system parameters G and K. In general, this cannot be accomplished. However, we note that as K approaches infinity these three equations reduce simply to

$$\lim_{K\to\infty}\frac{S}{G}=\lim_{K\to\infty} q=\lim_{K\to\infty} q_t=e^{-G} \tag{11}$$

Thus, we see that Eq. (2) is the correct expression for the throughput S only when K approaches infinity which corresponds to the case of infinite average delay; Abramson[8] gives this result and numerous others all of which correspond to this limiting case. Note that the large K case avoids the large delay problem if T is small (very high speed channels).

The numerical solution to Eqs. (7-9) is given in Figure 1 where we plot the throughput S as a function of the channel traffic G for various values of K. We note that the maximum throughput at a given K occurs when $G=1$. The throughput improves as K increases, finally yielding a maximum value of $S=1/e=.368$ for $G=1$, $K=$ infinity. Thus we have the unfortunate situation that the ultimate capacity of this channel supporting a large number of small users is less than 37 percent of its theoretical maximum (of 1). We note that the efficiency rapidly approaches this limiting value (of $1/e$) as K increases and that for $K=15$ we are almost there. The figure also shows some delay contours which we discuss below. In Figure 2, we show the variation of q and q_t with K for various values of G. We note how rapidly these functions approach their limiting values as given in Eq. (11). Also on this curve, we have shown Roberts' approximation in Eq. (10) which converges to the exact value very rapidly as K increases and also as G decreases.

Our next significant result is for packet delay as given by

$$D=R+1+\frac{1-q}{q_t}\left[R+1+\frac{K-1}{2}\right] \tag{12}$$

We note from this equation that for large K, the average delay grows linearly with K at a slope

$$\lim_{K\to\infty}\frac{\partial D}{\partial K}=\frac{1-e^{-G}}{2e^{-G}}$$

Using Eq. (11), we see that this slope may be expressed as $G-S/2S$ which is merely the ratio of that portion of transmitted traffic which meets with a conflict to twice the throughput of the channel; since $G-S/2S=\frac{1}{2}(G/S-1)$, we see that the limiting slope is equal to $\frac{1}{2}$ times the average number of times a packet is retransmitted. Little's well-known result[12] expresses the average number ($\bar{n}$) of units (packets in our case) in a queueing system as the product of the average arrival rate ($S_0=S$ in our case) and the average time in system (D). If we use this along with Eqs. (7) and (12), we get

$$\bar{n}=SD=G\left[R+1+\frac{K-1}{2}\right]-S\left[\frac{K-1}{2}\right] \tag{13}$$

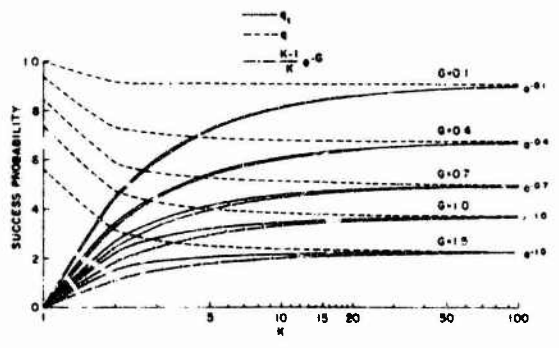


Figure 2—Success probabilities as a function of retransmission delay

In Figure 1 we plot the loci of constant delay in the S, G plane. Note the way these loci bend over sharply as K increases defining a maximum throughput $S_{max}(D)$ for any given value of D; we note the cost in throughput if we wish to limit the average delay. This effect is clearly seen in Figure 3 which is the fundamental display of the tradeoff between delay and throughput for Model I; this figure shows the delay-throughput contours for constant values of K. We also give the minimum envelope of these contours which defines the optimum performance curve for this system (a similar optimum curve is also shown in Figure 1). Note how sharply the delay increases near the maximum throughput $S=0.368$; it is clear that an extreme price in delay must be paid if one wishes to push the channel throughput much above 0.3' 0 and the incremental gain in throughput here is infinitesimal. On the other hand, as S approaches zero, D approaches $R+1$. Also shown here are the constant G contours. Thus this figure and Figure 1 are two alternate ways of displaying the relationship among the four critical system quantities S,G,K, and D.

From Figure 3 we observe the following effect. Consider any given value of S (say at $S=0.20$), and some given value of K (say $K=2$). We note that there are two possible values of D which satisfy these conditions ($D=21.8$, $D=161$). How do we explain this?* It is clear that the lower value is a stabie

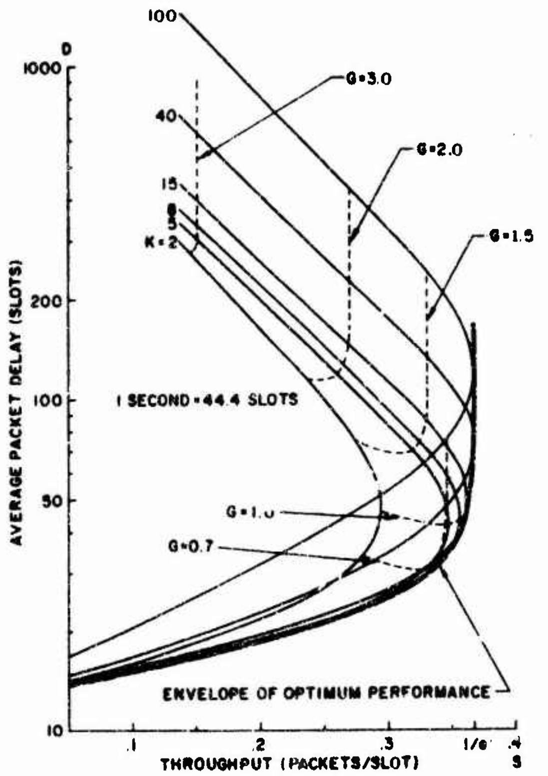


Figure 3—Delay-throughput tradeoff

* This question was raised in a private conversation with Martin Graham (University of California, Berkeley). A simulation of this situation is reported upon in Reference 13.

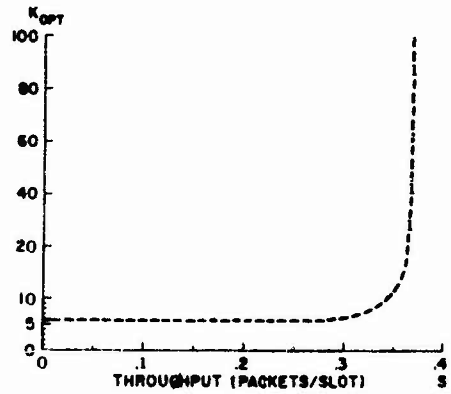


Figure 4—Optimum K

operating point since the system has sufficient capacity to absorb any fluctuation in the rate S_0. Suppose that we now slowly increase S_0 (the source rate); so long as we do not exceed the maximum value of the system throughput rate for this K (say, $S_{max}(K)$), then we see that $S=S_0$ and the system will follow the input. Note that $S_{max}(K)$ always occurs at the intersection of the $G=1$ curve as noted earlier. However, if we attempt to set $S_0>S_{max}(K)$, then the system will go unstable! In fact, the throughput S will drop from $S_{max}(K)$ toward zero as the system accelerates up the constant K contour toward infinite delay! The system will remain in that unfortunate circumstance so long as $S_0>S$ (where now S is approaching zero). All during its demise, the rate at which new packets are being trapped by the system is S_0-S. To recover from this situation, one can set $S_0=0$; then the delay will proceed down the K contour, round the bend at $S_{max}(K)$ and race down to $S=0$. All this while, the backlogged packets are being flushed out of the system. The warning is clear: one must avoid the knee of the K contour. Fortunately, the optimum performance curve does avoid the knee everywhere except when one attempts to squeeze out the last few percent of throughput. In Figure 4, we show the optimum values of K as a function of S. Thus, we have characterized the tradeoff between throughput and delay for Model I.

Model II. Background traffic with one large user

In this model the throughput equation is similar to that given in Eq. (7), namely,

$$S_i=G_i\frac{q_{it}}{q_{it}+1-q_i} \qquad i=1,2 \tag{14}$$

the quantities q_{it} and q_i are given in the appendix. Similarly the average delays for the two classes of user are given by

$$D_1=R+1+\frac{1-q_1}{q_{1t}}\left[R+1+\frac{K-1}{2}\right] \tag{15}$$

$$D_2=R+1+\frac{1-q_2}{q_{2t}}\left[R+1+\frac{K-1}{2}\right]+\frac{L+1}{2}\left[E_n+\frac{1-q_2}{q_{2t}}E_t\right] \tag{16}$$

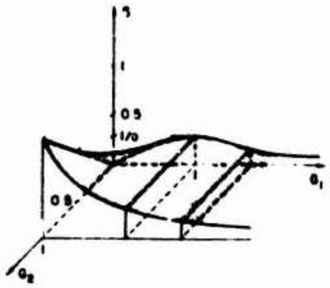


Figure 5 Throughput surface

where E_a and E_t are given in the appendix. It is easy to show that as K, L approach infinity,

$$q_1 = q_{1t} = e^{-G_1}(1-G_2) \tag{17}$$

$$S_1 = G_1 e^{-G_1}(1-G_2) \tag{18}$$

$$q_2 = q_{2t} = e^{-G_1} \tag{19}$$

$$S_2 = G_2 e^{-G_1} \tag{20}$$

$$S = (G - G_1G_2)e^{-G_1} \tag{21}$$

where we recall $G = G_1 + G_2$ and $S = S_1 + S_2$. From these last equations or as given by direct arguments in an unpublished note by Roberts, one may easily show that at a constant background user throughput S_1, the large user throughput S_2 will be maximized when

$$G = G_1 + G_2 = 1 \tag{22}$$

This last is a special case of results obtained by Abramson in Reference 8 and he discusses these limiting cases at length for various mixes of users. We note that,

$$\frac{\partial S}{\partial G_2} = e^{-G_1}(1-G_1) \tag{23}$$

$$\frac{\partial S}{\partial G_1} = -e^{-G_1}(G - G_1G_2 - 1 + G_2) \tag{24}$$

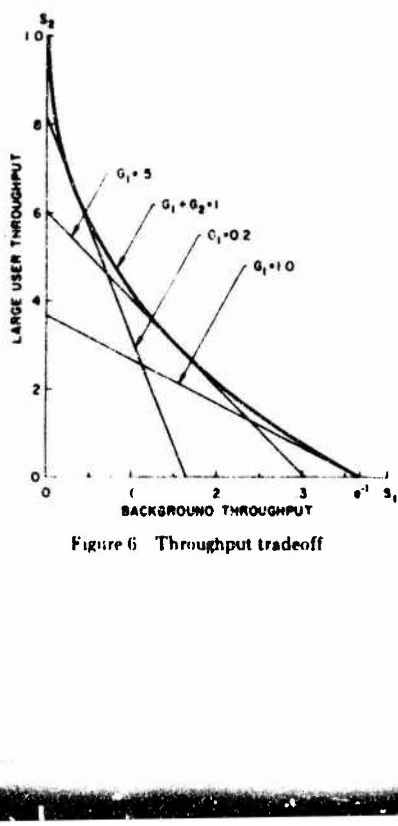


Figure 6 Throughput tradeoff

In Figure 5 we give a qualitative diagram of the 3-dimensional contour for S as a function of G_1 and G_2. We remind the reader that this function is shown for the limiting case K, L approaching infinity only. From our results we see that for constant $G_1 < 1$, S increases linearly with G_2 ($G_2 < 1$). For constant $G_1 > 1$, S decreases linearly as G_2 increases. In addition, for constant $G_2 < \frac{1}{2}$, S has a maximum value at $G_1 = 1 - 2G_2/1 - G_2$. Furthermore, for constant $G_2 > \frac{1}{2}$, S decreases as G_1 increases and therefore the maximum throughput S must occur at $S = G_2$ in the $G_1 = 0$ plane.

The optimum curve given in Eq. (22) is shown in the S_1, S_2 plane in Figure 6 along with the performance loci at constant G_1. We note in these last two figures that a channel throughput equal to 1 is achievable whenever the background traffic drops to zero thereby enabling $S = S_2 = G_2 = 1$; this corresponds to the case of a single user utilizing the satellite channel at its maximum throughput of 1. Abramson [8] discusses a variety of curves such as those in Figure 6; he considers the generalization where there may be an arbitrary number of background and large users.

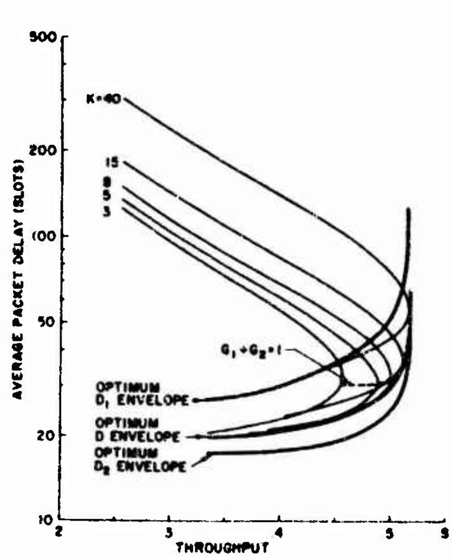


Figure 7 — Delay-throughput tradeoff at $S_1 = 0.1$

In the next three figures, we give numerical results for the finite K case; in all of these computations, we consider only the simplified situation in which $K = L$ thereby eliminating one parameter. In Figure 7 we show the tradeoff between delay and throughput similar to Figure 3. (Note that Figure 5 is similar to Figure 1.) Here we show the optimum performance of the average delay $D = S_1D_1 + S_2D_2/S$ along with the behavior of D at constant values of K and $S_1 = 0.1$ (note the instability once again for overloaded conditions). Also shown are minimum curves for D_1 and D_2, which are obtained by using the optimum K as a function of S. If we are willing to reduce the background throughput from its maximum at $S_1 = 0.368$, then we can drive the total throughput up to approximately $S = 0.52$ by introducing additional traffic from the large user. Note that the minimum D_1 curve is much higher than the minimum D_2 curve. Thus our net gain in channel throughput is also at the expense of longer packet delays for the small users. Once again, we see the sharp rise near saturation.

In Figure 8, we display a family of optimum D curves for various choices of S_1 as a function of the total throughput S. We also show the behavior of Model I as given in Figure 3. Note as we reduce the background traffic, the system capacity increases slowly; however, when S_1 falls below 0.1, we begin to pick up significant gains for S_2. Also observe that each of the constant curves "peels off" from the Model I curve at a value of $S = S_1$. At $S_1 = 0$, we have only the large user operating with no collisions and at this point, the optimal value of L is 1. This reduces to the classical queueing system with Poisson input and constant service time (denoted $M/D/1$) and represents the *absolute optimum performance* contour for any method of using the satellite channel when the input is Poisson; for other input distributions we may use the $G/D/1$ queueing results to calculate this absolute optimum performance contour.

In Figure 9, we finally show the throughput tradeoffs between the background and large users. The upper curve shows the absolute maximum S at each value of S_1; this is a clear display of the significant gain in S_2 which we can achieve if we are willing to reduce the background throughput. The middle curve (also shown in Figure 6 and in Reference 8) shows the absolute maximum value for S_2 at each value of S_1. The lowest curve shows the net gain in system capacity as S_1 is reduced from its maximum possible value of $1/e$.

CONCLUSIONS

In this paper we have analyzed the performance of a slotted satellite system for packet-switching. In our first model, we have displayed the trade-off between average delay and average throughput and have shown that in the case of traffic consisting of a large number of small users, the limiting

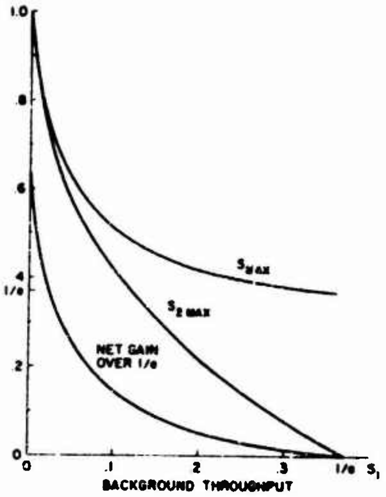


Figure 9—Throughput countours

throughput of the channel ($1/e$) can be approached fairly closely without an excessive delay. This performance can be achieved at relatively small values of K which is the random retransmission delay parameter. However, if one attempts to approach this limiting capacity, not only does one encounter large delays, but one also flirts with the hazards of unstable behavior.

In the case of a single large user mixed with the background traffic, we have shown that it is possible to increase the throughput rather significantly. The qualitative behavior for this multidimensional trade-off was shown and the numerical calculations for a given set of parameters were also displayed. The optimum mix of channel traffic was given in Eq. (22) and is commented on at length in Abramson's paper.[8] We have been able to show in this paper the relationship between delay and throughput which is an essential trade-off in these slotted packet-switching systems.

In Roberts' paper[9] he discusses an effective way to reserve slots in a satellite system so as to predict and prevent conflicts. It is worthwhile noting that another scheme is currently being investigated for packet-switching systems in which the propagation delay is small compared to the slot time, that is, $R = d/T \ll 1$. In such systems it may be advantageous for a user to "listen before transmitting" in order to determine if the channel is in use by some other user; such systems are referred to as "carrier sense" systems and seem to offer some interesting possibilities regarding their control. For satellite communications this case may be found when the capacity of the channel is rather small (for example, with a stationary satellite, the capacity should be in the range of 1200 bps for the packet sizes we have discussed in this paper). On the other hand, a 50 kilobit channel operating in a ground radio environment with packets on the order of 100 or 1000 bits lend themselves nicely to carrier sense techniques.

In all of these schemes one must trade off complexity of implementation with suitable performance. This performance must be effective at all ranges of traffic intensity in that no unnecessary delays or loss of throughput should occur due to

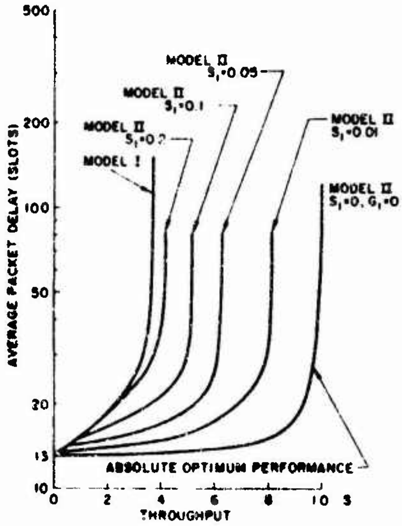


Figure 8 Optimum delay-throughput tradeoffs

complicated operational procedures. We feel that the slotted satellite packet-switching methods described in this paper and the reservation systems for these channels described in the paper by Roberts do in fact meet these criteria.

REFERENCES

1. Jackson, P. E., Stubbs, C. D., "A Study of Multi-access Computer Communications," Spring Joint Computer Conf., *AFIPS Conf. Proc.*, Vol. 34, 1969, pp. 491-504.
2. Syski, R., *Introduction to Congestion in Telephone Systems*, Oliver & Boyd, Edinburgh and London, 1960.
3. Chu, W. W., "A Study of Asynchronous Time Division Multiplexing for Time-Sharing Computer Systems," Spring Joint Computer Conf., *AFIPS Conf. Proc.*, Vol. 35, 1969, pp. 669-678.
4. Baran, P., Boehm, S., and Smith, P., "On Distributed Communications," series of 11 reports by Rand Corp., Santa Monica, Calif., 1964.
5. Davies, D. W., "The Principles of a Data Communication Network for Computers and Remote Peripherals," *Proc. IFIP Hardware*, Edinburgh, 1968, D11.
6. Roberts, L. G., "Multiple Computer Networks and Inter-Computer Communications," *ACM Symposium on Operating Systems*, Gatlinburg, Tenn., 1967.
7. Abramson, N., "The ALOHA System—Another Alternative for Computer Communications," Fall Joint Computer Conf., *AFIPS Conf. Proc.*, Vol. 37, 1970, pp. 281-285.
8. Abramson, N., "Packet Switching with Satellites," these proceedings.
9. Roberts, L. G "Dynamic Allocation of Satellite Capacity through Packet Reservation," these proceedings.
10. Kleinrock, L., Lam, S. S., Arpanet Satellite System Notes 12 (NIC Document #11294); 17 (NIC Document #11862); 25 (NIC Document #12734); and 27 (NIC Document #12756), available from the ARPA Network Information Center, Stanford Research Institute, Menlo Park, California.
11. Roberts, L., Arpanet Satellite System Notes 8 (NIC Document #11290, and 9 (NIC Document #11291), available from the ARPA Network Information Center, Stanford Research Institute, Menlo Park, California.
12. Kleinrock, L., *Queueing Systems: Theory and Applications*, to be published by Wiley Interscience, New York, 1973.
13. Rettberg, R., Arpanet Satellite System Note 11 (NIC Document #11293), available from the ARPA Network Information Center, Stanford Research Institute, Menlo Park, California.

APPENDIX

Define $G_s \triangleq$ Poisson arrival rate of packets to the transmitter of the large user

$$= S_2[1+E_n+E_2(1+E_1)] \tag{A.1}$$

The variables q_i, q_{ii} $(i=1, 2)$ in Eqs. (14–16) are then given as follows (see Reference 10 for details of the derivations):

$$q_1 = (q_0)^K (q_h)^L e^{-S} \tag{A.2}$$

$$q_{1t} = (q_s)^{K-1} q_{1e} (q_h)^L e^{-S} \tag{A.3}$$

where

$$q_0 = e^{-G_1/K} + \frac{1}{K}\left[(1-e^{-G_s})(e^{-G_1}-e^{-G_1/K}) + G_1 e^{-(G_1+G_s)}\right] \tag{A.4}$$

$$q_h = \begin{cases} (G_s+1)e^{-G_s} & L=1 \\ \dfrac{1}{L-1}\left(Le^{-G_s/L}-e^{-G_s}\right) & L\geq 2 \end{cases} \tag{A.5}$$

$$q_{1e} = \frac{1}{1-e^{-(G_1+G_s)}}\left[e^{-G_1/K}\left(1-\frac{1-e^{-G_s}}{K}\right)-e^{-(G_1+G_s)}\right] \tag{A.6}$$

Let us introduce the following notation for events at the large user:

SS = scheduling success (capture of the transmitter)
SC = scheduling conflict (failure to capture transmitter)
TS = transmission success (capture of a satellite slot)
TC = transmission conflict (conflict at the satellite)
NP = newly generated packet

Then,

$$q_2 = \frac{r_n + r_s E_n}{1+E_n} \tag{A.7}$$

$$q_{2t} = \frac{r_t + r_s E_t}{1+E_t} \tag{A.8}$$

where

$E_n \triangleq$ average number of SC events before an SS event conditioning on NP $= \dfrac{1-a_n}{a_s}$ (A.9)

$E_t \triangleq$ average number of SC events before an SS event conditioning on TC $= \dfrac{1-a_t}{a_s}$ (A.10)

The variables a_i, r_i $(i=n, t, s)$ are defined and given below:

$$a_n \triangleq \text{Prob}\,[SS/NP] = \left(\frac{q_0}{q}\right)^K (q_h)^L \left(\frac{1-e^{-S_2}}{S_2}\right) \tag{A.11}$$

$$r_n \triangleq \text{Prob}\,[TS/SS,\, NP] = q^K e^{-S_1} \tag{A.12}$$

$$a_t \triangleq \text{Prob}\,[SS/TC] = \frac{1}{K}\,\frac{1-(q_0/q)^K}{1-q_0/q} \tag{A.13}$$

$$r_t \triangleq \text{Prob}\,[TS/SS,\, TC] = q^{K-1} q_{2e} e^{-S_1} \tag{A.14}$$

$$a_s \triangleq \text{Prob}\,[SS/SC] = \left(\frac{q_0}{q}\right)^K \frac{q_{se}}{L}\,\frac{1-(q_h)^L}{1-q_h} \tag{A.15}$$

$$r_s \triangleq \text{Prob}\,[TS/SS,\, SC] = q^K e^{-S_1} \tag{A.16}$$

where

$$q = e^{-G_1/K} + \frac{G_1}{K} e^{-(G_1+G_s)} \tag{A.17}$$

$$q_{2e} = \frac{e^{-G_1/K} - e^{-G_1}}{1-e^{-G_1}} \tag{A.18}$$

$$q_{se} = \frac{1}{G_s - 1 + e^{-G_s}}\left(\frac{L}{L-1}\right)^2 \left[G_s\left(1-\frac{1}{L}\right)e^{-G_s/L} - e^{-G_s/L} + e^{-G_s}\right] \tag{A.19}$$

4. THE FLOW DEVIATION METHOD: AN APPROACH TO STORE-AND-FORWARD COMMUNICATION NETWORK DESIGN

by L. Fratta, M. Gerla, and L. Kleinrock

The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design

L. Fratta
Instituto di Elettrotecnia ed Elettronica
Politecnico di Milano, Italy

M. Gerla
Network Analysis Corporation
Glen Cove, New York

L. Kleinrock
University of California
Los Angeles, California

ABSTRACT

Two problems relevant to the design of a store-and-forward communication network (the message routing problem and the channel capacity assignment problem) are formulated and are recognized to be essentially non-linear, unconstrained multicommodity (m.c.) flow problems. A "Flow Deviation" (FD) method for the solution of these non-linear, unconstrained m.c. flow problems is described which is quite similar to the gradient method for functions of continuous variables; here the concept of gradient is replaced by the concept of "shortest route" flow. As in the gradient method, the application of successive flow deviations leads to local minima. Finally, two interesting applications of the FD method to the design of the ARPA Computer Network are discussed.

1. INTRODUCTION

In this paper we consider a procedure (the "flow deviation" method) for assigning flow within store-and-forward communication networks so as to minimize cost and/or delay for a given topology and for given external flow requirements. We begin by defining the basic model below and follow that with some examples. We then discuss various approaches to the problem and then introduce and describe the "flow deviation" method. This method is evaluated under some further restrictions and is then applied to various problem formulations for the ARPA network [6], [7].

Suppose we have a collection of nodes N_i, $(i=1,\ldots,n)$, and are required to route a quantity r_{ij} of type (i,j) commodity from N_i to N_j through a given network (Figure 1).

Networks, 3: 97-133

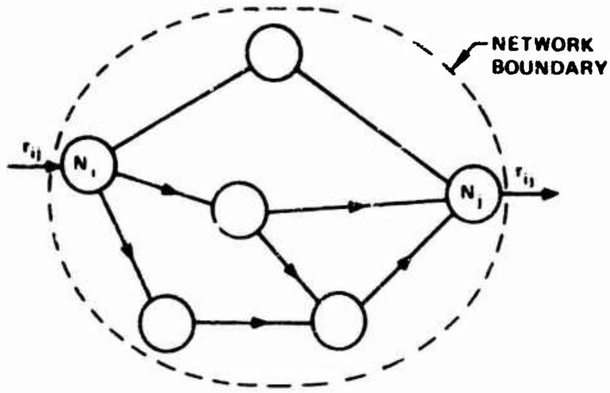


Fig. 1 Example of routing of the (i,j) commodity.

The multicommodity (m.c.) flow problem consists of finding the routes for all such commodities, which minimize (or maximize) a well-defined performance function (e.g., cost or delay), such that a set of constraints (e.g., channel capacity constraints) are satisfied.

The most general *multicommodity problem* can be expressed formally in the following way:

Given:	A network of n nodes and b arcs
	An $n \times n$ matrix $R = [r_{ij}]$, called the requirement matrix, whose entries are non-negative
Minimize:	(or maximize)* $P(\Phi)$
over Φ	where Φ is the flow configuration and P is a well-defined performance function

Furthermore, Φ must satisfy the following constraints:

Constraints:

1. Φ must be a multicommodity flow satisfying requirement R. For this, the following conditions must be verified:

Conservation of the flow at nodes, commodity by commodity:

$$\sum_{k=1}^{n} f_{k\ell}^{(ij)} - \sum_{m=1}^{n} f_{\ell m}^{(ij)} = \left\{ \begin{array}{ll} -\,r_{ij} & \text{if } \ell = i \\ +\,r_{ij} & \text{if } \ell = j \\ 0 & \text{otherwise} \end{array} \right] \forall i,j \qquad (1.1)$$

Without loss of generality, only the minimum problem is considered in the following.

Non-negativity of flow in directed arcs:

$$f_{k\ell}^{(ij)} \geq 0 \qquad \forall i,j,k,\ell \tag{1.2}$$

where $f_{k\ell}^{(ij)}$ is the portion of commodity (i,j) flowing on arc (k,ℓ).

2. Φ must satisfy some *additional* constraints,* different from problem to problem (e.g., capacity constraints on each channel and/or cost constraints).

Let us define the *(i,j) commodity flow* $\underset{\sim}{f}^{(ij)}$ as:

$$\underset{\sim}{f}^{(ij)} \triangleq \left(f_1^{(ij)}, f_2^{(ij)}, \ldots, f_b^{(ij)}\right)$$

where $f_m^{(ij)}$ is the portion of (i,j) commodity flowing in arc m, and define *the global flow* $\underset{\sim}{f}$ as:

$$\underset{\sim}{f} = \sum_{i=1}^{n} \sum_{j=1}^{n} \underset{\sim}{f}^{(ij)}$$

In the sequel, we restrict our analysis to m.c. problems in which the performance depends solely on the global flow:

$$P(\Phi) \equiv P(\underset{\sim}{f}) \tag{1.3}$$

However, most of the arguments and techniques presented in the paper can be extended to the general case of $P(\Phi)$ explicitly depending upon various types of commodities.

So far, we represented the flow configuration Φ in terms of $\underset{\sim}{f}^{(ij)}$, $\forall i,j$.

An equivalent representation is obtained by providing for each commodity (i,j) a set of routes π_{ij}^k, $k = 1, \ldots, k_{ij}$, from node i to node j, associated with some weights α_{ij}^k ($\alpha_{ij}^k > 0$, $\sum_{k=1}^{K_{ij}} \alpha_{ij}^k = 1$): by this we mean that commodity (i,j) is transferred from i to j along K_{ij} routes, and route π_{ij}^k carries an amount $\alpha_{ij}^k \cdot r_{ij}$ of commodity (i,j).

**If an m.c. flow problem has no additional constraints, we define it to be an unconstrained m.c. flow problem; such a definition will be motivated in one of the following sections.*

As a third representation, we can consider the global flow $\underline{f}$. It can very easily be seen that $\underline{f}$ does not completely characterize Φ: for instance, two different sets of routes might yield the same $\underline{f}$. However, from Equation (1.3), it turns out that such a representation is sufficient for many considerations, and is certainly more compact than the previous two. In the following we use whichever of these representations is most convenient.

It can be seen that the set of m.c. flows satisfying constraints (1.1) and (1.2) is convex. In particular, if we let $F \triangleq \{\underline{f} | \underline{f}$ is an m.c. flow satisfying constraints (1.1) and (1.2)$\}$, we have that F is a convex polyhedron. The global flows corresponding to the "corners" (extreme points) of F have an interesting property: they are shortest route* flows [9].

2. MULTICOMMODITY PROBLEMS IN THE DESIGN OF S/F NETWORKS

Let us now consider a store-and-forward (S/F) communication network [1]. In such a network, messages traveling from N_i to N_j are "stored" in queue at any intermediate node N_k, while awaiting transmission, and are sent "forward" to N_ℓ, the next node in the route from N_i to N_j, when channel (k,ℓ) permits. Thus, at each node there are different queues, one for each output channel. The message flow requirements between nodes arise at random times and the messages are of random lengths; therefore the flows in the channels and the queue lengths in the nodes are random variables. Under appropriate assumptions,† an analysis of the system can be carried out [1]; in particular, it is possible to relate the average delay T suffered by a message traveling from source to destination (the average is over time and over all pairs of nodes) to the average flows in the channels.

The result of the analysis is:

$$T = \sum_{i=1}^{b} \frac{\lambda_i}{\gamma} T_i \qquad (2.1)$$

**A shortest route flow is an m.c. flow whose routes can be decribed by a shortest route matrix, computed for an arbitrary assignment of lengths to the arcs.*

†Assumptions: Poisson arrivals at nodes, exponential distribution of message length, independence of arrival processes at different nodes, independence assumption of service times at successive nodes [1].

where

T = total average delay per message [sec/messg]
b = number of arcs in the network
λ_i = message rate on channel i [messg/sec]

$\gamma = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij}$ = total message arrival rate from external sources [messg/sec]

T_i = average delay suffered by a message waiting for channel i [sec/messg]

T_i is the sum of two components:

$$T_i = T_i' + T_i''$$

where

$T_i' = \dfrac{1}{\mu C_i - \lambda_i}$ = transmission and queueing delay

$T_i'' = p_i$ = propagation delay

and

C_i = capacity of channel i [bits/sec]
$1/\mu$ = average message length [bits/messg]

We can rewrite Equation (2.1) as follows:

$$T = \frac{1}{\gamma} \sum_{1}^{b} \left\{ \frac{\lambda_i/\mu}{C_i - \lambda_i/\mu} + (\lambda_i/\mu)\mu p_i \right\} \tag{2.2}$$

Letting $\lambda_i/\mu = f_i$, Equation (2.2) becomes:

$$T = \frac{1}{\gamma} \sum_{1}^{b} \left\{ \frac{f_i}{C_i - f_i} + f_i p_i' \right\} \tag{2.3}$$

where

f_i = average bit rate on channel i [bits/sec]
$p_i' = \mu p_i$

The average delay T is the most common performance measure for S/F networks, and the multicommodity problem consists of finding that routing, or flow pattern F, which minimizes T.

We may now pose two problems:

Problem A: "Routing assignment"

Given: Topology, channel capacities and a requirement matrix R

Minimize: $T(\underset{\sim}{f}) = \frac{1}{\gamma} \sum_{i=1}^{b} \left(\frac{1}{C_i - f_i} + p_i'\right) f_i$

over $\underset{\sim}{f}$

Constraints: (i) $\underset{\sim}{f}$ is an m.c. flow

(ii) $f_i \leq C_i$, $i = 1, \ldots, b$

The problem is in the standard multicommodity form* and the *additional* constraints are capacity constraints. Let F_A be the set of feasible flows for Problem A: $F_A = F \cap \{\underset{\sim}{f} | \underset{\sim}{f} \leq \underset{\sim}{C}\}$. Clearly F_A is a convex set (intersection of convex sets).

A second interesting problem in S/F networks is formulated below. Assume that we have a given network topology in which the channel capacities have to be assigned. A cost is associated with the values of the capacities, and the total cost of the network is given. In addition, the flow routes must be determined. The problem statement is:

Problem B': "Routing and capacities assignment, general cost-capacity function"

Given: Topology, requirement matrix R, number of dollars available D

Minimize: $T(\underset{\sim}{C},\underset{\sim}{f}) = \frac{1}{\gamma} \sum_{i=1}^{b} \left(\frac{1}{C_i - f_i} + p_i'\right) f_i$

over $\underset{\sim}{C},\underset{\sim}{f}$

Constraints: (i) $\underset{\sim}{f}$ is an m.c. flow

(ii) $f_i \leq C_i$, $i = 1, \ldots, b$

(iii) $\sum_{i=1}^{b} d_i(C_i) \leq D$

where

$\underset{\sim}{C} = (C_1, C_2, \ldots, C_b)$

$d_i(C_i)$ = arbitrary cost-capacity function for arc i

The minimization can be carried out first on $\underset{\sim}{C}$, keeping $\underset{\sim}{f}$ fixed, and then on $\underset{\sim}{f}$.

**The possibility of formulating the routing problem as a multicommodity flow problem was already recognized by Frank and Chou in [24]. An interesting linear programming approach is presented there.*

If the cost-capacity functions are linear (i.e., $d_i(C_i) = d_i C_i$), then the minimization over $\underset{\sim}{C}$ can easily be performed by the method of Lagrange multipliers and we get the following optimum capacities as functions of the flows [1]:

$$C_i = f_i + \frac{D_e}{d_i} \frac{\sqrt{f_i d_i}}{\sum_{j=1}^{b} \sqrt{f_j d_j}} \qquad (2.4)$$

where

$$D_e = D - \sum_{i=1}^{b} f_i d_i$$

By introducing Equation (2.4) into the expression of $T(\underset{\sim}{C},\underset{\sim}{f})$ we have:

$$T(\underset{\sim}{C},\underset{\sim}{f}) = T(\underset{\sim}{f}) = \frac{\left(\sum_{i=1}^{b} \sqrt{f_i d_i}\right)^2}{\gamma D_e} + \frac{1}{\gamma} \sum_{i=1}^{b} f_i p_i' \qquad (2.5)$$

Since

$$D \geq \sum_{i=1}^{b} d_i C_i \qquad \text{for (iii)}$$

and

$$\sum_{i=1}^{b} d_i C_i \geq \sum_{i=1}^{b} d_i f_i \qquad \text{for (ii)}$$

then

$$D \geq \sum_{i=1}^{b} d_i f_i$$

and

$$D_e = D - \sum_{i=1}^{b} d_i f_i \geq 0 \qquad \text{(iv)}$$

It is easy to see from Equation (2.4) that (iv) implies also (ii) and (iii); hence both (ii) and (iii) can be replaced by (iv).

By introducing Equation (2.5) into Problem B' and using result (iv), we obtain:

Problem B: "Routing and capacities assignment, linear cost-capacity function"

Given: Topology, requirement matrix R, number of dollars D

Minimize: over $\underset{\sim}{f}$

$$T(\underset{\sim}{f}) = \frac{\left(\sum_{i=1}^{b} \sqrt{f_i d_i}\right)^2}{\gamma D_e} + \frac{1}{\gamma} \sum f_i p_i'$$

Constraints: (i) $\underset{\sim}{f}$ m.c. flow

(ii) $D_e \geq 0$

Again the problem is reduced to an optimal flow problem of the standard multicommodity form. The *additional* constraint is now a cost constraint. Let F_B be the set of feasible flows for Problem B:

$$F_B = F \cap \{\underset{\sim}{f} \mid D - \sum_{i=1}^{b} d_i f_i \geq 0\}$$

Clearly F_B is convex.

The inspection of Problems A and B motivates the following important observation:

Observation:

In both Problems A and B, the performance $T(\underset{\sim}{f})$ goes to ∞ whenever f approaches the boundaries defined by the additional constraints (i.e., when any channel becomes saturated in A, or when the excess dollars D_e reduce to zero in B).

Using mathematical programming terminology, the performance $T(\underset{\sim}{f})$ incorporates the additional constraints as *penalty functions*. From a practical point of view, such a property is very important: it guarantees the feasibility of the solution (with respect to the additional constraints) during the application of usual non-linear minimization techniques, provided a feasible starting flow is found.

The property is quite general for S/F networks: when the additional constraints are satisfied with equality, usually some saturation occurs, the queues at nodes grow large and the delay T increases rapidly.

As a consequence of the above observation, if we assume that a feasible starting solution can be found,* we can disregard

Techniques for finding feasible starting solutions are shown in the applications section.

the additional constraints and approach Problems A and B as *unconstrained* m.c. flow problems. Problems A and B will be investigated further in later sections.

3. THE FD METHOD AS AN APPROACH TO THE SOLUTION OF NON-LINEAR M.C. FLOW PROBLEMS

In order to place the Flow Deviation (FD) method in the proper perspective in relation to the existing methods, it is convenient to classify the various m.c. flow problems into categories; for each category, the solution techniques available in the literature are reviewed and the contribution of the FD method is discussed.

a) Unconstrained M.C. Flow Problems

a.1) Linear performance. The linear min cost flow problem with no constraints on capacity has the well known shortest route solution (where the arc length is equivalent to the linear cost of the arc) [9,12]. Very efficient techniques are available for the evaluation of all shortest routes on a graph and for the routing of the commodities along such routes [9,16]; therefore it appears convenient to reduce complicated flow problems (i.e., non-linear, or constrained) to the linear, unconstrained form, which can be solved efficiently.

a.2) Non-linear performance. The most natural thing to do is to linearize the problem. Problems which are separable* and convex can be linearized by approximating the convex functions with piecewise linear functions and by introducing one supplementary variable and one constraint equation for each linearized segment [11,15,24]. This method has two serious drawbacks: first, it can be applied only to separable and convex problems; secondly, the number of variables and constraints becomes prohibitively large for large networks.

Another method, which applies to differentiable problems, consists of approximating the performance function with the tangent hyperplane, which is expressed in terms of the partial derivatives $\{\partial P/\partial f_i\}$. The min cost solution of the linearized problem is the shortest route flow, where the length of arc i is defined as $\partial P/\partial f_i$. As it will be shown later, such shortest route flow represents the direction of the *steepest descent* flow deviation.

---

*A separable m.c. flow problem has the form:

$$P(\underset{\sim}{f}) = \sum_{i=1}^{b} P_i(f_i)$$

The above idea is the essence of the *FD method*, which consists of repeated evaluations of steepest descent directions and of one variable minimizations along such directions; the method (described in Section 5) is conceptually very similar to the gradient method applied to non-linear minimization problems. If the problem is differentiable, the FD method is clearly superior to the supplementary variables method mentioned before: it does not add new variables and constraints, and can be applied to non-convex, non-separable cases.

In fact, the idea of using shortest routes (computed with partial derivatives) for the solution of non-linear problems is not new: using such techniques, Dafermos [17] solved various traffic problems, formulated as unconstrained, convex m.c. flow problems, and Yaged [18] solved a min cost capacity assignment for a communications network, which was formulated as an unconstrained, concave m.c. flow problem.

Dafermos stated the conditions for the optimality of the solution and proposed an algorithm for finding the optimal routing in the convex case; the algorithm, however, is impractical for large nets, as it requires the bookkeeping of all paths for all commodities [17]. Yaged's results, on the other hand, are very restricted: they apply only to a separable, concave problem [18].

In this paper, we attempt a more general, systematic investigation of the method; we introduce the main results in a more straightforward way and in a simpler formulation than in [17]. We indicate an algorithm which is applicable to non-separable problems and which has been efficiently applied to large nets.

b) Constrained M.C. Flow Problems

b.1) Linear performance, linear constraints. The classical, and most efficient, approach is the Dantzig-Wolfe decomposition [13,14], which reduces the solution of the main problem to the repeated solution of a Master Problem and a Subproblem. The Master is a linear program containing the additional constraints, and the Subproblem, which generates new columns to introduce into the Master, is an *unconstrained* linear min cost flow problem.

b.2) Non-linear performance, non-linear constraints. The general theory of non-linear problems with non-linear constraints is very hard. The special case of convex performance and concave non-negativity constraints, however, can be attacked efficiently with the Dantzig-Wolfe decomposition for convex programs [11]; the Master Problem is a linear program, and the column generating Subproblem is an *unconstrained* convex min cost flow problem. Here is another important area of application for the FD method.

We showed that the two design problems considered in the paper can be regarded as unconstrained m.c. flow problems; therefore, in the sequel, unless otherwise specified, we refer to unconstrained problems.

4. STATIONARITY CONDITIONS

Let us assume that $P(\underset{\sim}{f})$ is continuous with its first partial derivatives. We want to establish necessary and sufficient conditions for $\underset{\sim}{f}$ to be stationary.*

The most general perturbation (which we define as *flow deviation*) around $\underset{\sim}{f}$ can be obtained as a convex combination of $\underset{\sim}{f}$ with any m.c. flow $\underset{\sim}{v}$. The result of such flow deviation, $\underset{\sim}{f}'$, is expressed as:

$$\underset{\sim}{f}' = (1 - \lambda)\underset{\sim}{f} + \lambda\underset{\sim}{v} = \underset{\sim}{f} + \lambda(\underset{\sim}{v} - \underset{\sim}{f})$$

where

$$\underset{\sim}{v} \varepsilon F, \quad 0 \leq \lambda \leq 1$$

If $\lambda \to 0$, the flow deviation is *infinitesimal*. For $\lambda = \delta\lambda << 1$, we have:

$$\delta P(\underset{\sim}{f}) \triangleq P(\underset{\sim}{f}') - P(\underset{\sim}{f}) \doteq \delta\lambda \sum_{k=1}^{b} \ell_k (v_k - f_k) \tag{4.1}$$

where

$$\ell_k = \frac{\partial P}{\partial f_k}$$

From Equation (4.1) and from the definition of stationarity, $\underset{\sim}{f}$ is stationary if:

$$\sum_{k=1}^{b} \ell_k (v_k - f_k) \geq 0, \quad \underset{\sim}{v} \varepsilon F \tag{4.2}$$

We can also produce infinitesimal perturbations that involve only one of the commodities; $\underset{\sim}{f}$ must be stationary with respect to any one of them separately. It follows that $\underset{\sim}{f}$ is stationary if, for all (i,j) commodities:

$$\sum_{k=1}^{b} \ell_k (v_k^{(ij)} - f_k^{(ij)}) \geq 0, \forall \underset{\sim}{v} \varepsilon F^{(ij)} \tag{4.3}$$

*$\underset{\sim}{f}$ *is defined as stationary if, for any infinitesimal perturbation* $\delta\underset{\sim}{f}$ *(such that* $\underset{\sim}{f} + \delta\underset{\sim}{f}$ *is also m.c. flow) we have*

$$P(\underset{\sim}{f} + \delta\underset{\sim}{f}) \geq P(\underset{\sim}{f})$$

A local minimum is always stationary; the opposite, however, is not true.

where $F^{(ij)}$ is the set of the feasible (i,j) commodity flows. In fact, Equations (4.2) and (4.3) are equivalent, as will be seen from the subsequent derivations. Condition (4.2) can be rewritten as:

$$\min_{\underline{v} \in F} \sum_{k=1}^{b} \ell_k v_k \geq \sum_{k=1}^{b} \ell_k f_k \qquad (4.4)$$

But, as $\underline{f} \in F$, Equation (4.4) becomes:

$$\min_{\underline{v} \in F} \sum_{k=1}^{b} \ell_k v_k = \sum_{k=1}^{b} \ell_k f_k \qquad (4.5)$$

Similarly, Equation (4.3) becomes:

$$\min_{\underline{v}^{(ij)} \in F^{(ij)}} \sum_{k=1}^{b} \ell_k v_k^{(ij)} = \sum_{k=1}^{b} \ell_k f_k^{(ij)} \qquad (4.6)$$

Condition (4.5)* is easy to check: the right hand side can be directly evaluated, and the left hand side requires the computation of the shortest route flow under the metric $\{\ell_k\}$.

If we represent the m.c. flow as a collection of weighted routes (see Section 1), Equation (4.6) becomes:

$$\min_{\pi'} \sum_{k \in \pi'} \ell_k r_{ij} = \sum_{m=1}^{NP} \sum_{k \in \pi_m} \ell_k (\alpha_m r_{ij}) \qquad (4.7)$$

where

π' is any (i,j) route

π_m, m = 1, ..., NP, are the (i,j) routes used by commodity (i,j)

α_m, m = 1, ..., NP, are the associated weights

NP is the total number of routes used by commodity (i,j)

Let $\ell(\pi) \triangleq \sum_{k \in \pi} \ell_k$; Equation (4.7) becomes:

$$\min_{\pi'} \ell(\pi') = \sum_{m=1}^{NP} \alpha_m \ell(\pi_m) \qquad (4.8)$$

**A different derivation of Equation (4.5) is given in [19].*

Recalling that $\alpha_m > 0, \forall m$, and $\sum_{m=1}^{NP} \alpha_m = 1$, we obtain, for all commodities (i,j):

$$\ell(\pi_1) = \ell(\pi_2) = \ldots \ell(\pi_{NP}) \leq \ell(\pi') \qquad (4.9)$$

where π' is any (i,j) route.

Condition (4.9) is stated also in [17]; a similar equilibrium condition was mentioned by Wardrop [20]. In fact, the condition is very intuitive: it states that all non-zero weight routes must have the same marginal "gain," whereas the zero-weight routes must be less (or, at most, equally) convenient than the weighted ones. For an immediate interpretation of Equation (4.9), suppose there are two paths, π_p and π_q, both with non-zero weight, which do not satisfy Equation (4.9), i.e., $\ell(\pi_p) > \ell(\pi_q)$, say. An infinitesimal deviation of commodity (i,j) from π_p to π_q produces a variation $\delta P < 0$; therefore, the initial flow configuration was not stationary.

Notice that test (4.5) is computationally more convenient than test (4.9), as (4.5) only requires the knowledge of the global flow, while (4.9) requires the knowledge of all the paths [19].

The question remains, whether the stationary point is a local (or global) minimum. If $P(\underline{f})$ is strictly convex, the stationary point, if it exists, is unique and is a global min. If $P(\underline{f})$ is not convex, further considerations are required.

5. DESCRIPTION OF THE FD METHOD

The results of the previous section indicate that, if $\underline{f}$ is not a stationary flow, then the shortest route flow (evaluated under the metric $\ell_k = \partial P/\partial f_k$) represents the flow deviation of *steepest decrease* for P. This fact suggests a method, which we call *Flow Deviation method*, for the determination of stationary solutions of unconstrained, non-linear, differentiable flow problems $P(\underline{f})$.

The FD can be regarded as an operator (denoted by $FD(\underline{v},\lambda) \odot$) which maps an m.c. flow $\underline{f}$ into another m.c. flow $\underline{f}'$ and is defined as follows:

$$FD(\underline{v},\lambda) \odot \underline{f} \triangleq (1 - \lambda)\underline{f} + \lambda\underline{v} = \underline{f}' \qquad (5.1)$$

where

$\underline{v}$ is a properly chosen m.c. flow $\in F$

λ is the step size $(0 \leq \lambda \leq 1)$

Clearly FD is a map of F onto itself:

$$FD(\underset{\sim}{v},\lambda): \quad F \rightarrow F$$

Now, for each $\underset{\sim}{f} \in F$, we want to determine a pair $(\underset{\sim}{v},\lambda)$ in such a way that the repeated application of $FD(\underset{\sim}{v},\lambda)$ (starting from any flow $\underset{\sim}{f}^0$), produces a sequence $\{\underset{\sim}{f}^n\}$ which converges to a stationary flow. If we can define such a $FD(\underset{\sim}{v},\lambda)$, then we have an algorithm for the determination of stationary flows.

It can be shown [21] that, for a function $P(\underset{\sim}{f})$ which is continous, nondegenerate* and lower bounded, the following conditions† are sufficient for the convergence of an FD-mapping to a stationary flow:

$$\text{(i)} \quad \Delta P(\underset{\sim}{f}) \geq 0 \quad \forall \underset{\sim}{f} \in F$$
$$\text{(ii)} \quad \Delta P(\underset{\sim}{f}) = 0 \Rightarrow \underset{\sim}{f} \text{ stationary}$$

where

$$\Delta P(\underset{\sim}{f}) = P(\underset{\sim}{f}) - P(FD \odot \underset{\sim}{f})$$

Conditions (i) and (ii) require that the FD method be a true steepest descent method.

Again in [21] it was shown that under reasonable assumptions§ on $P(\underset{\sim}{f})$, the following definition of $FD(\underset{\sim}{v},\lambda)$ satisfies conditions (i) and (ii):

$$\underset{\sim}{v} \stackrel{\Delta}{=} \text{shortest route flow under metric } \ell_k{}^{\P}$$
$$\lambda \stackrel{\Delta}{=} \text{minimizer of } P[(1-\lambda)\underset{\sim}{f} + \lambda\underset{\sim}{v}],\ 0 \leq \lambda \leq 1 \qquad (5.2)$$

**P(f)* *is defined to be nondegenerate if, for any two distinct stationary flows, say* $\underset{\sim}{f}^1$ *and* $\underset{\sim}{f}^2$, *we have:*

$$P(\underset{\sim}{f}^1) \neq P(\underset{\sim}{f}^2).$$

†*Similar, but more restrictive conditions were stated by Dafermos in [17].*

§*The assumptions are:* $P(\underset{\sim}{f})$ *continuous and lower bounded; first partial derivatives continuous and nonnegative; second partial derivatives* $< +\infty$; $P(\underset{\sim}{f})$ *nondegenerate. The nonnegativity of the first partial derivatives is a reasonable assumption, as, in general, the performance that we want to minimize is an increasing function of the flow in each arc.*

¶*Notice that, by assumption,* $\ell_k = \partial P/\partial f_k \geq 0$; *this fact excludes the presence of negative cycles, which would have caused the failure of the shortest route computation (and therefore of the FD algorithm).*

Another valid definition of FD is the following.

Let:

$\pi_{ij}^{p} \triangleq$ shortest (i,j) path (under metric ℓ_k)

$\pi_{ij}^{q} \triangleq$ longest (i,j) path, with $\alpha_{ij}^{q} > 0$

Define (i,j) - deviation as the deviation of commodity (i,j) from π_{ij}^{q} to π_{ij}^{p}, which minimizes $P(\underset{\sim}{f})$. Define the FD operator as the composition of all (i,j) deviations: such a definition satisfies (i) and (ii).*

A general algorithm, based on the first definition of the FD operator, is outlined as follows:

1. Find a feasible starting flow $\underset{\sim}{f}^{0}$
2. Let $n = 0$
3. $\underset{\sim}{f}^{n+1} = FD(\underset{\sim}{v}^{n}, \lambda^{n}) \odot \underset{\sim}{f}^{n}$
4. If $\{P(\underset{\sim}{f}^{n}) - P(\underset{\sim}{f}^{n+1})\} < \varepsilon$, $\left(\text{or if } \sum_{k=1}^{b} \ell_k (f_k^n - v_k^n) < \varepsilon'\right)^{\dagger}$,

   where ε and ε' are acceptable positive tolerances, stop.

Otherwise, let $n = n + 1$ and go to 3.

The algorithm converges to stationary points; however, the only stationary points of stable equilibrium are the local minima, so we can assume that the algorithm converges to local minima.

In the case of $P(\underset{\sim}{f})$ strictly convex, the algorithm converges to the global min (see Appendix I for a proof of convergence and an upper bound on the error).

For $P(\underset{\sim}{f})$ non-convex, one should explore all local minima, in order to find the global minimum. However, a systematic search is impossible, for large-size networks, so heuristic approaches (like the repeated application of the FD algorithm to various initial flow configurations) have to be devised. In the case of $P(\underset{\sim}{f})$ concave (or quasi-concave [23]), the local minima correspond to extreme points of F, i.e., to shortest route flows [23]: this property, as shown later, greatly simplifies the FD algorithm and speeds up its convergence.

In the following sections, the FD method is applied to the solution of Problems A and B.

---

*Such an FD operator is essentially the "equilibration operator" defined by Dafermos [17].

†Such a test is obtained directly from the stationarity condition (3.5).

6. THE ROUTING ASSIGNMENT

Let us consider Problem A in Section 2. The performance $T(\underset{\sim}{f})$ (see Equation (2.3)) is strictly convex (separable sum of strictly convex functions), and the feasible set F_A is a convex polyhedron. Therefore, if the problem is feasible, there is a unique stationary point, which is the global minimum. The additional constraints are included in $T(\underset{\sim}{f})$ as penalties; therefore, if we can find a feasible starting flow $\underset{\sim}{f}^0 \in F_A$, Problem A can be regarded as an unconstrained m.c. flow problem and solved with the FD method.

Let us check if $T(\underset{\sim}{f})$ satisfies the conditions for the convergence (see Section 5). The first and second partial derivatives are:

$$\frac{\partial T}{\partial f_i} = \frac{1}{\gamma}\left[\frac{C_i}{(C_i - f_i)^2} + p_i'\right] \qquad (6.1)$$

$$\frac{\partial^2 T}{\partial f_i \partial f_j} = \begin{cases} 0 & \text{for } i \neq j \\ \dfrac{1}{\gamma}\dfrac{2C_i}{(C_i - f_i)^3} & \text{for } i = j \end{cases} \qquad (6.2)$$

From Equation (2.3), the optimal solution $\underset{\sim}{f}^*$, if it exists (i.e., if the problem is feasible), satisfies the capacity constraints as strict inequalities ($f_i^* < C_i \ \forall i$). Therefore, we can find an $\varepsilon > 0$ s.t.:

$$\underset{\sim}{f}^* \in F_A' \triangleq F \cap \{\underset{\sim}{f} \mid f_i \leq C_i - \varepsilon\} \qquad (6.3)$$

The application of the FD method can be restricted to $F_A' \subset F_A$; for $\underset{\sim}{f} \in F_A'$, the sufficient conditions on the first two derivatives of $P(\underset{\sim}{f})$ (as from Section 5) are satisfied; therefore the FD algorithm converges to the global minimum.

In order to find a flow $\underset{\sim}{f}^0 \in F_A$, several methods are available. One of them was described in [19]. Another method (applied below) consists of picking any $\underset{\sim}{f} \in F$, and then reducing the flows in all arcs by a scaling factor RE, until a feasible flow $\underset{\sim}{f}^0 = RE \cdot \underset{\sim}{f} \in F_A$ is obtained; $\underset{\sim}{f}^0$ satisfies a reduced requirement matrix $R_0 = RE \cdot R$. The FD method is applied using $\underset{\sim}{f}^0$ as starting flow and R_0 as starting requirement; after each FD iteration, the value of RE is increased up to a level very close to saturation. The search for a feasible flow terminates when one of the two following cases occurs: either RE $\geq$ 1, and a feasible flow is found; or the network is saturated, $T(\underline{f})$ is minimized and RE < 1. In the latter case the problem is infeasible and we are finished.

The FD algorithm for the solution of the routing problem consists of two phases, Phase 1 and Phase 2. In Phase 1 a feasible flow $\underline{f}^0$ is found (if it exists), or the problem is declared infeasible. In Phase 2 the optimal routing is obtained. The algorithm is outlined as follows:

Phase 1:

0. With $RE_0 = 1$, let $\underline{f}^0$ be the shortest route flow computed at $\underline{f} = 0$, i.e. with metric $\ell_k \triangleq [\partial T/\partial f_k]_{f_k=0} = 1/\gamma(1/C_k + p_k')$.*
   Let n = 0.

1. Let $\sigma_n = \max_k \left(\frac{f_k^n}{C_k}\right)$.

   If $\sigma_n/RE_n < 1$, let $\underline{f}^0 = \underline{f}^n/RE_n$ and go to Phase 2. Otherwise, let $RE_{n+1} = RE_n(1 - \varepsilon\ (1 - \sigma_n))/\sigma_n$, where ε is a proper tolerance, $0 < \varepsilon < 1$.

   Let $\underline{g}^{n+1} = \underline{f}^n(RE_{n+1}/RE_n)$.† Go to 2.

2. Let $\underline{f}^{n+1} = FD \odot \underline{g}^{n+1}$
   where FD is defined as in Equation (5.2).
3. If n = 0, go to 5.

The shortest route π_{ij} is therefore the route for which $\sum_{k\in\pi_{ij}} (p_k' + 1/C_k)$ is minimum. Notice that $1/C_k$ is the transmission delay per bit on channel k and p_k' is the propagation delay. No queueing delay is considered as the traffic is zero ($f_k = 0$). So, as we expect, for $f_k \to 0$, the shortest route π_{ij} minimizes the sum of transmission + propagation delay.

†*$\underline{g}^{n+1}$ is a feasible m.c. flow with requirement RE_{n+1}.*

4. If $\left| \sum_{k=1}^{b} \ell_k (v_k - g_k^{n+1}) \right| < \theta$ *and* $|RE_{n+1} - RE_n| < \delta$,
   where θ and δ are proper positive tolerances, and $\underset{\sim}{v}$ is the shortest route flow computed at g^{n+1}, stop: the problem is infeasible within tolerances θ and δ. Otherwise, go to 5.
5. Let n = n + 1 and go to 1.

Phase 2:

0. Let n = 0.
1. $\underset{\sim}{f}^{n+1} = FD \odot \underset{\sim}{f}^{n}$
2. If $\left| \sum \ell_k (v_k - f_k^n) \right| < \theta$, where θ is a proper positive tolerance, stop: $\underset{\sim}{f}^n$ is optimal within a tolerance θ. Otherwise, let n = n + 1 and go to 1.

The algorithm, in the form described above, provides only the optimum global flow $\underset{\sim}{f}$. If complete information about the routes taken by each commodity is required, a simple updating of routing tables at each FD iteration allows one to recover it at the end of the algorithm (see [19]).

7. NON-BIFURCATED ROUTING FOR LARGE AND BALANCED NETS

An m.c. flow is defined to be non-bifurcated if each commodity flows along one route only. Some applications require a non-bifurcated routing assignment; in some other applications the non-bifurcated solution is a very good approximation to the optimum bifurcated one, and is obtained with considerable saving in the amount of computation (see below). The above reasons motivate an investigation of the non-bifurcated routing assignment.

The introduction of the "non-bifurcation" constraint reduces the set of feasible m.c. flows to a discrete set: the number of elements in the set is equal to the number of all possible combinations of π_{ij} paths, $\forall$ i,j. Continuous techniques, like the FD method, cannot in general be used; discrete techniques, on the other hand, are very involved and computationally prohibitive already for networks of medium size (on the order of ten nodes). It is of interest to devise, therefore, efficient sub-optimum techniques. We will show that, in the important case of "large and balanced networks," a modification of the FD method can be successfully applied.

A network is said to be *large* if it has a large number of nodes; it is said to be *balanced* if the elements r_{ij} of the requirement matrix R are not highly diversified one from the other. For a more precise definition of "balanced," let r:

$$r \triangleq \frac{1}{(n-1)n} \sum_{ij} r_{ij}$$

be the average requirement per pair of nodes and let m:

$$m \triangleq \max_{(ij)} \; [r_{ij}/r]$$

be the ratio between the max and the average requirement.* Notice that $m \geq 1$ and that $m = 1$ corresponds to a uniform requirement matrix. A network is said to be balanced if m is close to 1.

We now combine these ideas into the notion of "large and balanced net." Let:

$$\eta \triangleq \frac{Km}{(n-1)\bar{p}'} \qquad (7.1)$$

where: $K \triangleq b/n$, the average arc to node density of the graph.

$\bar{p}' \triangleq \left(\sum_{ij} r_{ij} p'_{ij} \right) \Big/ \sum_{ij} r_{ij}$, where p'_{ij} is the length of the shortest (i,j) path (length of a path $\triangleq$ number of arcs in the path); $\bar{p}'$ is therefore the average path length, when all commodities are routed along the shortest paths.

A network is defined *large and balanced* if $\eta \ll 1$. In order to motivate such a definition, let us consider, for an arbitrary m.c. flow $\underset{\sim}{f}$, the ratio of the total flow f_k in arc k, versus the contribution $f_k^{(ij)}$ given by any commodity (i,j). Let us evaluate the average of this ratio, taken over all arcs:

$$\text{average} \left(\frac{f_k}{f_k^{(ij)}} \right) \triangleq \frac{1}{b} \sum_{k=1}^{b} \left(\frac{f_k}{f_k^{(ij)}} \right) \geq \frac{1}{bmr} \sum_{k=1}^{b} f_k \qquad (7.2)$$

**Many other appropriate definitions of m are possible, for example* $m' = \left[\sum \left(1 - \frac{r_{ij}}{r} \right)^2 \right]^{1/2}$, *in which case* $m' = 0$ *corresponds to the uniform traffic requirement.*

It was shown by Kleinrock [1] that:

$$\sum_{k=1}^{b} f_k = r(n-1)n \cdot \bar{p}$$

where: $\bar{p} \triangleq \left(\sum_{ij} r_{ij}p_{ij}\right) \Big/ \sum_{ij} r_{ij}$, and p_{ij} is the number of arcs in (i,j) route, relative to the routing assignment under consideration; $\bar{p}$ is therefore the average path length.*

Equation (7.2) becomes:

$$\text{average}\left(\frac{f_k}{f_k^{(ij)}}\right) \geq \frac{(n-1)n \cdot \bar{p}}{bm} \geq \frac{(n-1)\bar{p}'}{Km} = 1/\eta \qquad (7.3)$$

From (7.3) the following property holds:

Property (7.1): In a large and balanced net, on the average, the contribution of one single commodity in any arc can be considered infinitesimal, as compared to the total flow in that arc.

In order to show how the FD method applies to the non-bifurcated solution of large and balanced nets, let us consider a new version of flow deviation, defined as the composition of deviations involving only one commodity at a time. Suppose that the flow $\underset{\sim}{f}$ is non-bifurcated; that commodity (i,j) flows on π_{ij}; and that π'_{ij} is the shortest (i,j) route, under the usual metric $\{\ell_k\}$. The FD method deviates a proper amount $\lambda \cdot r_{ij}$, $(0 \leq \lambda \leq 1)$, of (i,j) commodity from π_{ij} to π'_{ij}, such that the performance $T(\lambda)$:

$$T(\lambda) \triangleq T(\underset{\sim}{f}(1-\lambda) + \underset{\sim}{v}\lambda) \qquad (7.4)$$

where: $\underset{\sim}{f}$ contains π_{ij}

$\underset{\sim}{v}$ contains π'_{ij}

is minimized. We can rewrite Equation (7.4) as follows:

$$T(\lambda) = T(0) + \lambda \sum_{k=1}^{b} \ell_k(v_k - f_k) + O[\lambda(\underset{\sim}{v} - \underset{\sim}{f})] \qquad (7.5)$$

**Notice that $\bar{p}$ depends on the particular routing assignment, while $\bar{p}'$ depends on requirement matrix and topology only; also notice that $\bar{p} \geq \bar{p}'$.*

where O( ) contains the terms of order higher than 1. Due to Property (7.1), the terms $(v_k - f_k)$ can be considered as infinitesimal, and the term O( ) is infinitesimal of order higher than 1. Therefore, as long as θ, defined as:

$$\theta \triangleq \sum_{k=1}^{b} \ell_k (v_k - f_k)$$

is sufficiently negative, the term O( ) can be disregarded and the minimizer of $T(\lambda)$ in Equation (7.5) is at the boundary $(\lambda_{min} = 1)$; hence the FD method preserves the non-bifurcated characteristic of the flow. On the other hand, if θ vanishes, the higher order terms become important and it might happen that $\lambda_{min} < 1$; however, $\theta = 0$ implies that $\underset{\sim}{f}$ is very close to optimum (see Appendix for bounds on the error). Therefore, the FD method provides non-bifurcated solutions which are very good approximations to the optimum bifurcated solution, and, as a consequence, very good approximations also to the optimum non-bifurcated solution.

The non-bifurcated FD algorithm is next introduced:

Non-Bifurcated FD Algorithm

Let $\underset{\sim}{f}^0$ be a starting feasible non-bifurcated flow.*
Let $n = 0$.

1. Compute $SR(\underset{\sim}{f}^n)$, defined as the set of shortest routes under metric $\{\ell_k\}$.

2. Let $\underset{\sim}{g} = \underset{\sim}{f}^n$.

   For each commodity (i,j):

   2.a Let $\underset{\sim}{v}$ be the flow configuration obtained from $\underset{\sim}{g}$ by deviating commodity (i,j) to the shortest route π'_{ij} given by $SR(\underset{\sim}{f}^n)$.
   2.b If [$\underset{\sim}{v}$ feasible *and* $T(\underset{\sim}{v}) < T(\underset{\sim}{g})$], go to 2.c. Otherwise, go to 2.d.
   2.c $\underset{\sim}{g} = \underset{\sim}{v}$
   2.d If all commodities (i,j) have been processed, go to 3. Otherwise, go to 2.a.

3. If $\underset{\sim}{g} = \underset{\sim}{f}^n$, stop. The FD method cannot improve the non-bifurcated solution any further. Otherwise, let $\underset{\sim}{f}^{n+1} = \underset{\sim}{g}$, $n = n + 1$ and go to 1.

**Such a starting flow can be found with a Phase 1 procedure, similar to that described in Section 6.*

The algorithm converges in a finite number of steps, as there are only a finite number of non-bifurcated flows, and repetitions of the same flow are excluded by the stopping condition.

An application of the algorithm to a large and balanced net is presented in the application section.

8. THE ROUTING AND CAPACITIES ASSIGNMENT

It was shown in Section 2, that F_B, the feasible set for Problem B, is a convex polyhedron; it was also shown that the additional constraint is included in the performance $T(\underline{f})$ as penalty function, so that Problem B can be regarded as an unconstrained m.c. flow problem.

Let us now investigate the properties of $T(\underline{f})$. Recall (see Equation 2.5):

$$T(\underline{f}) = \frac{\left(\sum_{i=1}^{b} \sqrt{f_i d_i}\right)^2}{\gamma\left(D - \sum_{i=1}^{b} f_i d_i\right)} + \sum f_i p_i' \tag{8.1}$$

Kleinrock, in [1], considered this case and also dealt extensively with a simplified version of Equation (8.1)* He showed that, whenever two routes, say π_{ij}^1 and π_{ij}^2, with the same number of intermediate arcs, are available for commodity (i,j), then $T(\underline{f})$ is minimized when the entire commodity is routed on one of the two routes only. Such a result, obtained under restrictive assumptions, suggests the conjecture that the optimal flow be, in general, non-bifurcated. In fact, further research has been done [21], [22], and it can be shown that $T(\underline{f})$ in in Equation (8.1) is *quasi-concave* on F_B, i.e., given any two feasible flows $\underline{f}^1$ and $\underline{f}^2$ [23]:

$$T(\underline{f}^1) \le T(\underline{f}^2) \Rightarrow T(\underline{f}^1) \le T[(1-\lambda)\underline{f}^1 + \lambda\underline{f}^2]$$

where: $0 \le \lambda \le 1$.

More generally, $T(\underline{f})$ can be shown to be quasi-concave for all "routing and capacities assignment" problems with concave cost-capacity functions [21]; the linear case is therefore a special case.

Essentially, $d_i = 1$ and $p_i' = 0$, $\forall i$.

As a consequence of such a property, the local minima are at extreme points of F_B, i.e., they correspond to shortest route flows (see Section 3), which are a subclass of the class of non-bifurcated flows.

The FD method, when applied to Problem B, can be greatly simplified: the step size λ is always equal to 1 (if we find a downhill direction, we go all the way down, due to the quasi-concavity of $T(\lambda)$), and the flow patterns generated are completely defined by just one ($n \times n$) matrix, the shortest route matrix.

A schematic description of the FD algorithm, as applied to Problem B, is as follows:

0. Suppose* $\underline{f}^0 \in F_B$; let $n = 0$.

1. Let $\underline{f}^{n+1} = FD \odot \underline{f}^n$.

2. If $(T(\underline{f}^{n+1}) \geq T(\underline{f}^n))$, stop; $\underline{f}^n$ local minimum. Otherwise let $n = n + 1$ and go to 1.

The convergence of the algorithm is guaranteed by the fact that there are only a finite number of shortest route flows, and repetitions of the same flow are not possible, as $T(\underline{f}^n)$ is strictly decreasing.

The partial derivatives, used for the shortest route computation, have the following expression:

$$\frac{\partial T}{\partial f_i} = \frac{1}{\gamma}\left(\frac{\sum\sqrt{f_j d_j}}{D_e}\right)\sqrt{\frac{d_i}{f_i}} + \frac{1}{\gamma}\left(\frac{\sum\sqrt{f_j d_j}}{D_e}\right)^2 d_i + \frac{p_i'}{\gamma}$$

Notice that $\frac{\partial T}{\partial f_i} \geq 0$; negative loops cannot exist. Also notice that:

$$\lim_{f_i \to 0} \frac{\partial T}{\partial f_i} = \infty$$

which means that, whenever the flow (and therefore the capacity, from Equation (2.4)) of an arc is reduced to zero at the end of

The problem of finding a feasible starting flow is discussed later in the section.

an FD iteration, then in such an arc, the flow and capacity are zero for all subsequent iterations, as the incremental cost of restoring the flow ($\equiv \partial T/\partial f_i$) is infinity.*

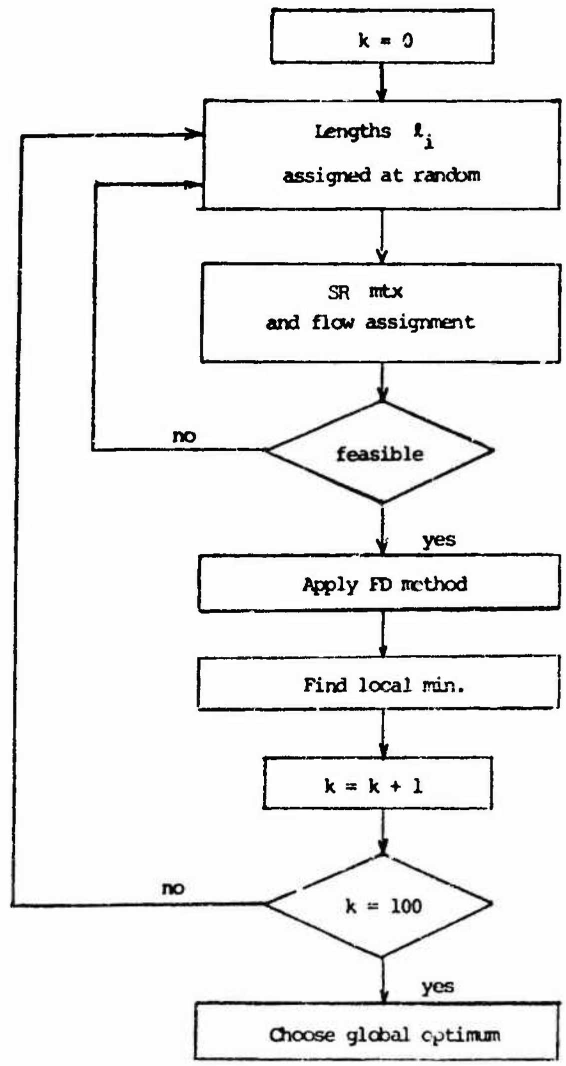


Fig. 2 Block diagram of the FD algorithm for Problem B.

This property suggests a method for the design of the topology: we can start from a topology which is highly connected, and eliminate arcs with the FD method, until a suboptimal configuration is obtained [21]. A similar approach is used by Yaged in [18].

The FD method leads to a local minimum, which depends on the choice of the feasible starting flow. In order to find several local minima, a mechanism that produces a large variety of feasible flows is required. We propose the following randomized procedure for the generation of feasible flows:*

1. Assign initial equivalent lengths $\{\ell_i^0\}$ to the arcs *at random*.
2. Compute the shortest route flow $\underset{\sim}{f}^0$ according to the metric $\{\ell_i^0\}$.
3. If $D - \sum_{i=1}^{b} f_i d_i > 0$, $\underset{\sim}{f}^0$ is feasible and can be used to start the FD algorithm. Otherwise $\underset{\sim}{f}^0$ is rejected.

The initial random choice of the lengths guarantees a certain randomness in the starting feasible flow, thus providing a method for finding several local minima. After a convenient number of iterations, the global minimum is chosen as the minimum of the local minima. This provides a "suboptimal" solution.

A block diagram of the method is given in Figure 2.

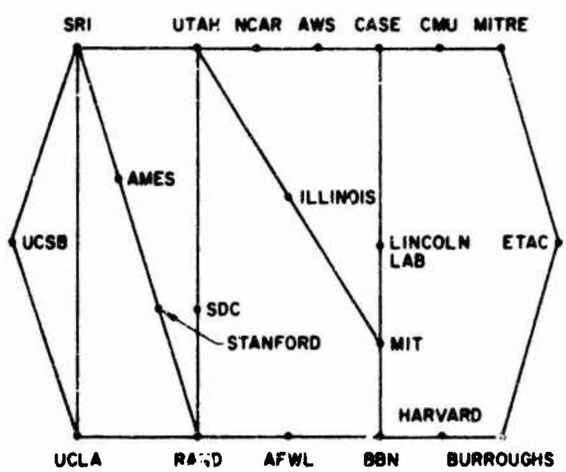


Fig. 3 A 21-node ARPA topology.

9. APPLICATIONS

As an application of the FD method, Problems A and B are solved for the ARPA Computer Network. The ARPA Computer Network is a S/F communication network connecting several computer

*Another procedure was proposed by Yaged [18].

facilities in the United States. A detailed description of the network is given in [3] - [8], [25] - [29]. Due to the fact that new computer centers are continually joining the network, its topology has been changing quite rapidly; in these applications we refer to one of the earlier proposed topologies, with 21 nodes connected by 26 full duplex channels (see Figure 3). We also assume that the traffic requirement is uniform between all pairs of nodes.

9.1 ARPA Network: The Routing Assignment

The traffic requirement $R = \{r_{ij}\}$ is assumed uniform:

$$r_{ij} = \begin{cases} r = 1.187 \text{ [kbits./sec.]*} & \text{for } i \neq j \\ 0 & \text{for } i = j \end{cases}$$

First, we show that, for the 21 node ARPA net with uniform requirement, the "large and balanced net" condition holds. From Equation (7.1), η is given by:

$$\eta = \frac{mb}{n(n-1)\bar{p}'}$$

In the present case:

$n = 21$

$\bar{p}' > 1$

$b = 52$ (each full duplex channel represents a pair of directed arcs: hence $26 \times 2 = 52$).

Hence: $\eta < 0.12 \ll 1$

The condition is satisfied. We can therefore apply both optimal and non-bifurcated FD algorithms and compare the results.

The result of the optimal FD algorithm is: $T_{min} = 0.2406$ sec., obtained after 80 shortest route computations, with an accuracy of 10^{-4} on T. The result of the non-bifurcated FD algorithm is: $T_{min} = 0.2438$ sec., obtained after 12 shortest path computations. The algorithms were programmed in Fortran and run on an IBM 360/91; the execution time was 30 sec. for

**The traffic requirement at saturation is $r_{sat} = 1.250$ [kbits./sec.] (see Figure 4). We chose $r = 0.95\, r_{sat} = 1.187$ in order to have a feasible, but difficult, requirement.*

the optimal algorithm and 4 sec. for the non-bifurcated one.* The error of the suboptimal non-bifurcated solution, with respect to the optimum, is less than 2 percent; the fact shows how powerful the non-bifurcated algorithm is for large and balanced nets, and suggests that a convenient modification of it could be useful for the solution of very large nets [21].

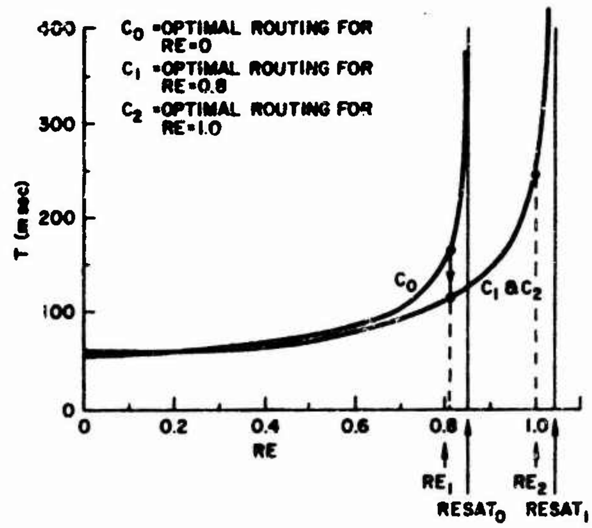


Fig. 4 Average delay T versus normalized traffic RE, using various routing schemes.

Figure 4 illustrates the application of the non-bifurcated algorithm. Recall that RE is the traffic level normalized to r = 1.187 kbits./sec. The traffic is first routed along the shortest routes computed for $RE_0 = 0$; curve C_0 plots the delay T versus RE, using such a routing scheme (which we refer to as RS_0). With RS_0, the saturation level for the traffic is $RESAT_0 = .85 < 1$; RE = 1 is infeasible, and therefore we are still in Phase 1. Let $\underset{\sim}{f}^1$ be the flow obtained by routing traffic level $RE_1 = .95\ RESAT_0 \simeq .8$, according to RS_0, and apply to $\underset{\sim}{f}^1$ the FD algorithm; a new routing scheme RS_1 is obtained, which improves $T(RE_1)$. Curve C_1, corresponding to RS_1, saturates at

We expect to be able to reduce considerably the computation time by optimizing the code and by improving some hard working subroutines, like the shortest route and flow assignment routines [16].

$RESAT_1 = 1.05 > 1$; RE = 1 is feasible and Phase 2 is initiated, with $RE_2 = 1$. At the end of Phase 2, the sub-optimal, non-bifurcated routing scheme RS_2 is found; curve C_2 corresponding to RS_2 practically coincides with curve C_1, in Figure 4, as the scale of T is not detailed enough to show differences in values. Notice that, as expected, the routing RS_0 gives the best results at low traffic levels; in fact, RS_0 is almost optimal up to RE = 0.5.

9.2 ARPA Network: Routing and Capacities Assignment

The set of channel capacities available for the ARPA Network is discrete: Table 1 contains the list of capacity options and corresponding costs considered in the present application [6]. In order to be able to apply the FD method, the discrete cost-capacity curves have been approximated with continuous, piece-wise linear curves (see Figure 5). We do not discuss the details of the approximation, but merely mention that they must be concave.* The concavity of the cost-capacity curves implies that the local minima are shortest route flows (see Section 8). The FD method can, therefore, be applied in a form similar to the one presented in Section 8; a few modifications are required due to the non-linearity of the cost-capacity curves.

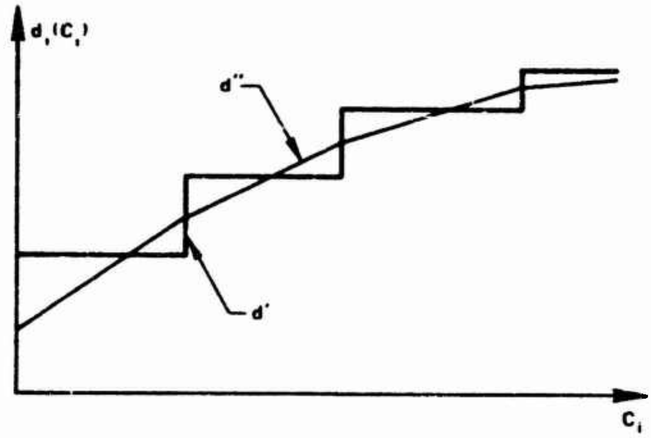
CHANNEL CAPACITIES AND CORRESPONDING
COSTS USED IN THE OPTIMIZATION

Capacity [kbits/sec]	Termination Cost [$/month]	Line Cost [$/month/mile]
7.2	810	.35
19.2	850	2.10
50	850	4.20
108	2400	4.20
230.4	1300	21.00

Table 1

Note: The total cost per month of a channel is given by: total cost = termination cost + (line cost) × (length in miles).

**Other concave approximations can be considered: see [6], [18].*



d': staircase corresponding to discrete capacity levels.
d": piece-wise linear approximation

Fig. 5 Cost-capacity curves for arc i.

A schematic description of the algorithm follows here:

0. Let D_0 be the total dollar investment.

   Let $\underset{\sim}{f}^0 \varepsilon F_B$

   Let $\underset{\sim}{C}^0$ be the optimal capacities assignment for fixed $\underset{\sim}{f}^0$.*
   Let $T_0(\underset{\sim}{f})$ be as from Equation (8.1), using linear approximations of the cost-capacity curves around $\underset{\sim}{C}^0$.
   Let n = 0.

1. Let:
   $\underset{\sim}{f}^{n+1}$ = shortest route flow computed at $\underset{\sim}{f}^n$
   (using metric $\ell_k = [\partial T_n(\underset{\sim}{f})/\partial f_k]_{\underset{\sim}{f}=\underset{\sim}{f}^n}$).

2. Let $\underset{\sim}{C}^{n+1}$ be the optimal capacities assignment for fixed $\underset{\sim}{f}^{n+1}$, and let $T_{n+1}(\underset{\sim}{f})$ be as from Equation (8.1), using linear approximations of the cost-capacity curves around $\underset{\sim}{C}^{n+1}$.

3. If $\left(T_{n+1}(\underset{\sim}{f}^{n+1}) \geq T_n(\underset{\sim}{f}^n)\right)$, stop; $\underset{\sim}{f}^n$ is a local minimum.
   Otherwise, let n = n + 1 and go to 1.

**The optimal assignment of capacities, given the flows and the total dollar investment, for concave cost-capacity functions, has been discussed by Kleinrock [6].*

The result of the above described algorithm is a local minimum for the continuous cost-capacity problem. In order to get a solution for the discrete problem, the capacities and flows given by the algorithm are "adjusted" in the following manner: in all arcs, the capacity is increased to the upper value of discrete capacity available (thus increasing the total investment to $D > D_0$); then, the routing is optimized once again with the FD routing algorithm.

The above described technique is clearly suboptimal. We cannot guarantee that the solutions so found are local minima; in fact, it is not even possible to *define* a local minimum in a discrete problem. Other suboptimal techniques have been proposed [7,10,21]; however, the optimization of a network with discrete capacities still remains a formidable (and basically unsolved) problem.*

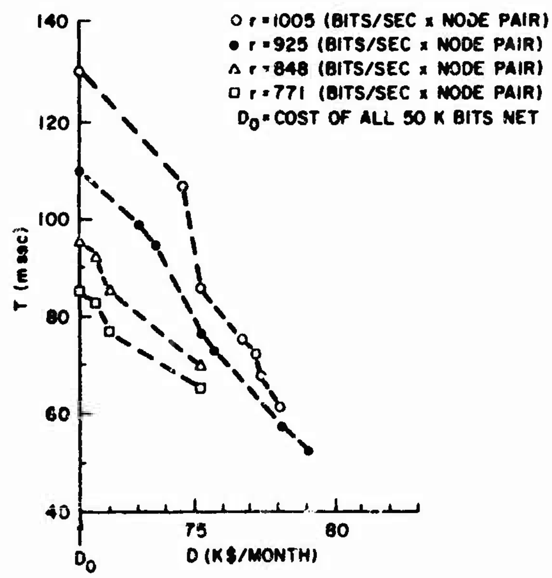


Fig. 6 Delay T versus cost D of various undominated capacity assignments for different traffic levels.

The optimum solution can be obtained, with dynamic programming techniques, in the special case of a centralized network [30]. In fact, for such a case, the problem reduces to the optimal assignment of capacities only, as the flows are already determined by the tree-structure topology.

The technique has been applied to the design of the ARPA Network. Four cases have been run, each with a different value of uniform requirement r (see Figure 5). The initial cost D_0 was made equal to the cost of the proposed network with all 50 kbit channels ($D_0 \approx 71{,}000$ \$/month). In order to be able to compare the 50 kbit capacities assignment to the assignments found with the FD method, the minimum delay T, with all 50 kbit capacities (i.e., with total cost $D = D_0$), was reported on the graph for each value of r (T was obtained from the curves in Figure 4). The delay T and the total cost D of the undominated* solutions are plotted in the graph of Figure 6.

10. CONCLUSION

The FD method can be applied to any *unconstrained* m.c. flow problem when some reasonable assumptions on $P(\underline{f})$ are satisfied. It also can be applied to *constrained* flow problems: in particular to problems that include the constraints as penalties in $P(\underline{f})$, or that have been decomposed with the Dantzig-Wolfe method. Local minima are in general attained; for convex problems, the global minimum is found.

The FD method seems to be an efficient tool for the design of S/F networks: for example, if we consider the optimal routing problem, it can be shown [19] that the amount of computation per iteration required by the FD method is comparable to that of the heuristic techniques so far proposed [16,24].† A general statement, however, about the effectiveness of the FD method as compared to other methods would not be appropriate: many factors, which depend on the specific application (like trade-off between precision and computational speed) should be considered in order to select the proper approach.

APPENDIX: CASE OF $P(\underline{f})$ STRICTLY CONVEX

If $P(\underline{f})$ is strictly convex, a direct proof of convergence of the FD algorithm, defined in Section 5, is available and a lower bound can be established.

**A solution* (T_i, D_i) *is said to be dominated by* (T_j, D_j) *if:*

$$(D_j < D_i) \text{ and } (T_j < T_i)$$

A solution is undominated if it is not dominated by any other solution.

†*The two bottlenecks, common to both approaches, are the shortest route computation and the flow assignment* [16].

Convergence

We want to show that:

$$\lim_{n\to\infty} \underset{\sim}{f}^n = \underset{\sim}{f}^* \qquad \text{(A.1)}$$

where $\underset{\sim}{f}^*$ is the global minimum of $P(\underset{\sim}{f})$ on F, and $\{\underset{\sim}{f}^n\}$ is the sequence generated by recursive application of the FD operator on a given starting flow $\underset{\sim}{f}^0$. The associated sequence $\{P(\underset{\sim}{f}^n)\}$ is monotonically non-increasing and lower bounded by $P^* \triangleq P(\underset{\sim}{f}^*)$, therefore it must converge:

$$\lim_{n\to\infty} P(\underset{\sim}{f}^n) = P' \geq P^* \qquad \text{(A.2)}$$

Also, recalling that:

$$P(\underset{\sim}{f}^n) - P' = \sum_{\ell=n}^{\infty} \Delta P(\underset{\sim}{f}^\ell)$$

where

$$\Delta P(\underset{\sim}{f}^\ell) \triangleq P(\underset{\sim}{f}^\ell) - P(FD \odot \underset{\sim}{f}^\ell) = P(\underset{\sim}{f}^\ell) - P(\underset{\sim}{f}^{\ell+1})$$

and recalling that:

$$\Delta P(\underset{\sim}{f}^\ell) \geq 0 \;\forall \ell$$

we have, from Equation (A.2):

$$\lim_{n\to\infty} \Delta P(\underset{\sim}{f}^n) = 0 \qquad \text{(A.3)}$$

Suppose (A.1) is false; this implies, since $P(\underset{\sim}{f})$ is strictly convex, that $P' > P^*$. However, in such a case, we are able to establish a relation which contradicts Equation (A.3) as follows.

Let us first establish a lower bound on $\Delta P(\underset{\sim}{f})$. Let:

$$P(\lambda) \triangleq P[(1-\lambda)\underset{\sim}{f} + \lambda \underset{\sim}{v}], \quad 0 \leq \lambda \leq 1$$

where: $\underset{\sim}{v}$ is the shortest route flow computed at $\underset{\sim}{f}$. Using Taylor's expansion:

$$P(\lambda) = P(0) + \lambda\left[\frac{dP}{d\lambda}\right]_{\lambda=0} + \frac{1}{2}\lambda^2\left[\frac{d^2P}{d\lambda^2}\right]_{\lambda=\xi} \qquad \text{(A.4)}$$

where ξ is a proper value in the interval $(0,\lambda)$ as usual. By assumption, the second partial derivatives of $P(\underset{\sim}{f})$ are upper bounded; therefore, the second directional derivative is also upper bounded, and Equation (A.4) becomes:

$$P(\lambda) - P(0) \leq \lambda\,\theta + \frac{1}{2}\lambda^2 M \tag{A.5}$$

where

$$\theta \triangleq \sum_{k=1}^{b} \ell_k (v_k - f_k) \leq 0 \tag{A.5)'$$

$$M: \text{ upperbound on } d^2P/d\lambda^2.^*$$

After minimizing both sides of Equation (A.5) over λ, and recalling that $\min\,[P(\lambda) - P(0)] \triangleq -\Delta P(\underset{\sim}{f})$, we get:

$$\Delta P(\underset{\sim}{f}) \geq \begin{cases} \theta^2/2M & \text{if } -\theta/M < 1 \\ M/2 & \text{if } -\theta/M \geq 1 \end{cases} \tag{A.6}$$

Equation (A.6) can be rewritten as follows:

$$\Delta P(\underset{\sim}{f}) \geq \frac{M}{2} \min \left\{ \frac{\theta^2}{M^2}, 1 \right\} \tag{A 6)'$$

Inequality (A.6)' represents a useful lower bound on $\Delta P(\underset{\sim}{f})$.

Consider now:

$$P(\lambda) \triangleq P[(1-\lambda)\underset{\sim}{f}^n + \lambda \cdot \underset{\sim}{f}^*]$$

$$\text{where: } 0 \leq \lambda \leq 1$$

$P(\lambda)$ is strictly convex, therefore it lies above its tangent line at $\lambda = 0$:

$$P(\lambda) \geq P(\underset{\sim}{f}^n) + \lambda \left[\sum_{k=1}^{b} \ell_k (f_k^* - f_k^n) \right] \tag{A.7}$$

$$\text{where: } \ell_k = \left[\frac{\partial P}{\partial f_k}\right]_{\underset{\sim}{f}^n}$$

Letting $\lambda = 1$ in (A.7) and recalling from (A.2) that $P(\underset{\sim}{f}^n) \geq P'$:

$$P(\underset{\sim}{f}^*) = P^* \geq P' + \sum_{k=1}^{b} \ell_k (f_k^* - f_k^n) \tag{A.8}$$

Notice that $M > 0$ as $P(\lambda)$ is strictly convex.

Let $\underset{\sim}{v}$ be the shortest route flow computed at $\underset{\sim}{f}^n$; we have, from Equation (A.8):

$$P^* \geq P' + \sum_{k=1}^{b} \ell_k (v_k - f_k^n) \tag{A.9}$$

From (A.9), using definition (A.5)', we have:

$$P' - P^* \leq |\theta| \tag{A.10}$$

Introducing (A.10) into (A.6)' we get:

$$\Delta P(\underset{\sim}{f}^n) \geq \frac{M}{2} \min \left\{ \frac{(P' - P^*)^2}{M^2}, 1 \right\} > 0 \tag{A.11}$$

The r.h.s. of Equation (A.11) is independent of n and strictly positive, therefore:

$$\lim_{n \to \infty} \Delta P(\underset{\sim}{f}^n) > 0 \tag{A.12}$$

Equation (A.12) contradicts Equation (A.3). Therefore (A.1) is true.

Lower Bound

By replacing $\underset{\sim}{f}^n$ with a generic $\underset{\sim}{f} \in F$ in (A.7), and letting $\lambda = 1$, we get, after a few steps:

$$P(\underset{\sim}{f}^*) \geq P(\underset{\sim}{f}) + \sum_{k=1}^{b} \ell_k (v_k - f_k) \tag{A.13}$$

where: $\underset{\sim}{f}^*$ is the global minimum

$\underset{\sim}{v}$ is the shortest route flow computed at $\underset{\sim}{f}$

From (A.13), lower and upper bounds on $P(\underset{\sim}{f}^*)$ are readily available:

$$P(\underset{\sim}{f}) \geq P(\underset{\sim}{f}^*) \geq P(\underset{\sim}{f}) + \sum_{k=1}^{b} \ell_k (v_k - f_k)$$

Notice that the test for optimality based on $\left| \sum_{k=1}^{b} \ell_k (v_k - f_k) \right|$ (see Section 5) is very powerful in the case of $P(\underset{\sim}{f})$ strictly convex, as it provides an upper bound on the optimal value error.

ACKNOWLEDGMENT

The authors are pleased to acknowledge Dr. D. G. Cantor for his valuable advice and his assistance in the development of the optimal routing algorithm.

REFERENCES

1. Kleinrock, L., *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill Book Co., New York, 1964.

2. Steiglitz, K., P. Weiner and D. J. Kleitman, "The Design of Minimum Cost Survivable Networks," *Trans. on Circuit Theory*, November 1969, pp. 455-460.

3. Roberts, L., "Multiple Computer Networks and Intercomputer Communications," *ACM Symposium on Operating Systems Principles*, Gatlinburg, Tennessee, October 1967.

4. Roberts, L. and B. Wessler, "Computer Network Development to Achieve Resource Sharing," *AFIPS Conference Proc.*, SJCC, May 1970, pp. 543-549.

5. Heart, R., R. E. Kahn, S. M. Ornstein, W. R. Crowther and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," *AFIPS Conference Proc.*, SJCC, May 1970, pp. 551-567.

6. Kleinrock, L., "Analytic and Simulation Methods in Computer Network Design," *AFIPS Conference Proc.*, SJCC, May 1970, pp. 569-579.

7. Frank, H., I. T. Frisch and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," *AFIPS Conference Proc.*, SJCC, May 1970, pp. 581-587.

8. Carr, S., S. Crocker and V. Cerf, "HOST-to-HOST Communication Protocol in the ARPA Network," *AFIPS Conference Proc.*, SJCC, May 1970, pp. 589-597.

9. Hu, T. C., *Integer Programming and Network Flows*, Addison-Wesley, 1969.

10. Meister, B., H. R. Mueller, and H. R. Rudin, Jr., "Optimization of a New Model for Message-Switching Networks," *ICC'71*, pp. 39-16 to 39-21.

11. Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1962.

12. Ford, L. K. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962.

13. Ford, L. K. and D. R. Fulkerson, "A Suggested Computation for Maximal Multicommodity Network Flows," *Management Science*, Vol. 5, 1958, pp. 97-101.

14. Tomlin, J. A., "Minimum Cost Multi-Commodity Network Flows," *Operations Research*, Vol. 14, No. 1, January 1966, pp. 45-47.

15. Charnes, A. and W. W. Cooper, "Multicopy Traffic Network Models," *Proc. of the Symposium on the Theory of Traffic Flow*, (R. Herman, ed.), 1961.

16. Network Analysis Corporation, "Research in Store and Forward Computer Networks," *Fourth Semiannual Technical Report*, December 1971, (Contract No. DAHC 15-70-C-0120).

17. Dafermos, S. C. and F. T. Sparrow, "The Traffic Assignment Problem for a General Network," *Journal of Research of the National Bureau of Standards - B*, Vol. 73B, No. 2, April 1969.

18. Yaged, B. Jr., "Minimum Cost Routing for Static Network Models," *Networks*, Vol. 1, No. 2, 1971, pp. 139-172.

19. Cantor, D. and M. Gerla, "The Optimal Routing of Messages in a Computer Network via Mathematical Programming," *IEEE Comp. Conference*, San Francisco, September 1972.

20. Wardrop, J. G. "Some Theoretical Aspects of Road Traffic Research," *Proc. Inst. Civ. Eng. Part II*, I, 1952, pp. 325-378.

21. Gerla, M., "The Design of Store-and-Forward Networks for Computer Communications," Ph.D. Dissertation, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, California, January 1973.

22. Fratta, L. and M. Gerla, "The Synthesis of Computer Networks: Properties of the Optimum Solution," *ACM-International Computing Symposium*, Venice, Italy, April 1972.

23. Mangasarian, O., *Nonlinear Programming*, McGraw-Hill Book Co., New York, 1969.

24. Frank, H. and W. Chou, "Routing in Computer Networks," *Networks*, Vol. 1, No. 2, 1971, pp. 99-112.

25. Ornstein, S. M., F. E. Heart, W. R. Crowther, H. K. Rising, S. B. Russell and A. Michel, "The Terminal IMP for the ARPA Computer Network," *AFIPS Conference Proc.*, SJCC, 1972, pp. 243-254.

26. Frank H., R. E. Kahn and L. Kleinrock, "Computer Communication Network Design -- Experience with Theory and Practice," *AFIPS Conference Proc.*, SJCC, 1972, pp. 255-270.

27. Crocker, S. D., J. Heafner, J. Metcalfe and J. Postel, "Function-Oriented Protocols for the ARPA Computer Network," *AFIPS Conference Proc.*, SJCC, 1972, pp. 271-279.

28. Thomas, R. H. and D. A. Henderson, Jr., "McROSS -- A Multi-Computer Programming System," *AFIPS Conference Proc.*, SJCC, 1972, pp. 281-293.

29. Roberts, L. G., "Extension of Packet Communication Technology to a Hand-Held Personal Terminal," *AFIPS Conference Proc.*, SJCC, 1972, pp. 295-298.

30. Frank, H., I. T. Frisch, W. Chou and R. Van Slyke, "Optimal Design of Centralized Computer Networks," *Networks*, Vol. 1, No. 1, pp. 43-57.

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense, Contract No. DAHC-15-69-C-0285, and in part by a 1971-72 John Simon Guggenheim Fellowship awarded to L. Kleinrock.

Paper received December 2, 1971.

5. ON NON-BLOCKING SWITCHING NETWORKS

by D. G. Cantor

On Non-Blocking Switching Networks

D. G. Cantor
Department of Mathematics
University of California
Los Angeles, California

ABSTRACT

A switching network may be informally described as a collection of single-pole, single-throw switches arranged so as to connect a set of terminals called inputs to another set of terminals called outputs. It is non-blocking if, given any set of connections from some of the inputs to some of the outputs, and an idle input terminal x and idle output terminal y, then it is possible to connect x to y without disturbing any of the existing connections. Denote by $\sigma(a,b)$ the minimal number of switches necessary to connect a inputs to b outputs using a non-blocking network. We are interested in studying the growth of $\sigma(a,a)$ as $a \to \infty$. Results of C. Clos show that $\sigma(a,a) \le C\, ae^{2\sqrt{\log a \cdot \log 2}}$. We show that $\sigma(a,a) \le 8a(\log_2 a)^2$.

1. INTRODUCTION

A *network* N consists of a graph G; two sets of vertices of G, denoted A and B and called, respectively, the (sets of) *inputs* and *outputs;* and a set P of paths of G. Each path in P connects an input to an output and meets no other inputs or outputs. We write $N = (G,A,B,P)$. A *state* of N is a subset S of P such that no two paths in S have a common vertex. A state S defines a bijection f_S from a subset of A to a subset of B as follows: Suppose $p \in S$ and p connects $x \in A$ to $y \in B$; put $f_S(x) = y$, and repeat this for each path in S. We shall say that a path p of G is *admissible* if $p \in P$. If x is a vertex of G we shall say that x is *busy* (in the state S) if x lies on a path $p \in S$; otherwise we shall say that x is *idle* (in the state S). If x is an input of G and y is an output of G, we shall

Networks, 1: 367-377

say that x has *access* to y (in the state S) if there exists a path $p \in P$ connecting x to · and such that $S \cup \{p\}$ is a state.

A network $N = (G,A,B,P$. may be interpreted as a switching device; under this interpretation, the elements of A are considered as input terminals, the elements of B are considered as output terminals, and the edges of G are considered as single-pole, single-throw switches which are normally open. Then a path p, which connects $x \in A$ to $y \in B$ may be thought of as a sequence of switches which, when closed, connect x to y. The state S yields a collection of switches (all edges on any path in S) which, when closed, connect inputs to outputs in the manner described by the function f_S.

The network $N = (G,A,B,P)$ is said to be *non-blocking* if given any state S of N and idle vertices $x \in A$, $y \in B$, then x has access to y in the state S. In terms of the switching network interpretation mentioned above, this means that if x and y are idle input and output terminals, respectively, then it is possible to establish a connection between them without disturbing the existing connections.

From now on, all the networks we study will have disjoint inputs and outputs (i.e. $A \cap B = \emptyset$).

Given positive integers a and b we are interested in finding those non-blocking networks $N = (G,A,B,P)$ with $|A| = a$, $|B| = b$ for which the number of edges of G is minimal. We shall denote this number by $\sigma(a,b)$. In terms of switching networks, this amount to finding non-blocking networks using a minimal number of switches. An obvious non-blocking network with a inputs and b outputs is the network whose graph is the complete bipartite graph on vertex sets A and B with $|A| = a$ and $|B| = b$. In this graph the set of vertices is $A \cup B$ and there is an edge connecting each vertex in A to each vertex in B. The set P consists of all paths consisting of exactly one edge. Thus P has ab elements. In the switching network interpretation, this amounts to an a by b crossbar switch. When the names of the sets A and B are unimportant, we shall denote this network by C_{ab}. The network C_{ab} shows that $\sigma(a,b) \leq ab$.

It was Clos [2] who showed that $\sigma(N,N) < N^2$ for all large N. His methods, which will be described later, show that $\sigma(N,N)$ $\leq C\, Ne^{2\sqrt{(\log N)\cdot(\log 2)}}$. We will show that $\sigma(N,N) \leq 8N(\log_2 N)^2$. We do not attempt to obtain the smallest possible constant multiplier, for it is not clear that the exponent 2 can not be reduced. In the opposite direction, an elementary argument shows that $\sigma(N,N) > C\, N \log_2 N$, and nothing stronger is known.

The author would like to acknowledge many stimulating discussions with Professors B. Gordon and C. B. Tompkins.

2. CONSTRUCTIONS

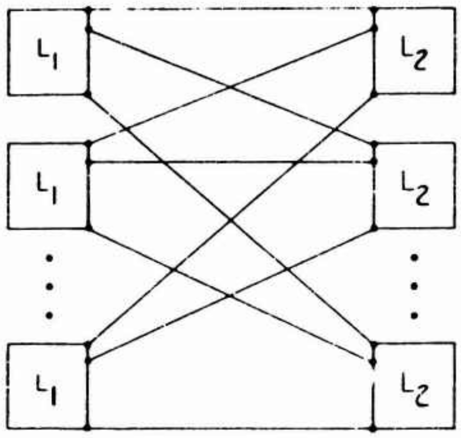
We shall say that networks $N = (G,A,B,P)$ and $N_1 = (G_1,A_1,B_1,P_1)$ are *isomorphic* (or *equivalent*) if there exists a graph isomorphism μ of G onto G_1 such that $\mu(A) = A_1$, $\mu(B) = B_1$, and $\mu(P) = P_1$. It is clear that the property of being non-blocking is preserved under isomorphism.

If $N = (G,A,B,P)$ is a network, we define its transpose N' to be the network $N' = (G,B,A,P)$; clearly $N'' = N$.

If G is a graph and C is a set, we define the graph $G \times C$ to be the graph whose vertices are the ordered pairs (x,c) with x a vertex of G and $c \in C$; $((x_1,c_1), (x_2,c_2))$ is an edge of $G \times C$ if $c_1 = c_2$ and (x_1,x_2) is an edge of G. If p is a path in G whose vertices, in order, are $x_0, x_1, \ldots, x_n$ then by $p \times c$ we mean the path in $G \times C$ whose vertices are (x_0,c), $(x_1 c)$, $\ldots$, (x_n,c). The product $C \times G$ is defined similarly.

Now suppose $L_i = (G_i,A_i,B_i,P_i)$ $(i = 1$ or $2)$ are networks; we are going to define the network product $L_1 \times L_2$. We shall denote this product by $N = (H,C,D,Q)$. Put $C = A_1 \times A_2$ and $D = B_1 \times B_2$. The graph H is obtained from the two graphs $G_1 \times A_2$ and $B_1 \times G_2$ by identifying the vertices in $B_1 \times A_2$, which appear in both graphs. All admissible paths $q \in Q$ of N are obtained as follows: Let $p_i \in P_i$ be an admissible path connecting $x_i \in A_i$ to $y_i \in B_i$ $(i = 1$ or $2)$. Then $p_1 \times x_2$ ends in the vertex (y_1,x_2) which is the first vertex of $y_1 \times p_2$. The path $q = (p_1,p_2)$ is defined to be the path obtained from the paths $p_1 \times x_2$ and $y_1 \times p_2$ by concatenating them and identifying the common vertex (y_1,x_2). Note that this maps $P_1 \times P_2$ onto Q.

In the switching network interpretation this construction amounts to taking $|A_2|$ copies of L_1 and $|B_1|$ copies of L_2, and connecting the outputs of each of the copies of L_1 to the inputs of all of the copies of L_2 (see Figure 1).



The lines do *not* represent edges; they connect output vertices of L_1 to the input vertices of L_2 with which they are identified.

Fig. 1 $L_1 \times L_2$

Let a_i, b_i, c, d denote, respectively, the cardinalities of A_i, B_i, C, D, and let g_i, h denote, respectively the number of edges of G_i and H. The following relationship between two by two matrices is easily verified

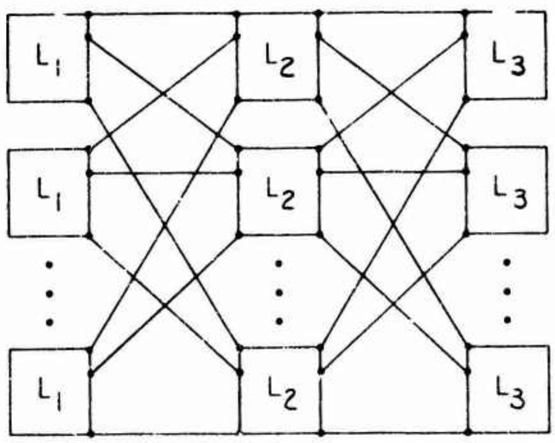
$$\begin{pmatrix} a_1 & 0 \\ g_1 & b_1 \end{pmatrix}\begin{pmatrix} a_2 & 0 \\ g_2 & b_2 \end{pmatrix} = \begin{pmatrix} c & 0 \\ h & d \end{pmatrix}. \tag{1}$$

If L_1 is isomorphic to M_1 and L_2 is isomorphic to M_2 it is easy to verify that $L_1 \times L_2$ is isomorphic to $M_1 \times M_2$. Furthermore $(L_1 \times L_2)' = L_2' \times L_1'$. Finally, we have associativity: $(L_1 \times L_2) \times L_3 = L_1 \times (L_2 \times L_3)$; we will usually write simply $L_1 \times L_2 \times L_3$. We will abbreviate the k-fold product $L \times L \times \cdots \times L$ by L^k.

We also define a triple product of the three networks $L_i = (G_i, A_i, B_i, P_i)$ $(i = 1,2,3)$ when $|B_1| = |A_3|$. Let τ be a bijection from A_3 onto B_1; the triple product of L_1, L_2, L_3 depends upon the choice of τ and will be denoted by $[L_1, L_2, L_3]_\tau$. (In many cases L_3 will be L_1' and in such cases we will choose τ to be the identity map. In any case those properties of the triple product which we will use will be independent of the choice of τ and we will frequently write $[L_1,L_2,L_3]$ instead of $[L_1,L_2,L_3]_\tau$.) Suppose then that $N = (H,C,D,Q)$ is $[L_1,L_2,L_3]_\tau$.

We put $C = A_1 \times A_2$ and $D = B_3 \times B_2$; H is defined as the graph obtained from the three graphs $G_1 \times A_2$, $B_1 \times G_2$, and $G_3 \times B_2$, by identifying $B_1 \times A_2$ in $G_1 \times A_2$ with $B_1 \times A_2$ in $B_1 \times G_2$, and by identifying $A_3 \times B_2$ in $G_3 \times B_2$ with $\tau(A_3) \times B_2 = B_1 \times B_2$ in $B_1 \times G_2$. The admissible paths $q \in Q$ are obtained in the following way: Let p_i be an admissible path of L_i connecting $x_i \in A_i$ to $y_i \in B_i$ $(i = 1,2,3)$ and suppose $\tau(x_3) = y_1$. Then $p_1 \times x_2$ ends at (y_1,x_2); $y_1 \times p_2$ begins at (y_1,x_2) and ends at (y_1,y_2); and $p_3 \times y_2$ begins at $(x_3,y_2) = (y_1,y_2)$. The path q is obtained by concatenating $p_1 \times x_2$, $y_1 \times p_2$, $p_3 \times y_2$ and identifying the vertices common to two segments of q.

Note that $[L_1,L_2,L_3]$ is, in general, different from $L_1 \times L_2 \times L_3$ (see Figure 2).



The lines do *not* represent edges; instead they connect vertices which are to be identified.

Fig. 2 $[L_1,L_2,L_3]$

Theorem 2.1: *Suppose* $L_i = (G_i, A_i, B_i, P_i)$ $(i = 1,2,3,4,5)$ *are networks. Suppose* τ_1 *is a bijection of* A_5 *onto* B_1. *Then*

$$[L_1, [L_2, L_3, L_4]_{\tau_1}, L_5]_{\tau_2} = [L_1 \times L_2, L_3, L_4 \times L_5]_{\tau},$$

where τ_3 *is the bijection from* $A_4 \times A_5$ *to* $B_1 \times B_2$ *given by* $\tau(a_4, a_5) = (\tau_2(a_5), \tau_1(a_4))$.

3. THE CLOS METHOD AND SOME VARIATIONS

The basic method, due to Clos [2] and quoted by Beněs [1] may be stated as the

Theorem (Clos): *Suppose* $L = (G,A,B,P)$ *is non-blocking and* $s \geq 2r - 1$. *Then* $N = [C_{rs}, L, C_{sr}]$ *is non-blocking.*

This is a special case of the following more general

Theorem 3.1: *Suppose* $L_i = (G_i, A_i, B_i, P_i)$ $(i = 1,2,3)$ *are non-blocking, that* $|B_1| \geq |A_1| + |B_3| - 1$, *and that* $|B_1| = |A_3|$. *Then* $N_\tau = [L_1, L_2, L_3]_\tau$ *is non-blocking for any bijection* τ *of* A_3 *onto* B_1.

Proof: Suppose $N_\tau = (H,C,D,Q)$ is in state S, and that $x \in C$, $y \in D$ are idle. We must show there exists a path $q \in Q$ connecting x to y and having no common vertices with any path in S. Suppose $x = (u_1, u_2) \in A_1 \times A_2$ and $y = (v_3, v_2) \in B_3 \times B_2$. There are $|A_1|$ vertices of the form $(u, u_2) \in A_1 \times A_2$ and at most $|A_1| - 1$ of them are busy. Hence at most $|A_1| - 1$ of the $|B_1|$ vertices of the form $(y, u_2) \in B_1 \times A_2$ are busy and hence at least $|B_1| - |A_1| + 1$ of them are idle. Denote these vertices by $(y_{i_1}, u_2), (y_{i_2}, u_2), \ldots, (y_{i_r}, u_2)$, so that $r \geq |B_1| - |A_1| + 1$. Similarly, there are vertices $(z_{i_1}, v_1), (z_{i_2}, v_2), \ldots, (z_{i_s}, v_s)$ in $A_3 \times B_2$ which are idle, and $s \geq |A_3| - |B_3| + 1$. The $r + s$ vertices $y_{i_1}, y_{i_2}, \ldots, y_{i_r}, \tau(z_{i_1}), \tau(z_{i_2}), \ldots, \tau(z_{i_s})$ all lie in B_1 and

$$\begin{aligned} r + s &\geq |B_1| - |A_1| + 1 + |B_1| - |B_3| + 1 \\ &\geq |B_1| + 1 + (|B_1| - |A_1| - |B_3| + 1) \\ &\geq |B_1| + 1. \end{aligned}$$

So two of them must be the same. Now the y_{i_j} are all distinct and so are the $\tau(z_{i_j})$. Thus there must be a y_{i_j} equal to a $\tau(z_{i_k})$, say $y_{i_1} = \tau(z_{j_1})$. Since L_1 is non-blocking there is a path p_1 connecting u_1 to y_{i_1} and such that $p_1 \times u_2$ has no common vertices with any vertex in S. Similarly there is a path p_2 from u_2 to v_2 in P_2 such that $y_{i_1} \times p_2$ has no common vertices with any path in S, and there is a path p_3 from z_{j_1} to v_3 in P_3 such that $p_3 \times v_2$ has no vertex in common with any path in S. Let q be the concatenation of $p_1 \times u_2$, $y_{i_1} \times p_2$, and $p_3 \times v_2$ with the appropriate vertices identified. Then q connects x to y and $S \cup \{q\}$ is a state of N_τ.

Remark 3.2: Suppose $a_i = |A_i|$, $b_i = |B_i|$ and g_i is the number of edges of G_i $(i = 1,2,3)$. It is easy to verify using (1) that $N = [L_1,L_2,L_3]$ has a_1a_2 inputs, b_2b_3 outputs and that its graph has $a_2g_1 + b_1g_2 + b_2g_3$ edges.

Clos [2] suggests using networks which may be described as

$$[L,[L,[L, \ldots, [L,M,L'],L'],L'], \ldots, L']$$

where $L = C_{n,2n-1}$ and $M = C_{n,n}$. By Theorem 2.1, this is the same as $[L^t,M,(L')^t]$, where $L^t = L \times L \times L \times \cdots \times L$ (t times). He shows that this non-blocking network, which has n^{t+1} inputs and outputs, has

$$\frac{n^2(2n - 1)}{n - 1} [(5n - 3)(2n - 1)^{t-1} - 2n^t]$$

edges. This follows immediately from the above remark. It is easy to verify that a non-blocking network with N inputs and outputs, constructed by this method, will require at least $C_0 N e^{2\sqrt{\log N \cdot \log 2}}$ edges, where $C_0 > 0$ is a constant.

Suppose that L_{ab} denotes a network with a inputs, b outputs, and whose graph contains a minimal number of edges, namely $\sigma(a,b)$. Using two copies of L_{aa} shows that $\sigma(a,2a) \leq 2\sigma(a,a)$. By Theorem 3.1, $[L_{a,2a}, L_{a,a}, L_{2a,a}]$ is non-blocking and by Remark 3.2, it has $\leq a\sigma(a,2a) + 2a\sigma(a,a) + a\sigma(2a,a) \leq 6a\sigma(a,a)$ edges. Thus

$$\sigma(a^2,a^2) < 6a\sigma(a,a). \qquad (2)$$

Iteration of (2) shows that $\sigma(N,N) \leq C\, N(\log N)^{\log_2 6}$. This result can be improved by considering $[L_{a,2a}, L_{a,2b}, L_{2a,a}]$; this network has ab inputs, 2ab outputs and its graph has $3b\sigma(a,2a) + 2a\sigma(b,2b)$ edges. This shows that

$$\sigma(ab,2ab) \leq 3b\sigma(a,2a) + 2a\sigma(b,2b). \qquad (3)$$

Putting a = b and iterating (3) shows that $\sigma(a,2a) \leq C\, a(\log_2 a)^{\log_2 5}$ and since $\sigma(a,a) \leq \sigma(a,2a)$ we find that

$$\sigma(N,N) \leq C\, N(\log_2 N)^{\log_2 5}$$

The exponent $\log_2 5$ can be decreased by choosing a and b differently. Let $\alpha > 1$ and $\beta > 2$ be the real solutions of the simultaneous equations

$$\left\{ \begin{aligned} \alpha^{\beta-1} &= 3 \\ (\alpha - 1)^{\beta-1} &= 3/2 \end{aligned} \right\} \qquad (4)$$

Numerical computation shows that $\alpha \approx 2.37638$ and $\beta \approx 2.26922$. Multiplying the second equation of (4) by $\alpha - 1$ and substituting from the first yields $2(\alpha - 1)^\beta = \alpha^\beta - 3$ or equivalently

$$3(1/\alpha)^\beta + 2(1 - 1/\alpha)^\beta = 1. \qquad (5)$$

We now show that if $\mu(x) = (\log x)^\beta$, then $\mu(x)$ satisfies the functional equation

$$\mu(z) = 3\mu(x) + 2\mu(y) \tag{6}$$

where $x = z^{1/\alpha}$ and $y = z/x$. Indeed,

$$3\mu(x) + 2\mu(y) = 3((\log z)/\alpha)^{\beta} + 2(\log z)^{\beta}(1 - 1/\alpha)^{\beta}$$
$$= (\log z)^{\beta}$$
$$= \mu(z)$$

using (5).

Now $\sigma(x,2x)/x$ satisfies a functional inequality similar to (6) where x and y must be integers. It follows that for each $\varepsilon > 0$, the exists $C_\varepsilon > 0$ such that

$$\sigma(N,2N) \leq C_\varepsilon \; N(\log N)^{\beta+\varepsilon}.$$

For comparison, $\log_2 5 = 2.32193$.

4. THE EXPONENT IS ≤ 2

Suppose $L = (G,A,B,P)$ is a network (not necessarily non-blocking). We shall say that L is of type $T(m,n)$ if, given any state S of L and m idle inputs $x_1, x_2, \ldots, x_m$ of L, then each has access, in the state S, to at least n outputs of L.

Lemma 4.1: Suppose $L = (G,A,B,P)$ is of type $T(m,m+n-1)$ for $1 \leq m \leq k$, that M is a non-blocking network with c inputs and d outputs, and that $nd \geq a(c-1)$. Then $L \times M$ is of type $T(m,m+n'-1)$ for $1 \leq m \leq k$ where $n' = nd - a(c-1)$ and a is the number of inputs of L.

Proof: Take $k \leq m$ idle inputs $z_1, z_2, \ldots, z_k$. Suppose, for example, that $z_1, z_2, \ldots, z_{k'}$ are of the form

$$(x_1,y_1),(x_2,y_1), \ldots, (x_{k'},y_1),$$

and $z_{k'+1}, z_{k'+2}, \ldots, z_k$ are of the form (x_h,y_i) where $i \geq 2$; here the x_j are inputs of L and the y_j are inputs of M. By hypothesis, (x_1,y_1) has access to at least $n + k' - 1$ vertices of the form (u_j,y_j) where the u_j are outputs of L. Since M is non-blocking, these have access to all idle vertices of the form (u_j, v_k) where v_k is an output of M. There are $(n + k' - 1)d$ such vertices. However, as many as $(c - 1)a - (k - k')$ of these could be busy; this would be the case if all inputs of the form (x_h, y_i), where $i \geq 2$, other than $z_{k'+1}, z_{k'+2}, \ldots, z_k$ were busy. Thus z_1 has access to at least

$$(n + k' - 1)d - (c - 1)a + (k - k') \geq nd - (c - 1)a + k - 1$$
$$= n' + k - 1$$

output terminals of $L \times M$.

The following theorem provides the motivation for defining the notion $T(m,n)$.

Theorem 4.2: Suppose M is a non-blocking network and L is a network with a inputs, b outputs, and of type T(1,n). If 2n > b, then [L,M,L'] is non-blocking.

The proof is similar to that of Theorem 3.1 and will be omitted.

Now choose an integer $k \geq 1$ and put $L_j = C_{2,2k} \times C_{2,2}^{j-1}$; if $1 \leq j \leq k$, then L_j has 2^j inputs, $k \cdot 2^j$ outputs, and inductively by Lemma 4.1, L_j is of type $T(1,2^{j-1}(2k - j))$ and $T(2,2^{j-1}(2k - j) + 1)$. Thus L_k is of type $T(2,k2^{k-1} + 1)$. Let M_k be obtained from L_k by omitting one input. Then M_k has $2^k - 1$ inputs $k \cdot 2^k$ outputs, is of type $T(1,k \cdot 2^{k-1} + 1)$, and its graph has no more edges than the graph of L_k. The associated matrix of L_k is

$$\begin{pmatrix} 2 & 0 \\ 4k & 2k \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 4 & 2 \end{pmatrix}^{k-1} = 2^k \begin{pmatrix} 1 & 0 \\ 2k^2 & k \end{pmatrix}.$$

Thus M_k has $\leq 2^k \cdot 2k^2$ edges and if N is any non-blocking network, then by Theorem 4.2, so is $[M_k, N, M_k']$. Thus putting, for example, $N = C_{2,2}$, we obtain a non-blocking network with $(2^{k+1} - 2)$ inputs and outputs whose graph has $\leq 2^{k+1}(4k^2 + 2k)$ edges. It is immediate that $\sigma(N,N) \leq 8N(\log_2 N)^2$ for all $N \geq 2$. It is not hard to see that the constant 8 could be considerably decreased, but the major open question is the value of the exponent.

ACKNOWLEDGMENT

The author would like to thank the National Science Foundation GP # 13164, the Sloan Foundation, and DoD Advanced Research Projects Agency Contract DAMC 15-69-C-0285 for support while writing this paper.

REFERENCES

1. V. E. Beneš, *Mathematical Theory of Connecting Networks and Telephone Traffic*, New York, Academic Press, 1965.

2. C. Clos, "A Study of Non-Blocking Switching Networks," *Bell System Tech. J. 32* (1953), pp. 406-424.

Paper received May 13, 1970.