August 1984

LIDS-P-1399

DISTRIBUTED ASYNCHRONOUS OPTIMAL ROUTING IN DATA NETWORKS*

John N. Tsitsiklis
Dimitri P. Bertsekas

Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We prove convergence of a distributed gradient projection method for optimal routing in a data communication network. The analysis is carried out without any synchronization assumptions and takes into account the possibility of transients caused by updates in the routing strategy being used.

1. INTRODUCTION

The most popular formulation of the optimal distributed routing problem in a data network is based on a multicommodity flow optimization whereby a separable objective function of the form

$$\sum_{(i,j)} \bar{D}^{ij}(F^{ij})$$

is minimized with respect to the flow variables F^{ij}, subject to multicommodity flow constraints ([1], [2], [3], [12]). Here (i,j) denotes a generic directed network link, and $\bar{D}^{ij}$ is a strictly convex differentiable, increasing function of F^{ij} which represents in turn the total traffic arrival rate on link (i,j) measured for example in packets or bits per second.

We want to find a routing that minimizes this objective. By a routing we mean a set of active paths for each origin-destination (OD) pair (set of paths carrying some traffic of that OD pair), together with the fraction of total traffic of the OD pair routed along each active path.

A typical example of a distributed routing algorithm operates roughly as follows:

The total link arrival rates F^{ij} are measured by time averaging over a period of time, and are communicated to all network nodes. Upon reception of these measured rates each node updates the part of the routing dealing with traffic originating at that node. The updating method is based on some rule, e.g. a shortest path method [2], [4], or an iterative optimization algorithm [1], [5], [6].

There are a number of variations of this idea - for example some relevant function of F^{ij} may be measured in place of F^{ij}, or a somewhat different type of routing policy may be used, but these will not concern us for the time being. The preceding algorithm is used in this paper as an example which is interesting in its own right but also involves ideas that are common to other types of routing algorithms.

Most of the existing analysis of distribution routing algorithms such as the one above is predicated on several assumptions that are to some extent violated in practice. These are:

a) The quasistatic assumption, i.e. the external traffic arrival rate for each OD pair is constant over time. This assumption is approximately valid when there is a large number of user-pair conversations associated with each OD pair, and each of these conversations has an arrival rate that is small relative to the total arrival rate for the OD pair (i.e. a "many small users" assumption). An asymptotic analysis of the effect of violation of this assumption on the stationary character of the external traffic arrival rates is given in [7].

b) The fast setting time assumption, i.e. transients in the flows F^{ij} due to changes in routing are negligible. In other words once the routing is updated, the flows F^{ij} settle to their new values within time which is very small relative to the time between routing updates. This assumption is typically valid in datagram networks but less so in virtual circuit networks where, existing virtual circuits may not be rerouted after a routing update. When this assumption is violated, link flow measurements F^{ij} reflect a dependence not just on the current routing but also on possibly several past routings. A seemingly good model is to represent each F^{ij} as a convex combination of the rates of arrival at (i,j) corresponding to two or more past routing updates.

c) The synchronous update assumption, i.e. all link rates F^{ij} are measured simultaneously, and are received simultaneously at all network nodes who in turn simultaneously carry out a routing update. However, there may be technical reasons (such as software complexity) that argue against enforcing a synchronous update protocol. For example the distributed routing algorithm of the ARPANET [4] is not operated synchronously.

In this paper we show that projection methods, one of the most interesting class of algorithms for distributed optimal routing, are valid even if the settling time and synchronous update assumption are violated to a considerable extent. Even though we retain the quasistatic assumption in our analysis we conjecture that the result of this paper can be generalized along the lines of another related study [7] whereby it is shown that a routing algorithm based on a shortest path rule converges to a neighborhood of the optimum. The size of this neighborhood depends on the extent of violation of the quasistatic assumption. A similar deviation from optimality can be caused by errors in the measurement of F^{ij}. In our analysis these errors are neglected.

In the next section we provide some background on distributed asynchronous algorithms and discuss the relation of the result of the present paper with earlier analyses. In section 3 we formulate our class of

* Work supported by DARPA under Contract No. ONR-N00014-75-C-1183 and an IBM Faculty Development Award.

distributed asynchronous routing algorithms, while Section 4 provides convergence analysis.

II. ASYNCHRONOUS OPTIMIZATION ALGORITHMS

We provide here a brief discussion of the currently available theory and tools of analysis of asynchronous distributed algorithms. In a typical such algorithm (aimed at solving an optimization problem) each processor i has in its memory a vector x^i which may be interpreted as an estimate of an optimal solution. Each processor obtains measurements, performs computations and updates some of the components of its vector. Concerning the other components, it relies entirely on messages received from other processors. We are mainly interested in the case where minimal assumptions are placed on the orderliness of message exchanges.

There are two distinct approaches for analyzing algorithmic convergence. The first approach is essentially a generalization of the Lyapunov function method for proving convergence of centralized iterative processes. The idea here is that, no matter what the precise sequence of message exchanges is, each update by any processor brings its vector x^i closer to the optimum in some sense. This approach applies primarily to problems involving monotone or contraction mappings with respect to a "sup"-norm (e.g. a distributed shortest path algorithm) [8,9]; it is only required that each processor communicates to every other processor an infinite number of times.

The second approach is based on the idea that if the processors communicate fast enough relative to the speed of convergence of the computation, then the evolution of their solution estimates x^i may be (up to first order in the step-size used) the same as if all processors were communicating to each other at each time instance [10,11]. The latter case is, however, mathematically equivalent to a centralized (synchronous) algorithm for which there is an abundance of techniques and results. Notice that in this approach, slightly stronger assumptions are placed on the nature of the communication process than in the first one. This is compensated by the fact that the corresponding method of analysis applies to broader classes of algorithms.

Unfortunately, the results available cannot be directly applied to the routing problem studied in this paper and a new proof is required. A main reason is that earlier results concern algorithms for unconstrained optimization. In the routing problem, the non-negativity and the conservation of flow introduce inequality and equality constraints. While equality constraints could be taken care by eliminating some of the variables, inequality constraints must be explicitly taken into account. Another difference arises because, in the routing algorithm, optimization is carried out with respect to path flow variables, whereas the messages being broadcast contain estimates of the link flows (see next section). In earlier results the variables being communicated were assumed to be the same as the variables being optimized.

III. THE ROUTING MODEL

We present here our basic assumptions, our notation and a simple model by which the nodes in a communication network may adjust the routing of the flow through that network.

We are given a network described by a directed graph $G = (V,E)$. (V is the set of nodes, E the set of directed links. For each pair $w = (i,j)$ of distinct nodes i and j (also called an origin-destination, or OD, pair) we introduce P_w, a set of directed paths from i to j, containing no loops. (These are the candidate paths for carrying the flow from i to j.) For each OD pair $w = (i,j)$, let r_w be the total arrival rate (at node i) of traffic that has to be sent to node j (measured, for example, in packets or bits per second). For each path $p \in P_w$, we denote by $x_{w,p}$ the amount of flow which is routed through path p. Naturally, we have the constraints

$$x_{w,p} \geq 0, \qquad \forall p \in P_w,\ \forall w, \tag{3.1}$$

$$\sum_{p \in P_w} x_{w,p} = r_w, \qquad \forall w. \tag{3.2}$$

Let us define a vector x_w with components $x_{w,p}$, $p \in P_w$. Constraints (3.1), (3.2) may be written compactly as $x_w \in G_w$, where G_w is a simplex (in particular, G_w is compact and convex).

Suppose that there is a total of M OD pairs and let us index them so that the variable w takes values in $\{1,\ldots,M\}$. Then, the totality of flows through the network may be described by a vector $x = (x_1,\ldots,x_M)$. Naturally, x is subject to the constraint $x \in G_1 x \ldots x G_M = G$.

For any link (i,j) in the network, let F^{ij} denote the corresponding traffic arrival rate at that link. Clearly,

$$F^{ij} = \sum_{w=1}^{M} \sum_{\substack{p \in P_w \\ (i,j) \in p}} x_{w,p}. \tag{3.3}$$

A cost function, corresponding to some measure of congestion through the network, is introduced. We assume the separable form

$$\bar{D} = \sum_{(i,j) \in E} \bar{D}^{ij}(F^{ij}). \tag{3.4}$$

We assume that for each link $(i,j) \in E$, $\bar{D}^{ij}$ is convex, continuously differentiable and has a Lipschitz continuous derivative.

We are interested in the case where the nodes in the network adjust the path routing variables $x_{w,p}$ so as to minimize (3.4). Since a set of path flow variables $\{x_{w,p}: p \in P_w,\ w \in \{1,\ldots,M\}\}$ determines uniquely the link flow variables F^{ij} (through (3.3)), it is more convenient to express the cost function in terms of the path flow variables. We are thus led to the cost function

$$D(x) = \sum_{(i,j) \in E} D^{ij}(x), \tag{3.5}$$

where

$$D^{ij}(x) = \bar{D}^{ij}(\langle e^{ij}, x \rangle) \tag{3.6}$$

and e^{ij} is a vector with entries in $\{0,1\}$, determined by (3.3). Clearly, D^{ij} inherits the convexity and smoothness properties of $\bar{D}^{ij}$.

Let us now consider the situation where the flows change slowly with time, due to re-routing decisions made by the nodes in the network. Accordingly, the flows at time n are described by a vector $x(n) = (x_1(n),\ldots,x_M(n)) \in G$. Let us assume that the routing decisions for the flow corresponding to a particular OD pair $w = (i,j)$ are made by the origin node i. In an ideal situation, node i would have access to the exact value of x(n) and perform the update

$$x_w(n+1) = [x_w(n) - \gamma\mu_w \frac{\partial D}{\partial x_w}(x(n))]^+ . \qquad (3.7)$$

(Here γ is a positive scalar step-size, μ_w a positive scaling constant and $[\cdot]^+$ denotes the projection on G_w with respect to the Euclidean norm.) In a practical situation, however, (3.7) is bound to be unrealistic for several reasons:

(i) It assumes perfect synchronization of all origin nodes.

(ii) It assumes that $x(n)$ (or, equivalently, the link flows $F^{ij}(n)$ at time n) may be measured exactly at time n.

(iii) Even if the origin node i is able to compute $x_w(n+1)$ exactly through (3.7), the actual flows through the network, at time n+1, will be different from the computed ones, unless the settling time is negligible. The above necessitate the development of a more realistic model, which is done below.

First, because of remark (iii) we will differentiate between the actual flows through the network (denoted by $x(n)$, $x_w(n)$, etc.) and the desired flows, as determined by the computations of some node; the latter will be denoted by $\bar{x}(n)$ and $\bar{x}_w(n)$. The routing decisions of some node at time n are determined by the desired flows $\bar{x}_w(n)$. However, due to transients, each component $x_{w,p}(n)$ of the actual flow $x(n)$ will have some value between $\bar{x}_{w,p}(n)$ and $x_{w,p}(n-1)$. Similarly, $x_{w,p}(n-1)$ will be a convex combination of $\bar{x}_{w,p}(n-1)$ and $x_{w,p}(n-2)$. Repeating this procedure, we conclude that $x_{w,p}(n)$ is in the convex hull of $x_{w,p}(0), \bar{x}_{w,p}(1),\ldots,\bar{x}_{w,p}(n)$. For n large enough, $x_{w,p}(0)$ should have negligible influence on $x_{w,p}(n)$ and will be ignored for convenience. We may thus conclude that there exist nonnegative coefficients $\alpha_{w,p}(n;k)$ such that

$$\sum_{k=1}^{n} \alpha_{w,p}(n;k) = 1, \quad \forall n,w,p\in P_w, \qquad (3.8)$$

$$x_{w,p}(n) = \sum_{k=1}^{n} \alpha_{w,p}(n;k)\bar{x}_{w,p}(k), \quad \forall n,w,p\in P_w. \qquad (3.9)$$

It seems realistic to assume that if $\bar{x}_{w,p}(k)$ is held constant, say equal to $\bar{x}$, the actual flows $x(n)$ should settle to $\bar{x}$ at a geometric rate. Accordingly:

Assumption: There exist constant $B\geq 0$, $\beta\in[0,1)$ such that

$$\alpha_{w,p}(n;k) \leq B\,\beta^{n-k}, \quad \forall n,k,w,p\in P_w. \qquad (3.10)$$

Concerning the computation of the desired flows we postulate an update rule of the form (cf. (3.7)).

$$\bar{x}_w(n+1) = [\bar{x}_w(n)-\gamma\mu_w\lambda_w(n)]^+. \qquad (3.11)$$

Here $\lambda_w(n)$ is some estimate of $\frac{\partial D}{\partial x_w}(x(n))$ which is, in general, inexact due to asynchronism and delays in obtaining measurements. However, it would be unnatural to assume that the computation (3.11) is carried out at each time instance for each OD pair. We therefore define a set T_w of times for which (3.11) is used. For all $n\notin T_w$, we simply let $\bar{x}_w(n+1) = \bar{x}_w(n)$. We only assume that the time between consecutive updates (equivalently, the difference of consecutive elements of T_w) is bounded, for each w.

We now describe the process by which $\lambda_w(n)$ is formed.

For each link (i,j), node i estimates from time to time the amount of traffic through that link. Practically, these estimates do not correspond to instantaneous measurements but to an average of a set of measurements obtained over some period of time. Accordingly, at each time n, node i has available an estimate

$$\bar{F}^{ij}(n) = \sum_{m=n-Q}^{n} c^{ij}(n;m)F^{ij}(m) \qquad (3.12)$$

Here, $c^{ij}(n;m)$ are nonnegative scalars summing to one (for fixed n), and Q is a bound on the time over which measurements are averaged plus the time between the computation of consecutive estimates of the flow. These estimates are broadcast from time to time (asynchronously and possibly with some variable delay). Let us assume that the time between consecutive broadcasts plus the communication delay until the broadcasted messages reach all nodes is bounded by some T. It follows that at time n each node k knows the value of $\bar{F}^{ij}(m_k)$, for some m_k with $n-T\leq m_k\leq n$. Combining this observation with (3.12) we conclude that at time n, each node k knows an estimate $\hat{F}_k^{ij}(n)$ satisfying

$$\hat{F}_k^{ij}(n) = \sum_{m=n-C}^{n} d_k^{ij}(n;m)F^{ij}(m) \qquad (3.13)$$

where C=T+Q and $d_k^{ij}(n;m)$ are (unknown) nonnegative coefficients summing to one, for fixed n.

For each OD pair w, the corresponding origin node (let us denote it by k) uses the values of $\hat{F}_k^{ij}(n)$ to form an estimate $\lambda_w(n)$ of $\frac{\partial D}{\partial x_w}(x(n))$ as follows. Note that

$$\frac{\partial D^{ij}}{\partial x_{w,p}}(x(n)) = \begin{cases} \frac{\partial \bar{D}^{ij}}{\partial F^{ij}}(F^{ij}(n)), & \text{if } (i,j)\in p \\ 0, & \text{otherwise.} \end{cases} \qquad (3.14)$$

Accordingly, a natural estimate is given (componentwise) by:

$$\lambda_{w,p}(n) = \sum_{(i,j)\in p} \frac{\partial \bar{D}^{ij}}{\partial F^{ij}}(\hat{F}_k^{ij}(n)) . \qquad (3.15)$$

The development of our model is now complete. To summarize, the basic equation is (3.11), where $x(n)$ is determined by (3.9), $\lambda_w(n)$ is determined by (3.15), $\hat{F}_k^{ij}(n)$ is given by (3.13) and F^{ij} is related to x by (3.3).

Let us close this section with a remark. A distributed version of the Bellman algorithm for shortest paths has been shown to converge appropriately [8], [9] even if the time between consecutive broadcasts is unbounded. In our model however, we have to assume boundedness because otherwise there are examples that demonstrate that convergence is not guaranteed. Of course such an assumption is always observed in practice.

A simple example is the following: consider the network of Figure 1. There are three origin nodes (nodes 1,2, and 3), with input arrival rate equal to 1 at each one of them, and a single destination node (node 6). For each OD pair there are two paths. For each origin node i, let x_i denote the flow routed through the path containing node 4. Let $\bar{D}^{ij}(F^{ij}) = (F^{ij})^2$ for (i,j) = (4,6) or (5,6) and $\bar{D}^{ij}(F^{ij}) = 0$ for all other links. In terms of the variables x_1, x_2, x_3, the cost becomes

$$D(x_1,x_2,x_3) = (x_1+x_2+x_3)^2 + (3-x_1-x_2-x_3)^2. \qquad (3.16)$$

We assume that the settling time is zero, so that we do not need to distinguish between actual and desired flows, and that each node i (i=1,2,3) knows x_i exactly and is able to transmit its value instantaneously to the remaining origin nodes. Suppose that initially $x_1=x_2=x_3=1$ and that each origin node executes a large number of gradient projection iterations with a small stepsize before communicating the current value of x_i to the other nodes. Then, effectively, node i solves the problem

$$\min_{0\le x_i\le 1} \{(x_i+2)^2 + (1-x_i)^2\},$$

thereby obtaining the value $x_i=0$. At that point the processors broadcast their current values of x_i. If this sequence of events is repeated, each x_i will become again equal to 1. So, (x_1,x_2,x_3) oscillates between (0,0,0) and (1,1,1) without ever converging to an optimal routing. The same behavior is also observed if the cost function (3.16) is modified by adding a term $\varepsilon(x_1^2 + x_2^2 + x_3^2)$, which makes it strictly convex, as long as $0<\varepsilon<<1$.

IV. RESULT AND CONVERGENCE PROOF

Theorem: With the algorithm and the assumptions introduced in the last section and provided that the stepsize γ is chosen small enough, D(x(n)) converges to $\min_{x\in G} D(x)$ and any limit point of $\{x(n)\}$ is a minimizing point. Moreover, $x(n)-\bar{x}(n)$ converges to zero. Finally, if each $\bar{D}^{ij}$ is strictly convex (as a function of the link flow F^{ij}) and if, for each OD pair w=(i,j), P_w contains all paths from i to j, then the link flows $F^{ij}(n)$ converge to their (unique) optimal values.

Lemma: Let $[\cdot]^+$ denote projection on a convex set G $\mathbb{R}^n$. Assume that $0\in G$. Then,

$$\langle a,[a]^+\rangle \ge ||[a]^+||^2, \quad \forall a \in \mathbb{R}^n. \qquad (4.1)$$

Proof: If $a\in G$, $[a]^+ = a$ and (4.1) holds trivially. So, let us assume that $[a]^+ \notin G$ and form a triangle with vertices at the points a, $[a]^+$ and the origin, denoted by A,B,0, respectively (see Figure 2). Let G_0 be the intersection of G with the plane defined by that triangle. Let us draw the normal to AB through point B. This line is a supporting hyperplane for G_0. Therefore, 0 and A lie at different sides of that line; hence the angle OBA is larger than 90 degrees. Let us now draw the normal to OB through B. It must intersect the segment OA at some point C, because, $\angle OBA \ge 90°$. Hence,

$$||[a]^+||^2 = ||\vec{OB}||^2 = \langle OB,OC\rangle \le \langle OB,OA\rangle = \langle a,[a]^+\rangle.$$

By translating the origin to an arbitrary point x, (4.1) becomes:

$$\langle a, [x+a]^+-x\rangle \ge ||[x+a]^+-x||^2, \quad x\in G,\ a\in\mathbb{R}^n. \qquad (4.2)$$

Proof of the Theorem: We define s(n) to be the vector with components

$$s_w(n) = \begin{cases} [\bar{x}_w(n)-\gamma\mu_w\lambda_w(n)]^+ - \bar{x}_w(n), & n\in T_w, \\ 0 & n\notin T_w \end{cases} \qquad (4.3)$$

so that

$$\bar{x}_w(n+1) = \bar{x}_w(n) + s_w(n). \qquad (4.4)$$

Using (4.2) with $a = -\gamma\mu_w\lambda_w(n)$, we obtain

$$-\langle\gamma\mu_w\lambda_w(n),\ s_w(n)\rangle \ge ||s_w(n)||^2,$$

or

$$\langle\lambda_w(n),\ s_w(n)\rangle \le -||s_w(n)||^2/\gamma\mu_w. \qquad (4.5)$$

Using (4.4), (3.9) and the assumption (3.10), it is easy to show that for some $A_1\ge 0$ (independent of γ or n)

$$||x(n) - \bar{x}(n)|| \le A_1 \sum_{k=1}^{n-1} \beta^{n-k}||s(n)||. \qquad (4.6)$$

Furthermore, comparing (3.14) to (3.15) and using the Lipschitz continuity of $\partial D^{ij}/\partial F^{ij}$, we conclude that for some constants $A_2,\dots,A_7$ (independent of γ)

$$\begin{aligned}
||\frac{\partial D}{\partial x_w}(x(n)) - \lambda_w(n)|| &\le A_2 \max_{i,j,k} |\hat{F}_k^{ij}(n)-F^{ij}(n)| \le \\
&\le A_3 \max_{i,j} \max_{n-C\le m\le n} |F^{ij}(m)-F^{ij}(n)| \le \\
&\le A_4 \max_{n-C\le m\le n} ||x(m)-x(n)|| \le \\
&\le A_4 \max_{n-C\le m\le n} \{||x(m)-\bar{x}(m)|| + ||\bar{x}(m)-\bar{x}(n)|| + \\
&\qquad + ||\bar{x}(n)-x(n)||\} \le \\
&\le A_5 \sum_{k=1}^{n-1} \beta^{n-k}||s(k)|| + A_6 \sum_{m=n-C}^{n-1} ||s(m)|| \le \\
&\le A_7 \sum_{k=1}^{n-1} \beta^{n-k}||s(k)|| .
\end{aligned} \qquad (4.7)$$

(The second inequality follows from (3.13), the third from (3.3), the fourth is the triangle inequality, the fifth uses (4.6).) Using Lipschitz continuity once more, (4.6) and (4.7) we finally obtain, for some $A_8\ge 0$ (independent of n,γ),

$$||\frac{\partial D}{\partial x_w}(\bar{x}(n))-\lambda_w(n)|| \le A_8 \sum_{k=1}^{n-1} \beta^{n-k}||s(k)||. \qquad (4.8)$$

Using a first order series expansion for D, we have

$$\begin{aligned}
D(\bar{x}(n+1)) &\le D(\bar{x}(n)) + \sum_w \langle \frac{\partial D}{\partial x_w}(\bar{x}(n)), s_w(n)\rangle + A_9||s(n)||^2 \le \\
&\le D(\bar{x}(n)) + \sum_w \langle\lambda_w(n), s_w(n)\rangle + A_8 \sum_{k=1}^{n-1} \beta^{n-k}||s(k)||\ ||s(n)|| + \\
&\qquad + A_9\ ||s(n)||^2 \le \\
&\le D(\bar{x}(n)) - \frac{A_{10}}{\gamma}\ ||s(n)||^2 + A_{11} \sum_{k=1}^{n} \beta^{n-k}||s(k)||^2 .
\end{aligned} \qquad (4.9)$$

(Here, the second inequality was obtained from (4.8); the third from (4.5).) Summing (4.9) for different values of n and rearranging terms we obtain

$$D(\bar{x}(n+1)) \leq D(\bar{x}(1)) - \sum_{k=1}^{n} \left[\frac{A_{10}}{\gamma} - A_{11} \sum_{i=k}^{n} \beta^{n-k} \right] ||s(k)||^2. \quad (4.10)$$

Suppose that γ is small enough so that $\frac{A_{10}}{\gamma} - \frac{A_{11}}{1-\beta} \geq 0$. Note that D is continuous on a compact set, hence bounded below. Let $n\to\infty$ in (4.10) to obtain

$$\sum_{k=1}^{\infty} ||s(k)||^2 < \infty, \quad (4.11)$$

and, in particular, $\lim_{k\to\infty} ||x(k)-\bar{x}(k)|| \leq A_1 \lim_{k\to\infty} ||s(k)|| = 0$. Let

$$f_w(x) = [x_w - \gamma\mu_w \frac{\partial D}{\partial x_w}(x)]^+ - x_w. \quad (4.12)$$

Using (4.7) we conclude that $||\lambda_w(n) - \frac{\partial D}{\partial x_w}(x(n))||$ converges to zero. Hence, $||f_w(x(n)) - s_w(n)||$ converges to zero along any integer sequence contained in T_w.

Let x^* be a limit point of $\{x(n)\}$. (At least one exists because G is compact.) Since $||s(n)|| \to 0$ and since the difference of consecutive elements of T_w is bounded, x^* is also a limit point of $\{x(n): n\in T_w\}$. Pick a subsequence $\{x(n_k)\}$, $n_k \in T_w$ which converges to x^*. Since $s_w(n_k) \to_{k\to\infty} 0$, we also have $f_w(x(n_k)) \to_{k\to\infty} 0$. Clearly, f_w is continuous, which implies that $f_w(x^*)=0$. Since this is true for all w, the sufficient conditions for optimality are satisfied at x^*. Finally, using (4.10), it is easy to see that $D(\bar{x}(n))$, and hence $D(x(n))$, converge; the limit can only be the optimal value of D.

The last part of the theorem is trivial.

V. CONCLUSIONS

Gradient projection algorithms for routing in a data network converge appropriately even in the face of substantial asynchronism and even if the time required for the network to adjust to a change in the routing policies (settling time) is non-negligible. While convergence is proved under the assumption that the input arrival rates r_w are constant, it is expected that the algorithm will be able to adjust appropriately in the face of small variations. If input variations become substantial, however, and the quasistatic assumption is violated, a more detailed analysis is required, incorporating stochastic effects.

Another idealization in our model arises in the measurement equation (3.12), which assumes that measurements are noiseless. This is a reasonable assumption if the time average runs over a sufficiently long period but may be unrealistic otherwise, necessitating again again a more elaborate stochastic model.

Finally, let us mention an important related class of distributed algorithms. In the present model the nodes measure and broadcast messages with their estimates of the link flows F^{ij}. Other nodes receive the broadcasted messages and use them to compute estimates of the expression $\frac{\partial \bar{D}^{ij}}{\partial F^{ij}}(F^{ij})$ which is required in the algorithm. An alternative possibility would be to let, say node j, to measure directly or compute the value of $\frac{\partial \bar{D}^{ij}}{\partial F^{ij}}(F^{ij})$ and broadcast that value to the other nodes. For certain special choices of the cost function $\bar{D}^{ij}$ and under certain assumptions, the partial derivative $\partial\bar{D}^{ij}/\partial F^{ij}$ equals the average delay of a packet traveling through link (i,j). In that case, it is very natural to assume that this derivative may be measured directly, without first measuring the flow F^{ij}. Our result may be easily shown to be valid for this class of algorithms as well.

REFERENCES

1. Gallager, R.G., "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. on Communications, Vol. COM-25, 1977, pp. 73-85.

2. Schwartz, M., and Stern, T.E., "Routing Techniques Used in Computer Communication Networks", IEEE Trans. on Communications, Vol. COM-28, 1980, pp. 539-559.

3. Bertsekas, D.P., "Optimal Routing and Flow Control Methods for Communication Networks", in Analysis and Optimization of Systems, A Bensoussan and J.L. Lions, Eds., Springer Verlag, New York, 1982, pp. 615-643.

4. McQuillan, J.M., Richer, I., and Rosen, E.C., "The New Routing Algorithm for the ARPANET", IEEE Trans. on Communications, Vol. COM-28, 1980, pp. 711-719.

5. Bertsekas, D.P., and Gafni, E.M., "Projected Newton Methods and Optimization of Multicommodity Flows", IEEE Trans. on Aut. Control, Vol. AC-28, 1983, pp. 1090-1096.

6. Bertsekas, D.P., Gafni, E.M., and Gallager, R.G., "Second Derivative Algorithms for Minimum Delay Distributed Routing in Networks", IEEE Trans. on Communications, Vol. COM-32, 1984, pp. 911-919.

7. Gafni, E.M., and Bertsekas, D.P., "Asymptotic Optimality of Shortest Path Routing", LIDS Report P-1307, M.I.T., July 1983.

8. Bertsekas, D.P., "Distributed Dynamic Programming", IEEE Trans. on Automatic Control, Vol. AC-27, 1982, pp. 610-615.

9. Bertsekas, D.P., "Distributed Asymchronous Computation of Fixed Points", Mathematical Programming, Vol. 27, 1983, pp. 107-120.

10. Tsitsiklis, J.N., Bertsekas, D.P., and Athans, M., "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms", Proceedings of the 1984 American Control Conference, San Diego, CA, 1984.

11. Tsitsiklis, J.N., "Problems in Decentralized Decision Making and Computation", Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1984.

12. Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay; McGraw-Hill, N.Y., 1964.

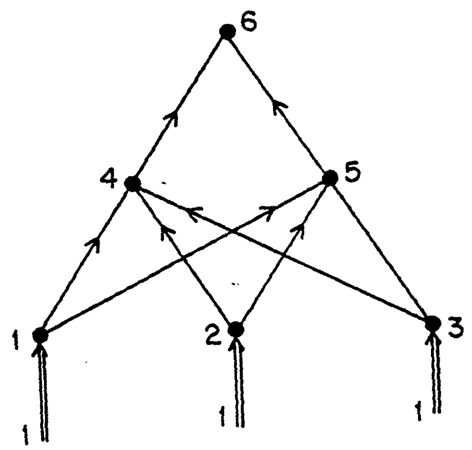


Fig. 1

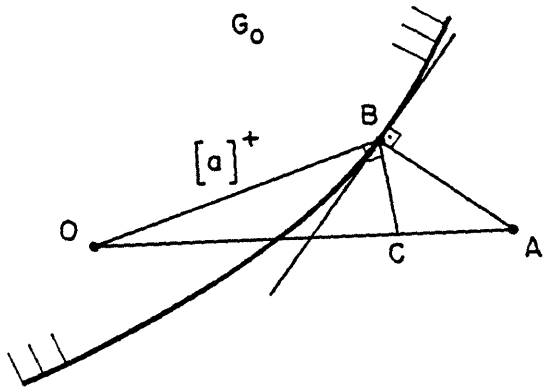
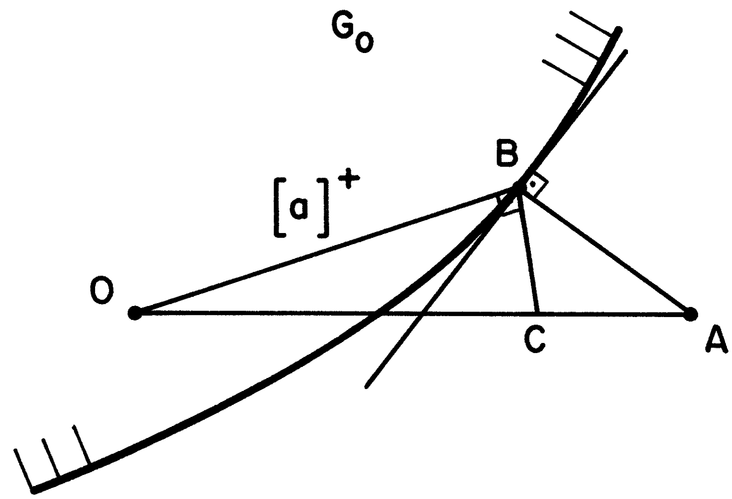
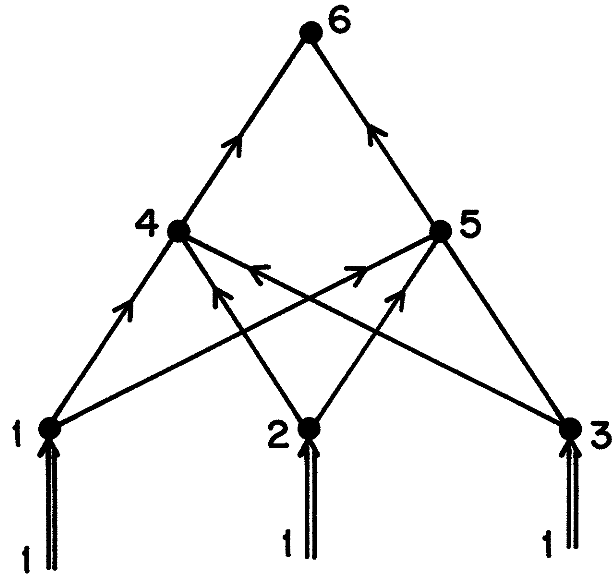


Fig. 2



G₀
B
[a]+
O
C
A


6
4
5
1
2
3
1
1
1