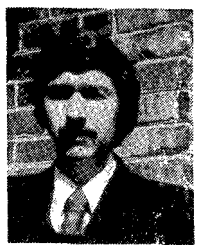
generalized vertex median of a weighted graph," *Operations Res.*, pp. 955–961, July 1967.
[39] R. R. Trippi, "The warehouse location formulation as a special type of inspection problem," *Management Sci.*, vol. 21, pp. 986–988, May 1975.
[40] L. S. Woo and D. T. Tang, "Optimization of teleprocessing networks with concentrators," in *Conf. Rec., Nat. Telecommun. Conf.*, Atlanta, GA, Nov. 26–28, 1973, pp. 37C1-37C5.
[41] C. T. Zahn, "Graph theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, pp. 68–86, Jan. 1971.

★



Patrick V. McGregor (S'67-M'73) received the B.S. degree in electrical engineering from Purdue University, Lafayette, IN, in 1968, and the M.S. and Ph.D. degrees from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1970 and 1973, respectively.

He spent two and a half years at Bell Laboratories, Murray Hill, NJ, developing telemetrized automatic surveillance and control systems. He is now Manager at Data Communications Systems, Vienna, VA, where he has major responsibilities in research and development of network analysis and design capabilities, and has applied these capabilities in the direction of projects ranging from feasability analysis and design of front end processors for the Navy to development of network architectures for the FAA.

★

Diana Shen received the B.S. degree in mathematics from Providence College, Taichung, Taiwan, in 1968 and the M.S. degree from the State University of New York, Stony Brook, in 1971.

From 1968 to 1969 she taught mathematics at Dominica High School, Kaohsuing, Taiwan. From 1972 to 1975 she was a Staff Scientist at Network Analysis Corporation, Glen Cove, NY, responsible for contributing to the ongoing research in the areas of large network design, topological optimization for terminal access, the concentrator location problem, and flow and congestion control strategies for packet switching networks. At present, she is with Graphnet Systems, Inc., Englewood, NJ, where, as a Staff Analyst, she is involved in improving the present Graphnet network.

A Minimum Delay Routing Algorithm Using Distributed Computation

ROBERT G. GALLAGER, FELLOW, IEEE

Abstract—**An algorithm is defined for establishing routing tables in the individual nodes of a data network. The routing table at a node *i* specifies, for each other node *j*, what fraction of the traffic destined for node *j* should leave node *i* on each of the links emanating from node *i*. The algorithm is applied independently at each node and successively updates the routing table at that node based on information communicated between adjacent nodes about the marginal delay to each destination. For stationary input traffic statistics, the average delay per message through the network converges, with successive updates of the routing tables, to the minimum average delay over all routing assignments. The algorithm has the additional property that the traffic to each destination is guaranteed to be loop free at each iteration of the algorithm. In addition, a new global convergence theorem for noncontinuous iteration algorithms is developed.**

Manuscript received March 16, 1976; revised September 15, 1976. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense under Grant N00014-75-C-1183, in part by the National Science Foundation under Grant NSF-ENG75-14103, and in part by Codex Corporation, Newton, MA 02195. This paper was presented at the International Conference on Communications, Philadelphia, PA, June 14-16, 1976.

The author is with the Department of Electrical Engineering and Computer Science and the Electronic Systems Laboratory/Research Laboratory for Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139.

INTRODUCTION

THE problem of routing assignments has been one of the most intensively studied areas in the field of data networks in recent years. These routing problems can be roughly classified as static routing, quasi-static routing, and dynamic routing. Static routing can be typified by the following type of problem. One wishes to establish a new data network and makes various assumptions about the node locations, the link locations, and the capacities of the links. Given the traffic between each source and destination, one can calculate the traffic on each link as a function of the routing of the traffic. If one approximates the queueing delays on each link as a function of the link traffic, one can calculate the expected delay per message in the network. The problem then is to choose routes in such a way as to minimize expected delay. This is a multicommodity flow problem, and the reader is referred to Cantor and Gerla [1] for a particularly elegant algorithm and for other references.

Quasi-static routing problems can be typified by the following situation. A data network is in operation, but over

time, new source-receiver pairs establish data transmission sessions and old sessions are terminated. It is necessary at the very least to establish routes for these new sessions and it might in addition be desirable to occasionally change routes for established sessions or to change the fraction of the traffic for a session that takes different routes. Over a longer range time scale, links or nodes fail, new links and nodes are added, and routings must be changed accordingly. The usual approach to this problem is to have a special node in the network that makes all decisions about routings. In principle such a node periodically gets information from all the other nodes about traffic requirements and uses this information to solve the current static routing problem. Such a strategy seems simple and straightforward, but in fact it is not. First there is the need for protocols for the nodes in the network to send updating information to the control node. Similarly protocols are required for the control node to send its routing decisions to the other nodes. There is also a serious problem about what to do when nodes or links in the network fail. The routes by which notification of such catastrophes are sent to the control node might in fact be destroyed by the catastrophe. Finally, there is the possibility that a failure of the control node may cause the whole network to fail. The point of this is not that central node routing is unworkable, but rather to convince the reader that the problems of communicating information about routing through a network is conceptually as difficult as making routing decisions once all the information is available.

Finally, dynamic routing refers to the kinds of problems that arise in a network when messages or packets are routed according to the instantaneous states of the queues at the links of the network. The routing of a particular message or packet is not determined when it enters the network; instead, each node that receives the message selects the next node to which the message is routed on its path to the destination. Here, in addition to the problem of determining an algorithm to make these decisions, there is also the problem of conveying information about queue lengths through the network and the problem of coping with lost messages and messages which arrive out of order at the destination node.

Our major interest here is in distributed algorithms for quasi-static routing, i.e., in algorithms in which each node constructs its own routing tables based on periodic updating information from neighboring nodes. We first develop a number of theoretical results that should be applicable to any such algorithm and then we develop a particular algorithm. The analysis is based on a static model with stationary traffic inputs and an unchanging network. We show that the average delay per message converges under these conditions to the minimum over all routing assignments. We have not addressed the problem of how well the algorithm adapts to variations in the input traffic or the network. Qualitatively, an algorithm's ability to adapt to variations is intimately connected with its speed of convergence in the static case and with its robustness. We feel that distributed algorithms have important advantages in both these areas. A distributed algorithm can react rapidly to a local disturbance at the point of the disturbance with slower "fine tuning" in the rest of the network. The robustness comes from lack of reliance on a central node that might fail and from avoiding the "chicken and egg" problem of centralized routing where one needs routes to transmit the routing information required to establish routes.

The algorithm here is quite similar to the algorithm used in the Advanced Research Projects Agency Network (ARPANET) [2]. The major difference is that the ARPANET attempts to send each packet over a route that minimizes that packet's delay with no regard to other packet's delays, whereas here packets are sent over routes to minimize the overall delay of all messages. This difference between "user optimization" and "system optimization" was evidently first noticed by Pigou [3], later used by Dafermos and Sparrow [4], and then by Agnew [5], [6]. Angew analyzed a network with a single source and destination and described an algorithm very similar to that described here. Kahn and Crowther [7] also developed a distributed algorithm which meters traffic so as to change routes slowly in response to quasi-static variations. Stern [8] developed another distributed algorithm based on an electrical network analogy of a communication network. Finally our algorithm has similarities to the centralized flow deviation strategy of Fratta *et al.* [8]. Their algorithm was the first to effectively exploit the marginal change in network delay with a change in link flow, a notion which we also use extensively.

One important characteristic of the algorithm, not possessed by any other routing algorithm to our knowledge, is its property of being loop free at every iteration. Aside from reducing delay, it appears that loop freedom can be important in simplifying higher level protocols. In fact, the major reason for building loop freedom into the algorithm was to prevent a potential deadlock in the protocol for communicating update information between the nodes.

FORMULATION OF THE MODEL

Let the nodes of an n-node network be represented by the integers 1, 2, $\cdots$, n and let a link from node i to node k be represented by (i,k). Let L be the set of links, $L = \{(i,k)$: a link goes from i to $k\}$. In order to discuss traffic flow, we distinguish link (i,k) from (k,i), but assume that if one exists the other does also.

Let $r_i(j) \geq 0$ be the expected traffic, in bits/s, entering the network at node i and destined for node j (see Fig. 1). We assume that this input traffic forms an ergodic process such as, for example, a Poisson process of message arrivals with a geometric distribution on message lengths. Let $t_i(j)$ be the total expected traffic (or node flow) at node i destined for node j. Thus $t_i(j)$ includes both $r_i(j)$ and the traffic from other nodes that is routed through i for destination j. Finally let $\phi_{ik}(j)$ be the fraction of the node flow $t_i(j)$ that is routed over link (i,k). We take $\phi_{i,k}(j) = 0$ for $(i,k) \notin L$ (i.e., no traffic is routed over nonexistent links). We also take $\phi_{ik}(j) = 0$ for $i = j$ (i.e., traffic which has reached its destination is not sent back into the network). Since the node flow $t_i(j)$ at node i is the sum of the input traffic and the traffic routed to i from other nodes,

$$t_i(j) = r_i(j) + \sum_l t_l(j)\phi_{li}(j), \qquad \text{all } i,j. \tag{1}$$

Equation (1) implicitly expresses the conservation of flow

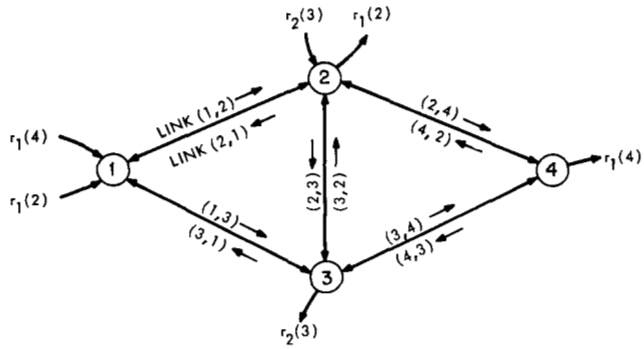


Fig. 1. Nodes, links, and inputs in a data network.

at each node; the expected traffic into a node for a given destination is equal to the expected traffic out of the node for that destination. Note that (1) deals with expected traffic and thus does not preclude the existence of traffic queues at the nodes.

Now let f_{ik} be the expected traffic, in bits/s, on link (i,k) (with $f_{ik} = 0$ if $(i,k) \notin L$). Since $t_i(j)\phi_{ik}(j)$ is the traffic destined for j on (i,k), we have

$$f_{ik} = \sum_j t_i(j)\phi_{ik}(j). \tag{2}$$

In what follows we refer to the set of expected inputs $\{r_i(j)\}$ as the *input* set r; the set of expected total node flows $\{t_i(j)\}$ as the *node flow* set t, the set of fractions $\{\phi_{ik}(j)\}$ as the *routing variable* set ϕ, and the set of expected link traffics $\{f_{ik}\}$ as the *link flow* set f. We have seen for an arbitrary strategy of routing (subject to the existence of the expectations $\{t_i(j)\}$ and the conservation of flow) that r, t, ϕ, and f all have meaning and satisfy (1) and (2). We are interested in distributed routing algorithms in which each node i chooses its own routing variables $\phi_{ik}(j)$ for each k, j. The question then arises whether the inputs r and the routing variables set ϕ uniquely specify t and f. Before answering this question, we define ϕ precisely, adding one additional constraint.

Definition: A routing variable set ϕ for an n-node network with links L is a set of nonnegative numbers $\phi_{ik}(j)$, $1 \leqslant i, k, j \leqslant n$, satisfying the following conditions.

1) $\phi_{ik}(j) = 0$ if $(i,k) \notin L$ or if $i = j$.

2) $\Sigma_k \phi_{ik}(j) = 1$.

3) For each i, j $(i \neq j)$ there is a routing path from i to j, which means there is a sequence of nodes $i, k, l, \cdots, m, j$ such that $\phi_{ik}(j) > 0, \phi_{kl}(j) > 0, \cdots, \phi_{mj}(j) > 0$.

Theorem 1: Let a network have input set r and routing variable set ϕ (according to the above definition). Then the set of equations (1) has a unique solution for t. Each component $t_i(j)$ is nonnegative and continuously differentiable as a function of r and ϕ.

This theorem is proved in Appendix A. It turns out that the constraint on the existence of routing paths in the definition of routing variables is necessary for this theorem. If this constraint were eliminated, one could still show, by the method in Appendix A, that $t_i(j)$ has a unique solution for each i, j for which a routing path from i to j exists. If no routing paths exist from some set of i to j, then there are two possible cases: 1) if no traffic for j comes into any of these nodes, either from inputs or from other nodes outside the set, then there are multiple solutions to (1); 2) otherwise there is no solution to (1). Physically, the first case above corresponds to a set of nodes which have no traffic for a given destination coming in or going out, but which might have some messages circulating around within the set. The second case corresponds to traffic coming into the set for a given destination, but none going out, leading to an infinite build-up of queues or lost traffic.

The more customary way to treat routing in a network is to regard it as a multicommodity flow problem (see, for example, Frank and Chou [9]). The traffic flow to each destination can be regarded as a commodity, and then (1) is equivalent to the multicommodity flow constraints. Our restrictions on the routing variables ϕ are somewhat more restrictive than the usual multicommodity flow constraints. In particular $\phi_{jk}(j) = 0$ prevents traffic at a destination j from looping back into the network, and the existence of routing paths prevents the isolated looping referred to in case 1) above.

We have seen that any routing policy, subject to the previously mentioned restrictions, leads to the sets t, ϕ, and f, and any distributed algorithm in which ϕ is selected by the individual nodes leads to a unique t, f. We now turn our attention to delay of messages in the network.

Let D_{ik} be the expected number of messages/s transmitted on link (i,k) times the expected delay/message (including queueing delays at the link input). Assume that D_{ik} is a function only of the link flow f_{ik}, i.e., that D_{ik} depends on the routing variables only through f_{ik}. We also make the assumption that messages are delayed only by the links of the network. This is reasonable if the processing time at an intermediate node is associated partly with the link on which the message arrives and partly with the link on which it departs.

It can now be seen with a little thought (or see Kleinrock [11]) that the total expected delay per message times the total expected number of message arrivals/s is given by

$$D_T = \sum_{i,k} D_{ik}(f_{ik}). \tag{3}$$

Since $f_{ik} = 0$ for $(i,k) \notin L$, we also take $D_{ik}(f_{ik}) = 0$ for $(i,k) \notin L$. Since the total message arrival rate is independent of the routing algorithm, we can minimize the expected delay/message on the network by minimizing D_T over all choices of routing variables (recall that f is a function of r and ϕ). The algorithm we describe subsequently will be an iterative algorithm for performing this minimization.

Before proceeding, however, we should point out some of the consequences of our assumption that D_{ik} is a function only of f_{ik}. Suppose that there are two paths from node i to j and that half the traffic is sent over each path, but the delay is greater on one path than the other. Then we could reduce the delay/message by sending the short messages over the small-delay path and the long messages over the long-delay path. Keeping the same traffic (in bits/s) on each path, we would have more messages on the short path than the long, and thus would reduce delay/message. The assumption that D_{ik} is a function only of f_{ik} restricts us from comparing such alternatives. Another consequence arises with dynamic routing, where one would hope to reduce the queueing delays on the links

without reducing the long-term expected link flow. This, however, would change the functions $D_{ik}(f_{ik})$. Thus our assumption effectively masks the distinctions between dynamic and quasi-static routing (and for this reason makes the problem analytically tractable).

Kleinrock [11] showed that if queueing delays are the only nonnegligible source of delay in a network, and if each link traffic can be modeled as Poisson message arrivals with independent exponentially distributed lengths, then $D_{ik}(f_{ik}) = f_{ik}/(C_{ik} - f_{ik})$ where C_{ik} is the capacity of link (i,k). This formula has also been refined to account for overhead and propagation delays (Kleinrock [12]). For our purposes, it is immaterial what function D_{ik} is, although we shall make the reasonable assumption that D_{ik} is increasing and convex $\cup$ in f_{ik}. Before describing the algorithm, we develop necessary and sufficient conditions on ϕ to minimize D_T.

NECESSARY AND SUFFICIENT CONDITIONS FOR MINIMUM DELAY

First we calculate the partial derivatives of the total delay D_T with respect to the inputs r and the routing variables ϕ. Assume a small increment ϵ in the input $r_i(j)$. For each adjacent node k, an increment $\epsilon\phi_{ik}(j)$ of this new incoming traffic will flow over (i,k), and to first order, this will cause an incremental delay on that link of

$$\epsilon\phi_{ik}(j)D_{ik}'(f_{ik}), \qquad \text{where } D_{ik}'(f_{ik}) = \frac{dD_{ik}(f_{ik})}{df_{ik}}. \tag{4}$$

If node k is not the destination node, then the increment $\epsilon\phi_{ik}(j)$ of extra traffic at node k will cause the same increment in delay from node k onward as an increment $\epsilon\phi_{ik}(j)$ of new input traffic at node k. To first order this incremental delay will be $\epsilon\phi_{ik}(j)\partial D_T/\partial r_k(j)$. Summing over all adjacent nodes k, then, we find[1] that, for $i \neq j$,

$$\frac{\partial D_T}{\partial r_i(j)} = \sum_k \phi_{ik}(j)\left[D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)}\right]. \tag{5}$$

We take $\partial D_T/\partial r_j(j) = 0$ in this and subsequent equations and also take terms for which $(i,k) \notin L$ to be 0. Theorem 2, which follows, gives a rigorous justification of (5).

Next consider $\partial D_T/\partial\phi_{ik}(j)$. An increment ϵ in $\phi_{ik}(j)$ causes an increment $\epsilon t_i(j)$ in the portion of $t_i(j)$ flowing on link (i,k). If $k \neq j$, this causes an addition $\epsilon t_i(j)$ to the traffic at k destined for j. Thus for $(i,k) \in L$, $i \neq j$,

$$\frac{\partial D_T}{\partial\phi_{ik}(j)} = t_i(j)\left[D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)}\right]. \tag{6}$$

Theorem 2: Let a network have inputs r and routing variables ϕ, and let each marginal link delay $D_{ik}'(f_{ik})$ be continuous in f_{ik}, $(i,k) \in L$. Then the set of equations (5), $i \neq j$, has a unique (and correct) set of solutions for $\partial D_T/\partial r_i(j)$. Furthermore, (6) is valid and both $\partial D_T/\partial r_i(j)$ and $\partial D_T/\partial\phi_{ik}(j)$ for $i \neq j$, $(i,k) \in L$ are continuous in r and ϕ.

This theorem is proved in Appendix A. The appendix also gives explicit expressions for $\partial D_T/\partial r_i(j)$ and $\partial D_T/\partial\phi_{ik}(j)$, but it turns out that the implicit forms in (5) and (6) are needed in the algorithm to be presented.

One might now hope that all that is required to minimize D_T is to find a stationary point for D_T with respect to variations in ϕ. Using Lagrange multipliers for the constraint $\Sigma_k \phi_{ik}(j) = 1$, and taking into account the constraint $\phi_{ik}(j) \geq 0$, the necessary conditions for a minimum of D_T with respect to ϕ are, for all $i \neq j$, $(i,k) \in L$,

$$\frac{\partial D_T}{\partial\phi_{ik}(j)} \begin{cases} = \lambda_{ij}, & \phi_{ik}(j) > 0 \\ \geq \lambda_{ij}, & \phi_{ik}(j) = 0. \end{cases} \tag{7}$$

This states that for a given i, j, all links (i,k) for which $\phi_{ik}(j) > 0$ must have the same marginal delay $\partial D_T/\partial\phi_{ik}(j)$, and that this marginal delay must be less than or equal to $\partial D_T/\partial\phi_{ik}(j)$ for the links on which $\phi_{ik}(j) = 0$. Unfortunately, as Fig. 2 illustrates, (7) is not a sufficient condition to minimize D_T (i.e., D_T can have inflection points as a function of ϕ).

In Fig. 2, the only input traffic goes from node 1 to 4. It is easy to verify that (7) is satisfied at each node. The trouble is that the traffic at node 2, $t_2(4)$, is zero, which automatically satisfies (7); one does not get a better routing by decreasing $\phi_{2,3}(4)$, but one does move to a point, when $\phi_{2,3}(4) < 1/2$, where the routing can be improved by increasing $\phi_{1,2}(4)$. After studying this example, it is not difficult to hypothesize that (7) would be sufficient to minimize D_T if the factor $t_i(j)$ were removed from the condition.

Theorem 3: For each $(i,k) \in L$ assume that $D_{ik}(f_{ik})$ is convex $\cup$ and continuously differentiable for $0 \leq f_{ik} < C_{ik}$ where the capacity C_{ik} satisfies $0 < C_{ik} \leq \infty$. Let ψ be the set of ϕ for which the link flows satisfy $f_{ik} < C_{ik}$ for all $(i,k) \in L$. Then (7) is necessary for ϕ to minimize D_T over ψ and (8), for all $i \neq j$, $(i,k) \in L$ is sufficient.

$$D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} \geq \frac{\partial D_T}{\partial r_i(j)}. \tag{8}$$

This theorem is proved in Appendix B. Note that the theorem does not assert the existence of a minimum; the conditions of the theorem do not even assert that ψ is nonempty. Note also that if we multiply both sides of (8) by $\phi_{ik}(j)$ and sum over k, then we see from (5) that (8) must be satisfied with equality for $\phi_{ik}(j) > 0$. Thus (8) is equivalent to

$$D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} - \min_{m:(i,m)\in L}\left[D_{im}'(f_{im}) + \frac{\partial D_T}{\partial r_m(j)}\right] \geq 0 \tag{9}$$

for all $i \neq j$, $(i,k) \in L$ with equality for $\phi_{ik}(j)$ greater than 0.

THE ALGORITHM

The general structure of an algorithm to minimize D_T (assuming stationary traffic inputs) should now be clear. Each

[1] Agnew [5], [6] develops an equation similar to (5) but omits the final term $\partial D_T/\partial r_k(j)$; his algorithm, however, effectively includes the effect of this term.

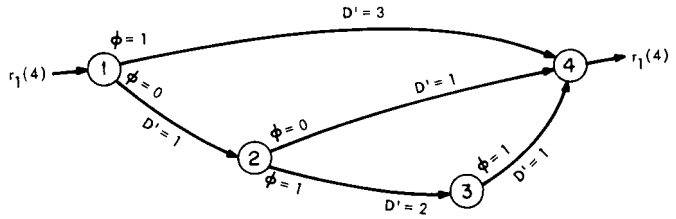


Fig. 2. Inflection point in $D_T(\phi)$.

node i must incrementally decrease those routing variables $\phi_{ik}(j)$ for which the marginal delay $D_{ik}'(f_{ik}) + \partial D_T/\partial r_k(j)$ is large, and increase those for which it is small. The algorithm breaks into two parts: a protocol between nodes to calculate the marginal delays and an algorithm for modifying the routing variables; we discuss the protocol part first.

Each node i can estimate, as a time average, the link traffic f_{ik} for each outgoing link. Thus with an appropriate formula for $D_{ik}(f_{ik})$, the node can also calculate $D_{ik}'(f_{ik})$. Since formulas for D_{ik} involve many assumptions which might be unwarranted, it might be preferable to estimate D_{ik}' directly; such estimation procedures are developed by Segall [13] and Bello [14].

In order to see how node i can calculate $\partial D_T/\partial r_j(k)$ for a neighboring node k, define node m to be *downstream* from node i (with respect to destination j) if there is a routing path from i to j passing through m (i.e., a path with positive routing variables on each link). Similarly, we define i as *upstream* from m if m is downstream from i. A routing variable set ϕ is *loop free* if for each destination j, there is no i, m ($i \neq m$) such that i is both upstream and downstream from m. Note that if such an i, m pair existed, there would be a routing path from i to j that looped from i to m and back to i on its way to j. If ϕ is loop free, then for each destination j, the downstream (and the upstream) relation form a partial ordering on the set of nodes.

The protocol used for an update, now, is as follows: for each destination node j, each node i waits until it has received the value $\partial D_T/\partial r_k(j)$ from each of its downstream neighbors $k \neq j$ (i.e., nodes k with $\phi_{ik}(j) > 0$). The node i then calculates $\partial D_T/\partial r_i(j)$ from (5) (using the convention that $\partial D_T/\partial r_j(j) = 0$) and broadcasts this to all of its neighbors (except to the destination node j which has no need of the information). It is easy to see that this procedure is free of deadlocks (i.e., a node waiting forever for updating information from a downstream neighbor) if and only if ϕ is loop free. In fact, for a given j, the nodes can broadcast their values in any order consistent with the downstream partial ordering. For this reason we will be careful to ensure that the algorithm generates only loop free ϕ.

It can be seen that in an update, each link (i,k) must transmit $\partial D_T/\partial r_i(j)$ for each $j \neq i$, $j \neq k$. The same amount of updating information is used in the ARPANET strategy, but there delays rather than marginal delays are sent, and the transmissions are unordered so that many updates are required for changes to propagate through the network. Here, of course, changes propagate completely in one update, and the only inaccuracies come from inaccuracies in the estimates of the link marginal delays. One might object to sending each value $\partial D_T/\partial r_i(j)$ separately on a link, and indeed the inefficiency would be high if each such number required an individual packet. However, the routing update information could easily be piggy-backed on other packets, requiring very little overhead. One might also object to the time required for the updating to propagate through the network, but speed is relatively unimportant in a quasi-static algorithm.

We shall later define one small but important detail that has been omitted so far in the updating protocol between nodes; a small amount of additional information is necessary for the algorithm to maintain loop freedom. It turns out to be necessary, for each destination j and each node i, to specify a set $B_i(j)$ of blocked nodes k for which $\phi_{ik}(j) = 0$ and the algorithm is not permitted to increase $\phi_{ik}(j)$ from 0. For notational convenience we include k such that $(i,k) \notin L$ in the set $B_i(j)$. We first define and discuss the algorithm and then define the sets $B_i(j)$.

The algorithm A, on each iteration, maps the current routing variable set ϕ into a new set $\phi^1 = A(\phi)$. The mapping is defined as follows. For $k \in B_i(j)$,

$$\phi_{ik}^1(j) = 0, \qquad \Delta_{ik}(j) = 0. \tag{10}$$

For $k \notin B_i(j)$, define

$$a_{ik}(j) = D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} - \min_{m \notin B_i(j)} \left[D_{im}'(f_{im}) + \frac{\partial D_T}{\partial r_m(j)} \right] \tag{11}$$

$$\Delta_{ik}(j) = \min\,[\phi_{ik}(j), \eta a_{ik}(j)/t_i(j)] \tag{12}$$

where η is a scale parameter of A to be discussed later. Let $k_{\min}(i,j)$ be a value of m that achieves the minimization in (11). Then

$$\phi_{ik}^1(j) = \begin{cases} \phi_{ik}(j) - \Delta_{ik}(j), & k \neq k_{\min}(i,j) \\ \phi_{ik}(j) + \sum_{k \neq k_{\min}(i,j)} \Delta_{ik}(j), & k = k_{\min}(i,j). \end{cases} \tag{13}$$

The algorithm reduces the fraction of traffic sent on nonoptimal links and increases the fraction on the best link. The amount of reduction, given by $\Delta_{ik}(j)$, is proportional to $a_{ik}(j)$, with the restriction that $\phi_{ik}^1(j)$ cannot be negative. In turn $a_{ik}(j)$ is the difference between the marginal delay to node j using link (i,k) and using the best link. Note that as the sufficiency condition (9) is approached, the changes get small, as desired. The amount of reduction is also inversely proportional to $t_i(j)$. The reason for this is that the change in link traffic is related to $\Delta_{ik}(j)t_i(j)$. Thus when $t_i(j)$ is small, $\Delta_{ik}(j)$ can be changed by a large amount without greatly affecting the marginal link delays. Finally the changes depend on the scale factor η. For η very small, convergence of the algorithm is guaranteed, as shown in Theorem 5, but rather slow. As η

increases, the speed of convergence increases but the danger of no convergence also increases.

It is not difficult to develop heuristic improvements on this algorithm to speed up its convergence; we have settled on this particular version since it allows us to prove convergence.

We now must complete the definition of algorithm A by defining the sets $B_i(j)$. First define a routing variable $\phi_{ik}(j)$ to be *improper* if $\phi_{ik}(j) > 0$ and $\partial D_T/\partial r_i(j) \leqslant \partial D_T/\partial r_k(j)$. We have already said that $B_i(j)$ includes only k for which $\phi_{ik}(j) = 0$, and thus, from (5),

$$\min_{m \notin B_j(j)} D_{im}'(f_{im}) + \frac{\partial D_T}{\partial r_m(j)} \leqslant \frac{\partial D_T}{\partial r_i(j)}. \tag{14}$$

Assuming positive marginal link delays, $\partial D_T/\partial r_i(j) < \partial D_T/\partial r_k(j) + D_{ik}'(f_{ik})$ if $\phi_{ik}(j)$ is improper, and we see that the algorithm always reduces improper routing variables. In fact, since $\partial D_T/\partial r_i(j)$ is the marginal delay from i to j, we would expect marginal delay to decrease as we move downstream, and improper routing variables should be rather atypical.

For a given destination node j, the set of marginal delays $\partial D_T/\partial r_i(j)$ $(i \neq j)$ forms an ordering of the nodes i. Note that if there are no improper routing variables, this ordering is consistent with the downstream partial ordering. Fig. 3 illustrates these orderings. The horizontal axis represents marginal delay (for the given destination node $j = 5$) and the solid lines show the downstream partial ordering by denoting the links for which $\phi_{ik}(5) > 0$. The dotted lines are examples of links (i,k) for which loops would form if $\phi_{ik}(5)$ were increased from 0. We now see that if ϕ is loop free and $\phi^1 = A(\phi)$ contains a loop for some destination j, then the following two conditions must hold.

1) The loop contains some link (i,k) for which $\phi_{ik}(j) = 0$, $\phi_{ik}{}^1(j) > 0$, and $\partial D_T/\partial r_i(j) > \partial D_T/\partial r_k(j)$.

2) The loop contains some link (l,m) for which $\phi_{lm}(j)$ is improper and for which $\phi_{lm}{}^1(j) > 0$.

The first condition reiterates that some routing variables must be increased from 0 to form a loop and that the algorithm only increases routing variables on links to nodes with smaller marginal delay. The second makes use of the fact that if nodes i have numbers associated with them $(\partial D_T/\partial r_i(j))$, then it is impossible to move around a loop of nodes and have those numbers monotonically decrease.

Definition: The set $B_i(j)$ is the set of nodes k for which either $\phi_{ik}(j) = 0$ and k is blocked relative to j or $(i,k) \notin L$. A node k is blocked relative to j if k has a routing path to j containing some link (l,m) for which $\phi_{lm}(j)$ is improper and

$$\phi_{lm}(j) \geqslant \eta \left[D_{lm}'(f_{lm}) + \frac{\partial D_T}{\partial r_m(j)} - \frac{\partial D_T}{\partial r_l(j)} \right] \Big/ t_l(j). \tag{15}$$

Note that the definition permits k to be identical to l. The reason for (15) can be seen from (14) and (12). If (15) is not satisfied, then $\Delta_{lm}(j) = \phi_{lm}(j)$ and $\phi_{lm}{}^1(j) = 0$, so that (l,m) can not be part of a loop for destination j.

Theorem 4: If the marginal link delays D_{ik}' are positive and ϕ is loop free, then $\phi^1 = A(\phi)$ is loop free.

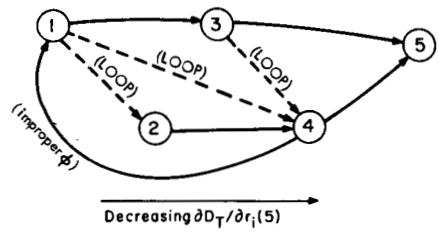


Fig. 3. Marginal delay ordering, downstream partial ordering, and possible loop formation.

Proof: Assume to the contrary that ϕ^1 has a loop, say with respect to destination j. Then from condition 2) above there is some link (l,m) on the loop for which $\phi_{lm}(j)$ is improper and $\phi_{lm}{}^1(j) > 0$. This implies that (15) is satisfied. Now move backward around the assumed loop to the first link (i,k) for which $\phi_j(i,k) = 0$; from condition 1), there must be such a link. Since (l,m) is on a routing path for ϕ from k to j, $k \in B_i(j)$. Thus, from the algorithm, $\phi_{ik}{}^1(j) = 0$, yielding the contradiction.

The protocol required for a node i to determine the set $B_i(j)$ is as follows. Each node l, when it calculates $\partial D_T/\partial r_l(j)$ determines, for each downstream m, if $\phi_{lm}(j)$ is improper and satisfies (15) (only the downstream neighbors could be improper). If any downstream neighbor satisfies these conditions, node l adds a special tag to its broadcast of $\partial D_T/\partial r_l(j)$. The node l also adds this special tag if the received value $\partial D_T/\partial r_m(j)$ from any downstream m contained a tag. In this way all nodes upstream of l also send the tag. The set $B_i(j)$ is then the set of nodes k for which either $(i,k) \notin L$ or the received $\partial D_T/\partial r_k(j)$ was tagged.

Theorem 5: Assume that for all $(i,k) \in L$, $D_{ik}(f_{ik})$ has a positive first derivative and nonnegative second derivative for $0 \leqslant f_{ik} < C_{ik}$ and that $\lim_{f_{ik} \uparrow C_{ik}} D_{ik}(f_{ik}) = \infty$. For every positive number D_0 there exists a scale factor η for A such that if ϕ^0 satisfies $D_T(\phi^0) \leqslant D_0$, then

$$\lim_{m \to \infty} D_T(\phi^m) = \min_{\phi} D_T(\phi) \tag{16}$$

where $\phi^m = A(\phi^{m-1})$ for all $m \geqslant 1$.

This is proved in Appendix C. Note that η depends on some upper bound D_0 to D_T; this is natural, since when the link flows are very close to capacity, small changes in the link flows cause large changes in marginal delay. The proof uses a ridiculously small value of η to guarantee convergence under all conditions and experimental work is necessary to determine practical values for η.

USE OF THE ALGORITHM FOR QUASI-STATIC ROUTING

We have shown in the last section that the algorithm A must eventually converge to the minimum average delay for a network with stationary inputs and links. The algorithm is really intended, however, for quasi-static applications where the input statistics are slowly changing with time and where occasionally links or nodes fail or are added to the network. Under these more general conditons, it is clear that the loop freedom of the algorithm is maintained since this is a mathe-

matical property that is independent of the marginal link delays and node flows, which are the only inputs to the algorithm (note that the inputs r plays a role in the theoretical development, but do not appear in the algorithm and need not be estimated).

The question of whether the algorithm can adapt fast enough to keep up with changing statistics is difficult and requires more study. Clearly, the faster the statistics change, the more frequently the algorithm should be updated, but frequent updating has two undesirable effects. First, frequent updates require more updating protocol, thus reducing the effective link capacities available for data, and second, frequent updates will necessitate noisier measurements of marginal link delays and node flows. Experimentation would be helpful both in determining update rates and the scale parameter η.

Another open question is that of a starting rule for the algorithm (finding a loop free ϕ to start with). One possibility is to start with shortest paths; that is, set $\phi_{ik}(j) = 1$ for the link (i,k) that leads to j from i with the smallest number of links. Such a strategy might well lead to link flows which mathematically exceed capacity, but in this case a well designed flow control would limit the input to the network, thus yielding large but finite marginal delays on such links and allowing the routing algorithm to gradually adapt.

The problem of dropped links or nodes is somewhat more complicated. Some of the problems here must be solved by higher order protocols, since if the network becomes disconnected, there is no way to route data between disconnected parts of the network. However the routing algorithm should still adapt by finding routes for any data that can be sent. Each node i at the end of a link (i,k) that has failed or whose opposite node has failed should signal the fact that an update should start throughout the network. In addition, i should no longer regard k as being downstream with respect to any destination j, and if k was the *only* downstream neighbor, then i should broadcast $\partial D_T/\partial r_i(j) = \infty$. This latter broadcast prevents upstream nodes from waiting indefinitely for update information to propagate through a failed link or node. The exact details of updating protocols in the presence of link and node failures is a subject for futher research.

APPENDIX A

Proof of Theorem 1:

Without loss of generality, take the destination node j to be the nth of the n nodes and drop the argument j from (1),

$$t_i = r_i + \sum_{l=1}^{n-1} t_l \phi_{li}, \qquad 1 \leqslant i \leqslant n. \tag{A1}$$

Summing both sides over i, we see that any solution to (A1) satisfies

$$t_n = \sum_i r_i. \tag{A2}$$

Temporarily let $\phi_{ni} = r_i/t_n$ and substitute this in (A1).

$$t_i = \sum_{l=1}^{n} t_l \phi_{li}. \tag{A3}$$

Any solution to (A3) and (A2) satisfies (A1) and vice versa. Let $\hat{\Phi}$ be the $n \times n$ matrix with components ϕ_{li}. $\hat{\Phi}$ is stochastic (i.e., $\phi_{li} \geqslant 0$ for all l, i and $\Sigma_i \phi_{li} = 1$ for all i) and (A3) is just the formula for steady-state probabilities in a Markov chain.

It is well known (see, for example, Gantmacher [15]) that if $\hat{\Phi}$ is irreducible, then (A3) has a unique solution, aside from a scale factor determined by (A2), and $t_i > 0$, $1 \leqslant i \leqslant n$. The matrix $\hat{\Phi}$ is irreducible; however, if for each i, k there is a path $i, l, m, \cdots, p, k$ such that $\phi_{il} > 0, \phi_{lm} > 0, \cdots, \phi_{pk} > 0$. If $r_i > 0$ for $1 \leqslant i \leqslant n-1$, then node n has a path to each i, $1 \leqslant i \leqslant n-1$. By the definition of routing variables, each i has a path to n and consequently $\hat{\Phi}$ is irreducible. Thus (A1) has a unique solution, with positive t_i, if $r_i > 0$ for $1 \leqslant i \leqslant n-1$.

Now let $t = (t_1, \cdots, t_{n-1})$, $r = (r_1, \cdots, r_{n-1})$, and let Φ be the $n-1 \times n-1$ matrix with components ϕ_{li} $(1 \leqslant i, l \leqslant n-1)$. Equation (A1) for $1 \leqslant i \leqslant n-1$ is then $t(I - \Phi) = r$. Since this equation has a unique solution for $r_i > 0$, $I - \Phi$ must have an inverse, and

$$t = r(I - \Phi)^{-1}. \tag{A4}$$

Since the components of t are positive when the components of r are positive, components of t are nonnegative when the components of r are nonnegative. Differentiating (A4), we get the continuous function of Φ,

$$\frac{\partial t_i}{\partial r_l} = [(I - \Phi)^{-1}]_{li}. \tag{A5}$$

Using (A5) in (A4), the solution to (A1) is conveniently expressed, for any r, as

$$t_i = \sum_l \frac{\partial t_i}{\partial r_l} r_l. \tag{A6}$$

Finally, differentiating (A1) with respect to ϕ_{km}, we get

$$\frac{\partial t_i}{\partial \phi_{km}} = \sum_{l=1}^{n-1} \frac{\partial t_l}{\partial \phi_{km}} \phi_{li} + t_k \delta_{im}$$

where $\delta_{im} = 1$ for $i = m$ and 0 otherwise. For fixed k, m, this is the same set of equations as (A1), so that the solution, continuous in ϕ, is

$$\frac{\partial t_i}{\partial \phi_{km}} = \frac{\partial t_i}{\partial r_m} t_k. \tag{A7}$$

Proof of Theorem 2

First we show that (5), repeated below with the destination node again taken to be n, has a unique solution.

$$\frac{\partial D_T}{\partial r_i} = \sum_{k=1}^{n} \phi_{ik} D_{ik}'(f_{ik}) + \sum_{k=1}^{n} \phi_{ik} \frac{\partial D_T}{\partial r_k}. \tag{A8}$$

Let $b_i = \Sigma_k \phi_{ik} D_{ik}'(f_{ik})$ and let $\boldsymbol{b}$ be the column vector $(b_1, \cdots,$

b_{n-1}). Let $\nabla \cdot D_T$ be the column vector $(\partial D_T/\partial r_1, \cdots, \partial D_T/\partial r_{n-1})$. Then (A8) can be rewritten as

$$\nabla \cdot D_T = b + \Phi(\nabla \cdot D_T). \tag{A9}$$

We saw in the proof of Theorem 1 that $I - \Phi$ has a unique inverse with components given by (A5). Thus the unique solution to (A9) is

$$\frac{\partial D_T}{\partial r_i} = \sum_l \frac{\partial t_l}{\partial r_i} \sum_m \phi_{lm} D_{lm}'(f_{lm}) \tag{A10}$$

$$= \sum_{l,m} \frac{\partial f_{lm}}{\partial r_i} D_{lm}'(f_{lm}). \tag{A11}$$

Differentiating D_T directly with (2) and (3), we get the same unique solution, which, from Theorem 1, is continuous in ϕ.

Finally we calculate $\partial D_T/\partial \phi_{ik}$ directly using (3) and (2),

$$\frac{\partial D_T}{\partial \phi_{ik}} = \sum_{l,m} D_{lm}'(f_{lm})\phi_{lm} \frac{\partial t_l}{\partial \phi_{ik}} + D_{ik}'(f_{ik})t_i$$

$$= t_i \left[\sum_{l,m} D_{lm}'(f_{lm})\phi_{lm} \frac{\partial t_l}{\partial r_k} \right] + t_i D_{ik}'(f_{ik})$$

$$= t_i \left[\frac{\partial D_T}{\partial r_k} + D_{ik}'(f_{ik}) \right]. \tag{A12}$$

We have used (A7) and (A10) to derive (A12), which is the same as (6). This is clearly continuous in ϕ given the continuity of t_i and $\partial D_T/\partial r_i$, and the proof is complete.

APPENDIX B

Proof of Theorem 3

First we show that (7) is a necessary condition to minimize D_T by assuming that ϕ does not satisfy (7). This means that there is some i, j, k, and m such that

$$\phi_{ik}(j) > 0, \qquad \frac{\partial D_T(\phi)}{\partial \phi_{ik}(j)} > \frac{\partial D_T(\phi)}{\partial \phi_{im}(j)}. \tag{B1}$$

Since these derivatives are continuous, a sufficiently small increase in $\phi_{im}(j)$ and corresponding decrease in $\phi_{ik}(j)$ will decrease D_T, thus establishing that ϕ does not minimize D_T.

Next we show that (8), repeated below, is a sufficient condition to minimize D_T.

$$D_{ik}'(f_{ik}) + \frac{\partial D_T(\phi)}{\partial r_k(j)} \geqslant \frac{\partial D_T(\phi)}{\partial r_i(j)}, \qquad \text{all } i, j, k. \tag{B2}$$

Suppose that ϕ satisfies (B2) and has node flows t and link flows f. Let ϕ^* be any other set of routing variables with node flows t^* and link flows f^*. Define

$$f_{ik}(\lambda) = (1 - \lambda)f_{ik} + \lambda f_{ik}^* \tag{B3}$$

$$D_T(\lambda) = \sum_{i,k} D_{ik}(f_{ik}(\lambda)). \tag{B4}$$

There is a set of routing variables $\phi(\lambda)$ which gives rise to $f(\lambda)$, but they are *not* linear in λ and their existence is not relevant to our proof. Since each link delay D_{ik} is a convex $\cup$ function of the link flow, $D_T(\lambda)$ is convex $\cup$ in λ, and hence

$$\left.\frac{dD_T(\lambda)}{d\lambda}\right|_{\lambda=0} \leqslant D_T(\phi^*) - D_T(\phi). \tag{B5}$$

Since ϕ^* is arbitrary, proving that $dD_T(\lambda)/d\lambda \geqslant 0$ at $\lambda = 0$ will complete the proof. From (B4) and (B3),

$$\left.\frac{dD_T(\lambda)}{d\lambda}\right|_{\lambda=0} = \sum_{i,k} D_{ik}'(f_{ik})[f_{ik}^* - f_{ik}]. \tag{B6}$$

We now show that

$$\sum_{i,k} D_{ik}'(f_{ik})f_{ik}^* \geqslant \sum_{j,k} r_k(j) \frac{\partial D_T(\phi)}{\partial r_k(j)}. \tag{B7}$$

Note from (B2) that

$$\sum_k D_{ik}'(f_{ik})\phi_{ik}^*(j) \geqslant \frac{\partial D_T(\phi)}{\partial r_i(j)} - \sum_k \frac{\partial D_T(\phi)}{\partial r_k(j)} \phi_{ik}^*(j). \tag{B8}$$

Multiplying both sides of (B8) by $t_i^*(j)$, summing over i, j, and recalling that $f_{ik}^* = \Sigma_j t_j^*(j)\phi_{ik}^*(j)$, we obtain

$$\sum_{i,k} D_{ik}'(f_{ik})f_{ik}^* \geqslant \sum_{i,j} t_i^*(j) \frac{\partial D_T(\phi)}{\partial r_i(j)} - \sum_{i,j,k} t_i^*(j)\phi_{ik}^*(j) \frac{\partial D_T(\phi)}{\partial r_k(j)}. \tag{B9}$$

From (1), $\Sigma_i t_i^*(j)\phi_{ik}^*(j) = t_k^*(j) - r_k(j)$. Substituting this into the rightmost term of (B9) and canceling, we get (B7). Note that the only inequality used here was (B8), and that if ϕ is substituted for ϕ^*, this becomes an equality from the equation for $\partial D_T/\partial r_i(j)$ in (5). Thus

$$\sum_{i,k} D_{ik}'(f_{ik})f_{ik} = \sum_{j,k} r_k(j) \frac{\partial D_T(\phi)}{\partial r_k(j)}. \tag{B10}$$

Substituting (B10) and (B7) into (B6), we see that $dD_T(\lambda)/d\lambda \geqslant 0$ at $\lambda = 0$, completing the proof. We note in passing that (B10) is valid for any set of routing variables and appears to be a rather fundamental conservation equation.

APPENDIX C

We prove Theorem 4 through a sequence of seven lemmas. The first five establish the descent properties of the algorithm,

the sixth establishes a type of continuity condition, showing that if ϕ does not minimize D_T, then for any ϕ^* in a neighborhood of ϕ, $D_T(A^m(\phi^*)) < D_T(\phi)$ for some m. The seventh lemma is a new global convergence theorem which does not require continuity in the algorithm A; Lemmas 6 and 7 together establish Theorem 4.

Let ϕ be an arbitrary set of routing variables satisfying $D_T(\phi) < D_0$ for some D_0. Let $\phi^1 = A(\phi)$ and let t, f, t^1, f^1 be the node and link flows corresponding to ϕ and ϕ^1, respectively. Let f^λ $(0 \leqslant \lambda \leqslant 1)$ be defined by $f_{ik}{}^\lambda = (1-\lambda)f_{ik} + \lambda f_{ik}{}^1$, and let

$$D_T(\lambda) = \sum_{i,k} D_{ik}(f_{ik}{}^\lambda). \tag{C1}$$

From the Taylor remainder theorm,

$$D_T(\phi^1) - D_T(\phi) = \frac{dD_T(\lambda)}{d\lambda}\bigg|_{\lambda=0} + \frac{1}{2}\frac{d^2D_T(\lambda)}{d\lambda^2}\bigg|_{\lambda=\lambda^*} \tag{C2}$$

where λ^* is some number, $0 \leqslant \lambda^* \leqslant 1$. The continuity of the second derivative above will be obvious from the proof of Lemma 4, which upper bounds that term. The first three lemmas deal with $dD_T(\lambda)/d\lambda\,|_{\lambda=0}$.

Lemma 1:

$$\frac{dD_T(\lambda)}{d\lambda}\bigg|_{\lambda=0} = \sum_{i,j,k} -\Delta_{ik}(j)a_{ik}(j)t_i{}^1(j). \tag{C3}$$

Proof[2]*:* Using the definitions of $a_{ik}(j)$ and $\Delta_{ik}(j)$ in (11) and (12)

$$\sum_k \Delta_{ik}(j)a_{ik}(j) = \sum_{k \neq k_{\min}(i,j)} [\phi_{ik}(j) - \phi_{ik}{}^1(j)]\left\{D_{ik}{}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} - \min_{m \notin B_i(j)}\left[D_{im}{}'(f_{im}) + \frac{\partial D_T}{\partial r_m(j)}\right]\right\}$$

$$= \sum_k [\phi_{ik}(j) - \phi_{ik}{}^1(j)]\left[D_{ik}{}'(f_{ik}) + \frac{\partial D_T(\phi)}{\partial r_k(j)}\right] \tag{C4}$$

$$= \frac{\partial D_T(\phi)}{\partial r_i(j)} - \sum_k \phi_{ik}{}^1(j)\left[D_{ik}{}'(f_{ik}) + \frac{\partial D_T(\phi)}{\partial r_k(j)}\right] \tag{C5}$$

In (C4), we have used (13) to extend the sum over all k and in (C5), we have used (5). Multiplying both sides of (C5) by $t_i{}^1(j)$, summing, and using (1) and (2), we get

$$\sum_{i,j,k} \Delta_{ik}(j)a_{ik}(j)t_i{}^1(j) = \sum_{i,j} t_i{}^1(j)\frac{\partial D_T(\phi)}{\partial r_i(j)} - \sum_{i,k} f_{ik}{}^1 D_{ik}{}'(f_{ik}) - \sum_{k,j}[t_k{}^1(j) - r_k(j)]\frac{\partial D_T(\phi)}{\partial r_k(j)}$$

$$= -\sum_{i,k} f_{ik}{}^1 D_{ik}{}'(f_{ik}) + \sum_{k,j} r_k(j)\frac{\partial D_T(\phi)}{\partial r_k(j)} \tag{C6}$$

$$= \sum_{i,k}(f_{ik} - f_{ik}{}^1)D_{ik}{}'(f_{ik}) \tag{C7}$$

$$= -\frac{dD_T(\lambda)}{d\lambda}\bigg|_{\lambda=0} \tag{C8}$$

We have used (B10) to get (C7), and (C8) follows from (C1), completing the proof.

Lemma 2:

$$\frac{dD_T(\lambda)}{d\lambda}\bigg|_{\lambda=0} \leqslant -\frac{1}{\eta(n-1)^3}\sum_{i,j}\Delta_i{}^2(j)t_i{}^2(j) \tag{C9}$$

where

$$\Delta_i(j) = \sum_k \Delta_{ik}(j). \tag{C10}$$

Proof: From the definition of $\Delta_{ik}(j)$ in (12), $-a_{ik}(j) \leqslant -t_i(j)\Delta_{ik}(j)/\eta$. Substituting this into (C3) yields

$$\frac{dD_T(\lambda)}{d\lambda}\bigg|_{\lambda=0} \leqslant -\frac{1}{\eta}\sum_{i,j,k}\Delta_{ik}{}^2(j)t_i(j)t_i{}^1(j) \leqslant -\frac{1}{(n-1)\eta}\sum_{i,j}\Delta_i{}^2(j)t_i(j)t_i{}^1(j) \tag{C11}$$

where (C11) follows from Cauchy's inequality, $(\Sigma_k \alpha_k\beta_k)^2 \leqslant (\Sigma\alpha_k{}^2)(\Sigma\beta_k{}^2)$, with $\alpha_k = 1$, $\beta_k = \Delta_{ik}(j)$, and the sum over $k \neq i$.

Now define $t_i{}^*(j)$ as the total flow at node i destined for j if the routing variables $\phi_{ik}(j)$ (for $k \neq k_{\min}(i,j)$) are reduced by $\Delta_{ik}(j)$ but $\phi_{ik}(j)$ for $k = k_{\min}(i,j)$ is not increased. Mathematically $t_i{}^*(j)$ satisfies

$$t_i{}^*(j) = \sum_l t_l{}^*(j)[\phi_{li}(j) - \Delta_{li}(j)] + r_i(j). \tag{C12}$$

This has a unique solution because of the loop freedom of ϕ. Subtracting (C12) from (1) results in

$$t_i(j) - t_i{}^*(j) = \sum_l [t_l(j) - t_l{}^*(j)]\phi_{li}(j) + \sum_l t_l{}^*(j)\Delta_{li}(j). \tag{C13}$$

[2] As mentioned before there is a ϕ^λ corresponding to f^λ, but it is nonlinear in λ, and $dD_T(\lambda)/\partial\lambda$ cannot be calculated in a straightforward way by differentiating with respect to ϕ^λ.

From (A6), using $\Sigma t_l^*(j)\Delta_{li}(j)$ for $r_i(j)$,

$$t_i(j) - t_i^*(j) = \sum_l \frac{\partial t_i(j)}{\partial r_l(j)} \sum_k t_k^*(j)\Delta_{kl}(j). \tag{C14}$$

Since ϕ is loop free, $\partial t_i(j)/\partial r_l(j) \leqslant 1$. Also if $\partial t_i(j)/\partial r_l(j) > 0$, then l is upstream of i for destination j and $\phi_{il}(j)$ (and hence $\Delta_{il}(j)$) is zero. Thus

$$t_i(j) - t_i^*(j) \leqslant \sum_l \sum_{k \neq i} t_k^*(j)\Delta_{kl}(j) = \sum_{k \neq i} t_k^*(j)\Delta_k(j). \tag{C15}$$

Multiplying the left side by $\Delta_i(j) \leqslant 1$ preserves the inequality, yielding

$$t_i(j)\Delta_i(j) \leqslant \sum_k t_k^*(j)\Delta_k(j). \tag{C16}$$

Since the right-hand side of (C14) is nonnegative, we also have $t_i(j)\Delta_i(j) \geqslant t_i^*(j)\Delta_i(j)$. We interrupt the proof now for a short technical lemma.

Lemma 3: Let α_i, β_i $(1 \leqslant i \leqslant m)$ be nonnegative numbers satisfying $\alpha_i \leqslant \Sigma_k \beta_k$; $\alpha_i \geqslant \beta_i$ for $1 \leqslant i \leqslant$ m. *Then*

$$\sum_{i=1}^{m} \alpha_i\beta_i \geqslant \frac{1}{m^2} \sum_i \alpha_i^2. \tag{C17}$$

Proof of Lemma 3:

$$\sum_i \alpha_i\beta_i \geqslant \sum_i \beta_i^2 \geqslant \frac{1}{m}\left(\sum \beta_i\right)^2 \tag{C18}$$

where we have used $\alpha_i \geqslant \beta_i$ and then Cauchy's inequality. Since $\Sigma\beta_i \geqslant \alpha_k$ for all k,

$$\sum_i \alpha_i\beta_i \geqslant \frac{1}{m}\alpha_k^2, \qquad \text{for all } k. \tag{C19}$$

This implies (C17), completing the proof of Lemma 3.

Now let $\alpha_i = t_i(j)\Delta_i(j)$ and $\beta_i = t_i^*(j)\Delta_i(j)$. Since these terms are nonzero only for $i \neq j$, we can take $m = n - 1$. Since the conditions of the lemma are satisfied for this choice,

$$\sum_i \Delta_i^2(j)t_i(j)t_i^*(j) \geqslant \frac{1}{(n-1)^2} \sum_i \Delta_i^2(j)t_i^2(j). \tag{C20}$$

Since $t_i^*(j) \leqslant t_i^1(j)$, we can substitute (C20) into (C11), getting (C9) and completing the proof of Lemma 2.

Lemma 4: Let M be an upper bound to $D_{ik}''(f_{ik}^{\lambda})$ over all i, k and over $0 \leqslant \lambda \leqslant 1$. Then for any λ, $0 \leqslant \lambda \leqslant 1$,

$$\frac{d^2D_T(\lambda)}{d\lambda^2} \leqslant M(n+2)(n-1)n \sum_{j,k} \Delta_k^2(j)t_k^2(j). \tag{C21}$$

Proof: The bound M must exist because $D_{ik}''(f_{ik}^{\lambda})$ is a continuous function of λ over the compact region $0 \leqslant \lambda \leqslant 1$. Taking the second derivative, we get

$$\frac{d^2D_T(\lambda)}{d\lambda^2} = \sum_{i,k} D_{ik}''(f_{ik}^{\lambda})[f_{ik}^1 - f_{ik}]^2 \leqslant \sum_{i,k} M[f_{ik}^1 - f_{ik}]^2. \tag{C22}$$

We now upper bound $|f_{ik}^1 - f_{ik}|$ by first upper bounding $|t_i^1(j) - t_i(j)|$. As in the proof of Lemma 2, we have

$$\begin{aligned} t_i^1(j) - t_i(j) &= \sum_l [t_l^1(j) - t_l(j)]\phi_{li}^1(j) \\ &\quad + \sum_l t_l(j)[\phi_{li}^1(j) - \phi_{li}(j)] \\ &= \sum_l \frac{\partial t_i^1(j)}{\partial r_l(j)} \sum_k t_k(j)[\phi_{kl}^1(j) - \phi_{kl}(j)]. \end{aligned} \tag{C23}$$

Since $0 \leqslant \partial t_i^1(j)/\partial r_l(j) \leqslant 1$, we can upper bound this by

$$t_i^1(j) - t_i(j) \leqslant \sum_k t_k(j)\Delta_k(j).$$

We can lower bound (C23) in the same way, considering only terms in which $\phi_{kl}^1(j) - \phi_{kl}(j) < 0$, and this leads to

$$|t_i^1(j) - t_i(j)| \leqslant \sum_k t_k(j)\Delta_k(j) \tag{C24}$$

$$\begin{aligned} f_{ik}^1 - f_{ik} &= \sum_j [t_i^1(j) - t_i(j)]\phi_{ik}^1(j) \\ &\quad + t_i(j)[\phi_{ik}^1(j) - \phi_{ik}(j)] \\ |f_{ik}^1 - f_{ik}| &\leqslant \sum_j \sum_l t_l(j)\Delta_l(j)\phi_{ik}^1(j) \\ &\quad + \sum_j t_i(j)\,|\,\phi_{ik}^1(j) - \phi_{ik}(j)\,|. \end{aligned} \tag{C25}$$

The double sum in (C25) has at most $(n-1)^2$ nonzero terms $(j \neq i,\ l \neq j)$ and the second sum at most $n - 1$ terms. Using Cauchy's inequality on both terms together, we get

$$\begin{aligned} |f_{ik}^1 - f_{ik}|^2 &\leqslant n(n-1)\left\{\sum_{j,l} t_l^2(j)\Delta_l^2(j)[\phi_{ik}^1(j)]^2 \right. \\ &\quad \left. + \sum_j t_i^2(j)[\phi_{ik}^1(j) - \phi_{ik}(j)]^2\right\} \\ \sum_k |f_{ik}^1 - f_{ik}|^2 &\leqslant n(n-1)\left\{\sum_{j,l} t_l^2(j)\Delta_l^2(j) \right. \\ &\quad \left. + 2\sum_i t_i^2(j)\Delta_i^2(j)\right\}. \end{aligned} \tag{C26}$$

Summing over i and substituting the result in (C22), we get (C21), completing the proof.

Lemma 5: For given D_0, define

$$M = \max_{i,k} \max_{f: D_{ik}(f) \leq D_0} D_{ik}''(f) \tag{C27}$$

$$\eta = [Mn^6]^{-1}. \tag{C28}$$

Then for all ϕ such that $D_T(\phi) \leq D_0$,

$$D_T(\phi^1) - D_T(\phi) \leq -\frac{1}{2\eta(n-1)^3} \sum_{i,j} \Delta_i^2(j) t_i^2(j). \tag{C29}$$

Proof: Temporarily let M be as defined in Lemma 4. Combining Lemmas 2 and 4,

$$D_T(\phi^1) - D_T(\phi) \leq \left[-\frac{1}{\eta(n-1)^3} + \frac{Mn(n-1)(n+2)}{2} \right] \sum_{i,j} \Delta_i^2(j) t_i^2(j). \tag{C30}$$

For $\eta = [Mn^6]^{-1}$, the second term in brackets above is less than half the magnitude of the first term, yielding (C29). It follows that $D_T(\phi^1) \leq D_T(\phi) \leq D_0$. By convexity then $D_{ik}(f^\lambda) \leq D_0$ for $0 \leq \lambda \leq 1$. Thus M as given in (C27) satisfies the condition on M in Lemma 4, compleing the proof.

Lemma 6: Let the scale factor η satisfy (C28) for a given D_0 and let ϕ be an arbitrary set of routing variables that does not minimize D_T and that satisfies $D_T(\phi) \leq D_0$. Given this ϕ, there exists an $\epsilon > 0$ and an m, $1 \leq m \leq n$, such that for all ϕ^* satisfying $|\phi - \phi^*| < \epsilon$,

$$D_T(A^m(\phi^*)) < D_T(\phi). \tag{C31}$$

Proof: We consider three cases. The first is the typical case in which no blocking occurs and $D_T(A(\phi)) < D_T(\phi)$, the second is the case in which blocking occurs, and the third is the situation typified by Fig. 2 in which $D_T(A(\phi)) = D_T(\phi)$.

Case 1: No blocking; $\Delta_i(j)t_i(j) > 0$ for some i, j. If no nodes are blocked for ϕ, then by the definition of blocking (15), there is a neighborhood of ϕ^* around ϕ for which no blocking occurs. In this neighborhood,

$$a_{ik}(j) = \left[D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} \right] - \min_{1 \leq m \leq n} \left[D_{im}'(f_{im}) + \frac{\partial D_T}{\partial r_m(j)} \right] \tag{C32}$$

which is continuous in ϕ. It follows from (12) that $\Delta_{ik}(j)$ is continuous in ϕ, and the upper bound to $D_T(A(\phi)) - D_T(\phi)$ in (C29) is continuous[3] in ϕ. Since by assumption the bound in (C29) is strictly negative, there is a neighborhood of ϕ^* around ϕ for which

$$D_T(A(\phi^*)) - D_T(\phi^*) < -\frac{1}{4\eta(n-1)^3} \sum_{i,j} \Delta_i^2(j) t_i^2(j) \tag{C33}$$

where $\Delta_i(j)$ and $t_i(j)$ correspond to the given ϕ. Choose ϵ small enough so that (C33) is satisfied for $|\phi - \phi^*| < \epsilon$ and also so that

$$|D_T(\phi^*) - D_T(\phi)| < \frac{1}{4\eta(n-1)^3} \sum_{i,j} \Delta_i^2(j) t_i^2(j).$$

Combining this with (C33), we have (C31) for $m = 1$.

Case 2: Blocking occurs. For any ϕ, we can use (5) to lower bound $a_{ik}(j)$ by

$$a_{ik}(j) \geq D_{ik}'(f_{ik}) + \partial D_T/\partial r_k(j) - \partial D_T/\partial r_i(j) \tag{C34}$$

$$\Delta_{ik}(j)t_i(j) \geq \min \left\{ \phi_{ik}(j)t_i(j),\ \eta \left[D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} - \frac{\partial D_T}{\partial r_i(j)} \right] \right\} \tag{C35}$$

The lower bounds above are continuous functions of ϕ. Since blocking occurs in ϕ, there is some i, j, k such that both

$$\frac{\partial D_T}{\partial r_k(j)} - \frac{\partial D_T}{\partial r_i(j)} \geq 0 \tag{C36}$$

and

$$\phi_{ik}(j)t_i(j) \geq \eta \left[D_{ik}'(f_{ik}) + \frac{\partial D_T}{\partial r_k(j)} - \frac{\partial D_T}{\partial r_i(j)} \right] \tag{C37}$$

Combining (C35) to (C37),

$$\Delta_{ik}(j)t_i(j) \geq \eta D_{ik}'(f_{ik}). \tag{C38}$$

Since the right-hand side of (C35) is continuous in ϕ, there is a neighborhood of ϕ^* around ϕ for which

$$\Delta_{ik}^*(j)t_i^*(j) \geq \frac{\eta}{2} D_{ik}'(t_{ik}). \tag{C39}$$

Equation (C31), for $m = 1$, now follows in the same way as in case 1.

Case 3: $\Delta_{ik}(j)t_i(j) = 0$ for all i, j, k. Let Φ_3 be the set of ϕ for which $\Delta_{ik}(j)t_i(j) = 0$ for all i, j, k. Let $\phi^{(l)} = A^l(\phi)$ for the given ϕ and let $m \geq 2$ be the smallest integer such that $\phi^{(m-1)} \notin \Phi_3$. We first show that $m \leq n$. Note first that for any $\phi \in \Phi_3$, A changes $\phi_j(i,k)$ only for i, j such that $t_i(j) = 0$ and thus the node flows and link flows cannot change. $\partial D_T/\partial r_i(j)$ can change, however, and as we shall see later, must change for some i, j if ϕ does not minimize D_T.

Now consider $\phi^{(l)}$ ($0 \leq l \leq m-2$, where $\phi^{(0)}$ denotes the original ϕ). Since $\phi^{(l)} \in \Phi_3$, $\Delta_{ik}{}^{(l)}(j) > 0$ implies that $t_i(j) = 0$. From (12), $\phi_{ik}{}^{(l)}(j) = \Delta_{ik}{}^{(l)}(j)$ and $\phi_{ik}{}^{(l+1)}(j) = 0$. For a given i, j all $\phi_{ik}{}^{(l)}(j)$ are reduced to 0 except for the k which minimizes $D_{ik}'(f_{ik}) + \partial D_T(\phi^{(l)})/\partial r_k(j)$. Thus, using (5),

[3] As a precaution against being too casual about these arguments, one should note that if the minimizing m in (C32) is not unique, then $A(\phi)$ is *not* continuous in ϕ.

$$\frac{\partial D_T(\phi^{(l+1)})}{\partial r_i(j)} = \min_k \left[D_{ik}'(f_{ik}) + \frac{\partial D_T(\phi^{(l)})}{\partial r_k(j)} \right]$$

$$\leqslant \frac{\partial D_T(\phi^{(l)})}{\partial r_i(j)} . \tag{C40}$$

Since this equation is satisfied for all l, $0 \leqslant l \leqslant m-2$, we see that $\partial D_T(\phi^{(l)})/\partial r_i(j)$ can be reduced on iteration l only if $\partial D_T(\phi^{(l-1)})/\partial r_k(j)$ is reduced on iteration $l-1$ for some k such that $\partial D_T(\phi^{(l-1)})/\partial r_k(j) < \partial D_T(\phi^{(l)})/\partial r_i(j)$. This reduction at node k however implies a reduction at some node k' of smaller differential delay at iteration $l-2$ and so forth. Since this sequence of differential delays is decreasing with decreasing l and since (from (C40)) the differential delay at a given node is nondecreasing with decreasing l, each node in the sequence must be distinct. Since there are $n-1$ nodes other than the given destination available for such a sequence, the initial l in such a sequence satisfies $l \leqslant n-2$. On the other hand, if $D_T(\phi^{(l)})/\partial r_i(j)$ is unchanged for all i, j, we see from (C40) that $\phi^{(l)}$ satisfies the sufficient conditions to minimize D_T' and then ϕ also minimizes D_T' contrary to our hypothesis; thus we must have $m \leqslant n$.[4]

Now observe that the middle expression in (C40), for $l = 0$, is a continuous function of ϕ and consequently $\partial D_T(\phi^{(1)})/\partial r_i(j)$ is a continuous function of ϕ for all i, j. It follows by induction that $\partial D_T(\phi^{(l)})/\partial r_i(j)$ is a continuous function of ϕ for all i, j and for $l \leqslant m-1$. Finally $\phi^{(m-1)} \notin \Phi_3$, so it must satsify the conditions of case 1 or 2; it will be observed that the analyses there apply equally to $\phi^{(m-1)}$ because of the continuity of $\partial D_T(\phi^{(m-1)})/\partial r_i(j)$ as a function ϕ. This completes the proof.

Our last lemma will be stated in greater generality than required since it is a global convergence theorem for algorithms that avoids the usual continuity constraint on the algorithm (see Luenberger [16] for a good discussion of global convergence).

Lemma 7: Let Φ be a compact region of Euclidean N space. Let A be a mapping from Φ into Φ and let D_T be a continuous real valued function in Φ. Assume that $D_T(A(\phi)) \leqslant D_T(\phi)$ for all $\phi \in \Phi$. Let $D_{\min}$ be the minimum of D_T over Φ and let $\Phi_{\min}$ be the set of $\phi \in \Phi$ such that $D_T(\phi) = D_{\min}$. Assume that for every $\phi \in \Phi - \Phi_{\min}$, there is an $\epsilon > 0$ and an integer $m \geqslant 1$ such that for all $\phi^* \in \Phi$ satisfying $|\phi - \phi^*| < \epsilon$, we have $D_T(A^m(\phi^*)) < D_T(\phi)$. Then for all $\phi \in \Phi$,

$$\lim_{m\to\infty} D_T(A^m(\phi)) = D_{\min}. \tag{C41}$$

Proof: Since Φ is compact, the sequence $\{A^m(\phi)\}$ has a convergent subsequence, say $\{\phi^l\}$, with

$$\phi' = \lim_{l\to\infty} \phi^l, \qquad \phi' \in \Phi. \tag{C42}$$

Since D_T is continuous,

$$D_T(\phi') = \lim_{l\to\infty} D_T(\phi^l). \tag{C43}$$

Furthermore, by assumption, $D_T(A^m(\phi))$ is nonincreasing in m, so that

$$D_T(\phi') = \lim_{m\to\infty} D_T(A^m(\phi)) \tag{C44}$$

$$D_T(\phi') = D_T(A^m(\phi)), \qquad \text{all } m \geqslant 1. \tag{C45}$$

To complete the proof, we must show that $\phi' \in \Phi_{\min}$; we assume the contrary and demonstrate a contradiction. By assumption then, there is an $\epsilon > 0$ and an m' associated with ϕ' such that $D_T(A^{m'}(\phi^*)) < D_T(\phi')$ for all $\phi^* \in \Phi$, $|\phi^* - \phi'| < \epsilon$. By (C42) there is an l such that $|\phi^l - \phi'| < \epsilon$, and thus $D_T(A^{m'}(\phi^l)) < D_T(\phi')$. Since $\phi^l = A^m(\phi)$ for some m, $D_T(A^{m'+m}(\phi)) < D_T(\phi')$, contradicting (C45) and completing the proof.

Proof of Theorem 4: Let Φ be the set of loop free routing variables ϕ such that $D_T(\phi) \leqslant D_0$. We have verified that A maps loop free routing variables into loop free routing variables, and from Lemma 5, $D_T(A(\phi)) \leqslant D_T(\phi)$ for $\phi \in \Phi$. Thus A is a mapping from Φ into Φ. It is obvious that Φ is bounded and easy to verify that any limit of loop free variables with $D_T(\phi) \leqslant D_0$ is also loop free with $D_T(\phi) \leqslant D_0$. Thus ϕ is compact. The final assumption of Lemma 7 is established by Lemma 6. Thus Lemma 7 asserts the conclusion of Theorem 4.

ACKNOWLEDGMENT

The author would like to thank L. Kleinrock, A. Segall, J. Wozencraft, and two anonymous reviewers for a number of helpful comments on an earlier version of this paper.

REFERENCES

[1] D. G. Cantor and M. Gerla, "Optimal routing in a packet switched computer network," *IEEE Trans. Comput.*, vol. C-23, pp. 1062-1069, Oct. 1974.
[2] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The interface message processor for the ARPA Computer Network," in *Conf. Rec. 1970 Spring Joint Comput. Conf., AFIPS Conf. Proc.*, 1970, pp. 551-566.
[3] A. C. Pigou, *The Economics of Welfare.* London, England: MacMillan, 1920.
[4] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *J. Res. Nat. Bureau of Standards–B Math. Sci.*, vol. 73B, no. 2, pp. 91-118, 1969.
[5] C. Agnew, "On the optimality of adaptive routing algorithms," in *Conf. Rec. Nat. Telecommun. Conf.*, 1974, pp. 1021-1025.
[6] —, "On quadratic adaptive routing algorithms," *Commun. Ass. Comput. Mach.*, vol. 19, no. 1, pp. 18-22, 1976.
[7] R. E. Kahn and W. R. Crowther, "A study of the ARPA Network design and performance," BBN rep. 2161, Aug. 1971.
[8] T. E. Stern, "A class of decentralized routing algorithms using relaxation," to be published.
[9] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: An approach to store-and-forward communication network design," *Networks*, vol. 3, pp. 97-133, 1973.
[10] H. Frank and W. Chou, "Routing in computer networks," *Networks*, vol. 1, pp. 99-122, 1971.
[11] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay.* New York: McGraw-Hill, 1964.
[12] —, "Analytic and simulation methods in computer network design," in *Conf. Rec., Spring Joint Comput. Conf., AFIPS Conf. Proc.*, 1970, pp. 569-579.

[4] It can be seen from this that the algorithm converges in at most n steps to a ϕ satisfying the sufficient conditions (8) if D_{ik} is linear in f_{ik} for each i, k (in this case, from (C28), $\eta = \infty$).

[13] A. Segall, "The modeling of adaptive routing in data communication networks," this issue, pp. 85–95.
[14] M. Bello, S. M. thesis, Dep. Elec. Eng. and Comput. Sci., Massachusetts Inst. Technol., Cambridge, Sept. 1976.
[15] F. R. Gantmacher, *Matrix Theory*, vol. 2. New York: Chelsea, 1959.
[16] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA: Addison Wesley, 1973.

★

Robert G. Gallager (S'58–M'61–F'68) was born in Philadelphia, PA on May 29, 1931. He received the S.B. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 1953 and the S.M. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1957 and 1960, respectively.

From 1953 to 1954 he was a member of the technical staff at Bell Laboratories and from 1954 to 1956 was in the signal corps of the U.S. Army. He has been at the Massachusetts Institute of Technology since 1956 and was Associate Chairman of the Faculty from 1973 to 1975. He is currently a Professor of Electrical Engineering and Computer Science and is the Associate Director of the Electronic Systems Laboratory. He is also a consultant to Codex Corporation, Newton, MA. He is the author of the text book *Information Theory and Reliable Communication* (New York: Wiley, 1968), and was awarded the IEEE Baker Prize Paper Award in 1966 for the paper "A Simple Derivation of the Coding Theorem and Some Applications."

Mr. Gallager was a member of the Administrative Committee of the IEEE group on Information Theory, from 1965 to 1970 and was Chairman of the group in 1971. His major research interests are data networks, information theory, and computer architecture.

The Modeling of Adaptive Routing in Data-Communication Networks

ADRIAN SEGALL, MEMBER, IEEE

Abstract—**Basic analytical models for problems of dynamic and quasi-static routing in data-communication networks are introduced. The models are intended to handle the quantities of interest in an algorithmic form, and as such require only a minimal number of assumptions. Control and estimation methods are used to construct algorithms for the solution of the routing problem.**

I. INTRODUCTION

THE problem of obtaining efficient routing procedures for fast delivery of messages to their destinations is of utmost importance in the design of modern data-communication networks. Although a large variety of routing algorithms have been developed and implemented in many existing networks, a lack of basic models and theories able to analyze a large variety of routing procedures has made it necessary to base these algorithms almost solely on intuition, heuristics, and simulation. It is the purpose of this paper to present several analytical models for various types of routing strategies and to indicate methods to analyze their performance.

For the purpose of this paper, we classify the routing procedures according to how dynamic they are, with the ends of the scale consisting of purely static and completely dynamic strategies. In a *purely static* situation, given fractions of the traffic at a node i of the network destined for each of the other nodes $j \neq i$ are directed on each of the links outgoing from node i. These fractions are decided upon before the network starts operating, are *fixed* in time, and depend only on the time and ensemble averages of the message flow requirements in the network. At the other end of the scale is the *completely dynamic* strategy which allows continuous changing of the routes. In particular, the routes can be varied not only as functions of time, but also according to congestion and traffic requirement changes in various portions of the network.

One can immediately see some of the advantages and drawbacks of each of the extreme strategies, but probably the most prominent ones are the following. The static routing procedure is relatively simple to implement, but on the other hand, if links or nodes in the network fail or build congestion, the messages intended to be transmitted over them will be completely blocked. The completely dynamic procedure is supposed to cope with the congestion and failure problems, but on the other hand, it requires large amounts of overhead per message for purposes of addressing, reordering at destinations, etc.

Given the advantages and disadvantages of the two extreme routing procedures, one should try in many practical situations to devise strategies that can possibly have some of the desired properties of both. One possibility is to use a *quasi-static* routing procedure, in which changes of routes will be allowed only at given intervals of time and/or whenever extreme situations occur. The time intervals between routing changes will be relatively long, so that most of the time messages will

Manuscript received March 10, 1976; revised June 30, 1976. This paper was presented at the National Telecommunications Conference, New Orleans, LA, December 1975. This work was supported by the Advanced Research Projects Agency of the Department of Defense (monitored by ONR) under Contract N00014-75-C-1183 and by the National Science Foundation under Grant ENG75-14103. Part of this work was performed at Codex Corporation, Newton, MA 02195.

The author was with the Electronic Systems Laboratory, Research Laboratory of Electronics and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02195. He is now with the Department of Electrical Engineering, Technion IIT, Haifa, Israel.