A Novel Burst Assembly Algorithm for Optical Burst Switched Networks Based on Learning Automata

T. Venkatesh, T. L. Sujatha and C. Siva Ram Murthy*

Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai - 600036, India
{tvenku,lsujatha}@cse.iitm.ernet.in, murthy@iitm.ac.in

Abstract. Optical Burst Switching (OBS) is widely believed to be the technology for the future core network in the Internet. Burst assembly time at the ingress node is known to affect the traffic characteristics and loss distribution in the core network. We propose an algorithm for adapting the burst assembly time based on the observed loss pattern in the network. The proposed Learning-based Burst Assembly (LBA) algorithm uses learning automata which probe the loss in the network periodically and change the assembly time at the ingress node to a favorable one. We use a discrete set of values for the burst assembly time that can be selected and assign a probability to each of them. The probability of selecting an assembly time is updated depending on the loss measured over the path using a Linear Reward-Penalty (L_{R-P}) scheme. The convergence of these probabilities eventually leads to the selection of an optimal burst assembly time that minimizes the burst loss probability (BLP) for any given traffic pattern. We present simulation results for different types of traffic and two network topologies to demonstrate that LBA achieves lower BLP compared to the fixed and adaptive burst assembly mechanisms existing in the literature.

1 Introduction

Wavelength Division Multiplexing (WDM) technology has become widely popular for deployment in the core network to meet the ever-increasing demand for bandwidth in the Internet. In WDM networks, a single fiber can support concurrent transmission of multiple wavelengths resulting in complexity of switching compared to the traditional networks. There are three important switching paradigms in WDM networks, namely, optical circuit switching, optical packet switching (OPS) and optical burst switching (OBS). Among these, OBS has attracted attention of the researchers due to its advantages among the other paradigms [1].

In an OBS network, IP packets are assembled into bursts at the ingress node and a control burst is sent out before each data burst. The control burst

* Author for correspondence. This work was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD Fellowship Award.

carries necessary information about the arrival time and the duration of the burst. Wavelengths are reserved only for the duration of the bursts. A time gap, called *offset time*, is maintained between the control burst and the data burst to enable reservation. At the egress node, the bursts are disassembled into IP packets again. Due to the dynamic reservation mechanism bursts are dropped without any information to the ingress node, whenever contention occurs at the core nodes. Thus, contention becomes the main source of the losses in an OBS network.

There are several factors responsible for contention losses in OBS networks such as, burst assembly, scheduling algorithm, offset time and routing and wavelength assignment algorithms. Though several solutions have been proposed for contention resolution in OBS networks (see [2] for a survey on this work), it is an active area for research due to inherent complexity of the problem. The main objective of research in OBS networks is often reduction of the burst loss probability (BLP) which is done by careful choice of mechanisms for the aforementioned factors. Burst assembly time at the ingress node is an important parameter in OBS networks that can affect the dynamics of the network. The assembly mechanism changes the burst size, inter-arrival time, distribution of traffic at the core node and hence the loss distribution [3].

In this work we try to reduce the BLP by a learning-based burst assembly mechanism at the ingress node instead of the fixed assembly schemes used earlier. The motivation for this work arises from the observation made in [4] that an adaptive burst assembly algorithm that considers the variation in the traffic, outperforms the schemes that set assembly time independent of the traffic. We use the loss rate in the path as a feedback to learn appropriate assembly time over the time and hence, achieve reduction in the BLP. The rest of the paper is organized as follows. Section 2 gives some links to the work in the literature on the burst assembly mechanisms and provides the motivation for our work. We give a brief introduction to the theory of Learning Automata and the Linear Reward Penalty (L_{R-P}) scheme in Section 3. In Section 4, we propose the Learning-based Burst Assembly (LBA) algorithm and demonstrate the improvement in BLP through simulations in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work and Motivation

The impact of burst aggregation on the BLP, primarily for TCP traffic, has been studied in the literature earlier [4], [5], [6] and it was observed that assembly time affects the BLP as well as throughput in the network. This is primarily because, the size of the bursts as well as their inter-arrival times at the core node depend on the assembly mechanism used at the ingress node. The impact of burst assembly time on the TCP throughput was studied in [7]. The existence of an optimal burst assembly time for which the TCP throughput is maximum was proved theoretically as well as through simulations.

Intuitively there are two ways to aggregate the bursts: either define a fixed burst size or define a burst aggregation time. In [4], the authors studied the effect of a time-based Fixed Assembly Period (FAP) algorithm and a combination of size-based and time-based algorithm, called Min-Burst Length-Max-Assembly-Period (MBMAP) algorithm on TCP traffic. Since the FAP is independent of the traffic rate, it was found to adversely impact the network performance. Therefore, the authors of [4] proposed an Adaptive Assembly-Period (AAP) algorithm that can dynamically change the value of assembly period (AP) at every ingress node according to the length of burst recently sent. It was shown that AAP is best among the three assembly algorithms, because it matches with the TCP rate control mechanism.

However, the AAP algorithm can be defined only for TCP traffic. It assumes that the rate of the incoming traffic varies linearly and uses an equation to adjust the assembly period proportional to the average burst length. The recently arrived burst is given a higher weight while computing the assembly period, to enable synchronization of the assembly algorithm with TCP. Such a scheme works well mainly with a linearly increasing traffic like TCP but not with other traffic sources like, ON/OFF, CBR or a combination of these. For non-TCP traffic, adapting burst assembly according to the arrival rate is infeasible because, the traffic does not follow a regular pattern. Instead of using rate of the traffic to change the assembly time, we observe the variation in the BLP along the path to learn the optimal assembly time. Since it was proved in [8] that any algorithm that arrives at an optimal policy over time through exploration outperforms the schemes based on fixed policy, we propose a learning automata-based solution for the burst assembly.

3 Learning Automata

In a learning automata system, a finite number of actions can be performed in a random environment. When a specific action is performed, the environment provides either a favorable or an unfavorable random feedback. The objective in the design of the learning automata system is to determine how the previous actions and responses should affect the choice of the current action to be taken, and to improve or optimize some predefined objective function. A learning automaton comprises of a learning module which learns from the feedback provided by the environment, and a decision module that makes decisions based on the assimilated knowledge of the environment. At any stage, the choice of action could be either deterministic or stochastic. In the latter case, probabilities are maintained for each possible action to be taken which are updated with each response from the environment.

A learning automaton can be formally described as a triple $\{\alpha, \beta, A\}$, where $\alpha(n)$ is the output or action of the automaton at time instant n, $\beta(n)$ is the input to the automaton at time instant n, and A is called the *updating algorithm* or the *reinforcement scheme*. The updating algorithm determines the $\alpha(n+1)$ in terms of the network state and $\beta(n)$, and could be either deterministic or stochastic. Determining the updating algorithm for a stochastic automaton makes for a very important design choice. The updating function can be either linear or non-linear. Well known linear updating functions include reinforcement schemes such as the Linear Reward-Penalty (L_{R-P}) scheme, the Linear Reward-Inaction (L_{R-I}) scheme, and the Linear Reward-ϵ-Penalty ($L_{R-\epsilon P}$) scheme [9]. The objective of the updating function is to enable the automaton to *learn* the state of the environment based on the feedback obtained and choose the best possible action at any point of time. It should be able to efficiently guide the automaton to quickly adapt to the changes in the environment. The updating function needs to be simple, and yet efficient, especially when the environment is known to change rapidly. For a detailed description of learning automata, readers are referred to [9] and references thereof.

The L_{R-P} Scheme: For the burstification problem, we use the L_{R-P} scheme [9] to update the automaton action probabilities. When a positive response is obtained for an action, its probability is increased and the probabilities of all the other actions are decreased. If a negative feedback is received for an action, the probability of that action is decreased and that of others is increased. For a multi-action system with S states, the updating algorithm can be written as follows, where $P_i(n)$ represents the probability of choosing action i at time instant n, and a and b are the reward and penalty parameters, respectively:

- When a positive feedback is obtained for action i,

$$P_i(n+1) = P_i(n) + a(1 - P_i(n))$$
$$P_j(n+1) = (1-a)P_j(n), j \neq i$$

- When a negative feedback is obtained for action i,

$$P_i(n+1) = (1-b)P_i(n)$$
$$P_j(n+1) = \frac{b}{S-1} + (1-b)P_j(n)), j \neq i$$

and similarly for all $i = 1, 2, \ldots, S$, where $0 < a < 1, 0 \leq b < 1$. In our case, an increased or decreased BLP are the two kinds of feedback returned to the automaton residing on an ingress node for the action of selecting a burst assembly time. Convergence of the S-model L_{R-P} scheme has been discussed in detail in [9].

4 Learning-based Burst Assembly Mechanism

Since the action selection is only made at the ingress node, the learning automata are placed only at the edge nodes. The burst manager at the ingress node and the learning agent comprise the decision module which decides the assembly time to be used. For every time period T, the learning agent decides the action to be chosen which here corresponds to selecting a burst assembly time from a discrete set of values. The bursts for the next period are transmitted with the selected assembly time. The BLP value for each ingress-egress pair obtained from the acknowledgment packet is used as the feedback from the environment.

Algorithm 1 Learning-based Burst Assembly Algorithm

```
Let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ be the vector of assembly times and $i$ be the index of the burst assembly time chosen.
Let $P_i$ be the probability of selecting the $i$th burst assembly time, $\mathcal{T}_i$.
Let $P_i = 1/n$ be the initial probabilities $\forall$, $1 \leq i \leq n$ .
        if (loss = zero) then
            $P_i = P_i + a * (1 - P_i)$
                for all $\mathcal{T}_j$, $\ni j \neq i$
                $P_j = 1 - a * P_j$
        endif
        elseif (loss > zero) then
            $P_i = (1 - b) * P_i$
            for all $\mathcal{T}_j$, $\ni j \neq i$
                $P_j = \frac{b}{n-1} + (1 - b) * P_j$
            endfor
        endif
where $a$ is the reward and $b$ is the penalty.
```

Depending on whether the response is favorable or not (decreased/increased BLP), the next assembly time (action) is selected. The period T for which the assembly time is changed, is selected such that the overhead incurred in collecting the feedback is minimized and any transient network conditions are ignored by the agent. A low value of T increases the probe overhead and also leads to transient conditions due to frequent changes in burst size. However, if T is set to a large value the response to congestion is slow but the overhead is low. Hence, T can be selected judiciously depending on the burstiness of the traffic. We assume that the time taken to collect feedback is much smaller than the time of adaptation. This is valid since the probe packets take negligibly small time for transmission and processing compared to the period in which decision is made.

We assume that a set of values for burst assembly time is available at the burst manager. Initially all the values have equal probability. The main idea in this mechanism, is to estimate the BLP for a selected burst assembly time and use the one with lowest BLP. Due to the L_{R-P} updation scheme, the assembly time corresponding to the lowest BLP has the highest probability. The estimated BLP is initially set to a predefined value. Each node in the network maintains the loss information for every connection (traffic request between an ingress node and an egress node). When a burst loss occurs at the node, the local BLP value for that flow is updated. The agent at every ingress node periodically sends a probe packet to the egress node. The probe packet collects information on the BLP along the path and the egress sends a negative feedback through acknowledgment packet if the BLP is higher than a threshold.

When a burst loss occurs at the node, the local BLP value for that flow is updated and this information is sent to the destination along with the probe packet. When the BLP is higher than a threshold, the source gets a negative feedback. On receiving the BLP value, the source node updates the probability for the corresponding burst assembly time using the L_{R-P} scheme and accordingly selects the assembly time for the next burst. We reserve a control wavelength for the probe and acknowledgment packets and assume no losses in control plane. This also avoids contention losses of control bursts. To avoid the implementation complexity at the core node to generate a control packet, we assume that the feedback is only sent from the egress node and the core nodes along the path update the cumulative BLP in the probe packet.

The LBA algorithm that uses L_{R-P} equations given in Section 3 to update the probability associated with a value of assembly time, depending on the resulting BLP is given in Algorithm 1. This problem wherein an agent selects an action from a finite set at each time instant to optimize an objective function over the long run is similar to the multi-armed bandit problem in the literature of reinforcement learning [10] for which learning automata provides an optimal solution. This is the motivation to design a learning automata-based solution for burst assembly. Here the reward for an action (selecting a burst assembly time) is characterized by a decreasing BLP value. So the learning automaton tries to select the optimal burst assembly time that minimizes the BLP over a period of time. Every time it selects an assembly time, depending on whether the BLP increases or decreases over the next time period, the agent selects an assembly time that minimizes BLP in the subsequent period.

5 Simulation Results

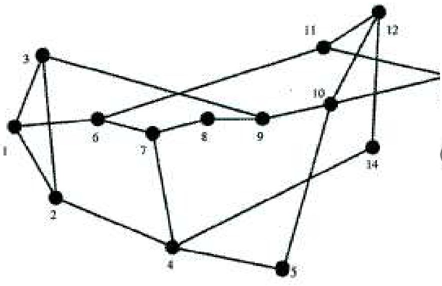


Fig. 1. NSFNET Topology.

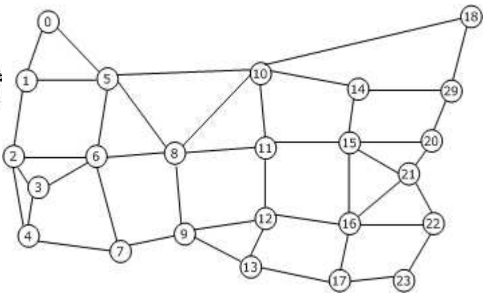


Fig. 2. USIP Backbone Topology.

In this section, we study the performance of LBA algorithm and compare it with AAP and MBMAP algorithms for Just-Enough-Time (JET) reservation scheme. Architectures for core node and edge node as proposed in [11] with Latest Available Unused Channel with Void Filling (LAUC-VF) scheduler were used. We use variation in the BLP with the network load (percentage of the maximum load), as a measure for the performance of the algorithms. A *connection* wherever used means a demand generated between a pair of edge nodes at higher layer that results in a flow of bursts for a time period between them. A set of 20 values equally spaced between 0.1 *ms* and 1 *ms* constitute the action set. The time period of updating assembly time was selected keeping the trade-off between the learning rate and control overhead. We assign +10 for reward and −10 for penalty. We use two topologies, NSFNET topology (Fig. 1) and USIP network (Fig. 2) with 32 *wavelengths per link* (8 control wavelengths) in all our simulations. All the results were obtained at 95% confidence level.

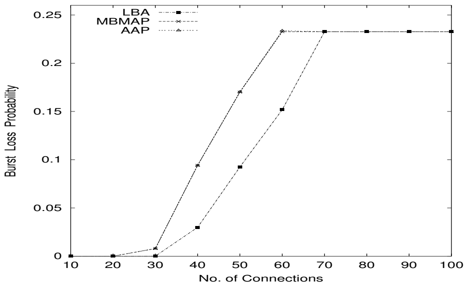


Fig. 3. Variation in the BLP for LBA, AAP and MBMAP as the load increases for CBR traffic on NSFNET.

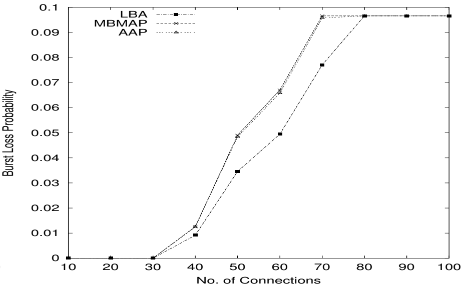


Fig. 4. Variation in the BLP for LBA, AAP and MBMAP as the load increases for CBR traffic on USIPNET.

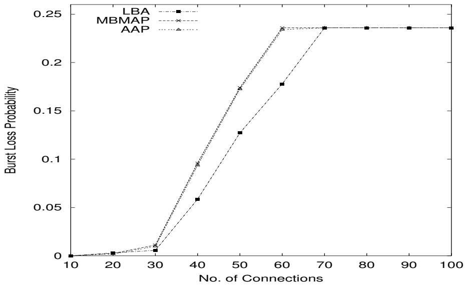


Fig. 5. Variation in the BLP for LBA, AAP and MBMAP as the load increases for ON/OFF traffic on NSFNET.

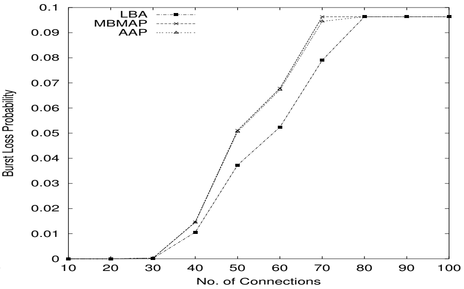


Fig. 6. Variation in the BLP for LBA, AAP and MBMAP as the load increases for ON/OFF traffic on USIPNET.

5.1 CBR Traffic

In this section we use several CBR connections generated randomly between source-destination pairs each with a rate selected randomly between 1 MB and

8 MB. Fig. 3 and Fig. 4 show the variation of BLP as the load increases for NSFNET and USIPNET, respectively. Initially when load is less there is no loss with all the algorithms. But as the load increases, contention losses increase in the network. We can observe that for upto 70% load LBA has a lower BLP than AAP. However, as the load grows beyond, both LBA and AAP have the same BLP. This is because contention losses cannot be controlled by varying the burst assembly time beyond a certain point. At this point contention losses occur only because of lack of wavelengths (severe congestion) and the burst size has lower effect on the contention losses after this.

5.2 ON/OFF Traffic

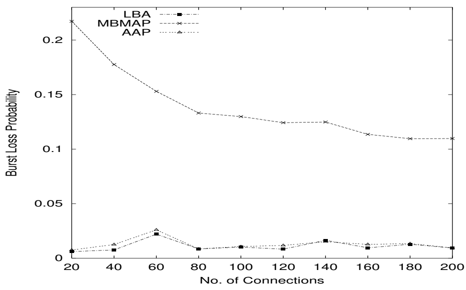
We use ON/OFF traffic with Pareto distribution with shape factor 1.5. Fig. 5 and Fig. 6 compare the BLP for the three algorithms in case of NSFNET and USIPNET, respectively as the load increases. Unlike in CBR traffic, where the packets are sent with constant rate, packets are sent in bursty fashion at certain times. So, even for a low load, there is contention loss in the network. However, we can observe that LBA has lower BLP than AAP. The reason is because the AAP algorithm is designed for TCP traffic and hence does not adapt with losses for any other traffic pattern. Even here we can observe that AAP and MBMAP have the same loss probability for ON/OFF traffic whereas LBA has the lowest BLP among all the schemes.

5.3 TCP Traffic

With increasing percentage load, the variation of BLP for all the three algorithms is compared in Fig. 7 and Fig. 8 for NSFNET and USIPNET topologies, respectively. In CBR and ON/OFF traffic, loss increases as the load increases. But, in TCP it is not the case because TCP controls the rate of flow of packets based on the loss events. However, we can observe that LBA and AAP have almost equal loss probabilities on an average. This shows that LBA performs atleast as good as AAP, which was specially designed for TCP traffic. We can observe that LBA and AAP algorithms are far better than MBMAP algorithm for TCP traffic. Hence, we can conclude that an adaptive assembly algorithm is essential to control the losses in an OBS network.

5.4 Mixed Traffic

In this section, the traffic is composed of all the types of traffic mentioned above (hence called mixed traffic). Fig. 9 and Fig. 10 compare the BLP for LBA, AAP and MBMAP with NSFNET and USIPNET, respectively. We see that LBA algorithm outperforms the AAP algorithm significantly at higher network load. We can observe that upto 30% load, AAP and LBA have almost equal loss probabilities but lower than MBMAP. When the load is between 30% and 60%, the BLP for AAP increases faster than that with LBA. Beyond that AAP cannot



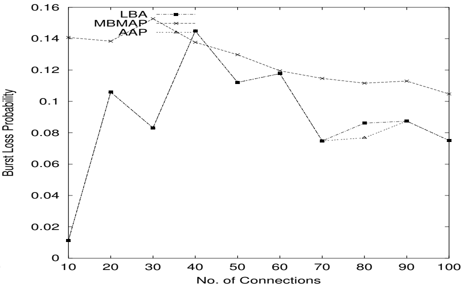
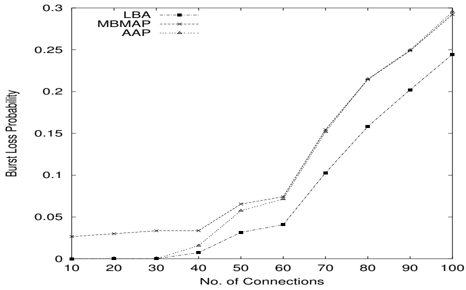


Fig. 7. Variation in the BLP for LBA, AAP and MBMAP as the load increases for TCP traffic on NSFNET.

Fig. 8. Variation in the BLP for LBA, AAP and MBMAP as the load increases for TCP traffic on USIPNET.



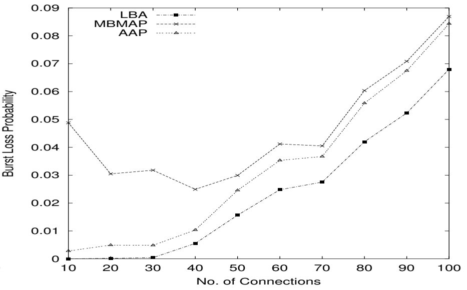


Fig. 9. Variation in the BLP for LBA, AAP and MBMAP as the load increases for mixed traffic on NSFNET.

Fig. 10. Variation in the BLP for LBA, AAP and MBMAP as the load increases for mixed traffic on USIPNET.

control the loss and it is as good as MBMAP. But LBA is better than AAP in spite of the higher loss rate for the mixed traffic type. In [4], the authors show that AAP performs better than MBMAP for TCP traffic. But we can observe from the results here that AAP and MBMAP have same loss rate for CBR and ON/OFF traffic and that AAP is advantageous only for TCP traffic and not for other types of traffic. Even for TCP traffic, LBA performs atleast as good as AAP algorithm.

6 Conclusion

In this paper, we proposed a new algorithm for adaptive burst assembly for OBS networks that use learning automata namely, Learning-based Burst Assembly (LBA). Since it was shown in the literature that the choice of burst assembly time affects both the throughput and the loss rates in OBS networks, we came up with a simple scheme to learn the optimal choice of burst assembly time. Upon convergence, the LBA algorithm learns the optimal burst assembly time for any traffic pattern. We demonstrated through simulations on two networks topologies that BLP is indeed reduced compared to the other adaptive assembly mechanism (AAP algorithm) in the literature. Further, the AAP algorithm available in the literature has advantage only for TCP traffic whereas, the LBA algorithm works well irrespective of the type of traffic (comparable to AAP for TCP). Since AAP algorithm uses the rate of incoming traffic to adapt the assembly time, it is not suitable to be used for non-TCP traffic and has almost same BLP as that of MBMAP algorithm for other traffic types. LBA algorithm performs well independent of the traffic type and captures the explicit dependence of the BLP on assembly time. We justified the motivation for designing a loss-aware assembly mechanism through this work and showed the advantage of an autonomic burst assembly algorithm.

References

[1] Qiao, C., Yoo, M.: Optical Burst Switching (OBS)-a new switching paradigm for optical Internet. Journal of High Speed Networks **8**(1) (1999) 69–84

[2] Farahmand, F., Jue, J.P.: Look-ahead window contention resolution in optical burst switched networks. In: Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR). (2003) 147–151

[3] Detti, A., Listanti, M.: Impact of segments aggregation of TCP Reno flows in optical burst switching networks. In: Proceedings of IEEE INFOCOM. (2002) 1803–1812

[4] Cao, X., Li, J., Chen, Y., Qiao, C.: Assembling TCP/IP packets in optical burst switched networks. In: Proceedings of IEEE GLOBECOM. (2002) 84–90

[5] Malik, S., Killat, U.: Impact of burst aggregation time on performance of optical burst switching networks. In: Proceedings of Optical Network Design and Modeling (ONDM). (2005) 19–25

[6] Hong, D., Poppe, F., Reynier, J., Bacelli, F., Petit, G.: Impact of burstification on TCP throughput in optical burst switching networks. In: Proceedings of 18th International Teletraffic Congress (ITC). (2003) 89–96

[7] Yu, X., Qiao, C., Liu, Y., Towsley, D.: Performance evaluations of TCP traffic transmitted over OBS networks (2003) Tech. Report 2003-13, CSE Dept., SUNY Buffalo.

[8] Littman, M., Boyan, J.: A distributed reinforcement learning scheme for network routing. In: Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications. (1993) 45–51

[9] Narendra, K.S., Thathachar, M.: Learning Automata: An Introduction. Prentice Hall, New Jersey (1989)

[10] Sutton, R.S., G.Barto, A.: Reinforcement Learning : An Introduction. The MIT Press, Cambridge, Massachusetts (1998)

[11] Xoing, Y., Vandenhoute, M., Cankaya, C.: Control architecture in optical burst-switched WDM networks. IEEE Journal of Selected Areas in Communications **18**(10) (2000) 1838–1851