

A Reinforcement Learning-Based Algorithm for Deflection Routing in Optical Burst-Switched Networks

Soroush Haeri
Simon Fraser University
Vancouver, British Columbia, Canada
shaeri@sfu.ca

Wilson Wang-Kit Thong and Guanrong Chen
City University of Hong Kong
Hong Kong SAR, China
wilsonthongwk@gmail.com, gchen@ee.cityu.edu.hk

Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
ljilja@sfu.ca

Abstract

In this paper, we propose a Q-learning based deflection routing algorithm that may be employed to resolve contention in optical burst-switched networks. The main goal of deflection routing is to successfully deflect a burst based only on a limited knowledge that network nodes possess about their environment. Q-learning, one of the reinforcement learning algorithms, has been proposed in the past to help generate deflection decisions. The complexity of existing reinforcement learning-based deflection routing algorithms depends on the number of nodes in the network.

The proposed algorithm scales well for larger networks because its complexity depends on the node degree rather than the network size. The algorithm is implemented using the ns-3 network simulator. Simulation results show that it has comparable performance to an existing reinforcement learning deflection routing scheme while having lower memory requirements.

1. Introduction

Optical burst switching (OBS) is an emerging technology designed to share optical fiber resources across data networks [1]. Current optical switching technologies for data communication such as Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) [2] reserve the entire light-paths from a source to a destination. Even though a light-path is not fully utilized, it may not be shared unless its reservation is explicitly released. The OBS technology overcomes these limitations. The *Just in*

Time (JIT) [3] and *Just Enough Time* (JET) [4] signaling protocols enable statistical resource sharing of a light-path among multiple traffic flows. Switching in OBS networks is performed optically, allowing optical/electrical/optical conversions to be eliminated in the data plane. This elimination enables high capacity switching with simpler switching architectures and lower power consumption [5].

In contrast to electrical switches, optical switches do not possess first-in-first-out (FIFO) buffers to queue data because optical signals may not be stored. In OBS networks, data packets are aggregated into bursts. Bursts contending for the same output link require novel contention resolution schemes. Wavelength conversion and fiber delay line schemes require deployment of special hardware modules while deflection routing requires only software modifications in the routers [6].

Various deflection routing architectures have been proposed in literature. Slotted and unslotted deflection schemes were compared [7], [8]. Performance of a simple random deflection algorithm and loss rates of deflected data were analyzed [9]–[12]. Various integrations of deflection routing with wavelength conversion and fiber delay lines were proposed [13], [14]. When an output link is busy, the deflection algorithm selects an alternate output link to deflect the burst. Deflection protocols were recently enhanced by enabling neighboring nodes to exchange traffic information. Hence, each node generates its deflection decisions based on a better understanding of its surrounding [15]–[17]. Furthermore, heuristic approaches may be used to process the information gathered from the neighboring nodes.

Reinforcement learning techniques have been recently employed to generate deflection decisions [18], [19]. Re-

inforcement learning is a trial-and-error based learning approach. An agent is an autonomous entity that learns either through direct interactions with the environment or through analysis of collected data. The agent makes a decision, receives reinforcement signals from the environment, and then processes them to improve its future decisions. Q-learning [20] is a reinforcement learning algorithm that may be employed by a learning agent that interacts with a dynamic environment whose statistical behavior is unknown and may change over time. A Q-learning agent learns an action-value function that evaluates the quality (Q) of taking an action. A node that needs to generate deflection decisions may employ such an agent.

In this paper, we propose a novel Node Degree Dependent (NDD) signaling algorithm. We have decoupled the design of the signaling infrastructure from the underlying learning algorithm and, therefore, the NDD signaling algorithm may employ a variety of reinforcement learning algorithms. We incorporate the NDD signaling algorithm and Q-learning into a deflection routing protocol named Q-NDD that is composed of two main modules: a signaling module that employs the NDD signaling algorithm and a learning module that employs the Q-learning algorithm. The complexity of the proposed signaling algorithm depends on the node degree rather than the network size. Therefore, it scales better compared to the existing deflection routing protocols that are based on reinforcement learning. The NDD signaling algorithm is designed to have a minimal number of parameters and, hence, does not require special routing configurations. We implement the proposed Q-NDD protocol in the ns-3 network simulator [21] and compare its performance with the reinforcement learning deflection routing scheme (RLDRS) [19]. Simulation results show that despite its lower complexity, performance of Q-NDD is comparable to RLDRS.

This paper is organized as follows. Reinforcement learning, its applications to deflection routing, introduction to Q-learning, and the scalability of the existing deflection routing protocols based on reinforcement learning are presented in Section 2. We describe the proposed Q-NDD deflection routing protocol in Section 3. The performance of the Q-NDD is evaluated in Section 4. We conclude with Section 5.

2. Reinforcement Learning Agents for Deflection Routing

An agent that learns how to interact with a dynamic environment through trial-and-error may use reinforcement learning techniques for decision-making [22]. Reinforcement learning consists of three abstract phases irrespective of the learning algorithm:

- An agent observes the state of the environment and selects an appropriate action.
- The environment generates a reinforcement signal and transmits it to the agent.
- The agent employs the reinforcement signal to improve its subsequent decisions.

Therefore, a reinforcement learning agent requires information about the state of the environment, reinforcement signals from the environment, and a learning algorithm. Enhancing a node in an OBS network with a reinforcement learning agent that generates deflection decisions requires three components:

- function that maps a collection of the environment variables to an integer (state)
- decision-making algorithm that selects an action based on the state
- signaling mechanism for sending, receiving, and interpreting the feedback signals.

Q-learning [20] is a simple reinforcement learning algorithm that has been employed for path selection in deflection routing. The algorithm maintains a Q-value $Q(s, a)$ in a Q-table for every state-action pair. Let s_t and a_t denote the encountered state and the action executed by an agent at a time instant t . Furthermore, let r_{t+1} denote the reinforcement signal that the environment has generated for performing action a_t in state s_t . When the agent receives the reward r_{t+1} , it updates the Q-value that corresponds to the state s_t and action a_t as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right], \quad (1)$$

where $0 < \alpha \leq 1$ is the learning rate and $0 \leq \gamma < 1$ is the discount factor.

The Q-learning path selection algorithm [18] calculates a priori a set of candidate paths $P = \{p_1, \dots, p_m\}$ for tuples (s_i, s_j) , where $s_i, s_j \in S$ and $S = \{s_1, \dots, s_n\}$ denotes the set of all edge nodes in the network. The Q-table stored in the i^{th} edge node maintains a Q-value for every tuple (s_j, p_k) , where $s_j \in S \setminus \{s_i\}$ and $p_k \in P$. S and P are the sets of states and actions, respectively. The Q-value is updated after each decision is made and the score of the path is reduced or increased depending on the received rewards. The algorithm does not specify signaling method or procedure for handling the feedback signals. RLDRS [19] also employs the Q-learning algorithm for deflection routing. The advantage of RLDRS is its precise signaling and rewarding procedures. The Q-learning path

selection algorithm and RLDRS are not scalable because their complexity depends on the size of the network. For example, in case of the Q-learning path selection algorithm [18], the size of the Q-table depends on the size of the network and the set of candidate paths. Therefore, it may be infeasible to store a large Q-table for larger networks. The proposed NDD signaling algorithm reduces the number of required feedback signals and its complexity depends only on a node degree.

3. The Node Degree Dependent Deflection Routing Algorithm

We describe here the proposed NDD signaling algorithm and the messages that need to be sent across the network in order to enhance an OBS node with decision-making ability. The NDD algorithm provides a signaling infrastructure that an OBS node may require in order to learn and optimally deflect the bursts in an OBS network.

The flowchart of the signaling algorithm is shown in Fig. 1. We consider an OBS network with N nodes. Each network node maintains a Q-table and all nodes are NDD compatible. A burst header that contains the control information associated with a burst is transmitted ahead of the burst. The burst header messages received by a node are passed to the NDD module. The module inspects the routing table for the next hop and then checks the status of the optical interfaces. If the desired optical interface is available, the optical cross-connects are configured according to the path defined by the routing table. If the interface is busy and the burst has not been deflected earlier by any other node, the current states of the optical interfaces and the output port defined by the routing table are passed to the Q-learning module. The states of the optical interfaces are mapped to an ordered string of 0s and 1s, where idle and busy interfaces are denoted by 0 and 1, respectively. We refer to the information passed to the Q-learning module as a *state*. The Q-learning module inspects the Q-table entry for the current *state*. If there is an entry, the learning module selects for deflection the output port that is associated with the maximum Q-value. However, if the learning module is unable to find a Q-table entry for the encountered state, it first initializes an entry for that *state* by assigning uniformly drawn random Q-values to all possible actions and then selects the action with the maximum Q-value. The Q-learning module returns to the NDD module the best selected output port for burst deflection. The following information is then added to the burst header:

- a unique *ID* number used to identify the feedback message that pertains to a deflection
- the address of the node that initiated the deflection,

to be used by other nodes as the destination for the feedback messages

- a deflection hop counter *DHC*, which is incremented each time other nodes deflect the burst.

When a burst is to be deflected at a node for the first time, the node records the current time as the deflection time *DfT* along with the *ID* assigned to the burst. The Drop Notification (DN) timer is initiated and the burst is deflected to the port that is selected by the Q-learning module. A maximum value for the DN timer is set to DN_{max} , which indicates expiration of the timer. The purpose of this timer is to reduce the number of feedback signals.

After a decision is made to perform a deflection, the Q-learning module waits for the feedback. The Q-learning module makes no new decisions during the idle interval. The deflected burst is discarded when either:

- its *DHC* reaches the maximum permissible number of deflections DHC_{max}
- it reaches a fully congested node.

The node that discards the deflected burst assembles a feedback message composed of the burst *ID*, *DHC*, and the time instant when the burst was discarded (drop time *DrT*). The feedback message is then sent to the node that initiated the deflection.

When the node that initiated the deflection receives the feedback message, it calculates the total travel time *TTT* that the burst has spent in the network after the first deflection:

$$TTT = DrT - DfT. \quad (2)$$

The *TTT* and *DHC* values are then used by the Q-learning module to update its statistics. If no feedback message is received until the *DN* timer expires, the node assumes that the burst has arrived successfully to its destination. The node may then update its learning module with the reinforcement signal that contains $TTT = 0$ and $DHC = 0$. A decreasing function with the global maximum at $(0, 0)$ may be used as a reward function to map *TTT* and *DHC* to a real value r . The Q-learning module updates the Q-value of the current state and the selected action as:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r - Q(s, a)). \quad (3)$$

An OBS node records the best action selected by the Q-learning module. These records are used if a node needs to deflect a burst:

- that has been deflected earlier

or

- during an idle interval.

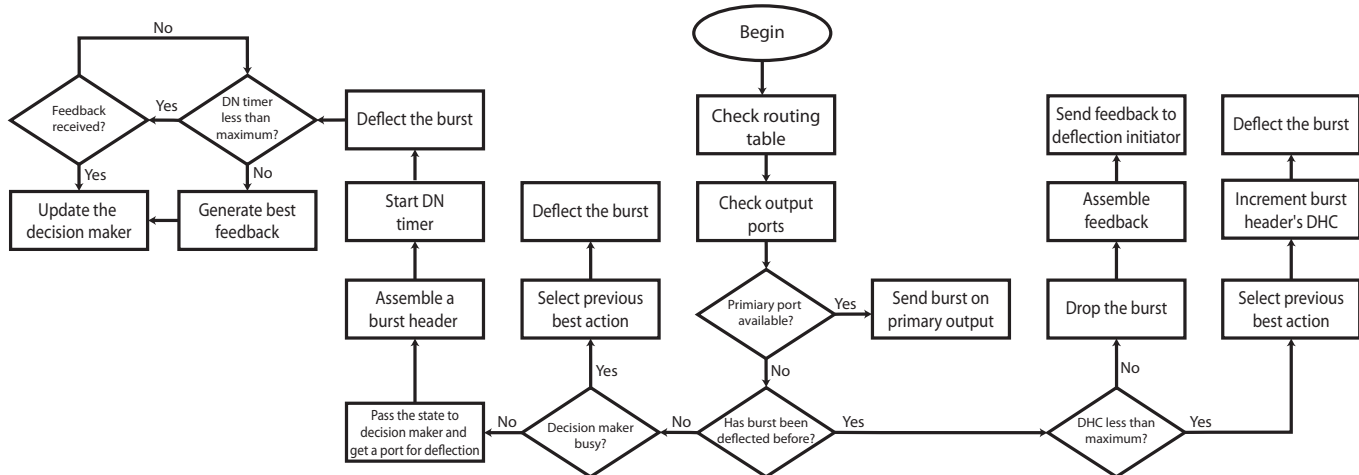


Figure 1. The flowchart of the proposed signaling algorithm. The DN timer denotes the drop notification timer. Nodes wait for feedback signals until this timer reaches DHC_{max} .

In order to reduce the excess traffic generated by the number of feedback messages, a node receives feedback messages only when it deflects bursts that have not been deflected earlier. Hence, deflecting a burst that has been deflected earlier does not enhance the node's decision-making ability.

4. Performance Evaluation

We evaluate performance of the proposed Q-NDD deflection routing protocol. We have also implemented RLDRS [19] in order to compare these algorithms based on burst loss probability, average end-to-end delay, and number of deflections. For simulation scenarios, we use the National Science Foundation (NSF) network topology shown in Fig. 2. We also compare these algorithms in terms of memory and Central Processing Unit (CPU) time by simulating randomly generated graphs of 100, 500, 1,000, and 2,000 nodes. In all simulation scenarios, we allow up to two deflections per burst ($DHC_{max} = 2$). The burst header processing time is set to 0.1 ms.

4.1. Simulation of the NSF Network

The topology of the NSF network shown in Fig. 2 was generated by extracting the geodetic coordinates [23] of the NSF network nodes from the Google Earth [24]. These coordinates are then transformed to Cartesian coordinates. The nodes are connected using bidirectional 1 Gbps fiber links with 8 or 64 wavelengths. The learning rate is set to $\alpha = 0.1$ and the maximum drop notification timer to $DN_{max} = 50$ ms.

Multiple Poisson traffic flows with a data rate of 0.5 Gbps are transmitted randomly across the network. Each

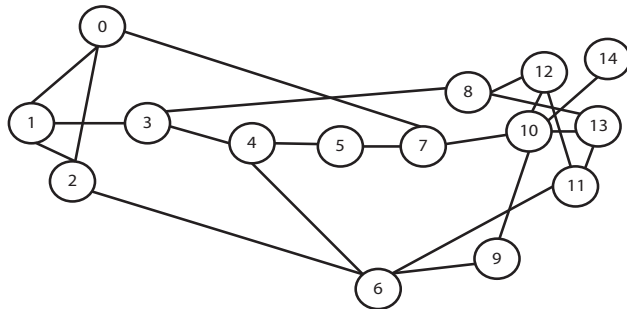
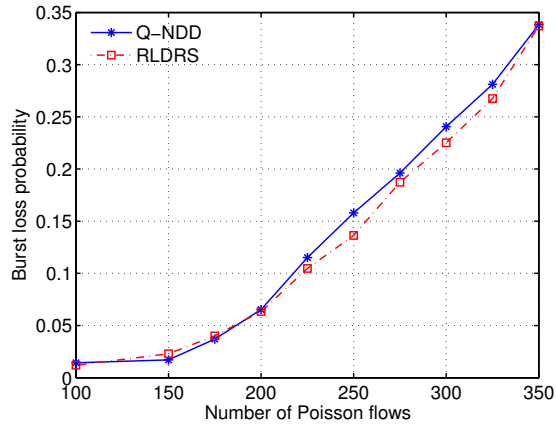


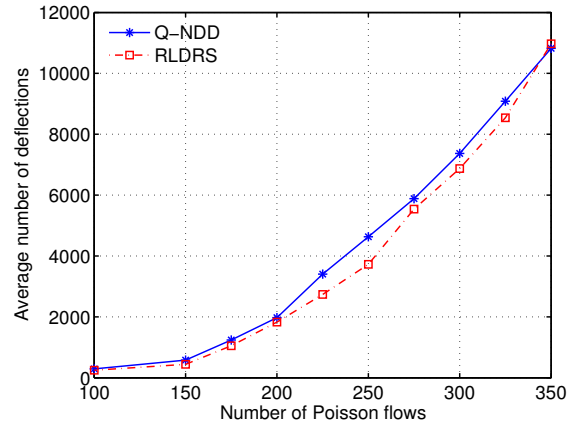
Figure 2. Topology of the NSF network after the 1989 transition. Nodes 9 and 14 were added in 1990.

Poisson flow is 50 bursts long with each burst containing 12.5 kB of payload. The burst arrival process depends on the aggregation algorithm [25] that is deployed in a node. However, the Poisson process has been widely used for performance analysis of OBS networks because it is mathematically tractable [26], [27]. We use the heavy-tailed Poisson-Pareto burst process (PPBP) with Hurst parameter $H = 0.7$ [28] to compare the Q-NDD and RLDRS algorithms. The simulation scenarios are repeated five times with various random assignments of nodes as sources and destinations. The simulation results are averaged over five simulation runs. The burst loss probability and the average number of deflections as functions of the number of Poisson flows for 8 and 64 wavelengths scenarios are shown in Fig. 3.

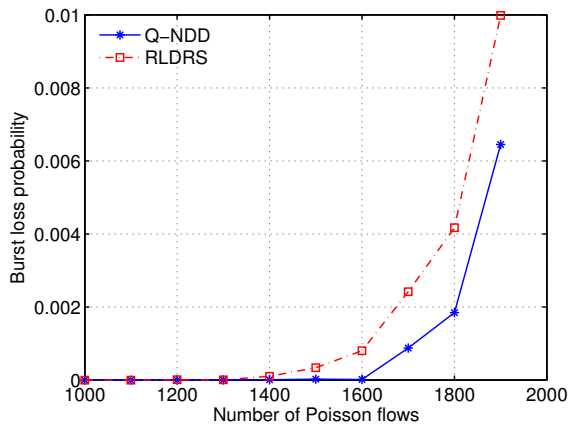
We also simulate the NSF network with 400 simultaneous PPBP traffic flows and 8 wavelengths. Performance of the Q-NDD protocol and RLDRS in terms of burst



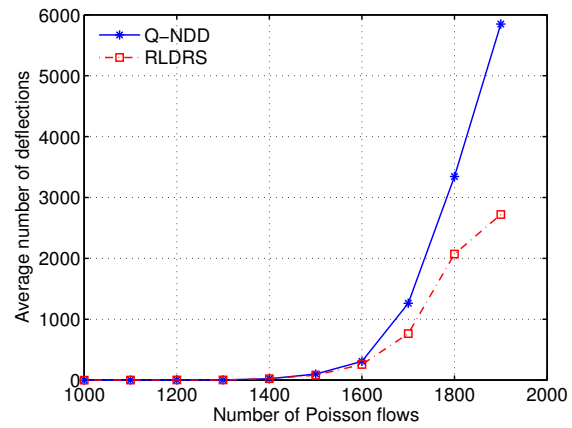
(a) 8 wavelengths



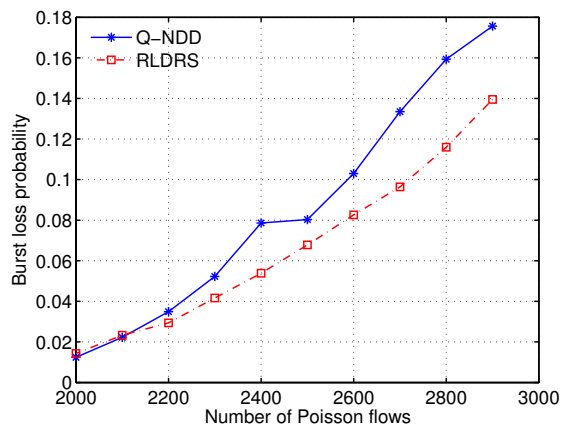
(b) 8 wavelengths



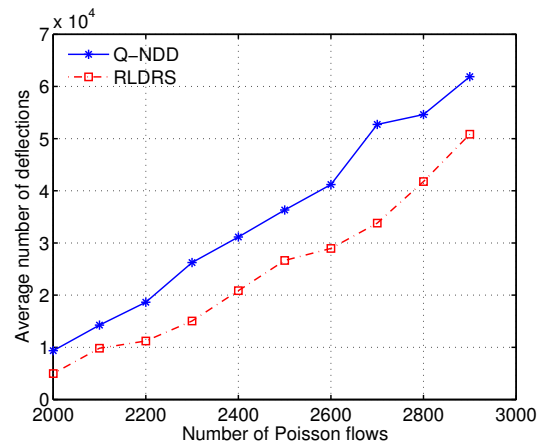
(c) 64 wavelengths: low to moderate load



(d) 64 wavelengths: low to moderate load



(e) 64 wavelengths: moderate to high load



(f) 64 wavelengths: moderate to high load

Figure 3. Burst loss probability (left column) and average number of deflections (right column) as a function of the number of Poisson flows in the NSF network simulation scenario with 8 wavelengths and 64 wavelengths. For readability, two cases were plotted for each 64 wavelength graph: 1,000 to 2,000 Poisson flows and 2,000 to 3,000 Poisson flows.

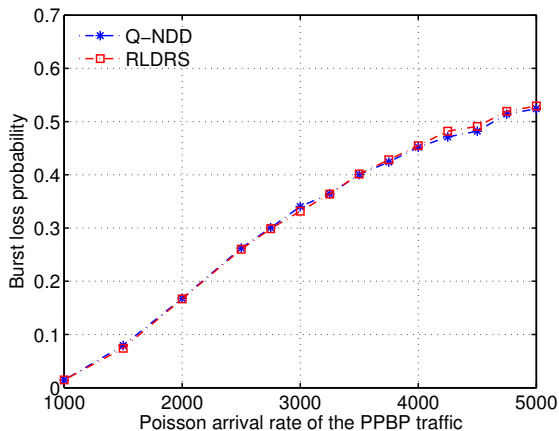


Figure 4. Burst loss probability as a function of the arrival rate of the Poisson component of the PPBP traffic with 8 wavelengths and 400 simultaneous traffic flows.

loss probability as a function of arrival rate of the Poisson component of the PPBP traffic is shown in Fig. 4. Even though the complexity of the Q-NDD protocol is reduced to the degree of a node, simulation results show that Q-NDD performs better than RLDRS in case of low to moderate traffic loads. However, Q-NDD initiates larger number of deflections compared to RLDRS.

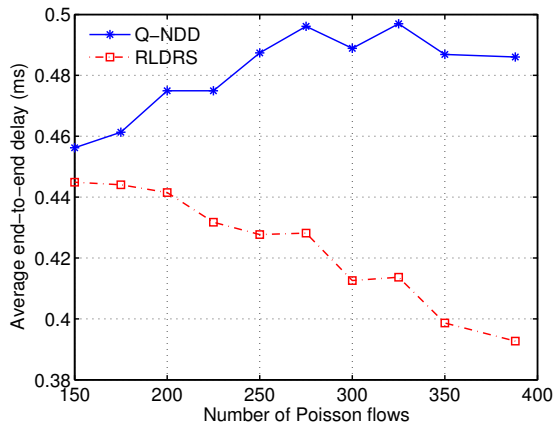
The RLDRS-based signaling algorithm takes into account the number of hops to the destination when generating the feedback signals. Therefore, RLDRS performs better in terms of average end-to-end delay and average number of hops traveled by bursts. Performance of Q-NDD and RLDRS in terms of average end-to-end delay and average number of hops is shown in Fig. 5 and Fig. 6, respectively.

4.2. Comparison of Memory Requirements and CPU Time

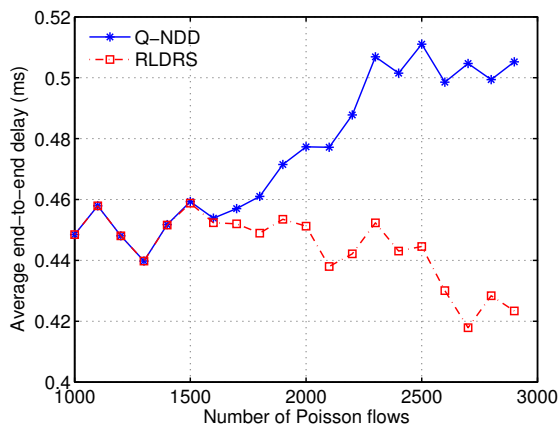
We use the Boston University Representative Internet Topology Generator (BRITE) [29] to generate random Waxman graphs [30] with 100, 500, 1,000, and 2,000 nodes. An edge that connects nodes u and v exists in a Waxman graph with a probability

$$\Pr(\{u, v\}) = \beta \exp\left(\frac{-d(u, v)}{L\delta}\right), \quad (4)$$

where $d(u, v)$ is the distance between nodes u and v , L is the maximum distance between any two nodes, and β and δ are parameters in the range $(0, 1]$. We set $\beta = 0.2$ and $\delta = 0.15$ in simulation scenarios. The nodes are randomly placed and each node is connected to three other nodes using bidirectional single wavelength fiber links. Sources



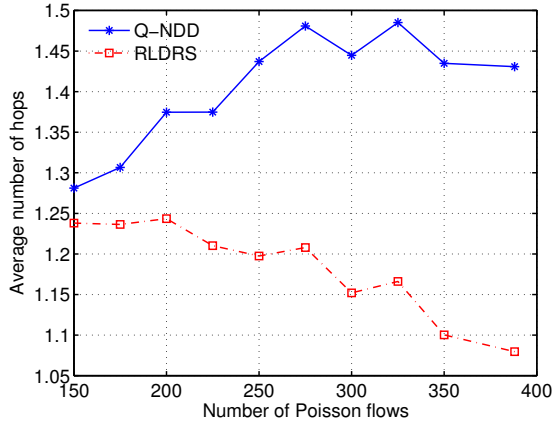
(a) 8 wavelengths



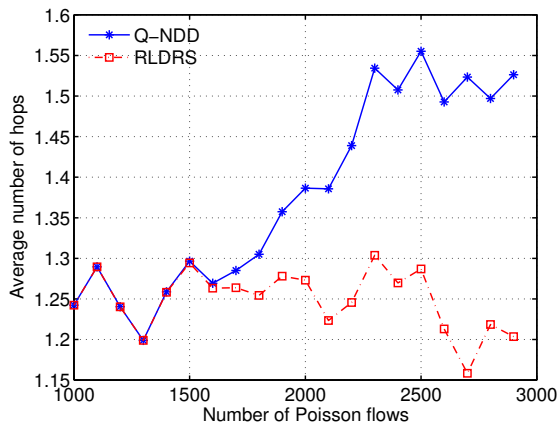
(b) 64 wavelengths

Figure 5. Average end-to-end delay as a function of number of flows in the NSF network scenario with 8 and 64 wavelengths.

and destinations of traffic flows are randomly selected. The number of the PPBP traffic flows for scenarios with 100, 500, 1,000, and 2,000 nodes are 600, 3,000, 6,000, and 12,000, respectively. Simulations are performed on the Dell Optiplex-790 with 16 GB memory and the Intel Core i7 2600 processor. The comparison of memory requirements and CPU time of Q-NDD and RLDRS is shown in Table 1. Both algorithms initially require the same memory. However, as the simulations proceed and new entries are populated into the Q-tables, the memory usage of RLDRS grows faster compared to Q-NDD. For example, in the 1,000 node case, both algorithms use 1,723 MB at the beginning of simulations. The maximum memory requirements for Q-NDD and RLDRS are 1,754 MB and 1,775 MB, respectively. The simulation results also show that Q-NDD requires less CPU time compared to RLDRS.



(a) 8 wavelengths



(b) 64 wavelengths

Figure 6. Average number of hops traveled by bursts as a function of number of flows in the NSF network scenario with 8 and 64 wavelengths.

The memory usage of Q-NDD and RLDRS is shown in Fig. 7.

5. Conclusion

In this paper, we introduced a low-complexity signaling algorithm that employs the Q-learning algorithm for deflection routing in OBS networks. The complexity of the proposed algorithm depends only on a node degree.

Simulation results indicate that the proposed NDD signaling algorithm requires less memory and CPU resources. These resource requirements become more significant as the size of the network grows. Furthermore, burst loss probability of the Q-NDD protocol is comparable to an existing reinforcement learning-based deflection routing algorithm.

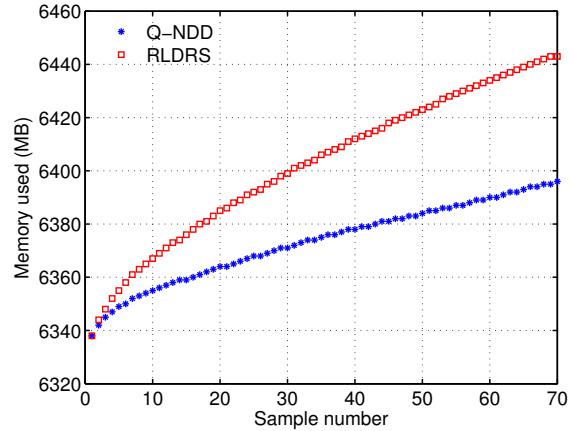


Figure 7. Memory used in the network with 2,000 nodes. The graphs were generated by using 70 equally spaced time instances over each simulation run.

References

- [1] C. Qiao and M. Yoo, "Optical burst switching (OBS) – a new paradigm for an optical Internet," *J. of High Speed Netw.*, vol. 8, no. 1, pp. 69–84, Mar. 1999.
- [2] H. G. Perros, *Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks*. Chichester, UK: John Wiley & Sons, 2005.
- [3] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Jumpstart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 82–89, Feb. 2002.
- [4] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," *IEEE Netw.*, vol. 18, no. 3, pp. 16–23, June 2004.
- [5] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1838–1851, Aug. 2000.
- [6] L. Xu, H. G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 136–142, Jan. 2001.
- [7] F. Borgonovo, L. Fratta, and J. Bannister, "Unslotted deflection routing in all-optical networks," in *Proc. IEEE GLOBECOM*, Houston, TX, Dec. 1993, pp. 119–125.
- [8] T. Chich, J. Cohen, and P. Fraigniaud, "Unslotted deflection routing: a practical and efficient protocol for multihop optical networks," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 47–59, Feb. 2001.
- [9] F. Borgonovo, "Deflection routing," in *Routing in Communications Networks*. New Jersey: Prentice-Hall, 1995, pp. 263–306.
- [10] A. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, no. 6, pp. 1070–1081, June 1992.

Table 1. Comparison of memory and CPU time of Q-NDD and RLDRS. Initially, the two algorithms use equal memory. The memory usage of RLDRS grows faster compared to Q-NDD.

| Algorithm | Number of nodes | Number of links | Number of flowss | Minimum memory usage (MB) | Maximum memory usage (MB) | Total CPU time (mm:ss) | Total simulation time (s) |
|-----------|-----------------|-----------------|------------------|---------------------------|---------------------------|------------------------|---------------------------|
| Q-NDD | 100 | 300 | 600 | 180 | 183 | 0:09.51 | 323.9 |
| | 500 | 1,500 | 3,000 | 561 | 578 | 1:25.61 | 1,832.8 |
| | 1,000 | 3,000 | 6,000 | 1,723 | 1,754 | 6:03.67 | 4,042.5 |
| | 2,000 | 6,000 | 12,000 | 6,338 | 6,397 | 32:58.17 | 9,661.5 |
| RLDRS | 100 | 300 | 600 | 180 | 185 | 0:14.26 | 470.2 |
| | 500 | 1,500 | 3,000 | 561 | 587 | 2:04.85 | 2,830.7 |
| | 1,000 | 3,000 | 6,000 | 1,723 | 1,775 | 8:02.16 | 6,026.8 |
| | 2,000 | 6,000 | 12,000 | 6,338 | 6,444 | 41:29.18 | 14,363.1 |

- [11] 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.
- [12] 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.
- [13] M. Yoo, C. Qiao, and S. Dixit, "Comparative study of contention resolution policies in optical burst-switched WDM networks," in *Proc. SPIE*, Boston, MA, Oct. 2000, vol. 4213, pp. 124–135.
- [14] A. Zalesky, H. Vu, Z. Rosberg, E. W. M. Wong, and M. Zukerman, "OBS contention resolution performance," *Performance Evaluation*, vol. 64, no. 4, pp. 357–373, May 2007.
- [15] X. Gao and M. Bassiouni, "Improving fairness with novel adaptive routing in optical burst-switched networks," *J. Lightw. Technol.*, vol. 27, no. 20, pp. 4480–4492, Oct. 2009.
- [16] J. Perelló, F. Agraz, S. Spadaro, J. Comellas, and G. Junyent, "Using updated neighbor state information for efficient contention avoidance in OBS networks," *Comput. Commun.*, vol. 33, no. 1, pp. 65–72, Jan. 2010.
- [17] W. W.-K. Thong, G. Chen, and Lj. Trajković, "RED-f routing protocol for complex networks," *Proc. IEEE Int. Symp. Circuits and Systems*, Seoul, Korea, May 2012, pp. 1644–1647.
- [18] Y. Kiran, T. Venkatesh, and C. Murthy, "A reinforcement learning framework for path selection and wavelength selection in optical burst switched networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 18–26, Dec. 2007.
- [19] 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.
- [20] C. J. C. H. Watkins and P. Dayan, "Technical note Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.
- [21] (2013, Jul.) The ns-3 network simulator. [Online]. Available: <http://www.nsnam.org/>.
- [22] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *J. of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.
- [23] E. J. Krakiwsky and D. E. Wells, *Coordinate Systems in Geodesy*. Fredericton, NB: Lecture Notes LN# 16, Department of Geodesy and Geomatics Engineering, Univ. of New Brunswick, 1971.
- [24] (2013, Jul.) The google earth. [Online]. Available: <http://www.google.com/earth/index.html/>.
- [25] X. Mountrouidou and H. Perros, "On the departure process of the burst aggregation algorithms in optical burst switching," *J. of Comput. Netw.*, vol. 53, no. 3, pp. 247–264, Feb. 2009.
- [26] A. Zalesky, H. Vu, Z. Rosberg, E. W. M. Wong, and M. Zukerman, "Modelling and performance evaluation of optical burst switched networks with deflection routing and wavelength reservation," in *Proc. INFOCOM*, Hong Kong SAR, China, Mar. 2004, vol. 3, pp. 1864–1871.
- [27] X. Yu, J. Li, X. Cao, Y. Chen, and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks," *J. Lightw. Technol.*, vol. 22, no. 12, pp. 2722–2738, Dec. 2004.
- [28] M. Zukerman, T. D. Neame, and R. G. Addie, "Internet traffic modeling and future technology implications," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 2003, vol. 1, pp. 587–596.
- [29] (2013, Jul.) BRITE. [Online]. Available: <http://www.cs.bu.edu/brite>.
- [30] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.