WDM Passive Star Networks: Receiver Collisions Avoidance Algorithms Using Multifeedback Learning Automata

GEORGIOS I. PAPADIMITRIOU and **DIMITRIOS G. MARITSAS**

Department of Computer Engineering, University of Patras, Greece.
&
Computer Technology Institute, Patras, Greece.

ABSTRACT

A new receiver collisions avoidance algorithm for WDM broadcast-and-select star networks is introduced. The proposed algorithm is based on the use of learning automata in order to reduce the number of receiver collisions and consequently, to improve the performance of the network. According to the proposed scheme, each station is provided with a learning automaton which decides about which of the packets waiting for transmission, will be transmitted at the beginning of the next time slot. The learning automaton used, is a multifeedback automaton, specially designed for the receiver collisions avoidance problem of WDM Broadcast-and-Select Star Networks. The asymptotic behavior of the system which consists of the automata and the network is analyzed and it is proved that the probability of choosing each packet, asymptotically tends to be proportional to the probability that no receiver collision will appear at the destination-node of this packet. Furthermore, extensive simulation results are presented, which indicate that a significant performance improvement is achieved when the proposed algorithm is applied on the basic DT-WDMA protocol.

I. INTRODUCTION

The Wavelength Division Multiplexing (WDM) technique [1], [2] is an efficient way to implement optical networks, capable of providing Gigabit data rates by using present-day optical and electronic technology. According to the WDM technique the available optical bandwidth is divided into multiple channels of lower bandwidth which can be easily supported by the stations' electronic circuits. Both, multiplexing and demultiplexing of the multiple channels, are performed in the optical domain - by using totally passive optical devices - without the need of optical to electronic translation and vice versa.

Broadcast-and-select star networks (fig.1) - which are the most commonly used WDM networks - use a passive star coupler in order to broadcast the transmitted packets to all the destination nodes. Since, each destination node is provided with only one tunable filter, it is unable to concurrently receive multiple packets coming to it from different wavelengths. Such a situation, which is termed as "receiver collision", causes the "loss" of all the collided packets, except one. A large number of receiver collisions would lead to a serious decrease of the network's performance.

In this paper, a receiver collisions avoidance algorithm, which is based on the use of learning automata, is introduced. According to the proposed scheme, each station is provided with a learning automaton which decides about which of the packets waiting for transmission, will be transmitted at the beginning of the next time slot. An analysis of the asymptotic behavior of the proposed automata-based algorithm, proves that the probability of choosing each packet, asymptotically tends to be proportional to the probability that no receiver collision will appear at the destination-node of this packet. In this way, the number of receiver collisions is

0742-1303/92 $3.00

reduced and consequently, both the delay vs throughput and the throughput vs offered load performance of the network are improved.

The structure of this paper is as follows. The presentation of the Multifeedback Learning Automaton in Section II is followed by a description of the Passive Star WDM network which is under consideration in Section III. The Receiver Collisions Avoidance Algorithm is presented in Section IV. An analysis of the asymptotic behavior of the system which is composed by both the Network and the Learning Automata is presented in Section V. It is followed by the presentation of extensive simulation results that indicate the superiority of the proposed automata-based algorithm in Section VI. Finally, in Section VII concluding remarks are given.

II. THE MULTIFEEDBACK LEARNING AUTOMATON

A Learning Automaton [4]-[7] consists of a variable structure automaton operating in a random environment, which tries to learn the optimal action this environment offers, via the following learning process (fig.2): The automaton chooses one of the actions which are offered, according to a probability vector, which at every instant contains the probability of choosing each action. The chosen action triggers the environment that responds with a reward or a penalty answer. The automaton takes into account this answer and modifies the probability vector by means of a learning algorithm. A learning automaton is one which learns the action that has the maximum probability to be rewarded by the environment and which ultimately chooses this action more frequently than other actions.

Many random environments have the capability of responding with a set of answers, whereby a separate answer is provided for each one of the possible actions. Environments of this kind can be characterized as "multifeedback environments". It is obvious that a multifeedback environment provides the automaton with much more feedback information than the well-known single-feedback environments. In this section we present a learning automaton which is capable of using the whole feedback information provided by a multifeedback environment. The proposed Multifeedback Learning Automaton (MLA) is defined by the quadruple $\langle A,B,P,T \rangle$ where:

$A=\{a_1,a_2,\ldots,a_r\}$ is the set of the r actions $(2 \le r < \infty)$. We define as $a(t)$ the action selected at a time instant t.

$B = \{0,1\}$ is the input set of the possible environmental responses, where 0 symbolizes a penalty and 1 a reward response. We define as $b(t)$ the environmental response at a time instant t. We have: $b(t)=\{b_1(t),b_2(t),\ldots,b_r(t)\}$ where $b_i(t) \in B$ for $i=1,2,\ldots,r$.

$\mathbf{P}$ is the probability vector. It is: $\mathbf{P}(t) = \{ P_1(t),P_2(t),\ldots,P_r(t) \}$ where $P_i(t) = \Pr[a(t)=a_i]$. Initially, $P_i(0)=1/r$ for $i=1,2,\ldots,r$.

T: $\mathbf{P}(t) \times B^r \rightarrow \mathbf{P}(t+1)$ is the probability updating scheme. It is also called "learning algorithm". Its formal description is presented below.

Let $K(t)=\{a_i \mid b_i(t)=1\}$ be the set of those actions which are rewarded at time instant t and $|K(t)|$ the number of elements of this set. The probability updating scheme is defined as follows (for $i=1,2,\ldots,r$):

$$P_i(t+1)=P_i(t)-L\ P_i(t) \qquad \text{if } b_i(t)=0$$

$$P_i(t+1)=P_i(t)+L\left(\frac{1}{|K(t)|}-P_i(t)\right) \qquad \text{if } b_i(t)=1$$

$L \in (0,1)$ is the "step size" parameter. It is an internal parameter of the automaton, which determines the step size of the probability updating scheme. Note, that the above learning algorithm requires $|K(t)| \neq 0$ for all t. Thus, at each time instant, there must be at least one rewarded action.

III. THE NETWORK DESCRIPTION

The architectural form of the network under consideration is the one presented in [3] (fig.1). Each transmitter is provided with two fixed-wavelength lasers: one at a station-specific wavelength which is used for data transmission and is called data wavelength, and one at a common wavelength which is used for the transmission of control signals and is called control wavelength. Both the wavelengths are combined by means of a 2x1 combiner. An optical fiber is used in order to connect the output of the combiner to one of the inputs of a nxn star coupler at the network hub. Thus, at each output port of the coupler, all the n data wavelengths, as well as the control wavelength are available. Each output port of the star coupler is connected to the corresponding receiver, by means of an optical fiber. At each receiver the optical signal is split into two parts by means of 1x2 splitter. One part of the signal goes to a fixed optical filter which passes only the control wavelength. The other part is fed to a tunable optical filter which is able to be tuned to pass any one of the n data wavelengths. In this way, the full connectivity of the network is guaranteed.

According to this scheme, each station transmits its packets at a station-specific wavelength. The receiving station - which has been informed of the imminent transmission by means of a signal which is sent via the control wavelength - tunes its optical filter at that wavelength, which corresponds to the transmitting station, in order to receive the incoming packet.

Since, each station is provided with only one tunable filter, it is unable to concurrently receive multiple packets coming to it from different wavelengths. Such a situation, which is termed as "receiver collision", causes the "loss" of all the collided packets, except one. The need for retransmission of the "lost" packets leads to a serious decrease of the network's performance.

IV. THE RECEIVER COLLISIONS AVOIDANCE LEARNING ALGORITHM

We propose a receiver collisions avoidance algorithm which is based on the use of learning automata and tends to minimize the probability of occurring concurrent transmissions from different nodes to the same destination node. According to the proposed Receiver Collisions Avoidance Learning Algorithm (RCALA), each station is provided with a learning automaton which decides about which of the packets waiting for transmission, will be transmitted at the beginning of the next time slot.

In the following description of the algorithm, we call a node of the network a "source node" if we consider it as a transmitting node and a "destination node" if we consider it as a receiving node. The operation of the Receiver Collisions Avoidance Learning Algorithm at a source node is described below.

IV.1 THE SELECTION OF THE DESTINATION NODE

Let n be the number of nodes (n is also the number of wavelengths). The set of nodes is defined as $U = \{1,2,\dots,n\}$.

The queue of packets which are waiting for transmission at each source node i is split into n separate subqueues; one subqueue for each destination node. Therefore, for each source node i, the queue of waiting packets is defined as a set of subqueues:

$Q_i(t) = \{ Q_{i,1}(t), Q_{i,2}(t), \dots, Q_{i,n}(t) \}$

where the $Q_{i,j}(t)$ subqueue contains those packets which are waiting at the source node i, and which are destined for the node j (with $Q_{i,i}(t)$ being always empty).

For each source node i the set $D_i(t)$, which contains all the possible destinations of node i at time instant t, is defined as follows:

$D_i(t)=\{\text{node } k \mid Q_{i,k}(t) \text{ is not empty}\}$ (1)

A learning automaton is placed at each source node i. Each destination node j corresponds to an action of the automaton. Therefore, at any time instant t the automaton contains a probability distribution over the set of destination nodes. For each destination node j we define the "destination node transmission probability" such as:

$$P_{i,j}(t) = \Pr[\text{node } i \text{ transmits to node } j \text{ at slot } t, \text{ given } |D_i(t)|=n]$$

Some of the subqueues $Q_{i,j}(t)$ may be empty. Thus, $|D_i(t)| \neq n$. Therefore, the probability of selecting each destination node $j \in D_i(t)$ must be scaled [5] in the following way:

$$r_{i,j}(t) = \Pr[\text{ node } i \text{ transmits to node } j \text{ at time slot } t] =$$

$$= \frac{P_{i,j}(t)}{\sum_{k \in D_i(t)} P_{i,k}(t)} \quad \text{for } j \in D_i(t) \quad (2)$$

At each time slot t, each source node i which has packets to transmit, chooses a destination node $j \in D_i(t)$, according to the scaled probabilities $r_{i,j}(t)$ and transmits the top (oldest) packet of the $Q_{i,j}(t)$ subqueue. The choice of the destination node is performed as follows: Since $\sum_{j \in D_i(t)} r_{i,j}(t)=1$, the (0,1) interval can be divided into k subintervals, with each subinterval corresponding to a destination node. A random number RND $\in$ (0,1) is generated according to the uniform probability distribution. This random number is located at one of the subintervals referred above. The destination node which corresponds to this subinterval is selected.

IV.2 THE PROBABILITY UPDATING SCHEME

All the transmissions that take place in the network (successful or unsuccessful) are preannounced to all the stations through the common control wavelength. After a round trip propagation delay of t_d slots from the nodes to the network hub and back, this information arrives to all the nodes of the network. Therefore, after a delay of t_d slots, each station is informed of the occurrence or not of a receiver collision at each one of the destination nodes. This information is used by the automata -which are placed at each node- as a network feedback information, in order to update the probability of selecting each destination node for transmission.

Obviously, the automata are fed with a separate feedback information for each destination node. Therefore, the use of the Multifeedback Learning Automata (MLA) which are presented in Section II, is the most appropriate one for this kind of environment. A Multifeedback Learning Automaton is placed at each node of the network. During each slot the automaton is fed with a set $\mathbf{b}(t)=\{b_1(t),b_2(t),..,b_n(t)\}$ of n feedbacks, with:

$b_j(t)=0$ (PENALTY) if a receiver collision has occurred at the destination node j during the $t-t_d$ time slot (j=1,2,...n).

$b_j(t)=1$ (REWARD) if no receiver collision has occurred at the destination node j during the $t-t_d$ time slot (j=1,2,...,n).

Since, the feedback information is common, all the automata throughout the nodes, always contain the same probability distribution. Thus, in order to simplify the notation, $P_{i,j}(t)$ is replaced by $P_j(t)$, and relation (2) is written as:

$$r_{i,j}(t) = \frac{P_j(t)}{\sum_{k \in D_i(t)} P_k(t)} \quad \text{for } j \in D_i(t) \quad (3)$$

According to the probability updating scheme of the MLA learning automaton which has been presented in Section II, we have (for $j=1,2,\ldots,n$):

$P_j(t+1) = P_j(t) - L\ P_j(t)$ if a receiver collision has occurred at the destination node j during the $t-t_d$ time slot.

$P_j(t+1)=P_j(t) + L\left(\frac{1}{|K(t)|} - P_j(t)\right)$ if no receiver collision has occurred at the destination node j during the $t-t_d$ slot.

where L is the "step size" parameter of the automaton and $K(t)=\{\text{node } k \mid b_k(t)=1\}$

Since the number of wavelengths is equal to the number of nodes, it follows that $\lceil n/2 \rceil \leq |K(t)| \leq n$ and consequently, $|K(t)| \neq 0$ for all t.

The occurrence of a receiver collision at a destination node j allows us to predict that the probability of occurring a receiver collision at the same node in the near future is high. This is due to the following two reasons:

1) The occurrence of a receiver collision at a destination node j implies that probably a large number of packets destined for this node are waiting for transmission at various source nodes.

2) All the packets, that take part in the receiver collision, except one, are rescheduled for transmission and consequently, they are capable of creating a new receiver collision at the same destination node in the near future.

Therefore, the choice probability of a node, where a receiver collision has been occurred, must be decreased in order to avoid loosing other packets, due a probable new receiver collision at the same destination node.

According to the learning algorithm which is presented above, the probability of choosing a destination node decreases when a receiver collision occurs at this node. Consequently, the number of packet losses due to receiver collisions is drastically decreased. In this way, the number of the retransmitted packets is reduced and consequently, both the delay vs throughput and the throughput vs offered load performance of the network is improved. Simulation results that confirm the above discussion are presented in Section VI.

V. PERFORMANCE ANALYSIS

The main factor which affects the performance of the proposed RCALA protocol is the choice of the probabilities with which each destination node is selected by the automata.

One could expect that the destination node, which has the minimum probability of receiver collision, must be selected with probability one, in order to optimize the performance of the network. Although, this proposal seems to be logical, it is wrong. If each source node would choose this destination node with probability one, then, the receiver collision probability of this node would be equal to the unity and consequently, this choice would be the worst one, rather than the optimal one!

It can be proved, that the receiver collision probability of each destination node is a monotonically increasing function of the probability with which this node is selected by the automata. The above proposition is formally expressed by the following theorem:

Theorem 1: If $c_j(t)$ is the probability that a receiver collision will occur at the destination node j, at time instant t, then, for a given distribution of packets waiting for transmission at the source nodes, $c_j(t)$ is a monotonically increasing function of the "destination node transmission probability", $P_j(t)$.

Proof: An outline of the proof is given in the Appendix.

According to the proposed RCALA protocol, the choice probability of each destination node, asymptotically tends to be proportional to the probability that no receiver collision will occur at this destination node. In other words, if the receiver collision probability of a destination node is relatively high, due to the existence of a large number of packets destined for this node, then the RCALA protocol tends to decrease the choice probability of this destination node. The formal expression of the above discussion is given by the following theorem:

Theorem 2: If $d_j(t)$ is the probability that no receiver collision will occur at the destination node j, at time instant t, for j=1,2,...,n, then, for a given distribution of packets waiting for transmission at the source nodes, for any two destination nodes i and j, the RCALA protocol asymptotically tends to satisfy the following relation:

$$\frac{P_i(t)}{P_j(t)} = \frac{d_i(t)}{d_j(t)}$$

Proof: An outline of the proof is given in the Appendix.

VI. SIMULATION RESULTS

In the following, the proposed RCALA protocol is compared to the well-known DT-WDMA protocol which is introduced in [3]. Protocol DT-WDMA was chosen for the following two reasons:

1) It is the most efficient access protocol reported in the literature up to now, for the architectural form under consideration. The superiority of the DT-WDMA among all the other well-known protocols has been demonstrated in [3].

2) The proposed Receiver Collisions Avoidance Learning Algorithm is applied on the basic DT-WDMA architectural form. Therefore, a performance comparison between the two protocols will clearly demonstrate the performance improvement which is due to the use of the proposed Receiver Collisions Avoidance Learning Algorithm.

Both protocols were simulated to operate in WDM broadcast-and-select star networks of the architectural form described in the previous sections.

We have used the following two broadly used performance metrics in order to compare the RCALA protocol with the DT-WDMA one:

1) The delay versus throughput characteristic.

2) The throughput versus offered load characteristic.

The protocols under comparison were simulated to operate in three different networks of the above described form. In all cases the total bandwidth was taken to be equal to 10 Gbps, while the packet size was equal to 1000 bits.

The first simulated network N1 connects 10 users and has a round-trip propagation delay from each node to the network hub and back equal to 4 slots. The queue size of each node is taken to be equal to 5 packets. In the RCALA case, the step size parameter L of the automata is taken to be equal to 0.30.

The second simulated network N2 connects 20 users. The round-trip propagation delay is equal to 8 slots, while the queue size is equal to 10 packets. Parameter L is equal to 0.25.

Finally, the third network N3 connects 30 users. The round-trip propagation delay is equal to 12 slots, while the queue size is equal to 15 packets. The step size parameter L is equal to 0.20.

The delay versus throughput characteristics of the RCALA and the DT-WDMA protocols when they are applied to the networks N1, N2 and N3, are appeared at figures 3, 5 and 7, correspondingly.

For low values of throughput, the mean packet delay of the two compared protocols is almost the same. Under low throughput conditions, the number of receiver collisions is very low and consequently, the use of a receiver collisions avoidance scheme does not result to a serious decrease of the mean packet delay.

For medium and high values of throughput, the use of the proposed Receiver Collisions Avoidance Learning Algorithm leads to a significant decrease of the mean packet delay. A network which operates under medium or high throughput conditions, suffers from a large number of receiver collisions. The use of the proposed Receiver Collisions Avoidance Learning Algorithm leads to a significant decrease of the number of receiver collisions. Therefore, the mean number of retransmissions of a packet before its successful transmission is seriously reduced. Consequently, the mean packet delay is also reduced.

The throughput versus offered load characteristics of the two compared protocols when they are applied to the networks N1, N2 and N3 are appeared at figures 4, 6 and 8, correspondingly.

From these graphs, it can be derived that the use of the proposed Receiver Collisions Avoidance Learning Algorithm leads to a significant increase of the achievable throughput when the network operates under medium and high load conditions. When the RCALA protocol is used, the number of receiver collisions is reduced and consequently, the number of successful transmissions per time slot is maximized.

The main result of the above discussion is that the use of the Receiver Collisions Avoidance Learning Algorithm leads to the reduction of the number of receiver collisions, and consequently, to a significant improvement of the network's performance.

VII. CONCLUDING REMARKS

The receiver collisions problem of WDM broadcast-and-select star networks has been a limiting factor of their performance.

In this paper we presented an easily implementable, learning automata based algorithm which reduces the number of receiver collisions and improves the performance of the network.

An analysis of the asymptotic behavior of the proposed Receiver Collisions Avoidance Learning Algorithm was presented. It was proved that according to the proposed algorithm, the choice probability of each destination node, asymptotically tends to be proportional to the probability that no receiver collision will occur at this destination node.

Furthermore, extensive simulation results were presented, which indicate that a significant performance improvement is achieved when the proposed algorithm is used.

VIII. REFERENCES

[1] C.A.Brackett, "Dense Wavelength Division Multiplexing Network: Principles and Applications", IEEE Journal on Selected Areas in Communications, vol.8, no.6, pp.948-964, August 1990.

[2] P.S.Henry, "Very High Capacity Lightwave Networks", IEEE ICC 1988, pp.1206-1209.

[3] M.S.Chen, N.R.Dono and R.Ramaswami "A Media Access Protocol for Packet Switched Wavelength Division Multiaccess Metropolitan Network", IEEE Journal on Selected Areas in Communications, vol.8, no.6, pp.1048-1057, August 1990.

[4] K.S.Narendra and M.A.L.Thathachar, "Learning Automata: An Introduction", Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[5] M.A.L.Thathachar and B.R.Harita, "Learning Automata with Changing Number of Actions", IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-17, no.6, pp.1095-2000, November/December 1987.

[6] G.I.Papadimitriou, "A new approach to the design of reinforcement schemes for learning automata: Stochastic Estimator Learning Algorithms", IEEE 1991 International Conference on Tools for Artificial Intelligence (TAI'91), November 5-8, 1991, San Hose California, pp.308-317.

[7] G.I.Papadimitriou et al., "Virtual Circuit Routing Algorithms Using Discretized Estimator Learning Automata for High-Speed Packet-Switched Networks", IFIP TC6/WG6.4 Third International Conference on High Speed Networking, March 18-22, 1991, Berlin, Germany, pp.309-323.

[8] M.F.Norman, "Markov Processes and Learning Models", New York and London, Academic Press, 1972.

[9] 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, pp.262-269, May 1980.

APPENDIX

Proof of Theorem 1 (Outline):

For each node j, the set $S_j(t)$ of the nodes which have packets waiting for transmission destined for node j at time instant t is defined as:

$$S_j(t) = \{ \text{ node } k \mid Q_{k,j}(t) \text{ is not empty } \}$$

The probability that a receiver collision will occur at the destination node j at time instant t, is:

$$c_j(t)=1- \Pr[\text{ no transmissions to node } j] - \Pr[\text{ one transmission to node } j] =$$

$$= 1 - \prod_{i \in S_j(t)} (1-r_{i,j}(t)) - \sum_{k \in S_j(t)} r_{k,j}(t) \prod_{\substack{i \in S_j(t) \\ i \neq k}} (1-r_{i,j}(t)) \qquad (A.1)$$

Now, we assume that all $r_{i,j}(t)$ for $i \in S_j(t)$ increase by a positive real number δ_i. Thus, $r'_{i,j}(t)=r_{i,j}(t)+\delta_i$ where: $\delta_i>0$. The new collision probability $c'_j(t)$ is given by relation (A.1), if $r_{i,j}(t)$ (for $i \in S_j(t)$) are substituted by $r_{i,j}(t) + \delta_i$. After some simple algebraic computations its is proved that:

$$c'_j(t) > c_j(t) \qquad (A.2)$$

Since, definition (3) guarantees that any increase of $P_j(t)$ leads to an increase of all $r_{i,j}(t)$ for $i \in S_j(t)$, and consequently (from relation (A.2)) to an increase of $c_j(t)$, it follows that the receiver collision probability $c_j(t)$ is a monotonically increasing function of $P_j(t)$.

Proof of Theorem 2 (Outline):

If $d_i(t)$ is the probability that no receiver collision will appear at the destination node i, then according to the proposed learning algorithm we have:

$$E[P_i(t+1)] = P_i(t) + (1-d_i(t)) (-L P_i(t))+ + d_i(t) L \left(\frac{1}{|K(t)|} - P_i(t) \right) =$$

$$= P_i(t) + L \left(-P_i(t) + \frac{d_i(t)}{|K(t)|} \right)$$

By using Theorem B of [9] it is proved that $\mathrm{var}[P_i(t)] = O(L)$ for all $t \geq 0$. We assume that the step size parameter L takes small values, close to 0. We have: $L \to 0 \Rightarrow \mathrm{var}[P_i(t)] \to 0$. Therefore,

$$P_i(t+1) = E[P_i(t+1)] =$$

$$= P_i(t) + L \left(-P_i(t) + \frac{d_i(t)}{|K(t)|} \right) \qquad (A.3)$$

Now, we prove that $P_i(t)\}_{t\geq 0}$ is a **Distance Diminishing Stochastic Process** by using the Criterion of Norman [8]. According to this criterion we have to show that for any $P_i(t)$ and $P'_i(t)$ with $P_i(t) \neq P'_i(t)$ the following inequality is satisfied:

$$\frac{| P_i(t+1) - P'_i(t+1) |}{| P_i(t) - P'_i(t) |} < 1$$

By using relation (A.3) and some simple algebraic computations, it is derived that the above inequality is satisfied when:

$$|K(t)|\ \frac{L-2}{L} < \frac{d_i(t)-d'_i(t)}{P_i(t)-P'_i(t)} < |K(t)| \qquad (A.4)$$

From Theorem 1 it is derived that the probability $d_i(t)$ that no receiver collision will occur at a node i is a monotonically decreasing function of $P_i(t)$. Thus, we can write:

$$d_i(t) = 1 - c_i(t) = f_i(P_i(t))$$

where $f_i(.)$ is a monotonically decreasing function of its argument. Therefore, relation (A.4) can be written as follows:

$$|K(t)|\ \frac{L-2}{L} < w < |K(t)| \qquad (A.5)$$

where w is the gradient of the line $y=f_i(x)$. From Theorem 1 we have that w is negative. Consequently, the right hand side of the above inequality is always satisfied. Furthermore, since L can be arbitrarily small, we can select a small enough value of L such that the left hand side of the above inequality is satisfied.

Therefore, for small enough values of L, inequality (A.5) is satisfied and $\{P_i(t)\}_{t\geq 0}$ is a **Distance Diminishing Stochastic Process**. Consequently, as t tends to infinity, $\{P_i(t)\}_{t\geq 0}$ converges to a random variable which is independent of the initial value $P_i(0)$.

Thus, $P_i(t+1)=P_i(t)$ and consequently, relation (A.3) gives:

$$L\left(-P_i(t) + \frac{d_i(t)}{|K(t)|}\right) = 0$$

which proves that:

$$P_i(t) = \frac{d_i(t)}{|K(t)|}$$

Since, the above relation holds for every destination node i, it follows that for any two destination nodes, i and j we have:

$$\frac{P_i(t)}{P_j(t)} = \frac{d_i(t)}{d_j(t)}$$

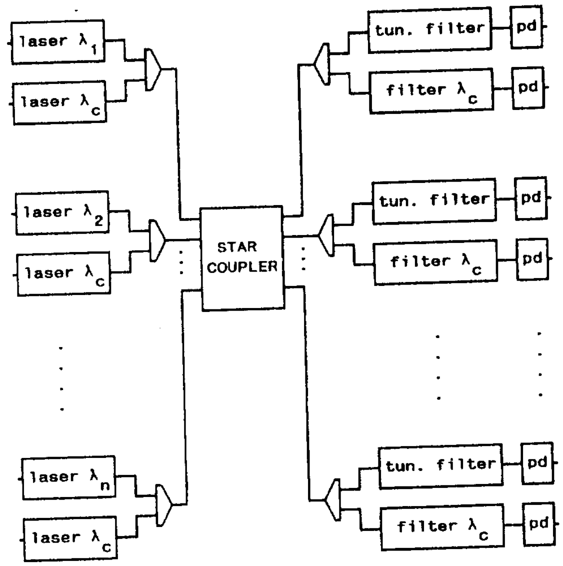

Fig.1: The DT-WDMA Passive Star Network.

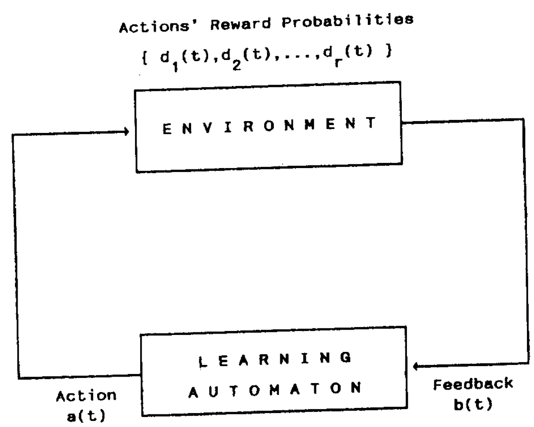

Fig.2: A Learning Automaton that interacts with a stochastic environment.

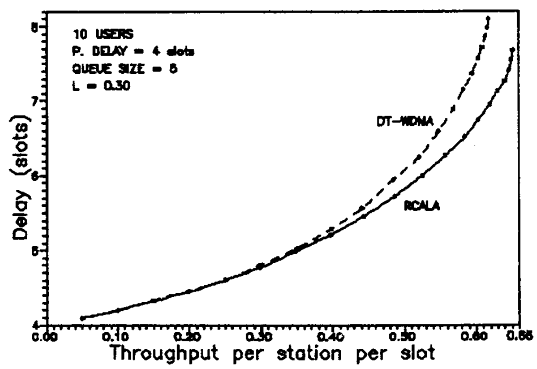

Fig.3: Delay vs Throughput (Network N1).

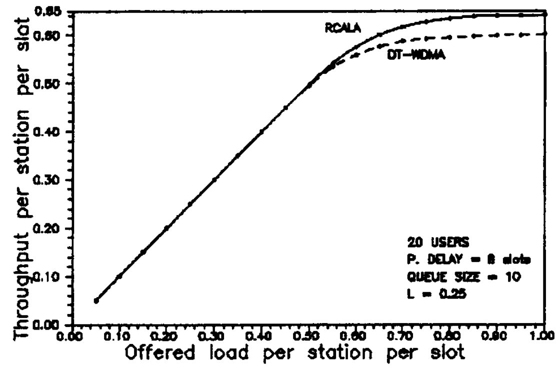

Fig.6: Throughput vs Load (Network N2).

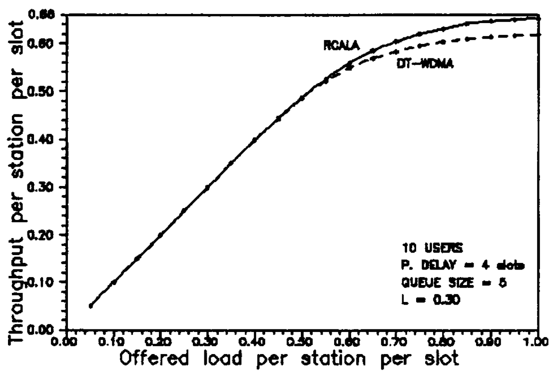

Fig.4: Throughput vs Load (Network N1).

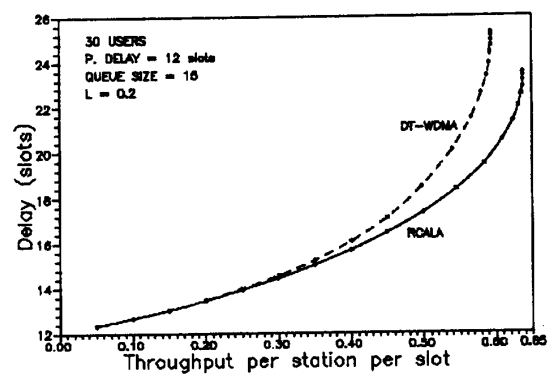

Fig.7: Delay vs Throughput (Network N3).

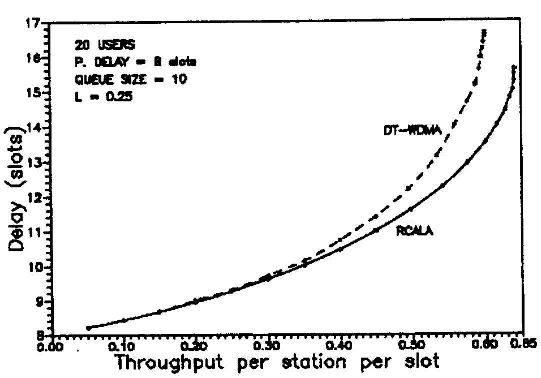

Fig.5: Delay vs Throughput (Network N2).

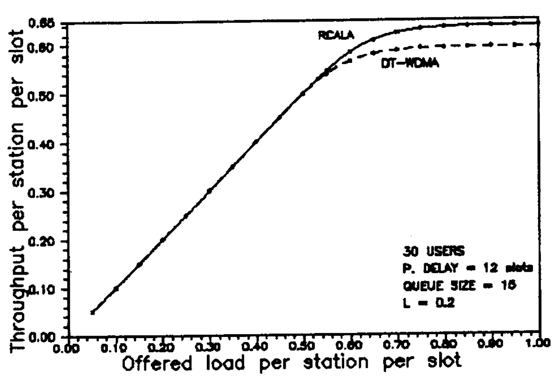

Fig.8: Throughput vs Load (Network N3).