Economides, A.A., Ioannou, P.A. & Silvester, J.A.: **Decentralized adaptive routing for virtual circuit networks using stochastic learning automata**. *Proceedings IEEE Infocom 1988, the Conference on Computer Communications*, pp. 613-622, IEEE 1988.

DECENTRALIZED ADAPTIVE ROUTING FOR VIRTUAL CIRCUIT NETWORKS USING STOCHASTIC LEARNING AUTOMATA*

Anastasios A. Economides, Petros A. Ioannou and John A. Silvester

Electrical Engineering-Systems Department
University of Southern California, SAL 300
Los Angeles, CA 90089-0781

Abstract

The problem of routing virtual circuits according to dynamic probabilities in virtual circuit packet switched networks is considered. Queueing network models are introduced and performance measures are defined. A decentralized asynchronous adaptive routing methodology, based on learning automata theory is presented. Every node in the network has a stochastic learning automaton as a router for every destination node. The routing probabilities that are assigned to the network paths are updated asynchronously on the basis of current network conditions. A new learning algorithm suitable for routing is used. Some initial simulation experiments, for a simple network, show convergence to optimal routing.

I. INTRODUCTION

The emergence of Integrated Services Digital Networks (ISDN) where the transmitted information is data, voice and images, as well as the need for interactive communication among users has lead to packet switched networks.

In packet switched networks, information is stored and transmitted from node to node in small groups of bits called packets. In other words, a message (data, voice or image) is decomposed into several packets, each with its own control and identification information, which are transmitted independently.

Packet switched networks can be operated using two main switching techniques - datagram and virtual circuit. In datagram networks, each packet is treated as a separate entity and may be routed differently from other packets belonging to the same message. Although this provides a very flexible way of information transfer, it is best suited for data, since packets may arrive out of order incurring resequencing delays, which cause problems in voice transmission. In virtual circuit networks, a call set-up packet, which may be part of the first packet of a message, initiates the establishment of a virtual circuit path from source to destination (Fig.1). All other packets belonging to this message follow the same route which remains fixed for the duration of the call. In this way, a virtual circuit provides a reliable logical channel with packets delivered in order.

Whether routing individual packets in datagram networks or the virtual circuits in virtual circuit networks, good route selection is very important for efficient communication and better network resource utilization. So it is not surprising that the routing problem in packet switched networks has received considerable attention, for example [1, 4, 7].

Routing can be classified according to how dynamic the route selection process is : 1) static, 2) quasi-static, and 3) adaptive or dynamic. In *static routing*, the route selection is independent of the current network conditions. Static routing algorithms can be : a) fixed, when there is a predetermined set of alternate paths between the source and the destination and all traffic arriving during the same time period follows the same route; b) random, where the traffic is split to several routes according to fixed probabilities. In *quasi-static routing*, route selection depends partially on the current network conditions but some network parameters are assumed to be stationary over time. In *adaptive routing,* route selection depends on the current network conditions (topology changes, traffic conditions). Since in reality, network conditions change over time the decisions in adaptive routing can be better and more accurate.

The most accurate representation of a real network is to consider a completely dynamic network environment where the conditions continually change over time. Analysis of adaptive routing algorithms has been attempted by several researchers, eg. [1]. The common problem is that the routing probabilities of selecting a specific route and hence the transition probabilities in a Markov model depend on the network state. The resulting Markov Chain does not have a product form solution and we are forced to use simulation or approximations.

*This work was supported by the National Science Foundation under Grant DMC-8452002 together with matching funds from AT & T Information Systems and a Greek State Fellowship.

Even in the simplest case of *datagram packet switched* networks, where packets are individually routed, approximations are needed. In order to deal with the difficulties of the system dynamics, many restrictive assumptions are used, such as: i) the "Quasi-Static" assumption [1]- that the external arrival traffic is stationary over time -, ii) the "Fast Settling Time" assumption - that after a new decision is made, the flows of traffic settle down to their new values instantaneously. Optimization techniques and flow approximations [1, 4 - 9, 14] are common approaches. The traffic follows the shortest path with respect to link lengths that depend on the flows carried by the links [1].

Ephremides, Varaiya and Walrand [4] considered a single node, with one incoming and two similar outgoing links. They proved that for exponential packet lengths: i) where the number of packets in each outgoing link is known, then the "send to shortest queue" policy is optimal and ii) where the initial queue lengths are known and equal and no further observations are made, then the "round-robin" policy is optimal.

Using flow dependent routing decisions, Gallager [7] shows that optimal routing is achieved by balancing the incremental (or marginal) delays on the routing paths.

The adaptive routing problem in *virtual circuit packet switched* networks is even more difficult. The difficulties arise because the routing decisions are made at virtual circuit arrival instants. Subsequently, no control is exercised over the packet routing process, i.e. the network state is affected for longer periods than in datagram networks. The system dynamics occur at two different time scales. The fundamental point is that although the control is exerted at the slower time scale, where the virtual circuit establishment / termination process occurs, the network performance is measured at the faster time scale, where the packet transport process occurs. Previous work on this problem [5, 6, 8, 14] assumes either independence of the virtual circuit establishment / termination process and packet transport process or limiting process rates.

For a comprehensive analysis and more precise control of the network, both the packet and virtual circuit levels should be considered. The fundamental question is how to appropriatelly characterize the network state such that optimal routing decisions can be made. In this paper, the available information about the network condition can be the current number of packets, the current number of virtual circuits or the current routing probabilities. Performance measures such as the unfinished work on a path, increase in the number of packets on a path due to the addition of a new virtual circuit on this path, and increase in the average packet delay on a path due to the addition of a new virtual circuit on this path are proposed and considered as path costs.

An interesting approach to the routing problem, is the use of learning automata [2, 10 - 13]. In these learning control algorithms, telephone call blocking [11, 13] and packet delay [2, 10, 12] are used to characterize the network state. Although slow, these adaptive algorithms are suitable for the control of very complex and random environments, such as a communication network.

In this paper we apply the stochastic learning automaton methodology to the routing problem in virtual circuit networks (Fig.3). A virtual call originating at a source node has to be routed through a path among many alternate paths to a destination node. At each intermediate node, a router probabilistically selects the best possible path to this destination node (Fig.2). The greatest potential of the learning automata methodology is that it permits the analysis of very complex dynamic systems, and global optimization is possible. Even when little information is available, they tend to stabilize a nonstationary system by predicting its behavior. A routing probability updating scheme is proposed that adapts itself to the network conditions so that, in steady state, the costs on the paths are equalized.

The paper is organized as follows : In section II, we introduce the Network Model. In section III, we propose a new stochastic learning automaton algorithm and apply it to the routing problem. In section IV, we describe simulation results that show the behavior of the proposed algorithm. In section V, we make improvements to the routing probabilities updating scheme and we define other performance measures that can be used by the virtual circuit routing algorithm to minimize the number of packets in the network or the overall network average packet delay. Finally, in section VI, we draw some conclusions on the approach and discuss extensions to the global optimization problem in decentralized adaptive virtual circuit routing algorithms.

II. NETWORK MODEL

The network is modeled as a directed graph G=(V,E), where V is the set of nodes and E is the set of directed links (Fig.1). At every network node k, a router $R_{[kd]}$ (or $R_{[skd]}$ if the virtual circuits are distinguished with regard to their source node s), selects the best possible path among $\Pi_{[kd]}$ alternate paths to the destination node d (Fig.4). The following assumptions are made for the performance evaluation of the virtual circuit packet switched network :

1. The packet interarrival times per virtual circuit are exponentially independent identically distributed (iid) with mean $1/r$.
2. The packet service times at link.ij are iid exponentially distributed with mean $1/\mu_{ij}$.
3. The virtual circuit interarrival times generated at node s with destination node d are iid exponentially distributed with mean $1/\gamma_{[sd]}$. The virtual circuit interrarival times entering the network are

iid exponentially distributed with mean $1/\gamma_o$, where $\gamma_o = \sum_s \sum_d \gamma_{[sd]}$.

4. The virtual circuit interarrival times either generated at node k or passing through node k with destination node d, are iid exponentially distributed with mean $1/\nu_{[kd]}$.

5. The virtual circuit durations are iid exponentially distributed with mean $1/\delta$.

6. There is enough buffering in every link to accomodate all packets.

7. All virtual circuits in a link can be served simultaneously.

8 The order in which packets and virtual circuits are served does not depend on their processing times and there are no priorities.

9. The virtual circuit set up time is negligible, since the first data packet is a path finder.

10. The processing delays within nodes are ignored.

Few analytical approaches to the routing problem in virtual circuit packet switched networks exist, [1, 5, 6, 8, 14], due to the complexity of the interaction between the virtual circuit and packet processes. For a complete analysis of the network, we must incorporate packet and virtual circuit metrics into the model simultaneously.

Consider a network node k that routes traffic to a destination node d and the embedded process at the arrival instants of virtual circuits to node k with the following definitions:

n_k : time instant of the arrival of the n^{th} virtual circuit at node k, (either generated at node k or passing through node k). From now on, we drop the subscript k and write n for n_k.

$V_o(n)$: total number of active virtual circuits in the network at time n.

$V_{ij}(n)$: number of virtual circuits using link.ij, at time n.

$N_{ij}(n)$: number of packets present in the queue (including those being transmitted, if any) on link.ij, at time n.

$\lambda_{ij}(n)$: packet arrival rate at link.ij at time n.

$q_{ij}(n)$: packet arrival rate at link.ij over total packet arrival rate entering the network at time n.

$T_{ij}(n)$: packet delay at link.ij at time n.

$U_{ij}(n)$: unfinished work on link.ij at time n.

$\Delta_\pi N_{ij}(n)$: increase of the number of packets on link.ij, due to the addition of a new virtual circuit on path.π at time n.

$\Delta_\pi(q_{ij}(n) * T_{ij}(n))$: increase of the portion of the overall network packet delay corresponding to link.ij, due to the addition of a new virtual circuit on path.π at time n.

$P_{\pi(kd)}(n)$: probability of routing an arriving virtual circuit at node k and destined for node d through path. $\pi(kd)$ at time n.

III. A STOCHASTIC LEARNING AUTOMATON AS A ROUTER

Since the network conditions change over time, route selection should track the changing conditions. Adaptive control algorithms that adjust the route selection to the current network conditions are needed. Therefore, we consider learning automata as routers, since they can operate in an asynchronous and decentralized mode.

The proposed adaptive routing algorithms are based on a "Probabilistic Selection of the Minimum Cost Path" idea. Instead of using a definitive decision as to where to send a newly arriving virtual circuit, we vary the path routing probabilities favoring the minimum cost path. In a rapidly changing system, even if we have some information about the system state at a time instant, this does not ensure that the same will hold for the next time instant. Also, since the information about the network condition needs some time to be transferred through the network, it may be obsolete when it is used by the router. A learning algorithm tends to the optimum path selection using past experience and attempts to predict future system behaviour.

The automaton selects action $a(n) = a_i$ with probability $P_i(n)$ at each instant n. Action $a(n)$ becomes input to the environment (Fig. 3). If this results in a favorable outcome for the network performance ($X(n) \to 0$), then the probability $P_i(n)$ is increased by $\Delta P_i(n+1) = P_i(n+1) - P_i(n)$ and the $P_j(n), j \neq i$, are decreased by $\Delta P_j(n+1) = P_j(n+1) - P_j(n)$. Otherwise, if an unfavorable outcome ($X(n) \to 1$) appears, then the $P_i(n)$ is decreased by $\Delta P_i(n+1) = P_i(n+1) - P_i(n)$ and and the $P_j(n), j \neq i$ are increased by $\Delta P_j(n+1) = P_j(n+1) - P_j(n)$.

Since communication networks are nonstationary environments, their state varies with time, but little work has been done on learning automata in nonstationary environments. Narendra and Thathachar [11], as well as Srikantakumar and Narendra [13] consider that the conditional probability of an unfavorable outcome when selecting a specific action is affected by the action probabilities. They show that either the telephone call blocking probabilities or the rates of the call blocking probabilities on the network links can be equalized in steady state.

Every network node k has a router for every destination node d that routes new arriving virtual circuits at node k (either genetated or passing through it) and destined for node d. These routers operate asynchronously and base their decision on the current network condition. The actions, $\alpha(n)$ of this router $R_{[kd]}$ are to select some particular path.$\pi(kd)$.

We define the cost for path.$\pi(kd)$ at time n to be the unfinished work on path.$\pi(kd)$:

$$\mathbf{C}_{\pi(kd)}(n) = U_{\pi(kd)}(n) = \sum_{\forall ij \in \pi(kd)} U_{ij}(n) = \sum_{\forall ij \in \pi(kd)} \frac{N_{ij}(n) + (r/\delta) * V_{ij}(n)}{\mu_{ij}} \quad (1)$$

where $\pi(kd) = 1(kd), ..., \Pi_{[kd]}(kd)$.

Note that the number of packets per virtual circuit is geometrically distributed (memoryless), with mean r/δ.

We propose the following adaptive algorithm at every network node k, for routing virtual circuits to a certain destination node d.

Probabilistic Selection of the Minimum Cost Path :

Suppose path.p was selected at time n-1, with $P_p(n-1)$.

Collect available traffic statistics.

Compute $\mathbf{C}_\pi(n)$ $\forall\pi, \pi$ *is a path from* k *to* d.

Set $\mathbf{C}_{Max}(n) = \max_\pi\{\mathbf{C}_\pi(n)\}$.

Set $X_\pi(n) = \frac{\mathbf{C}_\pi(n)}{\mathbf{C}_{Max}(n)}$ $\forall\pi$.

Set $X(n) = X_p(n)$.

Update the routing probabilities $P_\pi(n)$ $\forall\pi$, *(see below).*

Select the path for the n^{th} *virtual circuit probabilistically according to* $P_\pi(n)$.

Next, the routing probabilities updating scheme is described. At first, the selected path cost is compared to the cost of the other paths. If the selected path.p has the minimum cost (favorable outcome) among all alternative paths, then its routing probability increases by $\Delta P_p(n)$ and the routing probabilities of the other paths are decreased. Otherwise (unfavorable outcome), its routing probability is decreased by $\Delta P_p(n)$ and the routing probabilities of the other paths are increased. More specifically:

case 1: The selected path.p has the smallest cost

Then its routing probability increases in proportion to how small its cost was, i.e. if its cost was small, then its routing probability increase should be large. The routing probability should also depend on how small its previous routing probability was, i.e. if its previous routing probability was small, then its routing probability increase should be large. Therefore, we use $\Delta P_p(n) = \alpha * [1 - X(n)] * [1 - P_p(n)]$ $\quad 0 < \alpha < 1$.

Since we increase the routing probablity of the selected path, we decrease the routing probabilities of all other paths.

case 2: The selected path.p does not have the smallest cost

Then its routing probability decreases in proportion to how large its cost was, i.e. if its cost was large, then its routing probability decrease should be large. The routing probability should also depend on how large its previous routing probability was, i.e. if its previos routing probability was large, then its routing probability decrease should be large. Therefore, we use $\Delta P_p(n) = -\beta * X(n) * P_p(n)$ $\quad 0 < \beta < 1$.

Since we increase the routing probablity of the selected path, we should decrease the routing probabilities of all other paths.

Implementation of these concepts gives the following algorithm.

Stochastic Learning Automaton Updating Scheme

If $\quad X_p(n) = \min_\pi\{X_\pi(n)\}$, then

$$P_p(n) = P_p(n-1) + \alpha * [1 - X(n)] * [1 - P_p(n-1)]$$

$$P_\pi(n) = P_\pi(n-1) - \alpha * [1 - X(n)] * P_\pi(n-1) \quad \forall\pi \neq p$$

else

$$P_p(n) = P_p(n-1) - \beta * X(n) * P_p(n-1)$$

$$P_\pi(n) = P_\pi(n-1) + \beta * X(n) * [\frac{1}{\Pi - 1} - P_\pi(n-1)] \quad \forall\pi \neq p$$

where $\quad 0 < \alpha, \beta < 1$.

IV. SIMULATION RESULTS

In a simulation experiment we considered a single source node with two similar alternate paths to the same destination node (Fig. 2). Taking the unfinished work on each path as the response of the network, we applied the proposed adaptive routing algorithm for 10,000 virtual circuit arrivals. Our first aim was to investigate the behavior of learning automata algorithms and the effect of different values of α and β on the routing.

For initial routing probabilities $P_1(0) = P_2(0) = 0.5$; mean virtual circuit interarrival time $1/\nu = 1,000$ msecs;

mean packet interarrival time in the virtual circuits $1/r=$ 200 msecs; service time of packets $1/\mu = 50$ msecs; and mean virtual circuit duration $1/\delta = 4,000$ msecs; we noticed that large values of α and β, may be preferable (Fig.5, Fig.6, Fig.7, Fig.8, Fig.9, Fig.10, Fig.11). This happens because increasing the values of α and β, increases the adaptation speed to the network conditions, at the expense of the adaptation variance (although the simulation results show little fluctuation).

The mean packet delay is minimized for large values of α and β, and converges to a stable value of 750msecs (Fig.5). Large values of α and β exhibit more rapid convergence, without affecting the convergence variance too much.

The expected unfinished work on the two paths tend to be equal for all values of α and β, although large values of α and β work better (Fig.6, Fig.7). Also the absolute difference of the expected unfinished work (Fig.8, Fig.9), of the number of packets (Fig.10), and of the number of virtual circuits (Fig.11) on the two paths tend to zero for all values of α and β, although large values of α and β are better.

A routing algorithm may theoretically converge to optimality, but it is crucial how fast and how much instability is introduced. In the simulation results, we see that we can control the adaptation speed of the learning algorithm without introducing instability.

V. IMPROVEMENTS, EXTENSIONS AND GENERALIZATIONS

The optimal routing problem is a hard problem, since the decisions should adjust fast to track rapidly time varying traffic patterns, but simultaneously maintain stability. Next, we consider some improvements on the stochastic learning automaton updating scheme and on the performance measures used.

V.1 STOCHASTIC LEARNING AUTOMATON UPDATING SCHEME

1. Since in adaptive routing algorithms, there is a need to "react to large changes quickly and to small changes slowly", we can use different rates of adaptation for different network conditions. If the network cost is far away from the minimum, the algorithm should converge faster, while if the network cost is near to the minimum possible the algorithm should have smaller fluctuation.

This requirement can be incorporated into the reinforcement scheme by employing different parameters for different network responses ($X(n)$). We split the response possibilities into several regions, see below. Whenever the cost of the selected path is very small compared to the other paths (region 1), then its routing probability should increase very fast ($\alpha \to 1$). When the cost of the selected path is the smallest but is close to the other paths (region A), then its routing probability should increase slowly ($\alpha \to 0$). Correspondingly, whenever the cost of the selected path is very bad (region A+B), then its routing probability should decrease very fast ($\beta \to 1$). When the cost of the selected path is slightly greater than the smallest (region A+1), then its routing probability should decrease slowly ($\beta \to 0$).

Region.1 $\quad X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\} - \phi_1$

Region.2 $\quad \min_{\pi \ne p}\{X_\pi(n)\} - \phi_1 < X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\} - \phi_2$

....

Region.A $\quad \min_{\pi \ne p}\{X_\pi(n)\} - \phi_{A-1} < X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\}$

Region.A+1 $\quad \min_{\pi \ne p}\{X_\pi(n)\} < X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\} + \theta_{B-1}$

Region.A+2 $\quad \min_{\pi \ne p}\{X_\pi(n)\} + \theta_{B-1} < X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\} + \theta_{B-2}$

....

Region.A+B $\quad \min_{\pi \ne p}\{X_\pi(n)\} + \theta_1 < X_p(n) \le \min_{\pi \ne p}\{X_\pi(n)\}$

where $\quad 1 > \phi_1 > \phi_2 > ... > \phi_A > 0,$

$1 > \theta_1 > \theta_2 > ... > \theta_B > 0$, are constants.

A possible sequence for these numbers $\{\phi_i\}$ and $\{\theta_j\}$ could be a Fibonacci sequence (normalized to the (0,1) interval).

Therefore, in each of the above regions, we can use different $\alpha_i, i = 1, ..., A$ (favorable outcome), and $\beta_j, j = 1, ..., B$, (unfavorable outcome), with $1 > \alpha_1 > \alpha_2 > ... > \alpha_A > 0$, and $1 > \beta_1 > \beta_2 > ... > \beta_B > 0$.

2. In the simulation experiment, collection of the available statistics and update of the routing probabilities were done only at virtual circuit arrival instants. But this does not permit fast adaptation of the routing decisions to the network state. An idea for speeding the adaptation rate is to collect the available statistics at virtual circuit arrival instants, but update the routing probabilities more often. These multiple updates will be based on the single measurement at the virtual circuit arrival instants. Note that the updating scheme is composed of recursive equations. This leads us to extend the previously proposed updating sheme by using one network state measurement, but many (for example l_i in region.i) iterations of the scheme in one actual computing step (updating step from *n-1* to *n*). For clarity we show the transformation of only one network response region (the full detail is given in the appendix).

If $X_p(n) \leq \min_{\pi \neq p}\{X_\pi(n)\} - \phi_1$, then

$$P_p(n) = P_p(n-1) * \{1 - \alpha_1 * [1 - X(n)]\} + \alpha_1 * [1 - X(n)]$$

$$P_\pi(n) = P_\pi(n-1) * \{1 - \alpha_1 * [1 - X(n)]\} \quad \forall \pi \neq p$$

Since the measurements for P_p, P_π and X do not change between $n-1$ and n, call them P_p, P_π and X. By solving these recursive equations, we have the following equations

$$P_p(l_1) = P_p * \{1 - \alpha_1 * [1 - X]\}^{l_1} + \alpha_1 * [1 - X] * \sum_{i=0}^{l_1-1} \{1 - \alpha_1 * [1 - X]\}^i$$

$$P_\pi(l_1) = P_\pi * \{1 - \alpha_1 * [1 - X]\}^{l_1} \quad \forall \pi \neq p$$

The updating scheme becomes

If $X_p(n) \leq \min_{\pi \neq p}\{X_\pi(n)\} - \phi_1$, then

$$P_p(n) = P_p(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} + \alpha_1 * [1 - X(n)] * \sum_{i=0}^{l_1-1} \{1 - \alpha_1 * [1 - X(n)]\}^i$$

$$P_\pi(n) = P_\pi(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} \quad \forall \pi \neq p$$

We can use different $l_i, i = 1, ..., A$ and $m_j, j = 1, ..., B$ for different regions, where $l_1 > l_2 > ... > l_A > 0$, and $m_1 > m_2 > ... > m_B > 0$, are positive integers.

3. Another improvement of the adaptation speed, is to update the routing probabilities more often (eg. at every packet arrival instant). Then the routing algorithm will track the network state faster and the decisions will be better. Of course this will introduce more overhead of transmitting, selecting, storing and computing the traffic statistics.

4. Note that we have incorporated the normalized cost of the selected path $X(n)$ in the routing probability updating scheme. Although this concept produces a fairer routing probability updating scheme, it also slows the adaptation rate. In order to speed up the rate of convergence of the algorithm, we could simply set $X(n) = 1$ for a good decision and $X(n) = 0$ for a bad decision, but this might cause instability.

V.2 AVAILABILITY OF MEASURES

In order to investigate how much of the information about the network state is important for optimum routing, we consider three cases regarding the availability of the traffic measurements.

a) Measurements of the current number of active packets and virtual circuits on each path are possible

In equation 1, we use the actual measurements for $N_{ij}(n)$ and $V_{ij}(n)$.

b) Measurements of the current number of active virtual circuits on each link are possible, but the number of packets is unknown

Since the number of packets is unknown, we estimate the average number of packets on link.ij at time n, by modeling each link.ij at the packet level as an M/M/1 queue, with packet arrival rate $\lambda_{ij}(n) = r * V_{ij}(n)$, and $r * V_{ij}(n) < \mu_{ij}$

$$\bar{N}_{ij}(n) = \frac{r * V_{ij}(n)}{\mu_{ij} - r * V_{ij}(n)}$$

and the expected unfinished work is

$$\bar{U}_{\pi(kd)}(n) = \sum_{\forall ij \in \pi(kd)} \frac{\bar{N}_{ij}(n) + (r/\delta) * V_{ij}(n)}{\mu_{ij}} \tag{2}$$

c) No measurements of the traffic are possible

Here we must also estimate the number of virtual circuits. By modeling each link.ij at the virtual circuit level as an M/M/∞ queue, with virtual circuit arrival rate

$$\nu_{ij}(n) = \sum_{\forall k} \sum_{\forall d} \sum_{\forall \pi(kd), ij \in \pi(kd)} \nu_{[kd]}(n) * P_{\pi(kd)}(n)$$

the average number of virtual circuits on link.ij at time n is estimated to be

$$\bar{V}_{ij}(n) = \frac{\nu_{ij}(n)}{\delta}$$

Then the average number of packets on link.ij at time n is estimated to be

$$\bar{\bar{N}}_{ij}(n) = \frac{r * \nu_{ij}(n)}{\delta * \mu_{ij} - r * \nu_{ij}(n)}$$

and the expected unfinished work is

$$\bar{\bar{U}}_{\pi(kd)}(n) = \sum_{\forall ij \in \pi(kd)} \frac{\bar{\bar{N}}_{ij}(n) + (r/\delta) * \bar{V}_{ij}(n)}{\mu_{ij}} \tag{3}$$

V.3 ALTERNATE PERFORMANCE OBJECTIVES

The performance measure used in section IV is user oriented. For overall network optimal routing, the routing should try to equalize the derivative of the appropriate cost measure [1, 7]. Next, we consider the following objective functions :

1) Minimize the total number of packets in the network

$$min \lim_{n\to\infty} N(n) = min \lim_{n\to\infty} \sum_{\forall ij} N_{ij}(n)$$

Thus we define the cost for path.$\pi(kd)$ at time n to be the increase in the number of packets on path.$\pi(kd)$ due to the addition of a new virtual circuit on path.$\pi(kd)$ at time n :

$$\Delta N_{\pi(kd)}(n) = \sum_{\forall ij\in\pi(kd)} \Delta_\pi N_{ij}(n) \qquad (4)$$

2) Minimize the overall network average packet delay

$$min \lim_{n\to\infty} T(n) = min \lim_{n\to\infty} \sum_{\forall ij} q_{ij}(n) * T_{ij}(n)$$

Thus we define the cost for path.$\pi(kd)$ at time n to be the increase in the portion of the overall network average packet delay corresponding to path.$\pi(kd)$, due to the addition of a new virtual circuit on path.$\pi(kd)$ at time n :

$$\Delta T_{\pi(kd)}(n) = \sum_{\forall ij\in\pi(kd)} \Delta_\pi (q_{ij}(n) * T_{ij}(n)) \qquad (5)$$

ΔN and ΔT can be estimated from measurements or by analytical models as was done for the unfinished work in the previous section.

V.4 OTHER EXTENSIONS

In the above analysis, it is straightforward to also include the propagation and processing delays. Some approximations that would reduce the control traffic overhead can also be made :

Approximation 1 : Instead of modeling each link.ij as a queue, an approximation is to model each path.$\pi(kd)$ as an M/M/1 queue at the packet level, with packet arrival rate $\lambda_{\pi(kd)}(n) = r * V_{\pi(kd)}(n)$, where $V_{\pi(kd)}(n)$, is the number of virtual circuits using path.$\pi(kd)$ at time n, and service rate $\mu_{\pi(kd)}$, and as an M/M/∞ queue at the virtual circuit level, with virtual circuit arrival rate $\nu_{\pi(kd)}(n) = \sum \nu_{[kd]}(n) * P_{\pi(kd)}(n)$.

Approximation 2 : Instead of considering the traffic on the paths from node k to destination node d, an approximation will be to consider only the traffic on the outgoing links of node k.

Remark : The above analysis permits us to find $P^{opt}_{\pi(kd)}(n)\ \forall\pi$ at every time n. For medium or heavy traffic, all the alternative paths from k to d will be used. Optimal routing is achieved if the first derivative lengths of these paths is equal. The proposed learning algorithm will learn these $P^{opt}_{\pi(kd)}(n)$ by itself.

VI. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have presented *analytical models* of packet switched networks operating in virtual circuit mode, and a *decentralized asynchronous adaptive routing methodology* that uses queueing performance measures as the network response to learning automata actions. We have proposed a new *learning reinforcement scheme* and a class of flexible adaptive routing algorithms based on a "Probabilistic Selection of the Minimum Cost Path" concept.

The strength of our approach to the routing problem is that it is very suitable for extremely complex dynamic systems such as a communication network with many nodes and unpredictable behavior. The proposed decentralized asynchronous adaptive routing algorithms are simple and easy to implement. The routing decision at every node tends to equalize the costs on the network paths and requires small overhead of cpu-time, network bandwidth and buffer storage.

Further theoretical and simulation research is needed regarding the convergence conditions of the proposed routing algorithm. We are also working on comparing the proposed measures on this simple network. We are also interested in evaluating how much of the available information about the network state is useful for the proposed routing methodology.

APPENDIX

Stochastic Learning Automaton Updating Scheme

If $X_p(n) \le \min_{\pi\neq p}\{X_\pi(n)\} - \phi_1$, then

$$P_p(n) = P_p(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} + \alpha_1 * [1 - X(n)] * \sum_{i=0}^{l_1-1} \{1 - \alpha_1 * [1 - X(n)]\}^i$$

$$P_\pi(n) = P_\pi(n-1) * \{1 - \alpha_1 * [1 - X(n)]\}^{l_1} \quad \forall\pi \neq p$$

If $\min_{\pi \neq p} \{X_\pi(n)\} - \phi_1 < X_p(n) \leq \min_{\pi \neq p}\{X_\pi(n)\}$, then

$$P_p(n) = P_p(n-1) * \{1 - \alpha_2 * [1 - X(n)]\}^{k_2} + \alpha_2 * [1 - X(n)] * \sum_{i=0}^{k_2 - 1} \{1 - \alpha_2 * [1 - X(n)]\}^i$$

$$P_\pi(n) = P_\pi(n-1) * \{1 - \alpha_2 * [1 - X(n)]\}^{k_2} \quad \forall \pi \neq p$$

If $\min_{\pi \neq p} \{X_\pi(n)\} < X_p(n) \leq \min_{\pi \neq p}\{X_\pi(n)\} + \theta_1$, then

$$P_p(n) = P_p(n-1) * [1 - \beta_2 * X(n)]^{m_2}$$

$$P_\pi(n) = P_\pi(n-1) * [1 - \beta_2 * X(n)]^{m_2} + \frac{\beta_2 * X(n)}{\Pi - 1} * \sum_{i=0}^{m_2 - 1} [1 - \beta_2 * X(n)]^i \; \forall \pi \neq p$$

If $\min_{\pi \neq p} \{X_\pi(n)\} + \theta_1 < X_p(n)$, then

$$P_p(n) = P_p(n-1) * [1 - \beta_1 * X(n)]^{m_1}$$

$$P_\pi(n) = P_\pi(n-1) * [1 - \beta_1 * X(n)]^{m_1} + \frac{\beta_1 * X(n)}{\Pi - 1} * \sum_{i=0}^{m_1 - 1} [1 - \beta_1 * X(n)]^i \; \forall \pi \neq p$$

ACKNOWLEDGMENTS

The authors wish to thank V.Saksena of AT & T Bell Laboratories for suggesting the problem.

REFERENCES

1. Bertsekas, D. and Gallager, R. *"Data Networks"* Prentice Hall, 1987.
2. Chrystal, M. S. and Mars, M. *"Adaptive Routing in Computer Communication Networks Using Learning Automata"* Proc. IEEE Nat. Telecomm. Conf. 1981.
3. Epelman, M. and Gersht, A. *"Asymptotic Qualitative Analysis of a link imbedded into a Packet Switched Virtual Circuit Network"* Proc. IEEE Infocom 84 Conf.
4. Ephremides, A. ,Varaiya, P. and Walrand, J. *"A Simple Dynamic Routing Problem"* IEEE Trans. on Automatic Control, vol AC-25, no 4, Aug. 1980.
5. Gafni, E. M. and Bertsekas, D. P. *"Path Assignment for Virtual Circuit Routing"* Proc. ACM SIGCOMM 1983 Conf.
6. Gafni, E. M. and Bertsekas, D. P. *"Asymptotic Optimality of Shortest Path Routing Algorithms"* IEEE Trans. on Information Theory, vol IT-33, no 1, Jan. 1987.
7. Gallager, R. G. *"A Minimum Delay Routing Using Distributed Computation"* IEEE Trans. on Communications, vol COM-25, no 1, Jan. 1977.
8. Gersht, A. *"Analytical Model of Dynamic Routing in Virtual Circuit Packet Switching Network"* Proc. IEEE ICCC 1982 Conf.
9. Maglaris. B. S. *"An Optimal Local Policy for Two-Level Adaptive Routing in Computer Networks"* Proc. IEEE Infocom 84 Conf.
10. Mars, P., Narendra, K. S. and Chrystal, M. *"Learning Automata Control of Computer Communications Networks"* Proc. Third Yale Workshop on Applications of Adaptive Systems Theory, 1983.
11. Narendra, K. S. and Thathachar, M. A. L. *"On the behavior of a Learning Automaton in a Changing Environment with Application to Telephone Traffic Routing"* IEEE Trans. on Systems, Man and Cybernetics, vol SMC-10, no 5, May 1980.
12. Nedzelnitsky, O. V. *"Learning Algorithms in Data Communication Networks"* Proc. Third Yale Workshop on Applications of Adaptive Systems Theory, 1983.
13. Srikantakumar, P. R. and Narendra, K. S. *"A Learning Model for Routing in Telephone Networks"* SIAM J. Control and Optimization, vol-20, no 1, Jan. 1982.
14. Tsai, W. K., Tsitsiklis, J. N. and Bertsekas, D. P. *"Some Issues in Distributed Asynchronous Routing in Virtual Circuit Data Networks"* Proc. IEEE 25th Conf. on Decision and Control, Dec. 1986.

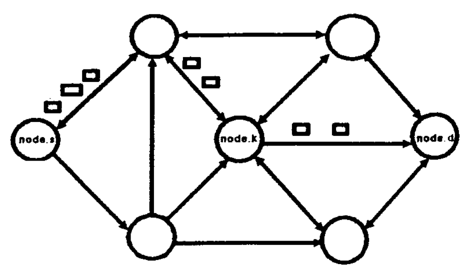

Figure 1: Virtual Circuit Routing

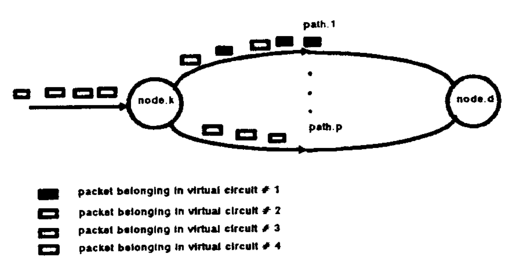

Figure 2: Route selection

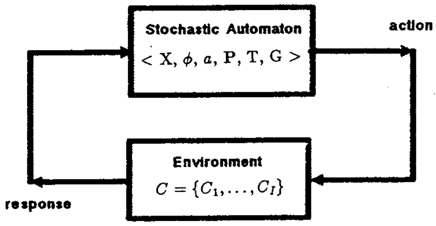

Figure 3: A learning automaton

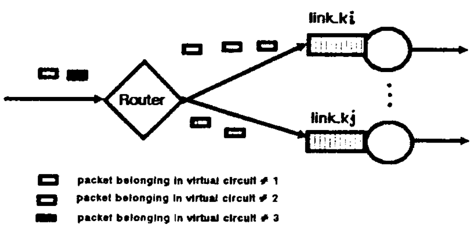

Figure 4: Router $R_{[kd]}$

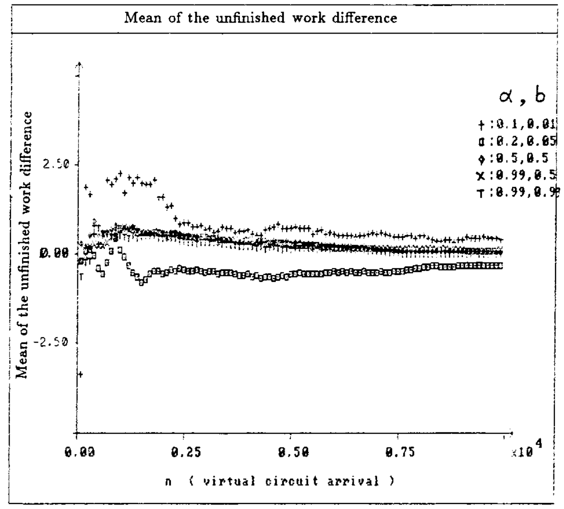

Figure 6: Mean of the unfinished work.difference on the two paths during the experiment, for different values of α and b.

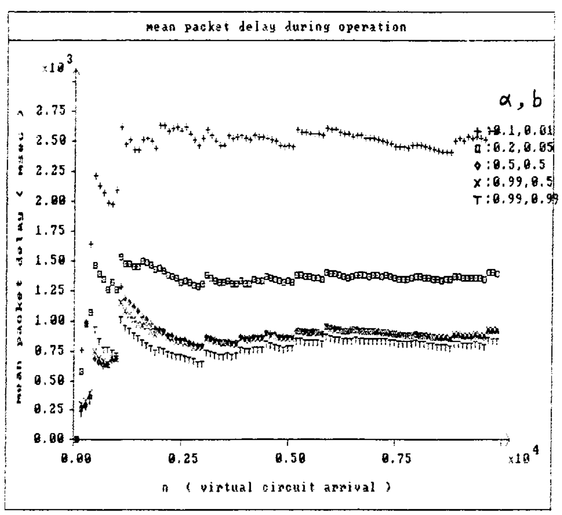

Figure 5: Mean packet delay during the experiment, for different values of α and b.

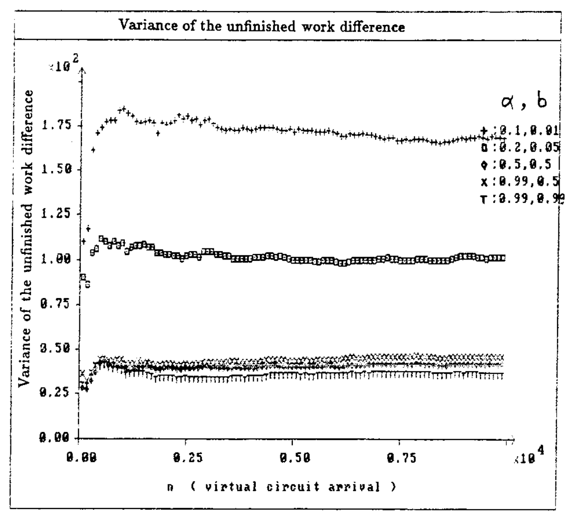

Figure 7: Variance of the unfinished work.difference on the two paths during the experiment, for different values of α and b.

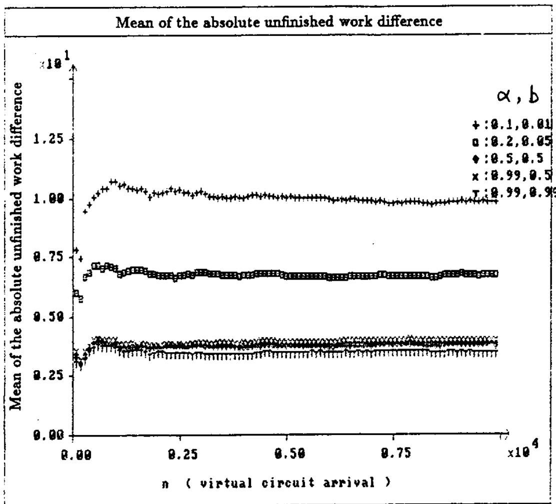

Figure 8: Mean of the absolute unfinished work.difference on the paths, during the experiment, for different values of α and b.

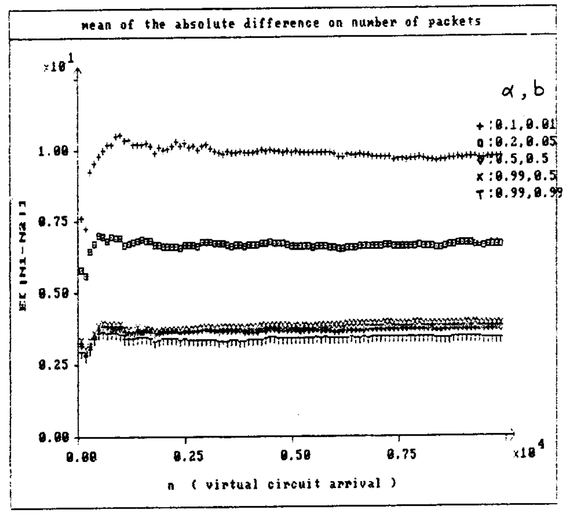

Figure 10: Mean of the absolute difference of the number of packets, on the two paths during the experiment.

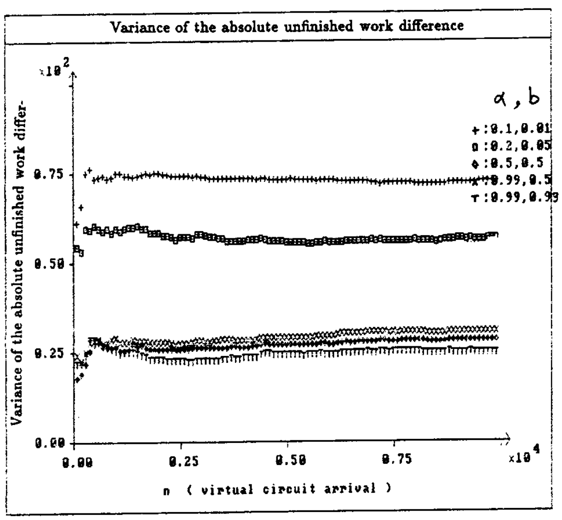

Figure 9: Variance of the absolute unfinished work.difference on the two paths during the experiment, for different values of α and b.

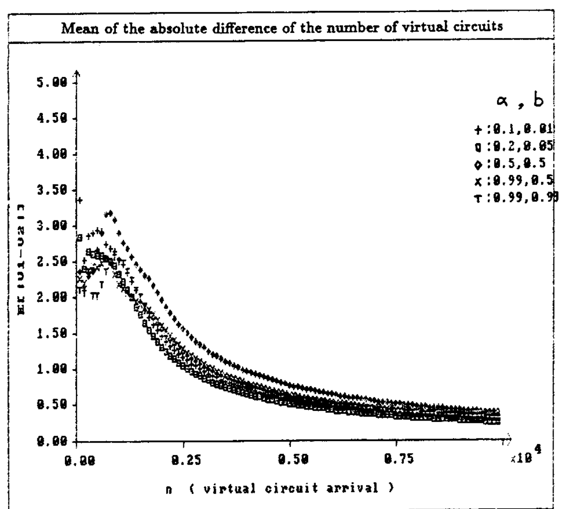

Figure 11: Mean of the absolute difference of the number of virtual circuits, on the two paths during the experiment.