KYBERNETIKA — VOLUME 25 (1989), NUMBER 6

LEARNING AUTOMATA FOR DATA COMMUNICATION ROUTING PROBLEM

ATHANASIOS V. VASILAKOS

A decentralized adaptive routing technique based on learning automata is presented for computer communication networks. Messages are routed by the automata selecting suitable outgoing links, with the delay experienced by a message used as a feedback response for updating future selection strategy. This simple feedback policy which is a realistic representation of the actual feedback received in communication networks is shown to give the minimum achievable delay.

1. INTRODUCTION

A computer network is a collection of nodes at which reside the computing resources which communicate with each other via a set of data links. At the nodal switching computers, the communications-oriented tasks or relaying messages (with appropriate routing flow controlling, queueing and so on) and of inserting and removing messages that originate and terminate at the terminals and main processors (hosts) at that node must be carried out. The transmission of digital information has shown tremendous growth over the last decade and will undoubtedly continue to expand in the future. This and the growth of digital electronic switching techniques have stimulate the development of more efficient network protocols and routing schemes.

The routing methods can be classified as static, quasi-static, and dynamic. The static case is the stationary situation when no adaptation is required. In quasi-static routing, data from various parts of the network is gathered during periods, and at the end of each period the control is recomputed using data of that period, which which will be effective for the next period (or session).

Quasi-static routing is suitable for slowly varying traffic conditions. In dynamic routing, the routing decisions are based upon the instantaneous values of the network states, and thus are better equipped for fast changes in traffic.

The optimal routing decisions at a node or at a region of the network usually depend upon the traffic and the network states in other regions of the network. For large sized networks (i.e. large number of nodes and links) collecting traffic

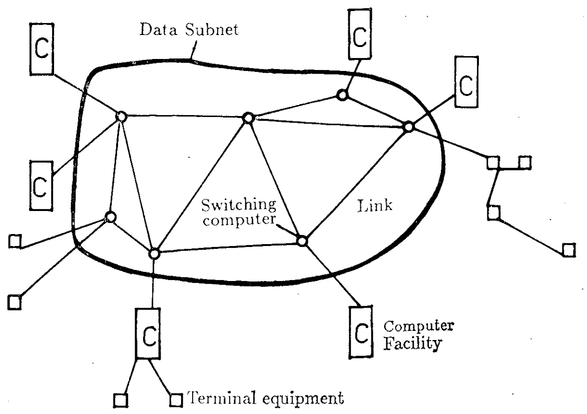

Fig. 1. Data and computer communication networks.

data from all parts of the network and making routing decisions at a single place or node possess difficult engineering computational problems. Also such a network with centralized decision making is dangerously vulnerable to failures of this central node, and failures of the links which carry control-information between this central node and other nodes of the network.

In order to circumvent this problem, the approach to routing that has gained considerable attention recently is that of distributed computation (or *decentralized control*). Typically in a decentralized control situation routing control algorithms are located at various nodes. These nodal controllers frequently may exchange some control-information among themselves through their neighbours. The routing decision at each node is made by the local controller depending upon the local traffic situation and also upon the global data that is summarized as the control-information passed from the neighbouring nodes.

The network routing problem – static, quasi-static or dynamic, centralized or decentralized – could be formulated in a deterministic framework, using the average values of the traffic parameters. The static centralized routing problem in a deterministic framework leads to multi-commodity flow problems [7] which have been well studied. The decentralized approach to routing is relatively new, and is currently a topic of intense investigation.

Most of the studies [1, 7, 8] treat the decentralized routing problem as a deterministic optimization problem, and give quasi-static routing methods.

Since the nature of the traffic is random, it is more appropriate to formulate the routing problem in a probabilistic framework. The probabilistic study of decentralized network routing, which has been studied by a few researchers recently, is still in its infancy.

So, the development of a theory, in a probabilistic framework, to systematically design and analyze learning schemes for adaptive routing in networks is needed.

In this paper some initial work is presented on automata routing in small networks.

The use of learning automata concepts in communication network routing has already been successfully studied in circuit-switched networks, including the telephone network [3].

The possible application of learning automata routing to message switched networks appears to be a very promising approach.

In such schemes messages are routed by the automata selecting suitable outgoing links, with the delay experienced by a message used as a feedback response for updating future selection strategy. This simple feedback policy can be very conventiently applied to real communication networks and is shown to give a routing performance close to the minimum achievable delay.

2. LEARNING AUTOMATON (LA)

An LA operates in such a manner as to choose (intelligently) an optimal action from an allowable set and to apply the selected action to a random environment. The environmental responds with a feedback signal, which initiates an updating of the internal state vector of the LA, responsible for the future action selection process. Consider the LA environment configuration of Figure 2.

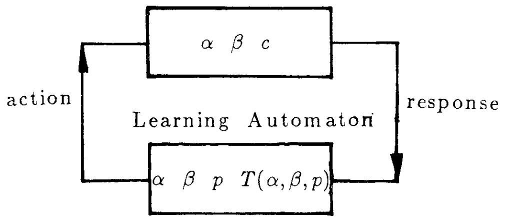

Fig. 2.

Consider $A(n)$, a variable structure Stochastic Automaton

$$A(n) = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{p}, T(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{p})\} \quad \text{where}$$

$\boldsymbol{\alpha}$: *The action set.* The performed action at stage n is

$$\alpha(n) \in \{\alpha_1, \ldots, \alpha_r\}$$

$\boldsymbol{\beta}$: *The response set.* At stage n $\beta(n) \in \{0, 1\}$ $\quad$ 0 – reward
1 – penalty

$\boldsymbol{p}$: *The action probability set.* The internal state of a variable structure automaton is specified by the action probability $\boldsymbol{p}$.

$$p(n) \in \{p_1, \ldots, p_r\} \quad \text{where} \quad p_i = \text{prob}\,\{\alpha(n) = \alpha_i\}$$

$T(\alpha, \beta, p)$: *The Reinforcement Algorithm.* It modifies the action probability vector in relation to the performed action and received response.

$$p(n+1) = T(\alpha, \beta, p(n))$$

The most popular reinforcement schemes are:
Linear Reward Penalty L_{RP}, and from this the
Linear Reward Inaction L_{RI}
Linear Reward ε-Penalty $L_{R-\varepsilon P}$

L_{RP} algorithm for r-action [2]
Reward on α_i

$$p_{j \neq i}(n+1) = (1-a)\,p_j(n)$$
$$p_i(n+1) = 1 - \sum_{j \neq i} p_j(n+1), \quad 0 < a < 1$$

Penalty on α_i

(1) $$p_{j \neq i}(n+1) = p_j(n) + [b/(r-1)]\,p_i(n)$$
$$p_i(n+1) = (1-b)\,p_i(n), \quad 0 < b < 1$$

L_{RI} algorithm is given by (1) when $b = 0$.
$L_{R-\varepsilon P}$ is given by (1) when $b = O[a]$ where $O[a]/a \to 0$ as $a \to 0$.

The Environment. It is described by the triple $E(n)$
$E(n) = \{\alpha, \beta, c\}$ where $c = \{c_1, \dots, c_r\}$ is the penalty set and
$c_i = \text{prob}\,\{\beta(n) = 1 \mid \alpha(n) = \alpha_i\}$
The constants a and b are the reward and penalty parameters, $0 < a, b < 1$.

Non-Stationary Environments. If the elements C_i are constants for all time, the environment is classified as stationary. However, in our case, the environmental characteristics will vary with time, and the response behaviour is influenced as a result of the actions performed by the automata.

In the dynamic routing problem the network represents a non-stationary environment, when an automata routing controller is used to select suitable outgoing links. By selecting a certain link with a high probability, the path obviously becomes less attractive due to increased delay caused by the higher traffic rate. On the other hand the alternative links are required to handle less traffic and consequently a more favourable response is obtained when a message is routed on any of these.

Convergence of the Learning Routing Schemes

Define $\Delta p_i(n) = p_i(n+1) - p_i(n)$
i) For the r action *L_{RI} automaton*

$$E[\Delta p_i(n) \mid p(n) = p] = a\,p_i\Big[\sum_{j \neq i} p_j\,C_j(p) - c_i(p)\sum_{j \neq i} p_j\Big]$$

$$\text{if} \quad \Delta P_i(n) \to 0 \quad \text{as} \quad n \to \infty \Rightarrow p_i \to 0 \quad or \quad \sum_{j \neq i} p_j\,c_j(p) = c_i(p)\sum_{j \neq i} p_j$$

which corresponds to an equalisation of the penalty probabilities $c_i(p)$. Note that in non-stationary environments the penalty probabilities c_i are functions of the action probabilities p_i, since the actions of automaton effect the distribution of the messages.

ii) For the r action L_{RP} *automaton*

$$\mathrm{E}[\Delta p_i(n) \mid p(n) = p] = a[1/(r-1)\sum_{j \neq i} p_j\, c_j(p) - p_i\, c_i(p)]$$

$$\text{since } a = b\,.$$

The equilibrium condition in this case results in an equalisation of the penalty rates $p_i\, c_i(p)$:

$$p_i\, c_i(p) = 1/(r-1)\sum_{j \neq i} p_j\, c_j(p)$$

3. LEARNING AUTOMATA FOR PACKET SWITCHING NETWORKS

A simple network/learning automaton combination is proposed with a view to establish an understanding of the basic routing behaviour in a packet switched network. Consider the four-node arrangement in Figure 3.

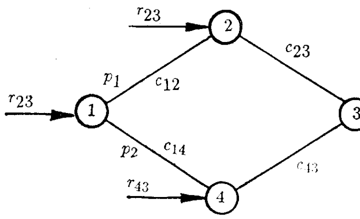

Fig. 3. 4-node network.

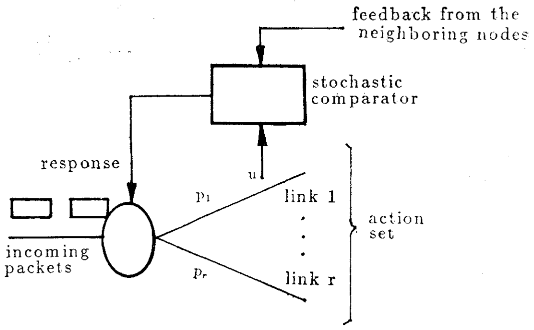

Fig. 4. Automaton routing controller.

Let r_{ij} be the average flow rate in packets/second between source i and destination j, $1/\mu$ the mean packet length in bits and C_{ik} be the capacity of the link (i, k) in bits/second.

An automaton at node 1 is used to select a suitable path for a packet, and the path options being performed with probabilities p_1 and p_2.

In a typical application (Fig. 4) the allowable set of outgoing links corresponds to the automaton action set, the routing operation selecting a link on the basis of the action probabilities. The response to the performed action is obtained by passing the delay experienced by a packet (or message) to the simple stochastic comparator which produces the necessary feedback to the reinforcement scheme.

Assuming Poisson/Exponential statistics and the independence assumption, the average time delay on the optional paths is given by

$$\overline{T}_1 = 1/(c_{12} - f_{12}) + 1/(c_{23} - f_{23})$$
$$\overline{T}_2 = 1/(c_{14} - f_{14}) + 1/(c_{43} - f_{43})$$

where

$$f_{12} = p_1 r_{13}/\mu\,, \quad f_{23} = f_{12} + r_{23}/\mu$$
$$f_{14} = p_2 r_{13}/\mu\,, \quad f_{43} = f_{14} + r_{43}/\mu$$

In the above formula the effect of a finite buffering size for the input and output queues of the nodes is neglected.

Two expediency criteria may be conceived for the operation of the automaton.

i) For the L_{RI} reinforcement algorithm, we expect an attempt to *equalise the penalty probabilities,*

$$\mathsf{E}[c_1(n)] = \mathsf{E}[c_2(n)] \quad \text{as} \quad n \to \infty$$

The penalty responses result from the operation of the stochastic comparator, hence

$$\mathsf{E}[c_i(n)] = \mathsf{E}[T_i > u] = \overline{T}_1$$

where u is a uniform deviate, T_i the delay experienced on path i, and $\overline{T}$ is the average delay. So the steady state operation of the L_{RI} *results to an equalisation of the average path delays.*

As shown by Gallager [1] the conditions for minimum delay are produced by a routing algorithm which equalises the differential delay over the alternative rouing options.

ii) For the L_{RP} reinforcement algorithm, we expect an attempt *to equalise the penalty rates.*

Indeed

$$\mathsf{E}[p_1 c_1] = \mathsf{E}[p_2 c_2] \quad \text{as} \quad n \to \infty$$

In the packet/message switched network this leads to

$$p_1\, \mathsf{E}[c_1] = p_2\, \mathsf{E}[c_2] \Rightarrow p_1 \overline{T}_1 = p_2 \overline{T}_2$$

So, the steady state behaviour of the L_{RP} *results to an equalisation of the delay rates.*

The two schemes L_{RI}, L_{RP} justify their applicability, giving promising results, but the equilibrium behaviour regarding the equalisation of average path or accummulated path delays, is suitable for *light traffic.*

The principal reason for using learning rules for routing packets or messages is to minimize the blocking probability of the network in the situation of *heavy traffic* taking into effect the buffers overflow and higher rejection probability.

Narendra and Thathachar [3] have shown that if $\{c_i(p)\}_{i=1}^{r}$ satisfy

A1. $c_i(p_1, \ldots, p_r)$ is continuous in p_j, $i, j = 1, \ldots, r$

A2. $\partial c_i(\cdot)/\partial p_i > 0 \ \forall i$ and $\partial c_j(\cdot)/\partial p_i \ll \partial c_i(\cdot)/\partial p_i$ for $j \neq i$,
where $\partial c_i(\ )/\partial p_i$ stands for partial derivatives.

A3. $c_i(\cdot)$ is continuously differentiable in all its arguments.

Then in steady state behaviour of the two-action L_{RP} automaton there exists p^* such that

$$p_1^* \, c_1(p_1^*) = p_2^* \, c_2(p_2^*)$$

If $p_i^B(p)$ is the blocking probability of action i for the automaton at node i and $p_i^B(\cdot)$ satisfies the above conditions (A1 – A3) and with increase in p_i then under equilibrium conditions the blocking rates

$$p_i \, p_i^B(p) = \text{constant} \quad \forall i$$

Back to our Problem

Let the nodal buffers be all of the same size q. Using the independence assumption we can write the probability that packets be blocked on the optional paths as follows

$$p_1^B(1) = p_{12}^B p_{23}^B$$
$$p_2^B(1) = p_{14}^B p_{43}^B$$

where $p_1^B(1)$, $p_2^B(1)$ = probabilities of a blocking at node 1 conditioned on the choice of paths 1 and 2, respectively.

Let p_{ij}^B denotes the blocking probability on the link (i, j). From [5] we have

(2) $$p_{ij}^B = (1 - \varrho_{ij}) \, p_{ij}^q/(1 - \varrho_{ij}^{q+1})$$

where ϱ_{ij} = link utilization f_{ij}/c_{ij}.

The equilibrium criterion for equalisation blocking rates on the optional paths then yields:

(3) $$p_1 p_1^B(1) = p_2 p_2^B(1) \Leftrightarrow p_1 p_{12}^B / p_2 p_{14}^B = p_{43}^B / p_{23}^B$$

So, the equilibrium is attained when the flow at node 1 is split in such a way that the rate of blocking on each link is inversely proportional to the blocking probability of the neighboring node.

4. EXAMPLE

Consider the network in Figure 3

$r_{13} =$ 1000 pack/sec	$c_{12} =$ 350kbits/sec
$r_{43} =$ 800 pack/sec	$c_{14} =$ 230 kbits/sec
$r_{23} =$ 600 pack/sec	$c_{23} =$ 550 kbits/sec
$1/\mu =$ 350 bits/pack	$c_{43} =$ 525 kbits/sec

So

$$\varrho_{12} = \varrho_{14} = 0{\cdot}6 \quad \varrho_{23} = 0{\cdot}7 \quad \varrho_{43} = 0{\cdot}8$$

From equations (3) $\Rightarrow p_1 = 0{\cdot}6$, $p_2 = 0{\cdot}4$.

The resulting flow splitting yields also *minimum average time delay*

$$\bar{T} = \bar{T}_1 + \bar{T}_2 =$$

$$= 1/(r_{13} + r_{23} + r_{43}) [p_1 r_{13}/(c_{12} - f_{12}) + (r_{23} + p_1 r_{13})/(c_{23} - f_{23}) +$$
$$+ p_2 r_{13}/(c_{14} - f_{14}) + (r_{43} + p_2 r_{13})/(c_{43} - f_{43})]$$

T vs routing strategy p_1 is shown below

Routing Strategy p_1	Average Delay $\bar{T}$ (mcsec)
0·2	23·1
0·4	19·23
0·5	10·1
$\rightarrow$0·6	$\rightarrow$8·69
0·7	9·52
0·8	12·67
0·9	22·5

5. CONCLUSIONS

Conclusions may be made concerning the behaviour of LA schemes in a simple message/packet switched network. Equalising blocking rates results to the minimum acheivable delay.

Future work will involve the application of LA to both virtual circuit and datagram complex networks, giving *new* learning automata reinforcement schemes which result in equalization of differential delays [1] and the collective behaviour of LA in a large network (i.e. stochastic games between LA) as well.

(Received December 21, 1987.)

REFERENCES

[1] R. G. Gallager: A minimum delay routing algorithm using distributed computation. IEEE Trans. Comm. *COM-25* (1977), 73—84.

[2] K. S. Narendra and M. A. L. Thathachar: Learning automata — a survey. IEEE Trans. Systems Man Cybernet. SMC-4 (1974), 323—334.

[3] K. S. Narendra and M. A. L. Thathachar: On the behaviour of a learning automaton in a changing environment with application to telephone traffic. IEEE Trans. Systems Man Cybernet. SMC-10 (1980), 262—269.

[4] M. Schwartz: Computer Communication Network Design and Analysis. McGraw-Hill, New York 1964.

[5] A. S. Tanenbaum: Computer Networks. Prentice Hall, Englewood Cliffs, N. J. 1980.

[6] D. P. Bertsekas: Optimal Routing and Flow Control Methods for Communication Networks. (Analysis and Optimization of Systems.) Springer-Verlag, Berlin—Heidelberg—New York 1982, pp. 615—643.

[7] A. Segall: The modelling of adaptive routing in data networks. IEEE Trans. Comm. *COM-25* (1977), 85—95.

Dr. Athanasios V. Vasilakos, School of Engineering — Dept. of Computer Engineering, University of Patras, 26500 Patras. Greece.