Economides, A.A.: **Real-time traffic allocation using learning automata.** *Proceedings 1997 International Conference on Systems, Man, and Cybernetics*, pp. 3307-3312, IEEE 1997.

REAL-TIME TRAFFIC ALLOCATION USING LEARNING AUTOMATA

Anastasios A. Economides
University of Macedonia
Thessaloniki 54006, GREECE

economid@macedonia.uom.gr

ABSTRACT

We present a new fixed structure, multi-action, multi-response learning automaton and use it to allocate arriving traffic at a multimedia network. For each source-destination pair, for each traffic type, a learning automaton allocates every new arriving call on one of the available routes from source to destination or rejects it.

The state diagram of the learning automaton has a star shape. Each branch of the star is associated with a particular route. Depending on how much "good" the traffic performance is on a route, the automaton moves deeper in the corresponding branch. On the other hand, depending on how much "bad" it is, the automaton moves out of this branch. Finally, we provide several performance metrics to characterize the traffic performance on a route as "good" or "bad".

1. INTRODUCTION

The successful deployement of broadband multimedia applications on high-speed networks depends on the appropriate network management and traffic allocation mechanisms. The integration of different traffic types (voice, video, data etc.) with diverse and complex traffic characteristics (average rate, peak rate, burstiness, activity duration, silence duration, peak duration etc.) and diverse Quality Of Service (QOS) requirements (constraints on delay, delay jitter, loss ratio etc.) create the need for sophisticated network management actions.

One such action is the Connection Admission Control (CAC) during the call set-up phase (or during call re-negotiation phases) in order to determine whether a virtual channel/virtual path connection request can be accepted or should be rejected (or whether a request for re-allocation can be accommodated) [1].

Multimedia applications, such as Tele-education, Tele-advertising, Tele-shopping, Tele-conference, etc. require communication via the transfer of voice, Hi-Fi sound, moving pictures, video-scanned still images and documents. They produce a wide range of bandwidth with traffic pattern uncertainties and burstiness that cause unpredictable statistical traffic fluctuations and transient phenomena. Therefore, real-time adaptive traffic allocation mechanisms are needed to control the network traffic and resources.

In such a random environment with unpredictable traffic behavior, it is necessary to have mechanisms which interact with the environment and learn dynamically the action that will produce the most desirable environment outcome. Learning automata are such mechanisms.

At times n=1, 2, ..., an automaton selects one of several available actions, according to action probabilities determined by its current state. The environment provides a random response to the action selected. Depending on the environment response, the automaton changes state. When the action probabilities of each state remain time-invariant, we have a *fixed-structure stochastic automaton (FSSA)*. When the action probabilities change in time, we have a *variable-structure stochastic automaton (VSSA)*.

Historically, the theory of learning automata was initiated with the study of FSSA (see [17]). Later, interest shifted to the study of VSSA (see [18]) which appeared to be more adaptable [12]. While VSSA's have attracted a lot of attention, FSSA's are easier to implement and require less computation per time step. Recently, several new FSSA have been introduced [13, 14, 15, 6].

In Section 2, we present a new FSSA, the MR-STAR$^{(D)}$ (Multiple Response - STAR$^{(D)}$) automaton. This automaton is a combination of the MRLA (Multiple Response Learning Automaton) [7] and of the STAR$^{(D)}$ automaton [6].

0-7803-4053-1/97/$10.00

The state diagram of this learning automaton has a star shape. Each branch of the star consists of D states, which are "committed" to one of the actions available to the automaton. In each branch, transitions from state to state depend on the degree of "goodness" and "badness" of the environment response to the associated action. For simplicity of the presentation, we shall consider the 4R-STAR$^{(D)}$ (4 Responses) automaton (Fig. 1). When the environment response to the selected action is "very good", the automaton goes at the edge of the branch associated with this action. When the response is "good", the automaton goes in a state deeper than its current state in the corresponding branch. When the response is "bad", the automaton goes in a state closer to the neutral state. Finally, when the response is "very bad", the automaton goes to the neutral state.

In Section 3, we use the 4R-STAR$^{(2)}$ automaton for allocating every new incoming call on a route or reject it from the network. Previous studies use VSSA learning automata for telephone routing [9, 10, 16, 11], datagram routing [2, 3] and virtual connection routing [4, 5], while in [8] FSSA are used for multimedia call routing.

Finally in Section 4, we summarize, present our conclusions and propose some directions for future research.

2. 4R-STAR$^{(D)}$

In this section we present a new multiple response FSSA scheme. At each instant n, the automaton selects probabilistically (according to the action probability vector $p(n)$) an action $\alpha(n) = i$ from the finite action set $\alpha = \{1, 2, ..., r\}$. The probability that the automaton selects action i, at time n is the action probability $p_i(n) = Prob[a(n) = i]$; we have $\sum_{i=1}^{r} p_i(n) = 1 \ \forall n$. The environment response can be "very good", "good", "bad" or "very bad". The environment response to action i is chosen according to the unknown probabilities: $Prob[\text{"very good" response} | a(n) = i]$, $Prob[\text{"good" response} | a(n) = i]$, $Prob[\text{"bad" response} | a(n) = i]$ and $Prob[\text{"very bad" response} | a(n) = i] \ \forall i$.

Learning takes place by repeated application of the following procedure: the automaton chooses an action according to the action probability vector of its current state. Depending on the response of the environment, the automaton moves to a new state and chooses a new action according to the action probability vector of its new state. Hopefully this procedure leads to the reduction of the average cost [6].

In the proposed 4R-STAR$^{(D)}$, the automaton can be in any of $D * r + 1$ states, $\{(0,0), (1,1), (1,2), ..., (1,D), \ldots, (r,1), (r,2), ..., (r,D)\}$. The state transition diagram is illustrated in Fig. 1.

When the automaton is in state $(i,1)$ or $(i,2)$ or ... (i,D), it performs action i with probability 1, i=1,2,...,r. So each one of these states is "committed" to a corresponding action. On the other hand, the state $(0,0)$ is a special, so-called "neutral" state: when in that state, the automaton choooses any of the r actions equiprobably.

Each of the four environment responses cause deterministic transitions according to the following rules ($0 \leq \epsilon_{VG}, \epsilon_G, \epsilon_B, \epsilon_{VB} \ll 1$):

1. When in state $(0,0)$ and chosen action is i,

   if the environment response is "very good" go to state (i,D) w.p. $1-\epsilon_{VG}$ or stay in state $(0,0)$ w.p. ϵ_{VG},

   if it is "good" go to state $(i,1)$ w.p. $1-\epsilon_G$ or stay in state $(0,0)$ w.p. ϵ_G,

   if it is "bad" stay in state $(0,0)$ w.p. $1-\epsilon_B$ or go to state $(i,1)$ w.p. ϵ_B,

   if it is "very bad" stay in state $(0,0)$ w.p. $1-\epsilon_{VB}$ or go to state (i,D) w.p. ϵ_{VB}.

2. When in state $(i,1)$ $i = 1, 2, ..., r$, and chosen action is i,

   if the environment response is "very good" go to state (i,D) w.p. $1-\epsilon_{VG}$ or go to state $(0,0)$ w.p. ϵ_{VG},

   if it is "good" go to state $(i,2)$ w.p. $1-\epsilon_G$ or go to state $(0,0)$ w.p. ϵ_G,

   if it is "bad" go to state $(0,0)$ w.p. $1-\epsilon_B$ or go to state $(i,2)$ w.p. ϵ_B,

   if it is "very bad" go to state $(0,0)$ w.p. $1-\epsilon_{VB}$ or go to state (i,D) w.p. ϵ_{VB}.

3. When in state (i,d) $i = 1, 2, ..., r$, $d = 2, ..D-1$ and chosen action is i,

if the environment response is "very good" go to state (i, D) w.p. $1-\epsilon_{VG}$ or go to state $(0,0)$ w.p. ϵ_{VG},

if it is "good" go to state $(i, d+1)$ w.p. $1-\epsilon_G$ or go to state $(i, d-1)$ w.p. ϵ_G,

if it is "bad" go to state $(i, d-1)$ w.p. $1-\epsilon_B$ or go to state $(i, d+1)$ w.p. ϵ_B,

if it is "very bad" go to state $(0,0)$ w.p. $1-\epsilon_{VB}$ or go to state (i, D) w.p. ϵ_{VB}.

4. When in state (i, D) $i = 1, 2, ..., r$ and chosen action is i,

   if the environment response is "very good" stay in state (i, D) w.p. $1-\epsilon_{VG}$ or go to state $(0,0)$ w.p. ϵ_{VG},

   if it is "good" stay in state (i, D) w.p. $1-\epsilon_G$ or go to state $(i, D-1)$ w.p. ϵ_G,

   if it is "bad" go to state $(i, D-1)$ w.p. $1-\epsilon_B$ or stay in state (i, D) w.p. ϵ_B,

   if it is "very bad" go to state $(0,0)$ w.p. $1-\epsilon_{VB}$ or stay in state (i, D) w.p. ϵ_{VB}.

In [6] we compare the performance of the 2 Response STAR$^{(D)}$ automaton to that of L_{R-P} and $L_{R-\epsilon P}$ (for various values of learning rates a and b). The automata operate in a switching environment where the best action changes periodically. We find that STAR$^{(D)}$ responds faster than L_{R-P} and $L_{R-\epsilon P}$ to environnment switchings and it also incurs smaller average cost. In addition to these advantages, it should be noted that STAR$^{(D)}$ implementation is simpler, since it requires no floating point computations.

3. 4R-STAR$^{(D)}$ AS A ROUTER

In this section we use 4R-STAR$^{(D)}$ automata at the source nodes of a multimedia network to allocate incoming calls on routes from source to destination. Every source node $[s.]$ in the network has several 4R-STAR$^{(D)}$ automata, each one for a particular destination node $[.d]$ and traffic class c (data, voice, video, etc.).

In the following, we consider the 4R-STAR$^{(D)}$ automaton that allocates the incoming calls of traffic class c on the $r-1$ routes between the source-destination pair $[sd]$ or rejects them. A call can only be accepted if sufficient network resources are available to meet its required QOS. Since everything is associated with traffic class c between source-destination $[sd]$, let ignore the indexes c and $[sd]$.

The actions of this automaton are to route a new incoming call to its destination through one of the $r-1$ routes or reject the call by sending it through a fictitious route r. After selecting a route, the automaton measures the performance of its traffic on all routes and goes to the next state. Depending on the traffic type, the performance measurements would be any of the following or their marginal values:

cell loss ratio: ratio of the number of lost cells to the sum of the number of lost plus successfully delivered cells

cell insertion rate: number of inserted cells within a specified time interval

cell error ratio: ratio of errored cells to the number of successfully delivered cells

cell transfer delay

mean cell transfer delay: arithmetic average of a specified number of cell transfer delays

cell delay variation: difference between a single observation of cell transfer delay and the mean transfer delay on the same connection

cell transfer capacity: the maximum possible number of successfully delivered cells occuring over a specified connection during a time unit

Next, let define the performance indexes to be used in the algorithm:

$l_i(n)$: length of route i for this traffic class at time n; this length can be the cost (delay, jitter of delay, loss ratio etc.) or the marginal cost on this route.

$QOS_i(n)$: Quality Of Service measurements on route i for this traffic class at time n,

θ: QOS constraint for this traffic class.

We classify the network responses to action i (selection of route $i = 1, ..., r-1$) at time n as:

"very good": when the length of route i is much smaller than that of all other routes and the QOS constraints are met

"good": when the length of route i is smaller (but not much smaller) than that of all other routes and the QOS constraints are met

"bad": when the length of route i is larger (but not much larger) than the minimum length of the other routes and the QOS constraints are met

"very bad": when the length of route i is much

larger than the minimum length of the other routes or the QOS constraints are not met.

Writting these concepts in mathematics, we have $(\zeta, \eta > 0)$:

$$
\begin{array}{lll}
\text{"}\mathit{very\ good}\text{"} & \equiv & l_i \leq \min_{j \neq i} l_j - \zeta \\
 & & and\ QOS_i < \theta \\
\text{"}\mathit{good}\text{"} & \equiv & \min_{j \neq i} l_j - \zeta < l_i \leq \min_{j \neq i} l_j \\
 & & and\ QOS_i < \theta \\
\text{"}\mathit{bad}\text{"} & \equiv & \min_{j \neq i} l_j < l_i \leq \min_{j \neq i} l_j + \eta \\
 & & and\ QOS_i < \theta \\
\text{"}\mathit{very\ bad}\text{"} & \equiv & \min_{j \neq i} l_j + \eta \leq l_i \\
 & & or\ QOS_i > \theta
\end{array}
$$

Similar, for action r (rejection of the call):

$$
\begin{array}{lll}
\text{"}\mathit{very\ good}\text{"} & \equiv & QOS_j > \theta \\
\text{"}\mathit{good}\text{"} & \equiv & l_r \leq \min_{j \neq r} l_j \\
 & & and\ QOS_j < \theta \\
\text{"}\mathit{bad}\text{"} & \equiv & \min_{j \neq r} l_j < l_r \leq \min_{j \neq r} l_j + \eta \\
 & & and\ QOS_j < \theta \\
\text{"}\mathit{very\ bad}\text{"} & \equiv & \min_{j \neq r} l_j + \eta \leq l_r \\
 & & and\ QOS_j < \theta
\end{array}
$$

$\forall \quad j = 1, ..., r-1$

Next, we present the routing and admission control algorithm using a simple case of the 4R-STAR$^{(D)}$ automaton with $\epsilon_{VG} = \epsilon_G = \epsilon_B = \epsilon_{VB} = 0$ and D=2 (Fig. 2).

Let a new call arrive at the source node.

1. When the automaton is in state $(0,0)$, it selects route i with propability $1/r$.

   if the network response is "very good", the automaton goes to state $(i,2)$ w.p. 1,

   if the network response is "good", it goes to state $(i,1)$ w.p. 1,

   if the network response is "bad", it stays in state $(0,0)$ w.p. 1,

   if the network response is "very bad", it stays in state $(0,0)$ w.p. 1.

2. When the automaton is in state $(i,1)$ $i = 1,2,...,r$, it selects action i w.p.1,

   if the network response is "very good", the automaton goes to state $(i,2)$ w.p. 1,

   if the network response is "good", it goes to state $(i,2)$ w.p. 1,

   if the network response is "bad", it goes to state $(0,0)$ w.p. 1.

   if the network response is "very bad", it goes to state $(0,0)$ w.p. 1.

3. When the automaton is in state $(i,2)$ $i = 1,2,...,r$, it selects action i w.p. 1,

   if the network response is "very good", the automaton stays in state $(i,2)$ w.p. 1,

   if the network response is "good", it stays in state $(i,2)$ w.p. 1,

   if the network response is "bad", it goes to state $(i,1)$ w.p. 1,

   if the network response is "very bad", it goes to state $(0,0)$ w.p. 1.

A slight modification of the above automaton is also described in Fig. 3.

4. CONCLUSIONS

In this paper we present a new class of FSSA with Multiple Responses from the environment, the so-called MR-STAR$^{(D)}$ automata. Then we emply them as routers at the source nodes of a multimedia network. They allocate incoming calls on the routes from source to destination or reject them from the network.

Extensions of the MR-STAR$^{(D)}$ automata would be the following: when the automaton is in a state (i,d), $d = 1,...,D$, it selects action i with some fixed high probability $p_{i(i,d)}$ and action j, $j \neq i$, with some fixed small probability $p_{j(i,d)}$, $\sum_k p_{k(i,d)} = 1$. Furthermore, simulation studies are needed in order to show the fast adjustment and stability of the automata when taking routing decisions in a real network.

References

[1] The ATM Forum, *"ATM User-Network Interface Specification Version 3.0"*, Prentice Hall 1993.

[2] M.S. Chrystall and P. Mars, "Adaptive routing in computer communication networks using learning automata", *Proc. of*

IEEE Nat. Telecomm. Conf., pp. A3.2.1-7, 1981.

[3] A.A. Economides and J.A. Silvester, "Optimal routing in a network with unreliable links", *Proc. of IEEE Computer Networking Symposium*, pp. 288-297, IEEE 1988.

[4] A.A. Economides, P.A. Ioannou and J.A. Silvester "Decentralized adaptive routing for virtual circuit networks using stochastic learning automata", *Proc. of IEEE Infocom 88 Conference*, pp. 613-622, IEEE 1988.

[5] A. A. Economides, "Learning automata routeing in connection-oriented networks", *International Journal of Communication Systems*, Vol. 8, pp. 225-237, 1995.

[6] A.A. Economides and Ath. Kehagias, "The STAR automaton: expediency and optimality properties", *submitted.*

[7] A.A. Economides, "Multiple response learning automata", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 1, pp. 153-156, Febr. 1996.

[8] A.A. Economides, "Learning automata for multimedia traffic control", *Proc. Int. Conf. Circuits, Systems and Computers '96*, pp. 555-560, Piraeus, 1996.

[9] K.S. Narendra and E.A. Wright and L.G. Mason, "Application of learning automata to telephone traffic routing and control", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-7, No.11, pp. 785-792, Nov. 1977.

[10] K.S. Narendra and M.A.L. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing", *IEEE Transactions on Systems, Man and Cybernetics,* Vol. SMC-10, No. 5, May 1980.

[11] K.S. Narendra and P. Mars, "The use of learning algorithms in telephone traffic routing - a methodology", *Automatica*, 19, (5), pp. 495-502, 1983.

[12] K. Narendra and M.A.L. Thathatchar, *Learning Automata*, Prentice-Hall, 1989.

[13] B.J. Oommen and E.R. Hansen "The asymptotic optimality of discretized linear reward-inaction learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-14, No. 3, pp. 542-545, May/June 1984.

[14] B.J. Oommen, "Absorbing and ergodic discretized two-action learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 2, pp. 282-293, March/April 1986.

[15] B.J. Oommen and J.P.R. Christensen, "ϵ-optimal discretized linear reward-penalty learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-18, No. 3, pp. 451-458, May/June 1988.

[16] P.R. Srikantakumar and K.S. Narendra, "A learning model for routing in telephone networks", *SIAM J. Control and Optimization,* vol-20, no 1, Jan. 1982.

[17] M.L. Tsetlin, "On the behavior of finite automata in random media", *Avtomatika i Telemekhanika*, Vol. 22, No. 10, pp. 1345-1354, Oct. 1961.

[18] V.I. Varshavskii and I.P. Vorontsova, "On the behavior of stochastic automata with a variable structure", *Avtomatika i Telemekhanika*, Vol. 24, No. 3, pp. 353-360, March 1963.

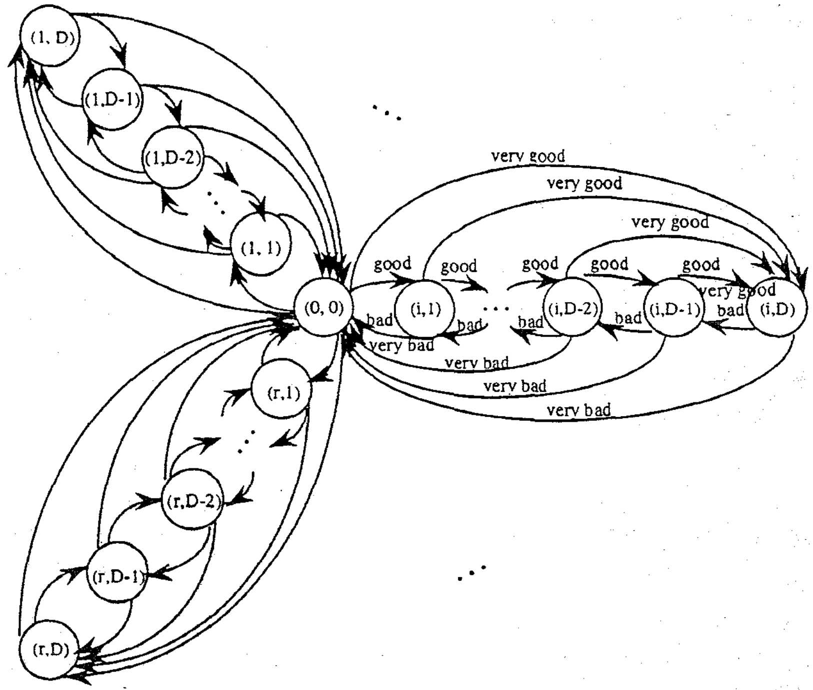

Fig. 1. 4R-STAR$^{(D)}$ state transition diagram

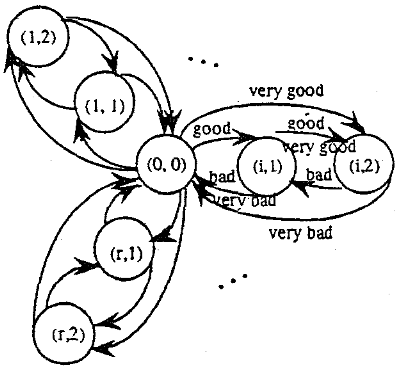

Fig. 2. 4R-STAR$^{(2)}$ state transition diagram

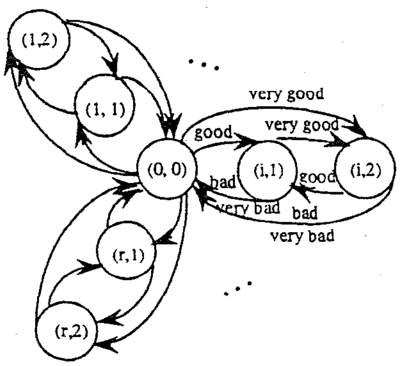

Fig. 3. Modified 4R-STAR$^{(2)}$