Equilibrium flows, routing patterns and algorithms for store-and-forward networks

L.G. MASON

INRS-Télécommunications, 3 Place du Commerce, Verdun, Québec H3E 1H6, Canada

1. Introduction

A considerable effort has gone into the synthesis and analysis of techniques for routing messages (packets) in store-and-forward (S/F) networks [1–39]. This work has been motivated by the importance of the routing protocol in respect to the delay performance, system cost and reliability. The area continues to attract attention due to the wide spectrum of choice in regard to routing protocol as well as the numerous and subtle behavioral characteristics which these schemes exhibit. Routing schemes have been implemented prior to a thorough understanding of their characteristics, with the result that system instability and deadlocks have occurred and only were corrected after the fact [12]. To obviate these difficulties, previous and ongoing research has sought an adequate theory for routing in store-and-forward networks.

Routing schemes in store-and-forward networks are described as being static (fixed), adaptive (or equivalently, quasi-static) and dynamic, respectively. While all schemes are dynamic to some degree, as pointed out by Schwartz and Stern [17], the static schemes employ a fixed routing, while the network topology remains invariant. When links or nodes are added or the link capacity is augmented, a possibly different static routing plan is employed. Adaptive or quasi-static schemes on the other hand do allow the routing pattern to change if the required traffic demand changes or if links or nodes fail. These changes occur much more slowly than the changes in the states of buffer occupancy. In a dynamic routing scheme, as the term is used in connection with store-and-forward networks, the routes are determined by the instantaneous state of the buffers.

This paper considers the quasi-static or adaptive routing problem. Studies on adaptive routing in S/F networks date back as far as Boehm and Mobley [38] who were concerned with reducing the vulnerability of message-switched military communication networks. Glorioso et. al. [39] applied learning techniques to adaptively route traffic in message-switched networks. With the advent of packet switching in the early 1970s a number of researchers again turned to adaptive routing techniques in the hope of improving network performance [4]. These early studies employed Monte Carlo simulation techniques to evaluate network behavior.

Fratta and Gerla and Kleinrock [5] posed the system optimal static routing problem as a convex programming problem and introduced the flow deviation algorithm as a means of computing the system optimal flow recursively in a centralized fashion. This marked the beginning of theoretical work of the routing problem in packet switching networks. In the mid-1970s Gallager [7] continued this trend with a classic paper on quasi-static routing. He formulated the optimal static routing problem as a convex programming problem in the space of routing variables as opposed to path or link flows, and derived necessary and sufficient conditions for its solution. He also proposed a protocol for on-line calculation of the system optimal routing pattern, and derived convergence properties of the scheme.

North-Holland
Large Scale Systems 8 (1985) 187–209

Recently, Mars, Narendra and Chrystal [24, 35] have again taken up the learning approach, armed with the recent results for learning automata operating in nonautonomous environments due to Narendra and Thatachar [36]. Extensive theoretical and simulation studies have demonstrated that a collection of L_{RI} automata controlling the routing of messages in a store-and-forward network provide a potentially effective means of adaptive routing. Moreover, in a number of simulation experiments the scheme tends to converge to a routing pattern which equalizes delays over the set of used paths [35, 37]. The present paper was inspired by the recent interest in the decentralized learning approach with the emphasis placed on analysis as opposed to a simulation. The paper can be viewed as a contribution to the area of learning in networks, as well as to the theory of nonlinear multicommodity flows.

Previous theoretical work on the routing problem in S/F networks emphasized the system optimization problem as opposed to the user equilibrium problem. While much of the current theory on routing can be adapted to the equilibrium problem with minor modification, as will be subsequently shown, we have placed particular emphasis on the equilibrium problem because of its relevance to learning automata routing schemes.

Previous algorithms for system optimal quasi-static routing assume or require the absence of loops or cycles in the flow of messages. Protocols are set up to exchange marginal delay information, which is then used to update the routing pattern or the path flows. The overall procedure is quite complex and difficult to implement in large distributed networks.

By contrast, the recursive schemes described in this paper, which models the L_{RI} learning automaton scheme as a special case, is very easy to implement, as only simple arithmetic operations are involved, and the delay measurements are obtained directly from the round trip time for messages and their acknowledgements. Hence no additional protocol for nodal information exchange is necessary. In addition the approach does not require loop-freedom to operate properly. This is an important attribute when only local topological information is available. Facility failures, for example, can introduce uncertainty in regard to network topology. To determine a loop-free routing it is necessary to possess global information on topology, which in turn requires a higher level protocol.

One of the potential drawbacks of our scheme is suboptimal performance under some conditions. It is therefore important to quantify the magnitude of this performance deficit relative to an optimal scheme as well as to characterize conditions under which the suboptimality can arise. Numerical studies and analyses indicate that user equilibrium routing and system optimal routing both tend to minimum hop routing under balanced or symmetric traffic conditions as well as under light asymmetric traffic conditions. Differences appear in the two routing methods for asymmetric networks under moderate and heave traffic, where bifurcated flows arise.

In Section 2 we summarize previous work on the system optimal routing problem. By modifying the objective function in the manner described by Dafermos and Sparrow [11] most of these available results and methods can be extended to the user equilibrium problem.

In Section 3 we describe the path and link flow formulations of the user equilibrium routing problem in S/F networks. Properties of the equilibrium flows and routing patterns are discussed, along with techniques for computing a solution.

In Section 4 we define and characterize various classes of routing patterns. Properties of the various classes are then derived. We then describe the route formulation of the user equilibrium problem and the associated optimality conditions. The complications resulting from the presence of zero elements in the traffic matrix are then discussed.

In Section 5 a recursion for updating the routing variables to compute an equilibrium routing and flow pattern is described, along with proofs of its behavior. Specifically, it is shown that the stable fixed points of the nonlinear recursion correspond to equilibrium routing patterns. Using the theory relating the user and system optimal problems, a technique is described by which the recursion can be used to compute the system optimal routing and its delay performance.

The network and controller models have been implemented in software and a numerical study of a store-and-forward network has been carried out. In Section 6 numerical results are presented which corroborate the theory and demonstrate the potential of a large class of learning schemes.

In Section 7 the issue of implementing the recursion in a network is discussed. The form of the recursion, suggests other implementations in addition to the L_{RI} scheme already mentioned.

2. Review of optimal static and quasi-static routing

There are a number of different problem formulations, representations and solution methods associated with the optimal static routing problem, in the transportation and communications literature.

Dafermos and Sparrow [11] have considered two traffic assignment problems originally posed by Wardrop [27] in connection with transportation networks. With minor modifications the formulations are suitable for the optimal static routing in S/F communication networks as well. The problems referred to are the system optimization and the user equilibrium problem mentioned previously. Since it has been shown [11] that a user equilibrium problem can be posed as an equivalent system optimization problem and vice versa by suitably modifying the cost functions, solution methods for either problem can be adapted to the other problem formulation as well.

There are two major representations for network routing problems, where in the first instance the problem is formulated in the space of flows, and in the second case in the space of the routing variables, which are the flow fractions allocated to the link. Within the flow formulations there are the path flow and the link flow forms. Representations in terms of routing variables are directly suitable for implementation; however, the optimization problem is convex as opposed to being strictly convex for the most general class of traffic matrix. This complicates the necessary and sufficient conditions (NSC) of optimality. For objective functions where the optimal routing depends on destination node only, rather than on the origin destination pair, the route and link flow formulations can be simplified thereby reducing the number of variables by a factor of $N-1$. The path formulations, on the other hand cannot be so reduced.

Path flow formulations are natural for virtual circuit operation as the path flow rates are the variables to be implemented. For datagram operation, a set of routing variables must be calculated once the path flows are determined. For the route formulation the routing variables can be used directly in the implementation. Link flow formulations, on the other hand, require additional auxiliary variables for implementation is either datagram or virtual circuit environments. The objective function in the space of path flows is, however, strictly convex, leading to simpler optimality conditions than is the case for the route formulations.

While necessary and sufficient conditions have been derived for optimal static routing policies which provide insight into the form of the optimal solution, closed form solutions have not been obtained. This is because the optimality conditions lead to large systems of coupled nonlinear equations in the routing variables which are notoriously difficult to solve.

As an alternative approach, network theorists have proposed recursive algorithms, which converge in the limit to the optimal static routing plan in stationary traffic conditions. Such algorithms can also be used on-line for adaptive (quasi-static) control provided the rate of convergence is sufficiently fast compared to changes in the traffic demand.

Finally, several solution techniques have been proposed for the different formulations and representations. In the first class of solution techniques, necessary and sufficient conditions for optimality are first derived, and following this an algorithm is proposed which adjusts the flows (or routing variables) to converge to a policy such that the NSCs are satisfied. In the second approach the optimization problem is tackled directly via some variation of the gradient search technique.

Within the class of approaches based on NSC, there are several solution techniques whose points of departure are different NSCs. For example, Dafermos and Sparrow [11] state the NSC in terms of the marginal costs along all paths where the cost derivatives are expressed in terms of the total link flows. They then propose a centralized recursive scheme based on their *equilibration operator*, which converges in the limit to a flow which satisfies these NSC.

Stern [9] on the other hand gives NSC in terms of marginal costs by commodity flow on all links. It is apparent that these differ from the NSC of [11] in that links are considered rather than paths, and the derivatives are by commodity rather than by the total link flows. Stern's NSCs lead to a large coupled system of nonlinear equations which he then solves by relaxation to obtain the optimal static flow pattern or equivalent routing plan. The method admits decentralized computation and is suitable for implementing a decentralized quasi-static routing scheme.

Yet another set of NSC is given by [5] which forms the basis of their flow deviation algorithm. Their NSCs are expressed in terms of the marginal link costs with respect to the total link flows. As a result these optimality conditions are more compact than those of Dafermos and Stern. These NSCs can be tested for a given flow by a shortest path calculation using the link marginal costs as the distance metric. The flow deviation method also employs this shortest path calculation to evaluate a direction for deviating a nonoptimal flow to obtain an improved flow. The flow deviation algorithm is a centralized recursive algorithm for calculating the optimal flow among the class of bifurcated flows. It can also be employed in problems with nonconvex objective function to obtain local minimum flows. Fratta et al. [5] have devised an efficient version of the flow deviation algorithm for computing the optimal nonbifurcated flow. They observe that for large balanced networks, the optimal flow among the class of bifurcated flows will be nonbifurcated. For the general class of networks, an optimal nonbifurcated solution is only a heuristic solution. It is nevertheless of interest as nonbifurcated routing schemes avoid certain sequencing and reassembly problems associated with bifurcated routing. In fact almost all existing networks use nonbifurcated routing.

Gavish and Hantler [23] have recently introduced a technique for finding heuristic solutions and tight lower bounds on the performance of nonbifurcated routing schemes. The method employs a Lagrangian relaxation of the mixed integer nonlinear programming problem to obtain the routes and bounds with a moderate amount of computation. This approach belongs to class where the optimization problem is tackled directly, in this case via a subgradient optimization of the relaxed problem, rather than by first determining the NSC.

Bertsekas [13] has considered gradient projection methods for solving the bifurcated system routing problem in the space of path flows. A formulation in the space of link flows has also been considered [14]. They claim a linear or superlinear convergence rate, and automatic scaling with respect to traffic level. The algorithm can be implemented in either centralized or decentralized fashion, and it is claimed to be the most efficient recursive solution, in terms of computational requirements, to the system optimal routing problem.

3. Flow formulations of the user equilibrium problem

There are two flow based formulations of the routing problem. The first is referred to as the link flow form, while the second is the path flow form. The link flow form admits a reduction in terms of the number of commodity types in that one need only distinguish commodities by their point of destination for minimum flow and minimum delay routing. In the path formulation both origin and destination are implicitly involved in the notion of a path.

The set of feasible link flows form a convex polyhedral set, where the extreme points correspond to minimum distance routing according to a metric. By varying the link lengths over all possible values the extreme points of the feasible set can be generated by application of a shortest path algorithm. Fratta et al. have used this property to advantage in their flow deviation method to be described in the sequel. The set of feasible path flows, on the other hand, is the Cartesian product of a set of simplexes with one simplex for each origin destination pair. Bertsekas [13] and Mason [18] have derived efficient search schemes for such sets, which are used to advantage in their recursion formulae.

3.1. Notation

Let $\mathcal{N} = \{1, 2, \dots N\}$ denote the set of nodes; $\mathcal{L} = \{1, 2, \dots, B\}$ be the set of oriented links; C_l is the capacity of link l in bits/second; λ_{ij} is the mean flow demand in bits/second, from source node i to destination node j. $\mathscr{W} = \{(i,j) | \lambda_{ij} > 0\}$. For each $(i,j) = w \in \mathscr{W}$ let $\lambda_{ij} = \lambda_w$. Let P_w be the set of directed loop free paths originating at node i and terminating at node j. This is the set of paths that may carry commodity w. Let f_p be the flow on path p. Define $f = \{f_p | p \in P_w, w \in \mathscr{W}\}$ to be the vector of path flows.

3.2. Flow sets

The *admissible* flow set $\mathscr{F}$ is defined by flow conservation or

$$\mathscr{F} = \left\{ f | f_p \geq 0, \sum_{p \in P_w} f_p = \lambda_w, \forall p \in P_w, w \in \mathscr{W} \right\}.$$

For a flow, f, to be *feasible*, it must satisfy the flow conservation constraints as well as the link capacity constraints. The total link flows can be expressed in terms of the path flows by the equation

$$\bar{f}_{ij} = \sum_{w \in \mathscr{W}} \sum_{p \in P_w} \delta_p(i,j) f_p \quad \forall (i,j) \in \mathscr{L},$$

where $\delta_p(i,j) = 1$ if path p contains link (i,j), and $\delta_p(i,j) = 0$ otherwise. For the flow, f, to be feasible, no total link flows $\bar{f}_l$ can exceed the link capacity or

$$\bar{f}_l \leq C_l \quad \forall l \in \mathscr{L}.$$

Here C_l is the capacity of link l in bits/second. The vector of total link flows is given by

$$\bar{f} = \{\bar{f}_l | l \in \mathscr{L}\}.$$

The link capacity constraints can be expressed explicitly in terms of the path flows f_p as

$$0 \leq \sum_{w \in W} \sum_{p \in P_w} \delta_{pl} f_p \leq C_l \quad \forall l \in \mathscr{L}.$$

Let $\mathscr{F}_f$ be the set of feasible flows, and $\overline{\mathscr{F}_f}$ be the set of feasible total link flows. $\mathscr{F}_f$ is convex since it is the intersection of the convex capacity constraint set with the convex flow requirement or admissible set. $\overline{\mathscr{F}}_f$ is also convex by virtue of the linear mapping from $\mathscr{F} \to \bar{\mathscr{F}}$.

3.3. Link cost functions

For S/F telecommunication networks, the measure of link cost is the mean link delay, Δ_l, in seconds/message. Kleinrock [20] has applied results for queuing theory to obtain expressions for the mean link delay and the total link delay in terms of the network and traffic parameters. Models have been proposed which include effects of nodal processing delay, link propagation and transmission delay as well as queuing delay. In the sequel we shall focus on the simplest and most commonly employed formula which neglects the node processing and propagation delays. It can be shown that the results can be easily extended to the more complex delay models as well:

$$\Delta_l = \frac{\alpha}{C_l - \bar{f}_l},$$

where α is the mean message length in bits. The total link delay in seconds is given by the expression

$$D_l = \bar{f}_l \Delta_l = \frac{\alpha \bar{f}_l}{C_l - \bar{f}_l}.$$

The total network or system delay is obtained by summing the total delay on all links:

$$D=\sum_{l=1}^{B} D_l=\sum_{l=1}^{B}\frac{\alpha \bar{f}_l}{C_l-\bar{f}_l}.$$

The mean delay of path p is given by

$$\Delta_p=\sum_{l=1}^{B}\Delta_l\delta_{pl}=\sum_{l=1}^{B}\delta_{pl}\frac{\alpha}{C_l-\bar{f}_l}.$$

3.4. *User equilibrium flows*

Dafermos and Sparrow define the user equilibrium flow as a Nash strategy for the noncooperative game where a player is associated with each origin destination pair. They also give an alternative characterization in terms of the path delays which is more useful for our purposes.

Definition. A flow f_e is an equilibrium flow iff f_e satisfies the following constraints expressed in terms of the path delays:

$$\Delta_{p\in P_w}=\mu_w \quad \forall\{p\,|\,f_p>0\},\quad p\in P_w,\quad w\in \mathcal{W},$$
$$\Delta_{q\in P_w}\geqslant \mu_w \quad \forall\{q\,|\,f_q=0\},\quad q\in P_w,\quad w\in \mathcal{W},$$

where μ_w is a positive multiplier associated with commodity w.

3.5. *Optimization formulations of user equilibrium problem*

In ref. [11] it is shown that equilibrium problems can be cast in an optimization form via the introduction of a modified link cost function $\Delta_l^*(\bar{f}_l)$ which is related to the mean link cost function by the equation

$$\Delta_l^*(\bar{f}_l)=\int_0^{\bar{f}_l}\Delta_l(x)\,\mathrm{d}x.$$

For the delay function considered previously we obtain:

$$\Delta_l^*(\bar{f}_l)=\alpha \ln\frac{C_l}{C_l-\bar{f}_l}.$$

The optimization formulation of the equilibrium problem in path flow form is then

$$\min_{f\in \mathcal{F}_f}\sum_{l=1}^{B}\alpha \ln\frac{C_l}{C_l-\bar{f}_l}$$

and the link flow form is

$$\min_{\bar{f}\in \bar{\mathcal{F}}_f}\sum_{l=1}^{B}\alpha \ln\frac{C_l}{C_l-\bar{f}_l}.$$

The link flow formulation has a unique equilibrium total flow vector $\bar{f}_e$, when $\bar{\mathcal{F}}_f$ is not empty.

This result follows from the fact that the objective function is strictly convex in $\bar{f}_l$ and the constraint set is a convex polytope. Identical results were reported by Fratta, Gerla and Kleinrock [5] for the system optimal problem, which differs from our problem only in the form of the objective function.

To obtain the optimality conditions in terms of the path flows, apply the Kuhn–Tucker results to the path flow formulation. The resulting necessary conditions are identical to the characterization of the equilibrium conditions. Sufficiency of these conditions follows from the fact that every path flow f satisfying the necessary conditions has a unique image in $\bar{\mathcal{F}}$ which satisfies the sufficiency condition.

The set of equilibrium path flow vectors is denoted by $\mathcal{F}_e$. This set is convex due to the linear mapping $\mathcal{F} \to \bar{\mathcal{F}}$, and the uniqueness of $\bar{\mathcal{F}}_e$.

While the equilibrium total link flow vector is unique, the set of equilibrium path flow solutions is not necessarily unique. This fact was first noted in ref. [11] where an example is given.

3.6. Solution methods

The literature describes several techniques which can be used to compute a solution to the above optimization formulations of the equilibrium problem. For example, flow deviation can be used to compute a $\bar{f}_e \in \bar{\mathcal{F}}_e$. Alternatively, Stern's relaxation technique [9] can be applied to the path flow formulation to compute an equilibrium flow. In both instances, once the equilibrium flow is determined the actual system performance is obtained in terms of this flow and the true link delay function by

$$\sum_{l=1}^{B} \frac{\alpha \bar{f}_{le}}{C_l - \bar{f}_{le}}.$$

It should be emphasized that the above-mentioned methods are suitable only for an off-line calculation as they depend upon the fictitious link cost function $\alpha \ln(C_l/(C_l - \bar{f}_l))$ which cannot be determined from local delay measurements. In Section 5 we introduce an alternative approach using a recursion which requires path delay information which is easily obtained from local measurements in an on-line setting, or can be computed from a link delay model in off-line applications.

4. Route formulations

We now consider programming models where the decision variables are expressed directly in terms of the fractions allocated to the outgoing links, the so-called routing variables. Models of this sort have been described by Gallager [7] and Segall [25]. The approach of Gallager is intended for message switched or datagram networks. It has been extended by Segall [25] to virtual circuit S/F networks as well. Mason [18] also studied such formulations for a class of decentralized control problems which includes the S/F network routing problem as a special case.

4.1. Classification of routing patterns

We define a *routing pattern* as a collection on N, $N \times N$ matrices

$$R = \{R^{(1)}, R^{(2)}, \ldots, N^{(N)}\},$$

where

$$R^{(d)} = [r_{ij}^{(d)}], \quad i, j \in \mathcal{N}.$$

The routing variables, $r_{ij}^{(d)}$ denote the fraction of messages (packets) arriving at node i with destination d which are routed directly to node j.

4.1.1. Admissible routing patterns

An *admissible* routing pattern satisfies the following conditions:

$$\sum_{j=1}^{N} r_{ij}^{(d)} = 1, \quad r_{ij}^{(d)} \geq 0, \quad \forall i \neq d, \quad \forall j \neq i$$
$$r_{ii}^{(d)} = 0, \quad \forall d,$$
$$r_{dj}^{(d)} = 0, \quad \forall j.$$

Let $\mathcal{R}$ be the set of all admissible routing patterns.

4.1.2. Feasible, equilibrium, and optimal routing patterns

To qualify as being *feasible* a routing pattern must be admissible and the resulting flow must be feasible, i.e. it must satisfy the flow conservation and the link capacity constraints. Let $\mathcal{R}_f$ denote the set of *feasible* routing patterns.

Complications arise when some off-diagonal elements in the demand matrix are zero. Here we restrict attention to the case where all point-to-point demands are strictly positive. Later we will discuss the more general case.

For nonzero demands we can define a routing pattern uniquely in terms of the link flows. The routing variables are simply the flow fractions or

$$r_{ij}^{(d)} = f_{ij}^{(d)} \Big/ \sum_{i} f_{ij}^{(d)},$$

where $f_{ij}^{(d)}$ is the flow on link ij with destination d. These multicommodity link flows are uniquely determined from the path flows by the equation:

$$f_l^{(d)} = \sum_{i=1}^{N} \sum_{p \in P_w, w \in (id)} f_p \delta_{pl}.$$

We can therefore define the feasible routing patterns $\mathcal{R}_f$ by the mapping $\mathcal{F}_f \to \mathcal{R}_f$. Equilibrium and optimal routing patterns can be similarly defined in terms of the corresponding preimages in the path flow space. Let $\mathcal{R}_e$ denote the set of equilibrium routing patterns and $\mathcal{R}^*$ denote the set of system optimal routing patterns.

Proposition 1: Convexity of routing pattern sets. *The sets $\mathcal{F}_f$, $\mathcal{F}_e$, and $\mathcal{R}$ are convex.*

Proof. Since the sets $\mathcal{F}_f$, $\mathcal{F}_e$, and $\mathcal{F}^*$ have been shown to be convex it is sufficient to show that a convexity in the space of path flows implies convexity in the image sets in the space of routing patterns. Consider any two distinct path flows, f' and f'', in any convex set of path flows. By definition the flow f''' is also an element of the set where

$$f''' = \lambda f' + (1 - \lambda) f''$$

and $0 \leq \lambda \leq 1$. Let R', R'' and R''' be the corresponding images in the space of routing patterns. To establish convexity of the image sets in $\mathcal{R}$ it is sufficient to show that there exist scalars $0 \leq \theta_i^{(d)} \leq 1$ $\forall i, d$ such that

$$r_{ij}^{(d)\prime\prime\prime} = \theta_i^{(d)} r_{ij}^{(d)\prime} + (1 - \theta_i^{(d)}) t_{ij}^{(d)\prime\prime}.$$

Notice that $\theta_i^{(d)}$ can vary with node and commodity index as each of the $r_{ij}^{(d)}$ are disjoint sets for distinct i and d. To simplify the presentation we shall delete the indices i and d in the following since the same analysis holds for all cases. Expressing the convexity condition for r_j in terms of the flows f_j we must show that

$$\theta\Big(f_i'\Big/\sum_i f_i'\Big)+(1-\theta)\Big(f_i''\Big/\sum_i f_i''\Big)=\frac{\lambda f_j'+(1-\lambda)f_j''}{\sum_j(\lambda f_j'+(1-\lambda)f_j'')}\,.$$

To verify the above choose

$$\theta=\frac{\lambda}{\lambda+(1-\lambda)\Big(\sum_j f_j''\Big/\sum_j f_j'\Big)}\,.$$

It is clear that $0\leqslant\theta\leqslant1$ since $0\leqslant\lambda\leqslant1$ and $\sum_j f''/\sum_j f_j>0$ by the premise of positive traffic. This completes the proof.

4.1.3. Loop-free and deadlocked routing patterns

A routing pattern is *loop free* iff the following is true for all cyclic paths in the network:

$$\prod_{(ij)\in\text{cycle}} r_{ij}^{(d)}=0\quad\forall d\,.$$

This obvious necessary and sufficient condition is difficult to verify due to the enormous number of distinct cycles in networks of practical size. An alternative characterization is given by the following proposition.

Proposition 2: Loop-free routing condition. *A routing is loop-free with respect to commodity d if the following matrix equation holds:*

$$[R^{(d)\prime}]^N=0\,,$$

where the prime denotes the transpose.

Proof. To verify this claim, consider the effect of the matrix $R^{(d)\prime}$ as an operator on traffic λ_{id}. For nodes $i\neq d$ the operator shifts a portion $r_{ij}^{(d)}$ from node i to node j. For $i=d$ all traffic at d leaves the network. Thus, we can view the matrix multiplication as a shift operation in the network. In an N node network there can be at most $N-1$ hops or shifts in a sequence if a cycle or loop is to be avoided. Hence, if $[R^{(d)\prime}]^N\geq0$ a loop would be formed. Thus a loop-free routing implies that the condition on R holds.

Proposition 3: Loop strength. *For a loop-free routing the following holds:*

$$\operatorname{Det}(d)=|I-R^{(d)}|=1\,.$$

The nodal flow vector,

$$\bar\gamma^{(d)}=[\gamma_1^{(d)},\ldots,\gamma_N^{(d)}]'\,,$$

satisfies the matrix equation:

$$\bar{\gamma}^{(d)} = [I - R^{(d)\prime}]^{-1}\bar{\lambda}^{(d)} ,$$

where $\bar{\lambda}^{(d)}$ is the column vector of traffic demand with destination d.

Proof. For a finite nodal flow vector to exist the matrix $[I - R^{(d)}]$ must be nonsingular. This condition is equivalent to the conservation of flow. When this matrix is singular for some d the routing pattern is said to be *deadlocked*. For deadlock-free routing patterns we can expand the inverse matrix as a geometric matrix series:

$$[I - R^{(d)\prime}]^{-1} = \sum_{k=0}^{\infty} [R^{(d)\prime}]^k .$$

In particular, for loop-free routing we have the finite sum:

$$[I - R^{(d)\prime}]^{-1} = \sum_{k=0}^{N-1} [R^{(d)\prime}]^k ,$$

where $[R^{(d)\prime}]^0 = I$, the identiy matrix. For ease of notation let us define the matrix $A = R^{(d)\prime}$. From matrix theory the inverse can be expressed by the following matrix identity:

$$[I - A]^{-1} = \frac{Adj[I - A]}{|I - A|} = \frac{A^{N-1} + (1 + a_1)A^{N-2} + \cdots (1 + a_1 + a_2 + \cdots a_{N-1})I}{1 + a_1 + a_2 + \cdots + a_N} .$$

But by our previous result this is equal to

$$\sum_{k=0}^{N-1} A^k .$$

Since this equality holds for any loop-free routing, this implies that $a_i = 0$, $i = 1, 2, \ldots, N$, which in turn implies:

$$|I - R^{(d)\prime}| = |I - R^{(d)}| = 1 ,$$

as claimed.

For deadlocked routing patterns, this determinant vanishes as the corresponding matrix is singular. This suggests the following measure for the strength or magnitude of the routing loop associated with commodity d.

Definition. The *loop strength* for commodity d is defined as $1 - |I - R^{(d)}|$. For loop-free routing the strength is zero while for deadlocked loops the strength is unity. For intermediate cases the determinant takes on values in the open interval $0 < |I - R^{(d)}| < 1$.

4.1.4. Bifurcated routing patterns

By definition a *bifurcated* routing pattern R_b is one in which for some i, j, d the following strict inequality holds:

$$0 < r_{ij}^{(d)} < 1 .$$

The complementary set of nonbifurcated routing patterns which are inherently deterministic or fixed has the following characterization:

$$r_{ij}^{(d)} \in \{0, 1\} \quad \forall i, j, d .$$

4.2. Route formulations of the equilibrium routing problem

$$\min_{R \in \mathcal{R}_f} \sum_{l=1}^{B} \alpha \ln \frac{C_l}{C_l - \bar{f}_l} .$$

Notice that the feasibility condition involves nonlinear inequalities in the routing variables since the link capacity constraints are

$$0 \leq \bar{f}_{ij} = \sum_d \gamma_i^d r_{ij}^d \leq C_{ij} \quad \forall i, j ,$$

and the flow conservation constraints are

$$\bar{\gamma}^d = [I - R^{(d)t}]^{-1} \bar{\lambda}^d .$$

In spite of these apparent difficulties, in view of the fact that $\mathcal{R}_f$ is convex the above problem is one of convex programming.

A relaxed version of this problem is based on the fact that the objective function implicitly includes the link capacity and flow conservation constraints, since the cost tends to infinity when these constraints are approached from the feasible side. Gallager [7] employed this approach for the system optimal static routing problem. Our formulation differs only in the form of the objective function:

$$\min_{R \in \mathcal{R}} \sum_{l=1}^{B} \alpha \ln \frac{C_l}{C_l - \bar{f}_l} .$$

The constraints $R \in \mathcal{R}$ are linear in the variables $r_{ij}^{(d)}$ and hence are convex.

The Kuhn–Tucker necessary conditions take the form:

$$\frac{\partial}{\partial r_{ij}^{(d)}} \left(\sum_{l=1}^{B} \alpha \ln \frac{C_l}{C_l - \bar{f}_l} \right) = \mu_i^d \quad \{i, d \mid r_{ij}^{(d)} > 0\}$$

$$\geq \mu_i^d \quad \{i, d \mid r_{ij}^{(d)} = 0\} .$$

For nonzero demands these conditions are also sufficient since every equilibrium routing pattern induces a total link flow which is unique and sufficient. For cases with zero demand elements, these necessary conditions are not sufficient, but can correspond to points of inflection. The Kuhn–Tucker conditions result in a large coupled nonlinear system in the routing variables. The recursion given in the next section is shown to converge to the equilibrium routing pattern, provided it is appropriately initialized.

Some demands are null

If $\lambda_{id} = 0$ for some $i \neq d$, then several complications arise. It is then possible for $\sum_j f_{ij}^{(d)} = 0$ and we cannot define a routing pattern in terms of the flows. On the other hand, any admissible assignment of the routing variables $r_{ij}^{(d)}$ produces the same flow, namely zero, so in a sense this is a 'don't care' condition. Interpreted in this way the property of convexity of the sets $\mathcal{R}_f$, $\mathcal{R}_e$, and $\mathcal{R}^*$ extends to the zero demand case.

A related complication concerns the sufficiency condition for the equilibrium routing patterns. As pointed out by Gallager [7], the necessary conditions may define inflection points rather than a minimum. Interestingly, this property has implications for the rate of convergence in the case where some demands are positive but small as 'near inflection' points reduce the rate of convergence of gradient-based search schemes.

A third complication concerns the definitions of feasible and loop-free routing patterns. For positive demand elements, singularity of the matrix $[I - R^{(d)}]$ implies that the flow conservation condition is not satisfied, while the presence of routing loops implies cyclic flows. Such is not the case if some demand elements are zero.

5. Distributed recursion for equilibrium routing

The following recursion was first described in [18] where it was applied to a general class of problems which includes the optimal S/F routing problem as a special case. Convergence results were given for the general problem. Here we apply the algorithm to the minimal system average delay and user equilibrium problems in a S/F network.

If the recursion is implemented on-line in a network, only equations (5) and (6) need be computed, since the delays are obtained from measurements of acknowledgement delays on the actual network. When the recursion is used off-line we will in addition compute the expected link, point-to-point, and network delays as well. It is significant that the bulk of the computation concerns the delay calculations, which are obviated in the case of on-line operation. The following equations consider the off-line case as this includes the on-line version as a subset.

First compute the nodal flow vector (messages/second) in terms of the current routing variables by the matrix equation:

$$\bar{\gamma}^{(d)} = [I - R^{d}]^{-1}\bar{\lambda}^{d} \quad \forall i, \forall d \neq i. \tag{1}$$

Then compute the total link flows (bits/second) on all links by equation (2):

$$f_{ij} = \sum_{d} \gamma_i^{(d)} r_{ij}^{d} \alpha \quad \forall i, \forall j \neq i. \tag{2}$$

Then compute the average delay on link ij for commodity d by the following equations. It is the same for all commodities:

$$\Delta_{ij}^{(d)} = \frac{\alpha}{C_{ij} - f_{ij}} \quad \forall i, \forall j \neq i. \tag{3}$$

Now compute the nodal delays as follows:

$$\Delta_i^{(d)} = [I - R^{d}]^{-1} \sum_{j} r_{ij}^{d} \Delta_{ij}^{(d)} \quad \forall i. \tag{4}$$

Define $S_{ij}^{(d)}$ as the reward strength associated with routing commodity d on link ij from node i. $\Delta_i^{(d)}$ is the average delay in seconds/message for commodity d from node i. Then compute the reward strengths which are defined as the 1's complement of the normalized delays associated with selecting link ij, or

$$S_{ij}^{(d)} = 1 - (\Delta_{ij}^{(d)} + \Delta_j^{(d)})/\Delta_{\max} \quad \forall i, \forall j. \tag{5}$$

Now update the routing variables according to the nonlinear recursion:

$$r_{ij}^{(d)}(t+1) = r_{ij}^{(d)}(t)\left(1 + G\left(S_{ij}^{d}(t) - \sum_{k} S_{ik}^{d}(t)\, r_{ik}^{d}(t)\right)\right), \tag{6}$$

where G is a positive constant. Return to step 1 and repeat until a stopping condition is satisfied.

The values of G and Δ_{max} remain to be defined. We choose the normalizing constant as follows:

$$\Delta_{max}^{(d)} = \max_{i}\left\{\Delta_{i}^{(d)} + \max_{j}\{\Delta_{ij}^{(d)}\}\right\}.$$

We choose the parameter G in the range, $0 < G < 1$. Actually, we can generalize the constant to make G a function of node number, commodity, and time. The effect of such extensions on convergence rate and stability remain to be investigated.

5.1. Initialization

The initial routing variables must be specified to start the procedure, as well as to compute the constant Δ_{max} as defined above. For reasons to be discussed below, it is desirable to choose an initial routing which is deadlock free, loop free and feasible, if possible. Deadlock freedom and feasibility are essential, while loop freedom is not mandatory for correct operation of the algorithm. Loop freedom is desirable, however, since it reduces the computational requirements. This point will be elaborated on below. The method of demand scaling [5] can be employed to assure feasibility if indeed a feasible solution exists.

5.2. System optimization

To compute the system optimal routing using the above recursion one need only replace equation (3) above with the following equation (3′):

$$\bar{\Delta}_{ij}^{(d)} = \alpha C_{ij}/(C_{ij} - f_{ij})^2 \quad \forall i, j \neq i. \tag{3′}$$

5.3. Applications and extensions

The above procedure can be used for other cost functions such as might arise in circuit switched networks, or transportation networks, for example by replacing equations (3) or (3′) and (4) with a suitable cost function. For example, the total link flow function might be used if one seeks minimal hop routing on either a capacitated or uncapacitated network.

The above algorithm updates all commodities at all nodes at a given stage before proceeding to the next. This is an all-parallel operation. One can also consider other variations, where nodes or commodities are computed to their stationary values, one at time, while others remain fixed. This corresponds to a purely sequential operation. Combinations of parallel and sequential operations are also possible. Studies of the stability and convergence of such schemes remain to be carried out.

5.4. Discussion of numerical aspects

The major computational effort is associated with the matrix inversion to model the performance. This complexity is not due to the proposed control algorithm but is common to all algorithms. It can be shown the matrix requiring inversion can be expressed as a geometric series by

$$[I - R']^{-1} = \sum_{n=0}^{\infty} (R')^n .$$

Since this series converges exponentially for any initial deadlock-free routing, an approximate expression to any desired degree of accuracy can be obtained by a sum of finite terms involving only matrix multiplications. In the case where the routing is loop free, the series contains only a finite number of terms. Hence, the sum of the first N terms, where N is the number of nodes in the network, yields the exact value of the required inverse. We also note that both deadlock freedom and loop freedom can be checked in the course of the series evaluation. Deadlock corresponds to periodic behavior in n of elements of $(R')^n$, while the system is loop free if the $(R')^N = 0$, the zero matrix.

5.5. Properties of the recursion

Proposition 4: Routing pattern class preservation. *Admissibility is preserved by the recursion iff* $0 \leq G \leq 1$. *The properties of loop freedom, deadlock freedom and bifurcation are preserved if* $0 \leq G < 1$.

Proof. In what follows the indices d and i have been deleted for notational convenience. The proof consists of two parts. First we show that the recursion preserves the condition $\sum_j r_j = 1$ for all t and second that it preserves the condition $r_j > 0$ for all t iff $r_j(0) > 0$. If $r_j(0) = 0$, then the condition $r_j = 0$ is preserved for all t.

The first part follows immediately from summing both sides of the recursion over j and using the premise that $\sum_j r_j(0) = 1$.

For proof of the second part define the first difference as follows:

$$\Delta r_j(t) = r_j(t+1) - r_j(t) = r_j G\Big(S_j - \sum_k S_k r_k\Big),$$

where the t has been omitted from the right-hand side of the expression to simplify the description. The second admissibility condition is equivalent to the inequality

$$-r_j \leq \Delta r_j \leq 1 - r_j .$$

We assert that

$$-(1 - r_j) \leq S_j - \sum_k S_k r_k \leq 1 - r_j .$$

The central expression takes on its maximum value when $S_j = 1$ and all other $S_{k \neq j} = 0$ resulting in the upper bound given above. The expression takes on its minimum value when $S_j = 0$ and all other $S_{k \neq j} = 1$, resulting in the lower bound since $\sum_{k \neq j} r_k = 1 - r_j$. From this inequality the following holds:

$$-r_j G(1 - r_j) \leq \Delta r_j \leq r_j G(1 - r_j) .$$

Hence, if $r_j G \leq 1$ and $-r_j G(1 - r_j) \geq -r_j$, then Δr_j is admissible. Both conditions are satisfied if $0 \leq G \leq 1$ as stated in the premise. This completes the proof regarding the preservation of the admissible property.

Using the argument given in connection with admissibility but limiting the gains to the semi-open interval $0 \leq G < 1$, it is apparent that the fixed or nonbifurcated, and the bifurcated properties are also preserved by the recursion, as in the latter case the extreme values $r_j(t) \in \{1, 0\}$ are only attained as $t \to \infty$. Fixed routings are preserved as these can be shown to correspond to fixed points of the recursion. The preservation of the bifurcated property in turn implies that the properties of loop-freedom and its complement, as well as deadlock freedom and its complement, are preserved under the recursion.

The preservation of routing pattern properties is a very important attribute of the routing algorithm as it allows the designer to eliminate undesirable routing patterns at the time of initiation. For example, it is relatively easy to produce an initial deadlock-free routing. The algorithm then assures that the routing will be deadlock free for all time. Also, if one desires to fix certain routing variables at 0 or 1 at initiation, they will remain fixed at these values for all time. In this way any initial information about the equilibrium routing pattern can be programmed in at initiation, which can accelerate convergence. On the other hand, one must be careful not to overconstrain the solution set as the best routing strategy can be eliminated at the outset. By choosing all routing variables to be initially bifurcated we are assured that the algorithm will converge asymptotically to the equilibrium solution. However, the rate of convergence can be decelerated due to the introduction of loops at initiation which are preserved although reduced in strength as time progresses. Only in the limit as $t \to \infty$ are these loops eliminated entirely. These and other claims will subsequently be verified. Before doing this, however, it is desirable to discuss the issue of feasibility.

In the relaxed versions of the optimization formulations of the equilibrium problem we have tacitly assumed that a feasible solution exists. It is a nontrivial problem to determine an initial feasible routing pattern. One approach is based on the method of demand scaling due to Fratta et al. [5]. Other approaches are conceivable and this issue is now under study. Once an initially feasible routing has been found it must be maintained at each step of the recursion. We have already noted that the recursion preserves deadlock freedom which implies that the flow conservation constraint is satisfied. On the other hand the link capacity constraints are not necessarily satisfied by the discrete time recursion. Since the feasibility conditions are complex when expressed in terms of the routing variables, we have elected to simply check the capacity constraint at each stage. If it is violated, the step size G is reduced. This is continued until a feasible routing pattern results. It should be emphasized that in an actual implementation of the recursion in a network, this problem will not arise as the measured delays can never be negative. The possibility of jumping over the delay barrier is an artifact of the stationary model employed for the queuing network in our numerical studies.

In the following we suppress the node and destination index for ease of presentation. However, it should be understood that we are considering the multicommodity, multinode case.

Proposition 5: Fixed points of the recursion. (1) *The extreme points of the simplex of admissible set of routing variables are fixed points. These vertices correspond to deterministic routing.* (2) *Bifurcated routing patterns such that*

$$S_q(R) = \mu \quad \{q \mid r_q > 0\},$$
$$S_p(R) \le \mu \quad \{p \mid r_p = 0\},$$

are fixed points of the recursion. Both claims are easily verified by substitution of the fixed points into the recursion. It is relatively easy to show that no other fixed points exist.

5.6. Stability analysis

The stability properties of the continuous time version of the recursion were investigated in [18] for the multicommodity case, while the stability properties of the discrete version of the recursion were given in [19] for a stationary environment. The method employed to establish whether a fixed point is stable or unstable involves linearization about the fixed point. Alternatively, nonlinear stability theory employing Lyapunov functions can be applied. We do not include a complete analysis here due to space limitations as the general proof requires considering many cases and is quite tedious. Instead, we will illustrate how the result is obtained in some of the cases. Before doing this, however, we wish to remark that a stability analysis is useful in establishing that the fixed points of the recursion, which correspond to user equilibrium routing patterns, are stable while fixed points that are not equilibrium routing patterns are unstable. Stability analysis is also useful in selecting values for the adaptive loop gains G to obtain good transient performance.

Proposition 6: On the stability of fixed points. *Among the set of extreme points of the admissible set of routing variables only the extreme points correponding to the maximum of S_j are stable.*

Proof. To prove this we shall show that all points $r_j = \delta_{kj}$ are unstable where

$$k \neq \text{argmax}_j\{S_j\}$$

and the extreme point $r_j = \delta_{j^*j}$ is stable where

$$j^* = \text{argmax}_j\{S_j\}\,.$$

First consider $j \neq j^*$ and set $r_j = 1-\varepsilon$, $r_{j^*} = \varepsilon$, all other $r_{i\neq j,j^*} = 0$. Direct substitution into the recursion reveals that

$$r_j(t+1) < r_j(t)\,,$$

proving the assertion that $\delta_{j\neq j^*}$ is unstable.

Now consider $r_j = \delta_{jj^*}$ and let $r_{j^*} = 1-\varepsilon$ and $\sum_{j\neq j^*} r_j = \varepsilon$, $r_j > 0$. Substitution into the recursion yields the inequality:

$$r_{j^*} \geq r_{j^*}\Big(1 + G(S_{j^*}(R)\varepsilon - \varepsilon \max_{j\neq j^*}(S_j(R))\Big)\,.$$

By hypotheses if the maximum S_{j^*} is unique:

$$S_{j^*}(R) > \max_{j\neq j^*}(S_j(R))\,.$$

Hence, $r_{j^*}(t+1) > r_{j^*}(t)$ as required for stability of the point $r_j = \delta_{j^*j}$. It can be shown that if the maximum strength is not unique, any convex combination of the set of maximizing choices is stable.

It can also be shown that the equilibrium bifurcated routing patterns are the only stable bifurcated fixed points of the recursion. The method of proof involves linearizing the recursion about its fixed points and applying discrete time linear stability theory. We summarize the main result in the following proposition.

Proposition 7: Equilibrium routing pattern – stable fixed point equivalence. *Equilibrium routing patterns map uniquely into the stable fixed points of the recursion and vice versa. This is an important result for it shows that if the initial routing pattern is bifurcated in all of its routing variables, then the routing pattern will converge asymptotically to an equilibrium routing pattern.*

6. Numerical results

The network delay model and recursion have been implemented in software and a variety of numerical calculations were performed to investigate the transient as well as the stationary behavior of the proposed system. Only a small but representative sample of these results is included here.

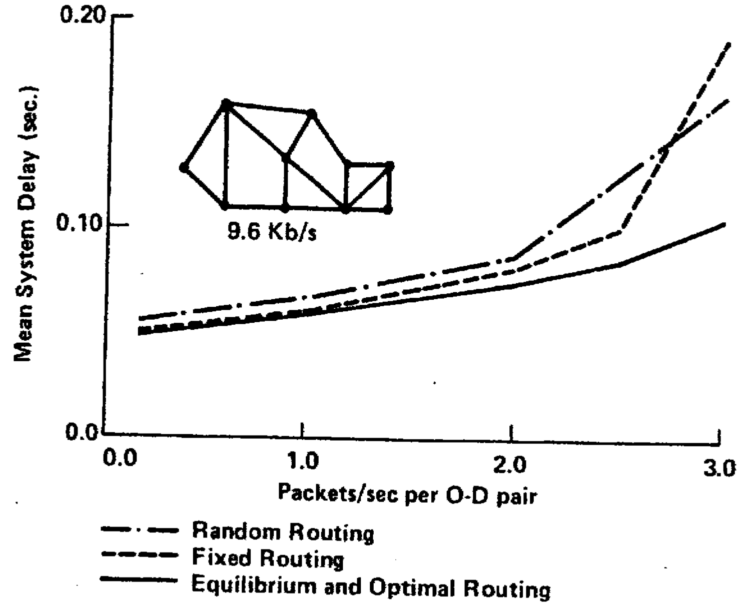

Fig. 1. Network performance.

The network topologies studied include the 10-node, 28-link network shown in Fig. 1 as well as two small subnetworks containing three, and four nodes, respectively. The 10-node network was selected since it has previously been used in studies of learning automata routing schemes [35]. This choice enabled certain comparisons to be made with our results. The small networks were selected to illustrate certain characteristics of the recursive scheme which can arise in the general case but were not evident in the 10-node networks studied. A range of traffic loads, both uniform and nonuniform, were tested. Various initial routing patterns and gain parameters were selected to exhibit the influence of these parameters of system behavior.

The first set of experiments attempted to reproduce the case studied by simulation in ref. [35] where a single nonzero demand λ_{19} was involved. System delay results for random routing, fixed minimum hop routing, equilibrium routing, and system optimum routing are shown in Table 1.

Some disparity in the results is to be expected since we have employed a single exponential distribution to model all packets, rather than two distinct constant packet sizes. The packet arrival rate was set at $\lambda_{19} = 21.48$ packets/second, with a mean packet size of 256 bits. This resulted in a traffic demand of 5500 bits/second.

The equilibrium and system optimal delays given for our model were those obtained after 25 iterations of the recursion. While there was some difference in the transient response for the user equilibrium and system optimal cases, the stationary values were essentially the same. Fig. 2 shows the transient response for the equilibrium case. We should mention that the system optimal curves were obtained from applications of our recursion to a model with the link costs modified as described in Section 5.

Table 1

Routing pattern	Average packet delay	Average message delay (35)
Fixed	250 ms	211.8 ms
Random	180 ms	193.0 ms
Equilibrium	144 ms	147.3 ms
Optimal	144 ms	–

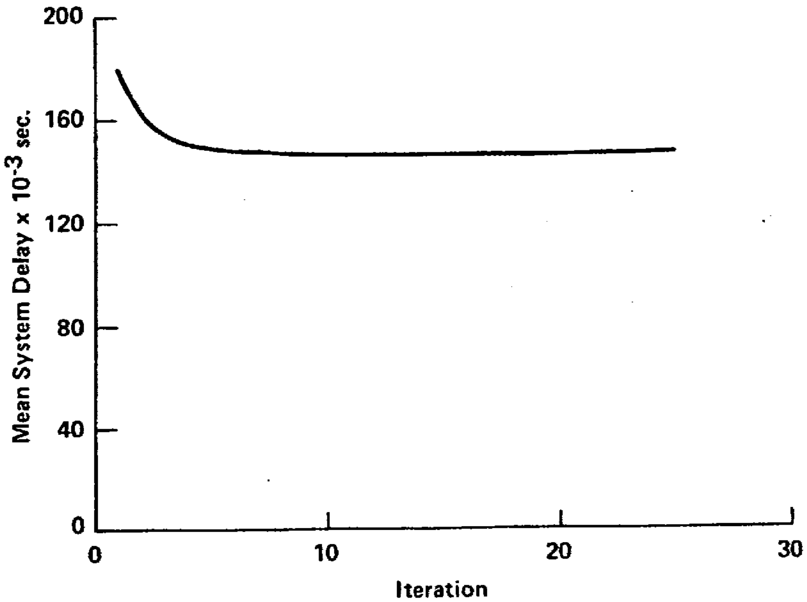

Fig. 2. Equilibrium routing transient response.

The next set of experiments was performed for a complete uniform traffic matrix. The demand per origin destination pair was varied over the range 0.1–3.0 packets/second. The system average delay versus load curves is displayed in Fig. 1 for the fixed, random, equilibrium, and system optimal routing. Again it is apparent that there is no significant difference in the delay performance of the equilibrium and the optimum routing patterns. Under light traffic conditions both schemes tend to minimum hop routing, while under moderate to heavy traffic the routing patterns are bifurcated and not minimum hop routing.

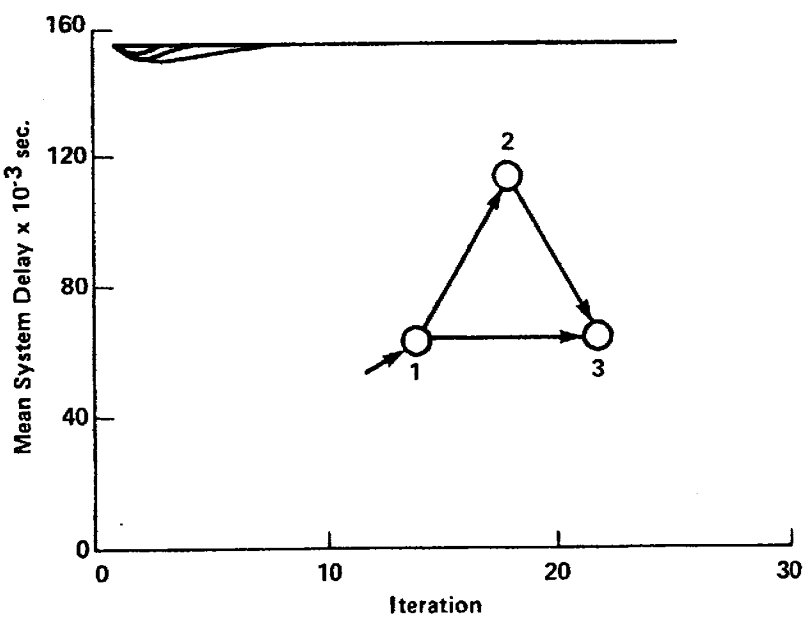

Fig. 3. Sub-optimal equilibrium routing.

To illustrate cases where the system optimal routing and the equilibrium routing differ significantly, an asymmetric three node network (shown in Fig. 3) was considered. Link capacity was set at 56 kb/second for all oriented links. The only nonzero demand is $\lambda_{13} = 90$ pkts/second with a mean packet length of 1024 bits. The initial routing matrix for commodity 3 was randomized, with each outgoing link being selected with a probability of 0.5.

The approach to equilibrium is shown in Fig. 3 for four different values of the gain parameter, namely $G = (0.0, 0.1, 0.2, 0.3)$. The effect of an increase in gain on the rate of convergence is apparent. For equilibrium routing, the convergence is not generally monotone. In this particular case the approach to equilibrium passed through the system optimal routing where delay is minimum and continued until the path delay was equalized for the direct and the alternate route. During the latter phase the system delay performance was deteriorating. Delay performance improved monotonically and exponentially for system optimization, as shown in Fig. 4, the final difference between optimal routing delay and equilibrium routing delay being of the order of 6%. The system optimum routing equalized the marginal delays on the direct and the alternate path which resulted in more traffic being carried on the alternate route.

The final set of experiments used the four node network shown in Fig. 5. These experiments were performed to assess the impact on convergence of an initial routing pattern which is a ***quasi-inflection*** point. A single nonzero traffic demand of $\lambda_{89} = 21.48$ packets/second was selected. The initial routing pattern for commodity 9 was as follows: $r^{(9)}_{89} = 0.9$, $r^{(9)}_{87} = 0.1$, $r^{(9)}_{79} = 0.01$, and $r^{(9)}_{7,10} = 0.99$.

The transient behavior of the equilibrium and optimal schemes is shown in Figs. 5 and 6, respectively. A plateau in the approach to the optimal routing pattern is apparent which is due to the quasi-inflection point previously mentioned. The equilibrium scheme shows an initial degradation in performance but ultimately improves performance to match that of optimal routing after 25 iterations. This particular example demonstrates the importance of initial routing pattern in regard to the transient response. In particular it suggests using an unbiased randomized pattern to avoid the quasi-inflection point conditions.

In the version of the algorithm used to produce the above results, there was no check for feasibility. In spite of this, in the vast majority of runs the algorithm did not violate the capacity constraints. There

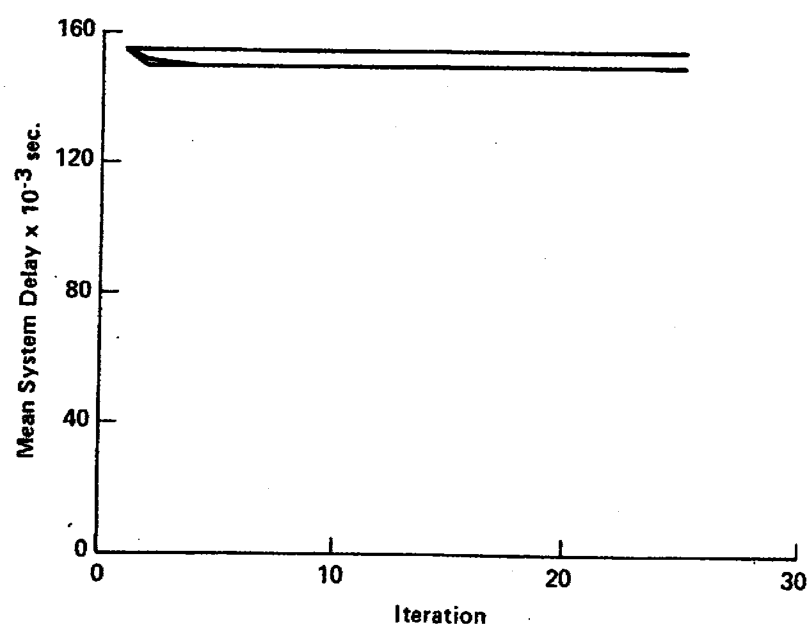

Fig. 4. System optimal routing transient response.

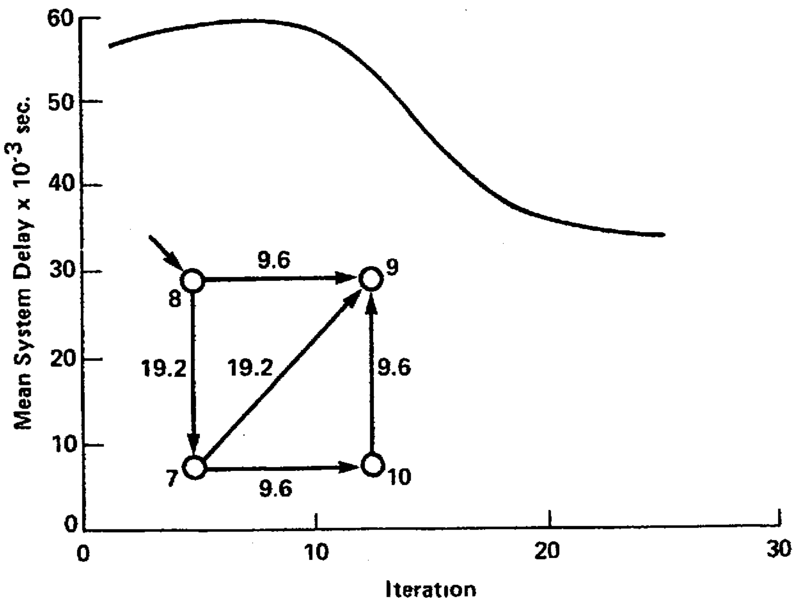

Fig. 5. Equilibrium routing transient response.

were exceptions, however, and in general it is necessary to reduce the step size parameter G sufficiently to avoid hopping over the capacity constraint barrier. While admissibility is only assured for $0 \leq G \leq 1$, it has been found experimentally that it is usually possible to increase the gain factor to 2 or 3 without violating admissibility or introducing instability. Larger gains resulted in oscillatory behavior which ultimately violated the feasibility or the admissibility constraints. This phenomenon should be studied in more depth as it can be used to accelerate convergence of the scheme. In all of the runs the gain parameter was fixed during the run, and equal for all nodes and commodities. The routing variables

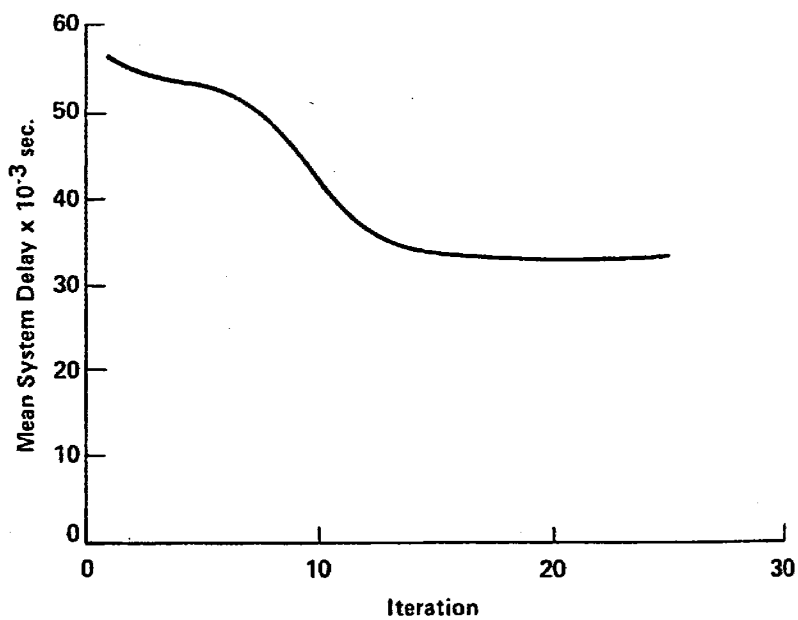

Fig. 6. System optimal routing transient response.

were updated in a synchronous fashion at all nodes. Recursive schemes with time, commodity, or node dependent gains, as well as asynchronous routing parameter updates, are areas which remain to be explored.

7. Implementations

The expected behavior of the cross-correlation learning algorithm under slow-learning conditions (small G) has the same form as the recursion described in Section 5 [18, 19]. Accordingly, this learning automaton scheme can be used to implement the recursion. Viswanathan and Narendra [38] have described a technique for extending the L_{RI} automaton to S-model environments by appropriately transforming the S-model responses into penalty strengths. It turns out that this results in the identical equations for the action probability updates as does the corss-correlation scheme which operates directly on the S-model responses. Accordingly, the recursion given in this paper models the behavior of the learning scheme reported in [24] and [35] under slow-learning conditions as well. Other implementations via sample mean automata are also possible, whereby several delay measurements are taken between routing updates and the sample mean delays are used to compute the updated routing pattern.

An alternative view of our recursion is, therefore, a model of the expected behavior of a class of learning automata routing schemes. The fixed points of the recursion correspond to the stationary distribution of the automata network combination. By using a sufficiently small G_i^d which is proportional to the mean nodal flow rate γ_i^d our recursion can be used to model the transient behavior of these learning schemes as well. This is felt to be a major contribution of this paper as our analytic approach is a cost-effective alternative to event simulation.

8. Conclusion

Analytic models for the equilibrium routing in store-and-forward networks have been described and properties of the equilibrium solution have been derived. A decentralized nonlinear recursion has been presented and its behavior has been characterized. The major result states that the set of equilibrium routing patterns corresponds to the stable fixed points of the recursion. By appropriately modifying the link delay function, the recursion has been used off-line to compute the system optimum routing pattern as well. This enables comparison in terms of mean system delay performance with the equilibrium routing pattern.

Numerical studies have been carried out which demonstrate the practicality of the approach. A number of learning automaton schemes provide a means of implementing the recursion and, conversely, the recursion can be used to model the expected transient and stationary performance of such schemes under slow-learning conditions. This provides a cost-effective alternative to Monte Carlo simulation when designing and evaluating such learning systems.

So far only fixed gains and synchronous updates have been studied numerically, although it is apparent that generalizations are possible. An area requiring further research involves the study of time-, node-, and traffic-dependent gain parameters as well as asynchronous updates in various orders. The effect of these variations on convergence and stability is the major point of interest.

Acknowledgements

The research was supported in part by NSERC grant A1344. The author also wishes to express his appreciation to students, G. Gauthier and T. Darveau who coded an earlier version of the model.

References

[1] R.G. Gallager and J. Golestaani, Flow control and routing algorithms for data networks, Proc. ICCC Atlanta, Ga. (Oct. 1980).

[2] M. Gerla and P.O. Nilsson, Routing and flow control interplay in computer networks, Proc. ICCC Atlanta, Ga. (Oct. 1980).

[3] H. Frank and W. Chou, Routing in computer networks, Networks 1 (1971) 99–112.

[4] J.M. Mcquillan, Routing algorithms for computer networks – a survey, NTC-77 28:1-1.

[5] L. Fratta, M. Gerla and L. Kleinrock, The flow deviation method: an approach to store-and-forward communication network design, Networks 3 (1973): 97–133.

[6] A. Segall, The modeling of adaptive routing in data-communication networks, IEEE Trans. on Com. COM-25 (1) (1977).

[7] R.G. Gallager, A minimum delay routing algorithm using distributed computation, IEEE Trans. on Com. COM-25 (1) (1977).

[8] F.H. Moss and A. Segall, An optimal control approach to dynamic routing in data communication networks, IEEE Trans on Automatic Control AC-27 (1982) 329–339.

[9] T.E. Stern, A class of decentralized routing algorithms using relaxation, IEEE Trans. on Com. COM-25 (10) (1977).

[10] M.A. Hall, Properties of the equilibrium state in transportation networks, Transportation Sci. 12 (3) (1978).

[11] S.C. Dafermos and F.T. Sparrow, The traffic assignment problem for a general network, J. N.B.S.-B. Mathematical Sci. 73B (2) (1969).

[12] J.M. McQuillan, I. Richer and E.C. Rosen, The new routing algorithm for the ARPANET, IEEE Trans. on Com. COM-28 (5) (1980).

[13] D.P. Bertsekas, A class of optimal routing algorithms for communication networks, ICCC Atlanta (Oct. 1980).

[14] D.P. Bertsekas and E.M. Gafni, Projection methods for variational inequalities with application to the traffic assignment problem, DARPA Report (Sept. 1980).

[15] G.J. Foschini and J. Salz, A basic dynamic routing problem and diffusion, IEEE Trans. on Com. COM-26 (3) (1978).

[16] L.G. Mason, Learning automata and telecommunications switching, Proc. of The Third Yale Workshop on the Applications of Adaptive Systems Theory, New Haven (June 1983).

[17] M. Schwartz and T.E. Stern, Routing techniques used in computer communication networks, IEEE Trans. on Com. COM-28 (4) (1980).

[18] L.G. Mason, Self-optimizing allocation systems, Ph.D. Thesis, Univ. of Saskatchewan, Saskatoon (1972).

[19] L.G. Mason, An optimal learning algorithm for S-model environments, IEEE Trans. Automatic Control (1973).

[20] L. Kleinrock, Communication Nets: Stochastic Message Flow and Delay (McGraw-Hill Book Co., New York, 1964).

[21] P.J. Burke, The output of a queueing system, Operations Research 4 (1966) 699–704.

[22] F. Baskett, F.M. Chandy, R.P. Muntz and F.G. Palacios, Open, closed and mixed networks of queues with different classes of customers, JACM 22 (1975) 248–260. April 1975.

[23] B. Gavish and S.L. Hantler, An algorithm for optimal route selection in SNA networks, IEEE Trans. on Commun. COM-31 (10) (1983).

[24] M.S. Chrystall and P. Mars, Adaptive routing in computer communication networks using learning automata, ICCC Atlanta (Oct. 1980).

[25] A. Segall, Optimal distributed routing for virtual line-switched data networks, IEEE Trans. on Commun. COM-27 (1) (1979).

[26] K.S. Narendra, A. Wright and L.G. Mason, Application of learning automata to telephone traffic routing and control, IEEE Trans. on Systems, Man and Cybernetics SMC-7 (11) (1977).

[27] J.G. Wardrop, Some theoretical aspects of road traffic research, Proc. Inst. Civ. Eng. Part II, I (1952) pp. 325–378.

[28] D.P. Bertsekas and E.M. Gafni, Projected Newton methods and optimization of multicommodity flows, Proc. 23rd Conference on Decision and Control, Orlando, Fla. (Dec. 1982).

[29] C.E. Agnew, Dynamic modeling and control of congestion-prone systems, Dept. Eng.-Econ. Systems, Stanford University, Stanford California, Tech. Report 10 (Jan. 1974).

[30] E.F. Wunderlich, L. Kaufman and B. Gopinath, The control of store and forward congestion in packet switching networks, Proc. ICCC, Atlanta, Ga. (Oct. 1980).

[31] D.P. Bertsekas, Dynamic behavior of shortest path routing algorithms for communication networks, IEEE Transactions on Automatic Control AC-27 (1982).

[32] B. Hajek and R.G. Ogier, Optimal dynamic routing in communication networks with continuous traffic, Proc. 23rd CDC, Orlando, Fla. (Dec. 1982).

[33] P.E. Sarachik, A dynamic alternate route strategy for traffic networks, Proc. IEEE Conference on Decision and Control, Orlando, Fla. (Dec. 1982).

[34] P. Chu, R. Boorstyn and A. Kershenbaum, A simulation study of a dynamic routing scheme, Proc. IEEE Conf. (1981).

[35] P. Mars, K.S. Narendra and M. Chrystal, Learning automata control of computer communications networks, Proc. Third Yale Workshop on Applications of Adaptive Systems Theory, New Haven (June 1983).

[36] K.S. Narendra and M.A.L. Thatachar, On the behavior of a learning automaton in a changing environment with application to telephone traffic routing, IEEE Trans. on Systems, Man, and Cybernetics SMC-10 (5) (1980).

[37] O. Nedzelnitsky, Learning automata routing in data communication networks, Proc. Third Yale Workshop on Applications of Adaptive Systems Theory, New Haven (June 1983).
[38] B.W. Boehm and R.L. Mobley, Adaptive routing techniques for distributed communications systems. Rand Corp. Memorandum RM-4781-PR, AD630271 (Feb. 1966).
[39] R.M. Glorioso, G.R. Gruenich and J.C. Dunn, Self-organization and adaptive routing for communication networks, Proc. EASCON, publication 69C31-AES (1969).