An Effective Deflection Routing Algorithm with Collision Avoidance Mechanism

Seshagiri Rao Ganta[1], Dr N Naga Malleswara Rao[2]
[1]Research Scholar, Department of CSE,
University College of Engineering, Acharya Nagarjuna University, Guntur, India
[2]Professor, Department of Information Technology, RVR and JC College of Engineering, Guntur, India.

Abstract- In the beginning days of internet, reinforcement learning (RL)-based approaches are implemented for routing policy rendering, where routing is a procedure of path choosing for linking couple of end points for transmission of packets. Q-learning is a such algorithm employed for routing policy rendering, which necessitates that nodes attain their determinations of routing locally with lesser computations. However, this sort of algorithms failed to render an optimal policy for routing that reckons the avoidance of collisions which is a major flaw in routing mechanism. Therefore, to address this issue this article proposed an enhanced Q-learning approach by implementing carrier sense multiple access with collision avoidance (CSMA/CA) mechanism to further improve the packet transmission and to mitigate the loss of packet by taking the routing policies appropriately for observing the flow of routing procedure in the buffer-less networks. Extensive simulation analysis shows the effectiveness of proposed approach with comparison to the conventional Q-learning based approaches. Further, the quantitative analysis also discussed with the help of end-to-end delay, energy consumption and throughput performance.

Keywords: *Computer networks, buffer-less networks, deflection routing, reinforcement learning, predictive Q-learning, Collision avoidance.*

I. INTRODUCTION

A prognosticating approach that effectively deal with the burst and active internet protocol traffic the optical networks is known as optical burst switching (OBS) [1], in which the data of user is combined into massive partition named a data burst which is transmitted by employing a unique path resource reservation. In general, burst header packet is defined as a burst that is prefaced in time by a control packet, this will be transmitted on a distinctive control wavelength and desire for assigning of resource at every switch. Accomplishment of the control packet reaches to the switch then there is a reservation of capacity in the cross-connect for the burst. If there is an adequate reservation of capacity at imparted time, then the burst can go across via the cross-connect without necessitate for the procedure or buffering. Therefore, the burst can be dropped due to association of resources or not enough time of offset if the burst grabs up the control packet. Hence, the approaches based on burst disputation acts an essential part to mitigate the ratio of burst loss (BLR) in the OBS networks [2].

In practice, RL-based algorithms validate the procedures of trial and error-based learning and it's four basic elements are:

- Agent or decision maker
- Environment of an agent
- Actions performed by an agent
- feedback signals attained by the environment.

The objective of an RL agent is to maximize/minimize the rewards/penalties that it receives from the environment. The agent remarks the variables of environment which are referred as state of the system and operates an action of rewarding concording to its environment knowledge. Later, environment produces a signal of reinforcement by assessing the action of agent. Next, this signal is employed by an agent to empower its potential to establish conclusions [3]. In [12-15], authors investigated the performance of simple random deflection approaches and their data loss rates. As mentioned earlier, the RL-based algorithms, which were implemented in the beginning days of internet for routing policy rendering are addressed in [4-7]. The Q-learning

Performance of a simple random deflection algorithm and loss rates of deflected data were analyzed [12–15]. RL-based algorithms were proposed in the early days of the Internet to generate routing policies [4–7]. The Q-learning is an algorithm employed for routing policy rendering, which necessitates that nodes attain their determinations of routing locally with lesser computations [4]. However, this sort of algorithms failed to render an optimal policy for routing with lesser loads nor it doesn't instruct novel optimal policies in mitigation of network load situations. Thus, to resolve these limitations by registering the excel experiences discovered in [6], which is known as predictive Q-routing in which the registered excel experiences further reutilized to estimate the behavior of traffic. Later, in [5] author presented an optimal routing policy based on the distributed gradient ascent policy search, where there exists a transmission of reinforcement signals when a packet is returned to its recipient successfully. However, none of the above mentioned RL-based approaches doesn't decide the mechanisms of routing when there is an occurrence of collision in buffer-less networks. Therefore, this article deals with the avoidance of collision with optimal routing policy generation in buffer-less networks. Rest of the paper is as follows: literature survey of the RL-based approaches is discussed in section II. Proposed framework is explained in section III. Section IV is given with the simulation results and discussion of proposed and conventional RL-based algorithms. Finally, section V concludes this work followed by references.

II. LITERATURE SURVEY

There are several researches works that addressed the issue of optimal routing policy generation with Q-learning algorithm. Author in [8] presented RL-deflection routing scheme (RLDRS) approach which utilizes the Q-learning approach for deflection routing in OBS networks with an accurate process of signaling and rewarding. However, this approach suffers from the lack of scalability since the path selection and complexity relies on the network size. In [9], Haeri et al. discussed an algorithm named as Q-NDD that utilizes the Q-learning approach for deflection routing as well. Though it is scalable due to that the complexity relies on the degree of node instead of the size of network as in RLDRS approach. However, this approach suffers from the lack of feedback signaling since it receives them only in the case of disposal of packet by another node. In [10], the authors studied routing and wavelength and timeslot assignment problem for a circuit witched time division multiplexed (TDM) wavelength routed optical WDM network, so as to overcome the shortcomings of non-TDM-based route and wavelength assignment (RWA). The algorithm was applied on a network where each individual wavelength is partitioned in the time domain into fixed-length timeslots organized as a TDM frame. Moreover, multiple sessions are multiplexed on each wavelength by assigning a subset of the TDM slots to each session. In the paper, a set of RWTA algorithms was proposed and evaluated in terms of blocking probability. In those algorithms, shortest path routing algorithm was used for the routing part of the algorithm. Least load wavelength selection scheme was used for wavelength assignment, while a least loaded timeslot technique was proposed for timeslot assignment. The researchers claimed that their proposed RWTA algorithm performs better than random wavelength and timeslot assignment schemes. The disadvantage of the algorithm is the use of shortest path (SP) as routing algorithm, which is a static route selection algorithm making the proposed RWTA not suitable for dynamic traffic of OBS.

The researchers in [11] proposed and evaluated a distributed dynamic RWTA algorithm based on dynamic programming approach. Their goal was to minimize blocking probability. The proposed algorithm consists of three distinct parts; each part solves a sub-problem of the RWTA: routing part and wavelength assignment section and finally timeslot assignment section. The results were compared with SP algorithm and were reported to perform better than that algorithm. The drawback of this solution is the static nature of the routing and the possibility of high delay that was not tested in the paper. In [16], author propose a predictive Q-learning deflection routing (PQDR) algorithm that combines the learning phase of the predictive Q-routing algorithm [6] and the signaling algorithm of the RLDRS [8].

III. PROPOSED FRAMEWORK

This section describes the proposed framework for optimal routing policy generation in buffer-less OBS networks.

3.1. The task for Q-learning

Assume that the computational agent is selected from a finite actions collection at each time stage that is roaming across a distinctive and finite world, which comprises a controlled procedure of Markov with the agent as a controller. The state of world $\boldsymbol{x_n (\in X)}$ is registered at $\boldsymbol{n^{th}}$ step by the equipment of an agent, accordingly their actions $\boldsymbol{a_n (\in \mathcal{A})^1}$. The probabilistic reward $\boldsymbol{r_n}$ is received by an agent whose average value $\boldsymbol{\mathcal{R}_{x_n}(a_n)}$ relies only on the action and state of the world then the world's state altered probabilistically to $\boldsymbol{y_n}$ as followed by:

$$\mathbf{Prob}[\boldsymbol{y_n = y[x_n, a_n] = P_{x_n y}[a_n]}] \qquad (1)$$

Finding an optimized policy is a task of an agent that can maximize the overall ignored reckoned reward. By ignored reward, we mean that rewards received $\boldsymbol{s}$ steps thus are deserving lesser rewards as received at present, by a factor of $\boldsymbol{\gamma_s (0 < \gamma < 1)}$. The state $\boldsymbol{x}$ value underneath $\boldsymbol{\pi}$ policy is expressed as,

$$\boldsymbol{V^{\pi}(x) = \mathcal{R}_x\big(\pi(x)\big) + \gamma \sum_y P_{xy}\,[\pi(x)] V^{\pi}(y)} \qquad (2)$$

As the agent anticipates obtaining $\boldsymbol{\mathcal{R}_x\big(\pi(x)\big)}$ right away for executing the action $\boldsymbol{\pi}$ advocates, subsequently proceeds to a state that is deserve $\boldsymbol{V^{\pi}(y)}$ to it, with a probability of $\boldsymbol{P_{xy}[\pi(x)]}$. The DP theory implemented in 1980's ensured us that there is not less than unitary optimal static policy $\boldsymbol{\pi^*}$ which is such that $\boldsymbol{V^*(x) = V^{\pi^*}(x) = \max_{a}\{\mathcal{R}_x(a) + \gamma \sum_y P_{xy}[a] V^{\pi}(y)\}}$ is in addition to a node of an agent can do from the state $\boldsymbol{x}$. In spite of the fact that this might visible circular, it is considerably determined in reality, and numerous approaches were rendered by DP for computing $\boldsymbol{V^*}$ and single $\boldsymbol{\pi^*}$, considering that there are known $\boldsymbol{\mathcal{R}_x(a)}$ and $\boldsymbol{P_{xy}[a]}$ are known. The task confronting a $\boldsymbol{Q}$ learner is that of finding a $\boldsymbol{\pi^*}$ in the absence of these values. For learning $\boldsymbol{\mathcal{R}_x(a)}$ and $\boldsymbol{P_{xy}[a]}$, several state-of-art approaches are there which also execute DP at the same time, but any premise of certainty comparability, i.e., computing actions as if the present model were exact, costs dearly in the beginning learning stages. Specify values of $\boldsymbol{Q}$ for a $\boldsymbol{\pi}$ policy as:

$$\boldsymbol{Q^{\pi}(x, a) = \mathcal{R}_x(a) + \gamma \sum_y P_{xy}[\pi(x)] V^{\pi}(y)} \qquad (3)$$

Put differently, the value of $\boldsymbol{Q}$ is the anticipated discounted reward for performing action $\boldsymbol{a}$ at state $\boldsymbol{x}$ and adopting policy $\boldsymbol{\pi}$ form that time on. The aim of Q-learning is to reckon the values of $\boldsymbol{Q}$ for an optimized policy. Conveniently, define these as $\boldsymbol{Q^*(x, a) = Q^{\pi^*}(x, a), \forall (x, a)}$. It is free from ambiguity to disclose that $\boldsymbol{V^*(x) = \max_{a} Q^*(x, a)}$ and that if $\boldsymbol{a^*}$ is an action at which the maximum is acquired then formation of optimal policy is done as $\boldsymbol{\pi^*(x) = a^*}$. In this place consists the usefulness of the values of $\boldsymbol{Q}$- if an agent can experience form them, it can simply conclude what it is optimal to do. Even though there are numerous optimal policies or $\boldsymbol{a}^*$, the values of $\boldsymbol{Q}^*$are unique.

In the procedure of Q-learning, the agent's experience comprises of a distinctive sequence steps or episodes. In the $\boldsymbol{n^{th}}$ stage:

- Notices its present state x_n,
- Choose and execute an action a_n,
- Observes the posterior state y_n,
- Attains a contiguous payoff r_n, and
- Alters its values of Q_{n-1}employing a factor of learning α_n, according to:

$$\boldsymbol{Q_n(x,a)} = \begin{cases} \boldsymbol{(1-\alpha_n)Q_{n-1}(x,a) + \alpha_n[r_n + \gamma V_{n-1}(y_n)], x = x_n \text{ and } y = y_n} \\ \boldsymbol{Q_{n-1}(x,a),} \quad \textbf{otherwise} \end{cases} \quad (4)$$

Where,

$$\boldsymbol{V_{(n-1)}(y) = \max_{b}\{Q_{n-1}(y,b)\}}$$

is the excel that the node of agent recalls it can do from the state $\boldsymbol{y}$. Naturally, in the beginning learning stages the values of $\boldsymbol{Q}$ may not exactly reverberate the policy they determine without doubting (the maximizing actions in eq. (2)). The initial values of $\boldsymbol{Q}$ and $\boldsymbol{Q_0(x,a)}$ for all the states and actions are considered given. Note that this explanation considers a look-up table representation for the $\boldsymbol{Q_n(x,a)}$ and demonstrated that the Q-learning may not converge properly for some other representations.

3.2 Calculating Congestion Level (CL)

Consider link l and assume $(\boldsymbol{l,t})$ denotes the set of packets that have successfully traversed link l during a period $\boldsymbol{t}$ and let $\mathbf{size}_{\boldsymbol{i}}$ denotes the size of packet $\boldsymbol{i}$. Then the utilization $\boldsymbol{U}$ of link l is defined as follows:

$$\boldsymbol{U(l,t)} = \frac{\sum_{i\in \mathbf{succ}(l,t)} \mathbf{size}_i}{\boldsymbol{BW(l)}} \quad (5)$$

Where, $\boldsymbol{BW(l)}$ is the bandwidth capacity of link l.

3.3. CSMA/CA framework

Proposed approach shown in figure 1, employed a routing mechanism called CSMA/CA that avoids the collision and enhances the transmission rate of packet by reducing its packet loss. In CSMA/CA, the transmission is hold up by the node when it is found that the broadcast channel is busy, and it waits for a frame with random time which is cited as back-off factor (BOF) that examines the entire network once again to discover that the channel if free or not. Afterwards, the packet with data is transmitted once there is an availability of channel. Further, it sends back an acknowledgement packet (ACK) once the data is received by the recipient. When it doesn't receive the ACK then it is a considered as there is a loss of packet or the packet is discarded without receiving by the recipient then it sets up the retransmission automatically.

Pseudo code

```
S – State of the node, A- action, Q() – Q-table of the nodes
####
Initialize the Q-values table, Q(s, a).
Observe the current state, s.
Choose an action, a,
For all nodes
Read the congestion c,
Observe the reward, r,
Observe the new state, s′.
Update the Q-value for the state using the observed reward
Maximum reward possible for the next state
Set the state to the new state,
Repeat the process
```

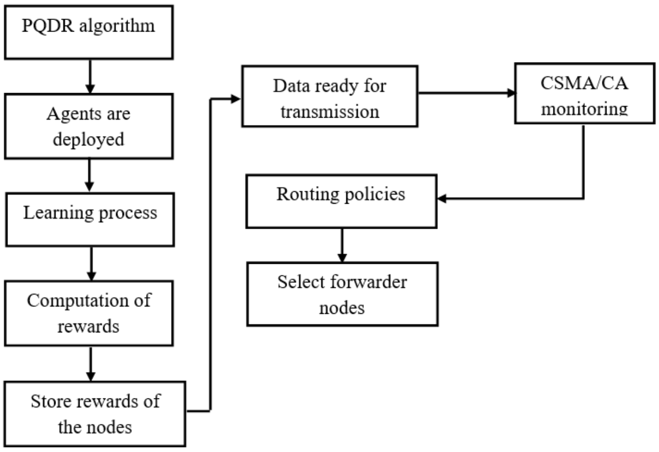


Fig. 1 Architecture diagram of proposed system

As mentioned in figure 1, the nodes in the network are deployed once the PQDR algorithm is applied. In that time, the procedure of learning starts then computes the rewards for every node level and store it in system level. Now, the datta is ready for transmisison. Next, apply CSMA/CA approach to the transmitted data to avoid any collision occurance checking and then employ routing policies. Further, choose the forwarder nodes to take the decisions.

IV. RESULTS AND DISCUSSION

This section explains the simulation results and discussion of proposed RL-based deflection routing mechanism in buffer-less OBS networks with comparison to the existing RL-based approaches. Network simulator 2 (NS2) environment is employed for testing these routing policies, where 30 sensor nodes are randomly distributed over a 1000x500 m^2 field and accept that no gap exists in the detecting field. In addition, static sensors are the same in their abilities. In the meantime, also accept that the destination is located in the top-left corner of the two-dimensional (2-D) territory and its coordinates are (50m, 50m). Collision avoidance PQDR (CA-PQDR) algorithm is employed to conduct numerous experiments in the generated sensing field. According to the network lifetime and the movement of every node, experimental results of the algorithm are presented below. Table 1 shows the system parameters used in simulations.

Table 1: Simulation parameters

PARAMETER	VALUE
Application Traffic	CBR
Transmission rate	1024 bytes /1.0 sec
Radio range	250m
Packet size	1024 bytes
Maximum speed	25m/s
Simulation time	10000ms
Number of nodes	30
Area	1000x500
Routing Protocol	AODV

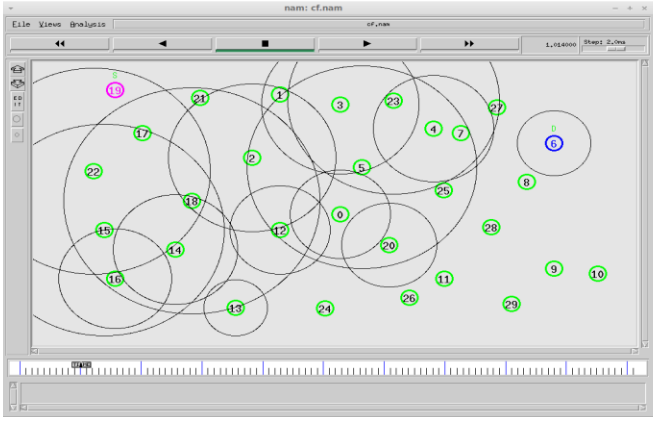


Fig. 2 Network deployment

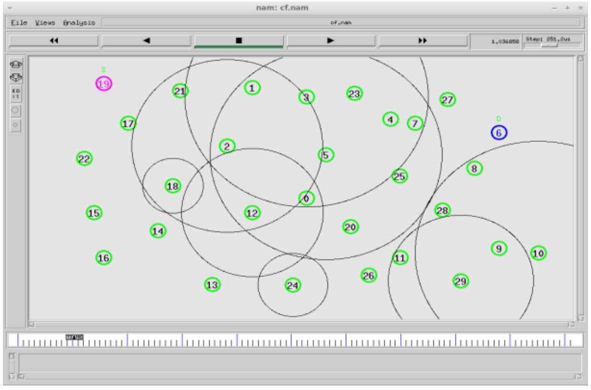


Fig. 3 Broadcasting process in network

Figure 2 shows all nodes placed in network and proper deployment of nodes in the network. Here all nodes displayed based on topology values and all properties of NAM window it should be mentioned. Figure 3 shows the broadcasting occur throughout the network, which occurs for the purpose of communication and all the nodes of network will be involved in this process. Data transmission procedure of network is disclosed in figure 4, where the maximum number of packets are transmitted form the source to destination during the process of communication. In addition, it is shown that the transmitted data with its interval of time through traffic protocol.

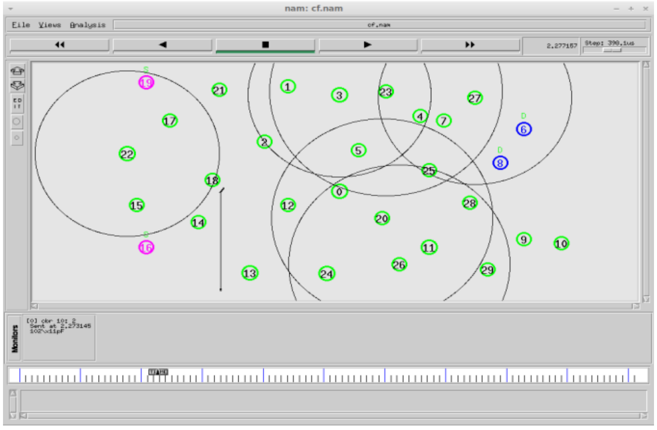
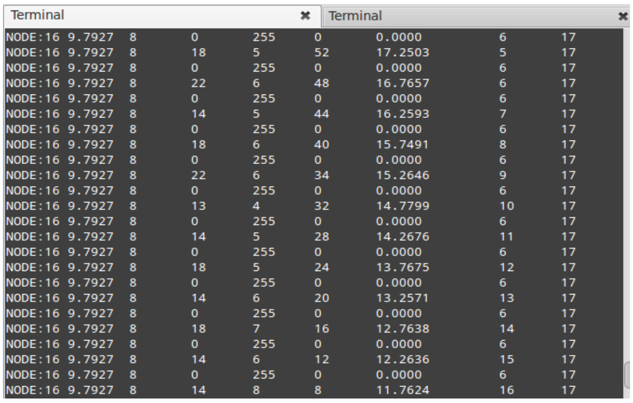


Fig. 4 Data transmission process in network



```
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      18     5      52     17.2503      5      17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      22     6      48     16.7657      6      17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      14     5      44     16.2593      7      17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      18     6      40     15.7491      8      17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      22     6      34     15.2646      9      17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      13     4      32     14.7799      10     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      14     5      28     14.2676      11     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      18     5      24     13.7675      12     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      14     6      20     13.2571      13     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      18     7      16     12.7638      14     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      14     6      12     12.2636      15     17
NODE:16 9.7927  8      0      255    0      0.0000       6      17
NODE:16 9.7927  8      14     8      8      11.7624      16     17
```

Fig. 5 Routing table of node level

Figure 5 represents routing table of users participating in network. In this, the updating values of nodes are represents as per time schedule with their ID of node, current time, destination, next hop, hops, sequence number, expire time, route order and last route. Performance analysis end-to-end delay is demonstrated in figure 6, where the graph represents the end-to-end delay with respect to the simulation time. It is shown that the proposed CA-PQDR obtained an enhanced delay time which led to the mitigation of delay between the nodes of communication. It also shown that the comparison of proposed CA-PQDR with the RLDRS [8] and PQDR [12]. Figure 7 shows and represents the energy consumption with respect to the simulation time. The performance of energy consumption is decreased in proposed CA-PQDR as compared to the conventional RLDRS [8] and PQDR [12] algorithms. Similarly, packet delivery ratio (PDR) and throughput performance of network is disclosed in figure 8 and figure 9 respectively where the proposed CA-PQDR obtained superior PDR and throughput performance over the existing algorithms.

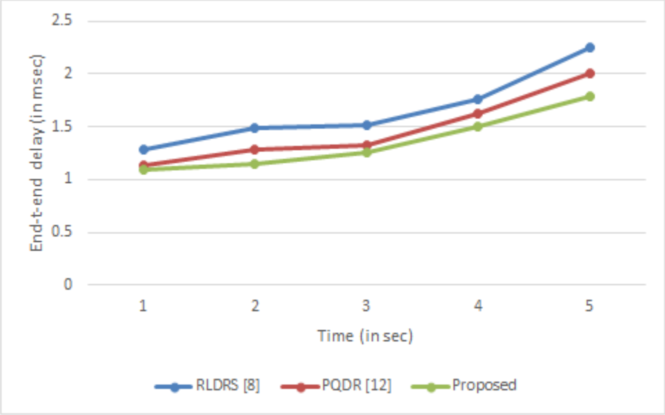


Fig. 6 performance of proposed and existing Q-learning based deflection routing with end-to-end delay

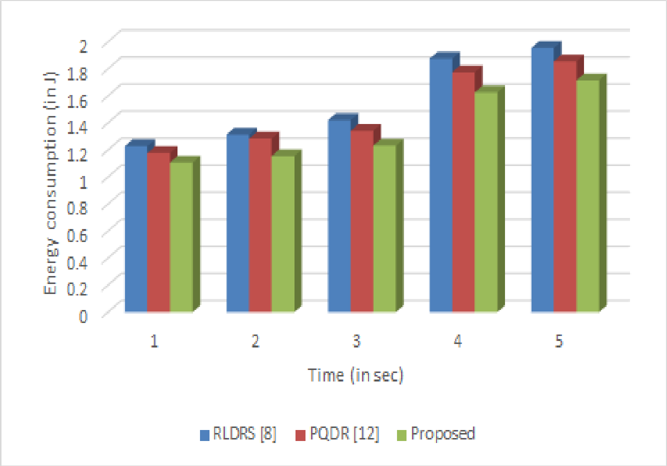


Fig. 7 Energy consumption comparison with the proposed and existing Q-learning based deflection routing algorithms

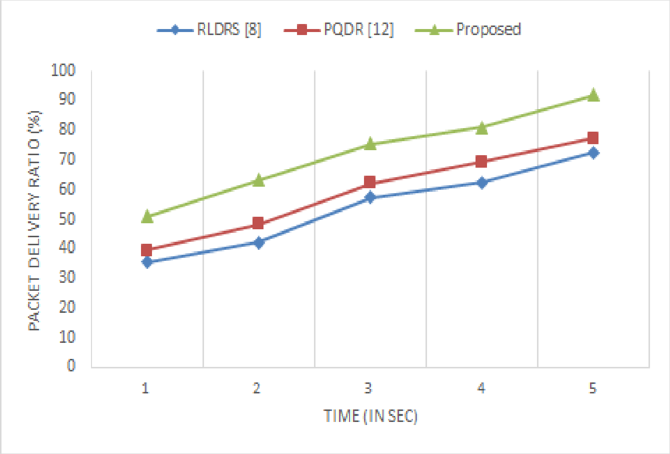


Fig. 8 Comparison with the proposed and existing Q-learning based deflection routing algorithms with packet delivery ratio

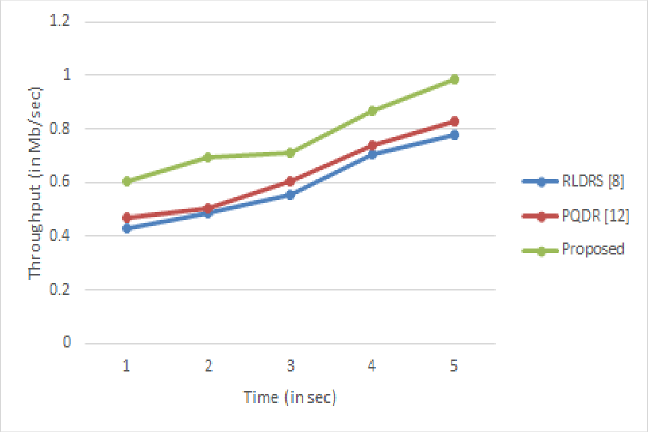


Fig. 9 Throughput performance with the proposerd and existing Q-learing based deflection rotuing algorithms

V. CONCLUSION

This article explained the collision avoidance based predictive Q-learning algorithm for deflection routing in buffer-less networks. Proposed approach combines the Q-routing with the dual RL and CSMA/CA technique and obtained enhanced explorative capabilities where the PQDR algorithm is implemented for faster and enhanced routing mechanism at lower congestion levels. At higher loads, the routing policy learnt by PQDR performs comparatively less in terms of average packet delivery time. Moreover, CA-PQDR can sustain higher load levels than PQDR and shortest-path routing.

REFERENCES

[1] C. Qiao and M. Yoo, "Optical Burst Switching – A New Paradigm for an Optical Internet", Journal of High-Speed Networks, vol. 8, no 1, pp- 69-84, 1999.

[2] Christoph M. Guager, Martin Kohn, and Joachim Scharf, "Performance of Contention Resolution Strategies in OBS Network Scenarios", Proceedings of the 9th Optoelectronics and Communications Conference/3rd International Conference on the Optical Internet (OECC/COIN2004), 2004.

[3] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *J. of Artificial Intell. Research*, vol. 4, no. 5, pp. 237– 285, May 1996.

[4] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," in *Advances in Neural Inform. Process. Syst.*, vol. 6, pp. 671–678, 1994.

[5] L. Peshkin and V. Savova, "Reinforcement learning for adaptive routing," in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, USA, May 2002, vol. 2, pp. 1825–1830.

[6] S. P. M. Choi and D. -Y. Yeung, "Predictive Q-routing: a memory-based reinforcement learning approach to adaptive traffic control," in *Advances in Neural Inform. Process. Syst.*, vol. 8, pp. 945–951, 1996.

[7] A. Nowe, K. Steenhaut, M. Fakir, and K. Verbeeck, "Q-learning for adaptive load-based routing," in *Proc. IEEE Int.*

Conf. Syst., Man, and Cybern., San Diego, CA, USA, Oct. 1998, vol. 4, pp. 3965–3970.

[8] A. Belbekkouche, A. Hafid, and M. Gendreau, “Novel reinforcement learning-based approaches to reduce loss probability in buffer-less OBS networks,” *Comput. Netw.*, vol. 53, no. 12, pp. 2091–2105, Aug. 2009.

[9] S. Haeri, W. W-K. Thong, G. Chen, and Lj. Trajkovi´c, “A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks,” in *IEEE Int. Conf. Inf. Reuse and Integration*, San Francisco, USA, Aug. 2013, accepted for publication.

[10] Wen B, Sivalingam K. Routing, wavelength and timeslot assignment in time division multiplexed wavelength routed optical WDM networks. IEEE INFOCOM. Twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE 2002; 3: 1442–1450.

[11] Yang W, Hall T. Distributed dynamic routing, wavelength and timeslot assignment for bandwidth on demand in agile all-optical networks. In Canadian Conference on Electrical and Computer Engineering, 2006. CCECE ’06., IEEE, Ottawa Ont, 2007; 136–139.

[12] F. Borgonovo, “Deflection routing,” in Routing in Communications Networks. New Jersey: Prentice-Hall, 1995, pp.263–306.

[13] A. Greenberg and B. Hajek, “Deflection routing in hyper-cube networks,” IEEE Trans. Commun., vol. 40, no. 6, pp. 1070–1081, June 1992.

[14] A. Zalesky, H. Vu, Z. Rosberg, E. W. M. Wong, and M. Zukerman, “Stabilizing deflection routing in optical burst switched networks,” IEEE J. Sel. Areas Commun., vol. 25, no. 6, pp. 3–19, Aug. 2007.

[15] E. W. M. Wong, J. Baliga, M. Zukerman, A. Zalesky, and G. Raskutti, “A new method for blocking probability evaluation in OBS/OPS networks with deflection routing,” J. Lightw. Technol., vol. 27, no. 23, pp. 5335–5347, Dec. 2009.

[16]. S. Haeri, M. Arianezhad and L. Trajkovic, "A Predictive Q-Learning Algorithm for Deflection Routing in Buffer-less Networks," *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, 2013, pp. 764-769.