Hindawi
Journal of Computer Networks and Communications
Volume 2020, Article ID 2708357, 14 pages
https://doi.org/10.1155/2020/2708357

Review Article

A Node-Regulated Deflection Routing Framework for Contention Minimization

Bakhe Nleya[1] **and Andrew Mutsvangwa**[2]

[1]*Department of Electronic Engineering, DUT, Durban 4001, South Africa*
[2]*Faculty of Education, NWU, Potchefstroom, South Africa*

Correspondence should be addressed to Bakhe Nleya; bmnleya@gmail.com

Received 30 April 2019; Revised 9 December 2019; Accepted 28 December 2019; Published 8 June 2020

Guest Editor: En Wang

Optical Burst Switching (OBS) paradigm coupled with Dense Wavelength Division Multiplexing (DWDM) has become a practical candidate solution for the next-generation optical backbone networks. In its practical deployment only the edge nodes are provisioned with buffering capabilities, whereas all interior (core) nodes remain buffer-less. In that way the implementation becomes quite simple as well as cost effective as there will be no need for optical buffers in the interior. However, the buffer-less nature of the interior nodes makes such networks prone to data burst contention occurrences that lead to a degradation in overall network performance as a result of sporadic heavy burst losses. Such drawbacks can be partly countered by appropriately dimensioning available network resources and reactively by way of deflecting excess as well as contending data bursts to available least-cost alternate paths. However, the deflected data bursts (traffic) must not cause network performance degradations in the deflection routes. Because minimizing contention occurrences is key to provisioning a consistent Quality of Service (QoS), we therefore in this paper propose and analyze a framework (scheme) that seeks to intelligently deflect traffic in the core network such that QoS degradations caused by contention occurrences are minimized. This is by way of regulated deflection routing (rDr) in which neural network agents are utilized in reinforcing the deflection route choices at core nodes. The framework primarily relies on both reactive and proactive regulated deflection routing approaches in order to prevent or resolve data burst contentions. Simulation results show that the scheme does effectively improve overall network performance when compared with existing contention resolution approaches. Notably, the scheme minimizes burst losses, end-to-end delays, frequency of contention occurrences, and burst deflections.

1. Introduction

Dense Wavelength Division Multiplexing (DWDM) does support speeds in the terabit ranges in a single fiber; hence, it can adequately handle the massive amounts of heterogeneous data in present and future service networks. The terabit range speeds are primarily achieved by way of individually modulating several wavelengths before multiplexing them into a single fiber. In order to match the ultrahigh transmission speeds, the OBS paradigm was proposed as a candidate solution. OBS generally combines as well as represents a trade-off between other rival switching paradigms such as Optical Circuit Switching (OCS) and Optical Packet Switching (OPS). Whereas OCS may be relatively easy to implement, it however has a disadvantage of low network resources utilization and coarse granularity. On the other hand, OPS generally features excellent resources utilization as well as fine granularity even though it would be too costly to implement [1].

The edge (ingress and egress) nodes interface directly with user networks such as subscriber access, individual home metropolitan access rings, cloud centers, and smart grids. Core and edge nodes are mesh interconnected via DWDM links as illustrated in Figure 1. The primary functions of these two types of nodes are summarized in Figure 2. In practical operation, an ingress (edge) node assembles data packets destined for a common egress (edge) node into data bursts. Upon completion of a data burst

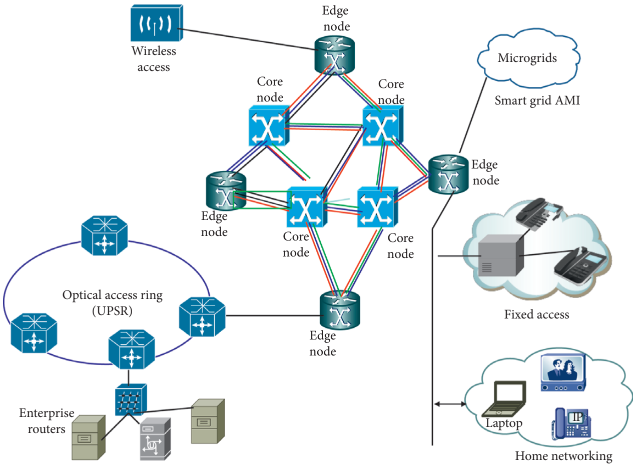


Figure 1: OBS backbone network [1].

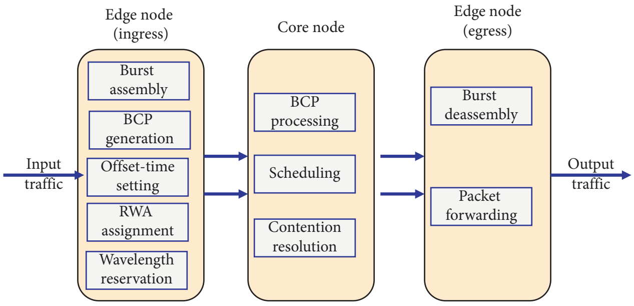


Figure 2: OBS network functional block diagram.

assembling, a signalling burst control packet (BCP) is generated and dispatched with an offset time ahead of the data burst. This will be delivered to the next (intermediate) node. The set offset time is critical in ensuring that required resources are configured at the next node prior to the data burst's arrival. Particularly, the offset timing is a time allowance for *processing* of the BCP at the next and subsequent nodes, reservation of a wavelength at the desired output port link, and preconfiguring the switching network with respect to the incoming and outgoing ports to be used [1]. The data burst later cuts through the node and shortly afterwards resources previously reserved for its switching are freed so that they can be utilized by other lightpath connection demands. The cut-through approach in handling the burst alleviates the necessity for any node buffering as this would otherwise escalate CAPEX/OPEX, as well as overall end-to-end delays for the data bursts. The momentary usage of resources enhances utilization as well as improves adaptability to highly heterogeneous traffic. When establishing an end-to-end lightpath connection, it is important to carefully select the set of candidate routing links as well as wavelengths at each ingress node so as to reduce the possibilities of data burst contentions occurring at subsequent nodes as a result of improper routing and wavelength assigning (RWA). In order to resolve any possible contention occurrences, limited extra resources in the form of wavelength converters (WCs) and optical (fiber) delay lines (FDLs) are provided. Once a data burst has successfully traversed the network, it will eventually be disassembled into individual packets which are then routed to their respective final destinations [1]. A key step in the design and deployment of OBS backbone networks is to optimize the limited available links as well as wavelengths such that a maximum possible number of lightpath connections can be established simultaneously. In essence, the RWA problem centers on establishing an end-to-end lightpath using a single wavelength. In so doing, this wavelength continuity constraint forbids the establishment of more than one lightpath using a particular wavelength concurrently on the same fiber link.

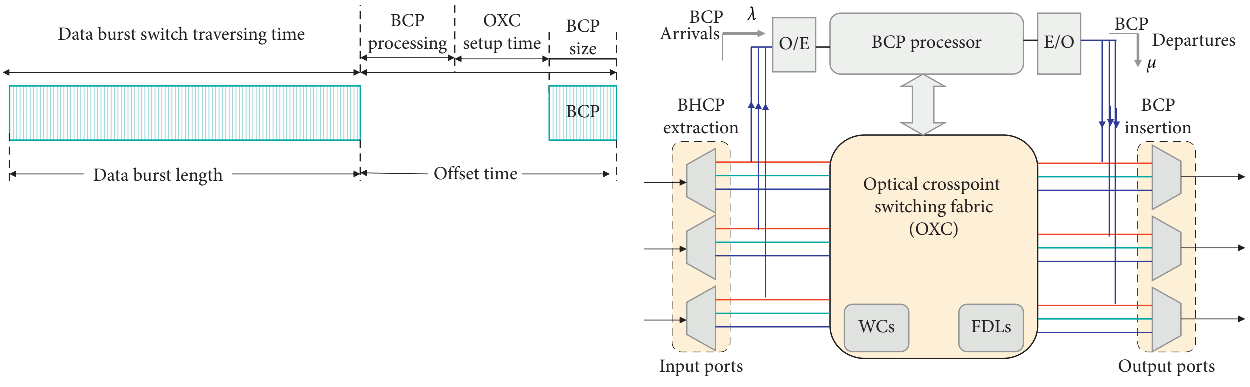
As core nodes do not directly interface with access networks, their primary functions are therefore restricted to BCP processing, scheduling, and contention resolution. Its functional components include input ports terminating each incoming fiber, output ports feeding to outgoing fiber links, a BCP processor, an optical cross-connect (OXC), wavelength converters (WCs), and FDLs. BCPs from ingress nodes are extracted at the input ports of intermediate nodes along the routing path then processed electronically by BCP processors. The offset timing relationship between a data burst and a BCP is illustrated as shown in Figure 3(a).

As was mentioned earlier, core nodes (Figure 3(b)) do not offer any buffering capabilities; thus, the data bursts cut through such nodes wholly in the optical domain. A performance drawback with regard to this being that some data bursts may be lost whenever contention occurs. The latter will always occur whenever more than one data burst destined for the same output port overlap in time and frequency.

Contention is regarded as a significant hindrance to the smooth operation of an OBS network as it often leads to significant burst losses and consequently QoS degradation. It arises when two or more data bursts utilizing an identical wavelength and destined for the same output port partially or wholly overlap in time. Both *reactive* and *proactive* contention resolution approaches are being implemented. Reactive contention resolution approaches attempt to resolve the contentions only after occurrence mainly in the core network, whereas proactive schemes are designed to avoid any contentions taking place in the network and effectively preventing any data burst losses.

Appropriate dimensioning of available network resources is regarded as a proactive contention prevention measure. Generally, it is relatively easier to implement proactive approaches at the ingress nodes routing the data on a carefully chosen optimal path to an intended destination coupled with optimal wavelength assigning to it. Network level performance metrics such as burst loss probability, end-to-end delays, and frequency of deflections serve as measures of optimality in the path and wavelength assigning.

Key reactive approaches include the use of limited available buffering in the form of FDLs, converting one of the contending data burst's wavelengths to any other available one using a WC, burst segmentation in which the overlapping sections of contending data bursts are discarded, *or* where one of the contending data bursts is deflected to any other available output port.



(a)

(b)

Figure 3: OBS core node.

2. Related Work on Contention Resolution

Accurately dimensioning network resources in relation to actual traffic volumes can be essential towards minimizing contention. However, not much work has been done towards accurately spelling out actual traffic flows throughout a given network. The authors in [2, 3] propose and evaluate an algorithm for obtaining a detailed end-to-end traffic matrix of a given network. Their proposed algorithm is based on fractal and cubic spline interpolation reconstruction approaches. The proposed algorithms will enable the attainment of detailed information about traffic flows on all active links of the network. However, both algorithms may still generate errors and hence this leads to inaccuracies in determining actual traffic flows. In a bid to further improve the reconstruction accuracy, the authors go on to propose the weighted geometric method whose simulation results [2] show relatively better accuracy in determining the network traffic flows. End-to-end traffic flow estimation based on measured source-destination flows is also explored in [3]. The proposed method herein, however, is deemed generally costly in implementation even though its measurement associated overheads are relatively low. In all cases, the various proposed methods attempt to acquire a better global view of the network's traffic flows. The traffic flow measurements obtained will enable network designers to accurately dimension all available network resources. In that way they alleviate possible congestion and contention occurrences since more effective RWA problem can be better resolved.

The RWA problem constitutes simultaneously setting up end-to-end lightpaths across the optical backbone transport network as well as routing and assigning a unique wavelength to each lightpath connection setup. In so doing, the wavelength continuity constraint must be maintained and at the same time designers are thriving to maximize the number of simultaneous connections with minimal network resources possible [4]. Once the network is operational, contentions will always occur in the intermediate nodes primarily because of their buffer-less nature. Extensive research work is focusing on minimizing the frequency of contention occurrences. The authors in [5] proposed and evaluated an algorithm that utilizes voids so as to minimize contentions as well as burst losses in interior nodes. The algorithm initially identifies all possible candidate void channels on which a data burst can be scheduled from before finally selecting one that maximizes the void utilization factor. Similarly, the authors in [6] propose a modified OBS paradigm that adapts assembled data burst sizes as a function of network traffic load. In this case when network loads are high, longer data bursts are assembled by the ingress nodes. Triangular estimator-based burst scheduling algorithms are proposed in [7]. All sections of the network that are currently prone to contention occurrences are identified as well as avoided when scheduling bursts. The authors in [8] studied adverse effects of deflection routing load balancing on general TCP performance. In their work, they suggest source ordering as a means of improving TCP throughput performance. Based on earlier findings, the authors in [9] extended the work by proposing a modified Horizon scheduling algorithm with minimum reordering effects (MHS-MOE). Artificial intelligence techniques are utilized to enhance the network's routing decisions by the authors in [10] who propose and analyze a Reinforcement Learning-Based Deflection Routing Algorithm (RLDRA). Their aim was to reduce data loss probabilities when the rate of contention occurrences at intermediate nodes surges significantly as a result of deflection traffic. Their scheme tries to control the count of authorized deflections for each burst to reduce the extra traffic generated by deflection routing. It also does not generate significant amounts of signalling and other computational overheads.

A multiclass preemptive scheduling-based scheme on deflection paths (routes) is proposed in [11] in which an attempt is made to improve general QoS of existing and future connections by implementing preemption policies on the onset of contention in the network. The proposed scheme's complexity is in the involvement of multitudes of parameters for determining and defining preemption probabilities and policies. Deflection routing in an anycast-based OBS grid is proposed by the authors in [12]. However, the accompanying proposed enhanced deflection routing algorithm does not appear to address or alleviate the contention problem satisfactorily. Fairness and data burst loss owing to cascading constraint when bursts have longer hop count value in OBS networks is explored in [13]. The authors herein propose a preemptive scheduling technique for next-generation OBS networks in which newly arriving bursts with higher priority may preempt already scheduled ones when contention occurs.

It is on the strength of the earlier cited weakness that in this paper we propose a controllable deflection routing scheme which couples with a simple wavelength and routing assignment (WRA) algorithm to enhance overall network performance, by way of minimizing contention, wavelength congestion, and consequently bursts blocking probabilities. The scheme attempts as much as possible to deflect either of the contending bursts to paths that have been chosen based on the minimization of performance measures such as delay and blocking. It aims at controlling deflection traffic by way of selective path routing upon wavelength congestion onset. It is backed by a very simplified distributed RWA approach that ensures minimal contention in the primary (original) chosen route(s).

Summarily, the paper's contributions are as follows:

We propose a three-step approach to reducing contention occurrences in the core OBS network: (1) offset time regulation at burstification, (2) regulated deflection routing, and (3) neural network-based robust network state updating. Therefore, in this regard, the following steps are carried out:

(1) We describe a novel adjustable offset time coupled with a segmented burst assembly algorithm that regulates the offset timing such that extra delays are not incurred by traffic in the network. The algorithm suits both delay and non-delay-sensitive applications data.

(2) We develop a regulated deflection routing (rDr) scheme that is primarily concerned with the selection of the best-choice deflection routes at (from) each node when contentions have occurred, and deflection is necessitated. It is a proactive multipath neural network reinforced routing approach that seeks to avoid contentions occurring, as well as maintain load balancing in the overall network.

(3) For more accurate routing decision making by the node, we propose that neural network (NN)-based agents be incorporated to each routing node. The agents will form a distributed signalling network that periodically updates the entire network resources to routing tables. In that way, nodes can make better routing decisions. Each agent has a simple architecture; namely, it utilizes a single hidden as well as an associative learning algorithm to periodically update others regarding the state of network resources within the neighbourhood.

The rest of the paper is organized as follows:

In Section 2, we provide a brief description of deflection routing-based contention resolution in the core network. We introduce an adjustable offset timing burst assembly algorithm in Section 3. Neural network-based reinforcement learning is reviewed in Section 4, and in Section 5, we introduce a distributed regulated deflection routing framework. The same section discusses the NN-S algorithm. Section 6 is dedicated to evaluation, discussions, and conclusions.

3. Adjustable Offset Timing

Various traffic classes will always require some degree of QoS guarantees. During data burst assembling, QoS parameters such as jitter, blocking probabilities, and end-to-end delays can be minimized by reducing burstification delays as well as adaptively regulating the offset time [13]. Generally, burst assembly algorithms strive to strike balances between the rate at which BCPs are generated versus individual data burst sizes. The two are complimentary for moderate-to-high traffic loads in that maintaining huge data burst sizes during the assembly process would result in lesser numbers of BCPs generated per unit time. However, maintaining huge burst sizes will also lead to increased burstification delays when network traffic is low. Thus, the magnitude of offset timing chosen will directly influence end-to-end delays. Fixed offset timing gives extra time allowances for short data bursts but will be problematic for huge-sized bursts. In this paper, we adopt an adjustable offset timing algorithm during burst assembly whereby the actual offset time assigned is set as a function of an estimate of the current data burst size during its assembly [13, 14].

The adjustable offset time algorithm assumes the segmented burst assembly approach [15]. For a given source (s) to destination (d) node pair, the total end-to-end delay ($t_{d,s}$) is expressed as

$$t_{d,s} = t_{\text{ba}} + t_{\text{offset}} + h \times \left(t_{\text{proc}} + t_{\text{prop}}\right), \tag{1}$$

where t_{ba} is the burst assembly time at the ingress node, h is the hop count between s and d, t_{proc} is the processing and switching time at each node, and t_{prop} is the propagation time per hop distance.

Upon arrival of packets at an ingress node, the packets are assembled into class segments [15]; the maximum time delay tolerance of the packet class ($t_{\max\_\text{class}}$) is set according to its QoS requirements.

$$t_{\max\_\text{class}} \geq t_{d,s}. \tag{2}$$

If the $t_{\max(\text{app})}$ is less than the end-to-end delay time, $t_{d,s}$, then the offset time is calculated as

$$t_{\text{offset}} = \left(h \times t_{\text{prop}} + t_{\text{proc}}\right) - t_{\text{odv}}. \tag{3}$$

From (3), t_{odv} is an offset-time deducted value which is typically 90 nanoseconds [13].

The next burst assembly time (t_{out}) is computed from

$$t_{\text{out}} = \Delta t, \tag{4}$$

where

$$\Delta t = t_{\max\_\text{class}} - t_{d,s}. \tag{5}$$

4. ANN Agents and Reinforcement Learning

Artificial Neural Networks (ANNs) are becoming a very popular tool for solving issues at operational level in networking.

The primary challenge in the use of ANN agents in this regard is for them to learn situation-to-action mappings in a particular network scenario and once the learning is successfully accomplished, the agent autonomously carries out the task. By availing appropriate data, an agent will quickly learn how to control a specific task in a networked environment. By leveraging complex statistical as well as mathematical tools, a given ANN set is capable of independently performing complex tasks that traditionally could only be solved by humans [16].

The adoption of ANN techniques in network routing was triggered by the unprecedented surge in network complexity. The large number of interdependent adjustable network system parameters such as modulation formats, coding schemes, routing configurations, and symbol rates contributes to overall network complexity. Figure 4 illustrates the principles of applying artificial intelligence in an OBS network. The agent routinely monitors the network (environment) for any state change(s) and actions to each of them accordingly. For every agent's action, the environment reciprocates with a reward signal. The agent in turn uses the reward signal for self-performance as well as on improving its subsequent decisions. For this reason, the agent requires a learning algorithm to enhance its interactions with the environment. It is expected to demonstrate intelligence capabilities by way of incorporating planning, reasoning, and knowledge. Because events in an OBS network are often nondeterministic, an agent must operate robustly under uncertainty. It should also make use of key decision-making algorithms so that they can maximize the expected utility. Practical OBS network environments mostly deal with uncertainties and hence the precision of an agent will depend on a multitude of interactions with the environment rather than a single isolated one. Learning is also a key issue that enables an agent to improve its accuracy on future decisions, based on previous experience. Learning will also help it to be adaptable to a changing environment. The learning phase is commonly referred to as machine learning (ML). The various learning algorithm categories include supervised learning, unsupervised learning, hybrid learning, and reinforced learning. In this section, we will elaborate more on reinforcement learning. In this case, the agent requires inputs such as jitter, tolerable network delays, and loss probabilities of various output ports (links) from the OBS network environment before responding accordingly with the most optimal link (path) choice towards a particular destination. The network eventually sends a positive or negative evaluative reinforcement feedback signal to the agent. The latter signal will be used by the agent to adjust its parameters accordingly so that it could improve on its future decisions regarding the same network aspect [17]. The reinforcement learning system also requires a strict policy so that the agent's behaviour can be regulated. It will also utilize reward function maps on each state–action pair of the environment to a numerical value which indicates the desirability of that state or the desirability of an action at that state [17, 18].

4.1. Extreme Learning Algorithm.

4.1. Extreme Learning Algorithm. In this subsection, we summarily describe an example ANN set learning algorithm called the extreme learning algorithm [10]. It facilitates the following key steps in order to arrive at a decision.

(i) Each ANN set accepts an input signal from its environment (the OBS network).

(ii) The input data set now traverses the entire set (via the hidden layer) and is eventually mapped to the output layer as an action.

(iii) The output of the ANN set is fed to the environment where the latter will evaluate it before subsequently responding to the NN set with a reinforcement signal r.

(iv) The NN set uses r to further improve on future decisions. This is achieved by modifying its weights accordingly.

We hereby take an example of estimating the blocking probability in an OBS network [19].

Th training set can be represented by a vector:

$$D = \left\{ (x_k, p_k) : x_k \in \mathfrak{R}^d, p_k \in \mathfrak{R}, k = 1, \ldots, N \right\}, \tag{6}$$

where x_k is the ANN input vector and y_k is a targeted output vector of k^{th} sample.

By assuming the ANN set comprises a single hidden layer, then

$$f_n(x) = \sum_{i=1}^{n} \beta_i g_i(x), \tag{7}$$

where g_i denotes the i^{th} hidden node's output, whereas β_i is the weight between the i^{th} hidden node and the output node.

The output function $g_i(x)$ of the i^{th} hidden node can thus be defined as

$$g_i(x) = \frac{1}{1 + e^{-\left(a_i^T x + b_i\right)}}, \tag{8}$$

where a_i is i^{th} hidden layer's input weight vector and β_i is the corresponding bias term.

If n hidden nodes are used for the approximation, then the mean square error (MSE) is

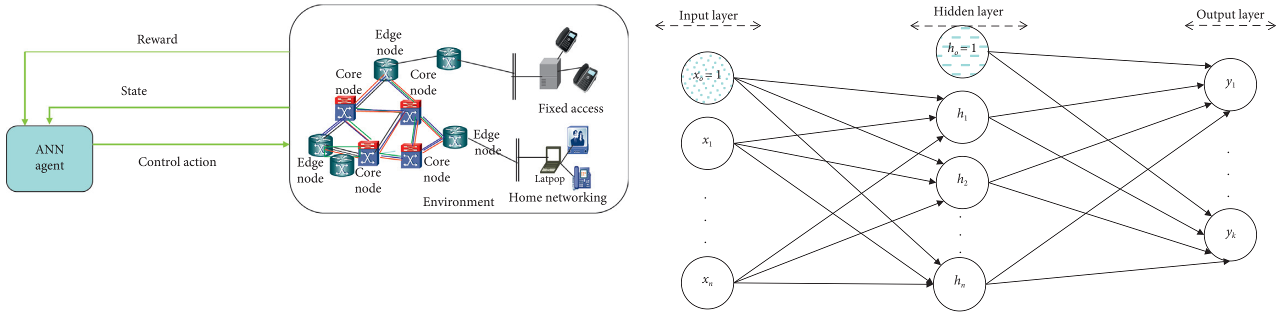
$$\varepsilon = \sum_{k=1}^{N} \left[p_k - \sum_{i=1}^{n} \beta_i g_i(x_k) \right]^2 = \left\| p - \sum_{i=1}^{n} \beta_i g_i \right\|_2^2. \tag{9}$$

In the above, $p = [p_1, \ldots, p_N]^T$ and $g_i = [g_i(x_i), \ldots, g_i(x_N)]^T$. At the n^{th} step, the training error is

$$\varepsilon_n = \left\| p - \sum_{i=1}^{n} \beta_i^2 g_i \right\|_2^2 = \left\| e_{n-1} - \beta_n g_n \right\|_2^2, \tag{10}$$

where $e_{n-1} = p - \sum_{i=1}^{n-1} \beta_i g_i$.

The output weight can thus be minimized using



(a)

(b)

Figure 4: Reinforcement learning principles.

$$\beta_n = \frac{e_{n-1}^T g_n}{\|g_n\|_2^2}. \tag{11}$$

5. Node-Regulated Deflection Routing Framework

We describe our proposed scheme as follows. Firstly, we discuss its principles followed by the learning scheme that would be used by the incorporated agent at a node in making key routing decisions. The ultimate objective of the scheme is to find an optimal route to deflect a contending burst if deflection routing is the only contention resolution scheme implemented in the particular network. The scheme thrives to ensure that each core node where contention is likely to occur learns to select an optimal deflection routing link(s) in terms of minimal blocking and delay with respect to the original route. In so doing, it thrives to minimize both signalling and computational overheads.

5.1. The Proposed Scheme. The proposed scheme's focus is on regulating selection of a deflection output port such that the chosen deflection route does not degrade the QoS of both the deflected bursts and existing traffic on the deflection link/path. We assume that should any two data bursts destined contend for the same output port, only one of them will be routed on the original path and the other will be deflected onto a selected least-cost deflection route. This is further illustrated by the queuing model in Figure 5(a). As defined in [20], deflection path server #1 queue represents a deflection path that offers a QoS nearing that of the original path.

A data burst will be deflected to path server #1 only if the controller buffer's capacity exceeds a threshold state q_1; otherwise, it will always be routed on the original path. Similarly, path server #2 would be the next-choice deflection route after the threshold exceeds q_2. Otherwise, in the absence of contention, the original path will always be preferred. The overall performance of a given link is time dependent; hence, Figure 5(b) illustrates these state changes for the two selected deflection paths. When busy, a selected deflection path exits the projected performance bounds at a rate α_j and conversely its performance improves at a rate β_j.

Summarily, the key steps are as follows:

(i) Sending node dispatches a BCP whose offset time is calculated according to (3) to the next intermediate node.

(ii) The BCP is processed upon its arrival at the next intermediate node. If desired primary route's output port and link wavelength are available, the burst will be accepted.

(iii) However, if contention or its imminence is detected, then the contention is resolved before the data bursts arrive as follows:

  (a) If the current node is the ingress, its BCP is dropped and retransmission attempted at a later time.

  (b) The other burst(s) can be either routed via the primary route, deflected to an alternate path, or worst case be discarded. This is done according to the set of rules in step iv as follows:

(iv) Data burst assigned to the original path: there are two or more contending bursts that have arrived and are all in transit. The controller buffer is in state $q < q_1$, and there are enough free wavelengths to accommodate all the contending bursts. Note that a link may have more than one fiber.

(a) deflected to path #1: the controller buffer is in state

$$q_1^* \le q \le q_2^*, \tag{12}$$

(b) deflected to path #2: the controller buffer is in state

$$q_2^* \le q \le q_N, \tag{13}$$

where q_N is the maximum possible aggregated deflection capacity on the two links.

Note that the threshold values q_1^* and q_2^* are set by taking into the key QoS metrics such as end-to-end delay and blocking.

5.2. Reinforcement Learning. We assume that the routing decisions pertaining to which deflection paths are determined by a routing agent. The Q-learning algorithm [21, 22]

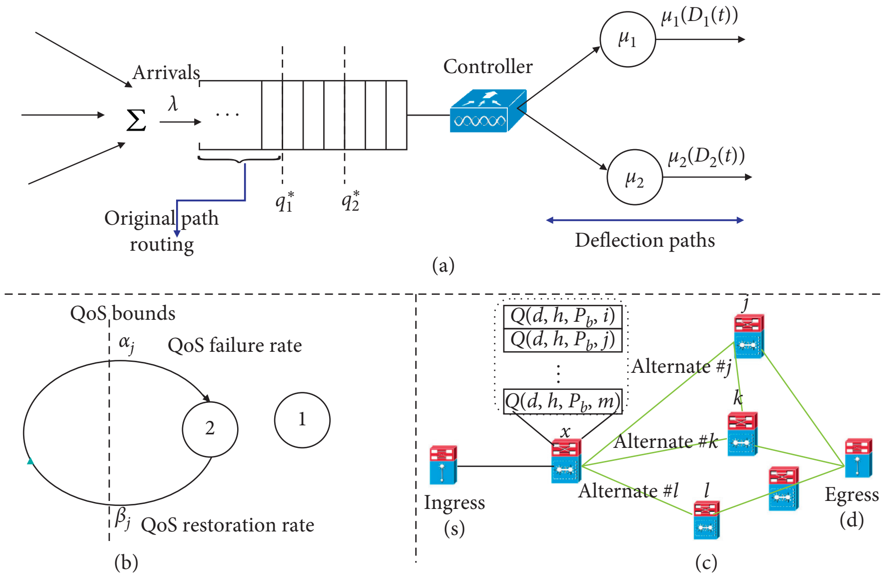


Figure 5: (a) Proposed scheme queuing model. (b) QoS state transitions. (c) Routing tables.

is adopted for this scheme. The Q-learning agent will gradually learn the best deflection route choice given a particular network state (s) after several trials of all possible actions (a) and by evaluating the corresponding reward $Q^{\pi}(s, a)$. The algorithm can be summarized in equation form as follows:

$$Q^{\pi}(s_t, a_t) \longleftarrow Q^{\pi}(s_t, a_t) + \alpha[r_{t+1} + \gamma_{a_{t+1}} \max Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi} s_t, a_t], \quad (14)$$

where r is the discount rate, α is the learning speed, and s_t, a_t, and r_t are the state, action, and reward at a given time t, respectively.

Each core node's incorporated agent always learns the states of all links connected to the node and updates this information in the node's routing tables. In so doing, it stores Q-values for each outgoing/incoming link, each represented by a vector with entries such as blocking probability (P_b) and hop count (h). In addition, each entry is further indexed by the (destination, neighbouring node); (d, k), where d is the destination and k is the neighbouring node.

When it is desired to deflect a burst to a particular destination, the node will choose a link with the highest Q; that is,

$$y = \underset{z \in N(x), z \neq p}{\arg\max}\, Q_x^{\pi}(d, k, h, P_b), \quad (15)$$

where $Q_x^{\pi}(d, k, h, P_b)$ is the Q-values associated with neighbouring node z with respect to the destination (d) and $N(x)$ is the set of nodes (links) neighbouring x.

Subsequently, the node controller chooses the best two performing links to a particular destination, path #1, (Q_{1x}^{π}) and path#2, (Q_{2x}^{π}), where $Q_{1x}^{\pi} \geq Q_{2x}^{\pi}$. It then sets their threshold buffers q_1 and q_2.

After node x successfully deflects a data burst to a neighbouring node y, its agent receives a feedback signal $Q_y^{\pi}(d, z, h, P_b)$ that it uses to compute a reinforcement numerical as

$$f_{yx} = Q_y^{\pi}(d, z, h, P_b) \times Q_x^{\pi}(d, z, h, P_b). \quad (16)$$

Node x uses this value to update its deflection tables as

$$Q_x^{\pi}(d, y) \longleftarrow Q_y^{\pi}(d, y) = \frac{\alpha f_{yx}(1 - B_{xy})}{D_x(d, y)} - Q_x^{\pi}(d, y), \quad (17)$$

where B_{xy} is the burst lost probability along link $x \longleftrightarrow y$;

$$B_{xy} = \begin{cases} \dfrac{N_{\text{drop}_{xy}}}{\text{sent}_{xy} + N_{\text{drop}_{xy}}}, & \text{if } \text{sent}_{xy} + N_{\text{drop}_{xy}} > 0, \\ 0, & \text{if } \text{sent}_{xy} + N_{\text{drop}_{xy}} = 0. \end{cases} \quad (18)$$

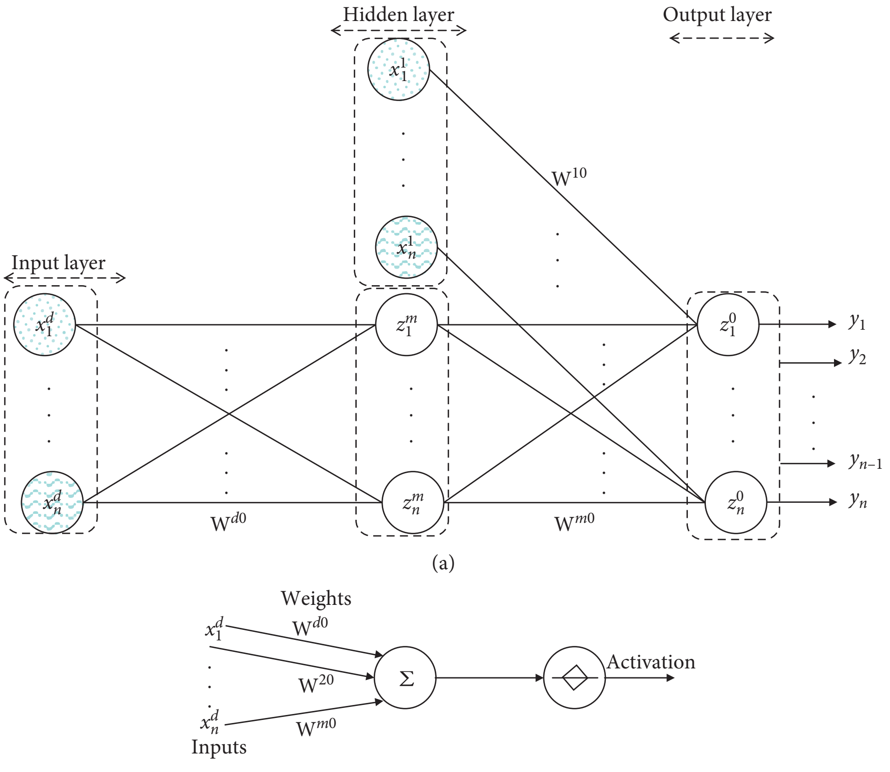
In (18), $N_{\text{drop}_{xy}}$ is the total number of dropped bursts, while sent_{xy} is the total number of successfully transmitted bursts on the given link.

The deflection path loss probability from the point of deflection to destination d is

$$P_{xp} = 1 - \prod_{1 \leq i \leq |P|-1} (1 - B_{h,h_{+1}}), \quad (19)$$

where $h_{1,\ldots,h|p|}$ is the number of remaining hops to the intended egress node.

5.3. Nodal Agent. In this section, we assume a feedforward single hidden ANN set for deflection routing. Its structure is shown in Figure 6.



(a)

(b)

Figure 6: ANN for deflection routing.

It relies on its look-up table to store Q-values representing the states of all output links at a node to a particular egress node. This Q entry takes into account QoS metrics such as burst loss probability and remaining hope count as in (15). When a node receives BCPs and detects contention, its controller deflects one of the contending bursts to a neighbouring node with the highest Q-value with respect to intended destination. The agent has three layers, namely input, middle, and output layers. We assume that the OBS node has n outgoing ports. The input layer has two sublayers; namely,

$$\begin{aligned}\mathbf{I}^l &= \left[x_1^l, \ldots, x_n^l\right], \\ \mathrm{I}^d &= \left[x_d^l, \ldots, x_n^d\right].\end{aligned} \tag{20}$$

If a port is blocked, then it is set to 1; otherwise, it is always set to 0.

The $\mathbf{I}^l$ input section's weights to the output layer can be expressed as a $n \times n$ matrix as follows:

$$W^{lo} = \begin{bmatrix} w_{11}^{lo} & 0 & \ldots & 0 \\ 0 & w_{22}^{lo} & \ldots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & & w_{nm}^{lo} \end{bmatrix}. \tag{21}$$

If $Z^m = [z_1^m, \ldots, z_n^m]$ is a binary vector representing the middle layer, then it maps the input and the output layers using the matrix

$$W^{dm} = \begin{bmatrix} \omega_{11}^{dm} & \omega_{12}^{dm} & \ldots & \omega_{1n}^{dm} \\ \omega_{21}^{dm} & \omega_{22}^{dm} & \ldots & \omega_{2n}^{dm} \\ \vdots & \vdots & & \vdots \\ \omega_{n1}^{dm} & \omega_{n2}^{dm} & & \omega_{nn}^{dm} \end{bmatrix}, \tag{22}$$

$$W^{mo} = \begin{bmatrix} \omega_{11}^{mo} & \omega_{12}^{mo} & \ldots & \omega_{1n}^{mo} \\ \omega_{21}^{mo} & \omega_{22}^{mo} & \ldots & \omega_{2n}^{mo} \\ \vdots & \vdots & & \vdots \\ \omega_{n1}^{mo} & \omega_{n2}^{mo} & & \omega_{nn}^{mo} \end{bmatrix}. \tag{23}$$

The matrices given in (21) and (22) reflect deflection route preferences and are continuously updated and new reinforcement signals are received from the OBS network (environment).

Therefore, for any given $h \times k$ matrix $\mathbf{Q}$, a Bernoulli semilinear operator $\Phi(\mathbf{Q})$ can be defined as

$$\Phi(\mathbf{Q}) = \begin{bmatrix} \dfrac{1}{1+e^{q_{11}}} & \cdots & \dfrac{1}{1+e^{q_{1k}}} \\ \vdots & \vdots & \vdots \\ \dfrac{1}{1+e^{q_{h1}}} & \cdots & \dfrac{1}{1+e^{q_{hk}}} \end{bmatrix}. \tag{24}$$

In each decision making, a probability vector

$$\mathbf{P}^m = \lfloor p_1^m \ \dots \ p_n^m \rfloor, \tag{25}$$

is computed using the Bernoulli semilinear operator as

$$\mathbf{P}^m = \Phi\left(\mathbf{I}^d \times \mathbf{W}^{mo}\right). \tag{26}$$

Each element $x_k^m \in \mathbf{Z}^m$ is calculated as follows:

$$x_k^m = \begin{cases} 0, & \text{if } p_k^m < 0.5, \\ U(0,1), & \text{if } p_k^m = 0.5, \\ 1, & \text{if } p_k^m > 0.5, \end{cases} \tag{27}$$

where $U(0,1)$ assumes 0 or 1 from a uniform distribution.

We further let $1 \times 2n$ row vector $\mathbf{I}^o = [\mathbf{I}^l\mathbf{Z}^m]$ represent an input to the output layer $\mathbf{Z}^o$.

The $n \times 2n$ matrix $\mathbf{W}^o = \lfloor \mathbf{W}^{lo}\mathbf{W}^{mo} \rfloor$ represents the weight matrix of the output layer $\mathbf{Z}^o$. The probability vector is calculated as

$$\mathbf{P}^o = \lfloor p_1^0, \dots, p_n^0 \rfloor = \Phi\left(\mathbf{I}^o \times (\mathbf{W}^o)^T\right). \tag{28}$$

Finally, the binary output vector $\mathbf{Z}^o = [z_1^o, \dots, z_n^0]$ can now be calculated. Indices of 1*s* appearing in the output vector indicate availability of that output for deflection.

6. Simulation Results and Analysis

In this section, we evaluate the performance of the proposed rDr scheme by way of simulation. The simulation focuses on QoS metrics such as loss probability, frequency of contentions occurring, number of deflections, mean burst end-to-end latencies, number of deflections, and deflection path lengths (measured in hops).

Our simulations are carried out on an online NS-2 simulator [23], together with OBS modules that are implementable in the same simulator [24]. Only the 15-node mesh topology network shown in Figure 7 was considered for the simulations. Each fiber link terminates with a standard port on either end and is bidirectional and all fibers have equal numbers of wavelengths. Only the edge nodes on the network can be ingress (source) and egress (destination) points for the traffic, while the rest route traffic. The sources and destination pairs are randomly chosen from among the designated edge nodes. The traffic load normalized to the maximum link capacities, i.e., the ratio of incident traffic to a node and the aggregated capacities of all the wavelengths constituting the link.

The dynamic segmented burst assembly approach coupled with adjustable offset timing algorithm as described in Section 2 is assumed. For the sake clarity, the dynamic segmented burst assembly algorithm's model is illustrated in Figure 8, which is presented in detail in [15]. Two traffic type streams whose arrival rates are λ_1 (high priority) and λ_2 (low priority) are considered. The assembly approach is summarized as follows:

(i) The buffer sizes are H_1 (high priority) and H_2 (low priority)

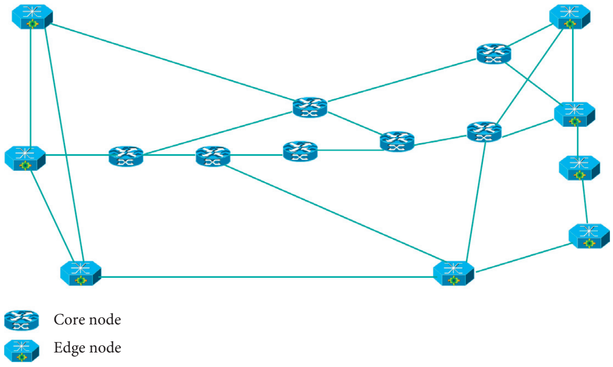


Figure 7: NSFNET topology network.

(ii) A new HP segment arrival will always be accepted in the HP buffer provided it is not full; otherwise, it is discarded

(iii) If upon arrival of a new LP segment, the total number of segments in the LP buffer is i, $i < H_2$ and the number of segments in the HP buffer is j, $j < H_1$, the LP segment jumps to the HP buffer with probability $p_i(j)$ and will now be served as a HP segment; as such an arriving LP segment joins the LP buffer with probability $1 - p_i(j)$

(iv) A new LP segment arrival will be discarded if the HP and LP buffers are in states H_1 and H_2, respectively

(v) A new LP segment arrival finding the LP buffer in state (H_2), but the HP buffer in state j, $j < H_1$ will be admitted to the latter with probability $p_{H_2}(j)$, or else be discarded with probability $1 - p_{H_2}(j)$

In our performance evaluations, we use both delay-sensitive and non-delay-sensitive sources. Since all networks that employ (adopt) deflection routing may suffer from insufficient offset timing problem, we first simulate the end-to-end delays as well as burst loss ratios of the proposed adjustable offset timing scheme coupled with segmented burstification. Here we restrict ourselves to the HP class (delay-sensitive traffic type) as the two QoS parameters (blocking and end-to-end delay) are relatively critical to this class than the LP one. Note that when LP segments (bursts) are lost, they can always be retransmitted.

Each link has one GHz bandwidth and BCP processing time at each intermediate node is 1.5 μ secs. The propagation delay averages 1 μ sec on each hop. Each segment size is 50 packets and that of the segmented burst is 4000 kB. Only HP class data segments are generated and their arrival at the burstification buffers follows a Poisson distribution. We also set three different values for the maximum burst transfer delay ($t_{d,s}$) to 90 μ secs, 120 μ secs, and 135 μ secs. The traffic intensities will be varied from low to high. Shown in Figure 9 are the average end-to-end delays for the high-class segments (or packets) for three different values of maximum burst transfer delay limits. In this case, the offset time is computed according to (3) subject to (2).

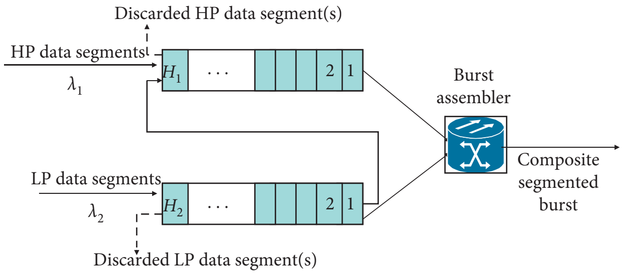


Figure 8: Dynamic segmented burst assembly approach.

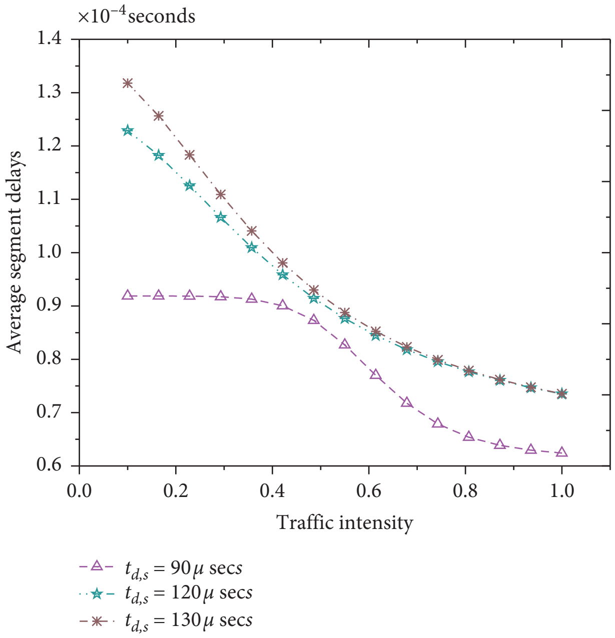


Figure 9: Average segment (packet) delays end to end.

As can be observed for low traffic levels, the average delay is determined by the preset maximum burst transfer delay values which are 90, 120, and 130 microseconds, respectively. The plotted results overall demonstrate that the Segmented Burst assembly approach coupled with the offset time approach will significantly reduce the end-to-end delays incurred by the HP traffic. It is also observed that for low traffic intensities, setting the maximum end-to-end delay ($t_{d,s}$) to a low value yields overall less delays. This is because the segmented bursts are dispatched for scheduling without waiting for the completion of burst aggregation. At high traffic intensities, the aggregation time is still less than the maximum delay.

We now turn to the burst loss ratio, in which we compare the proposed adjustable offset time scheme to the conventional OBS (where the offset time is fixed). The simulation results are shown in Figure 10. For traffic intensities ranging from zero to about 0.5, the two approaches perform comparably the same. However, the proposed adjustable offset approach still performs relatively better at peak traffic intensities. This is because as traffic increases, the conventional offset time scheme generates more bursts and this leads to more contentions (leading to burst losses). On the other hand, the proposed scheme generates relatively fewer but longer bursts and this coupled with adjustable offset timing still results in fewer contentions as well as sufficient burst processing times.

We further evaluate the performance of the proposed rDr scheme with respect to the number of deflections, burst loss probabilities, and average end-to-end delays. In so doing, we compare the scheme to existing similar schemes such as the following:

(i) Shortest Path Deflection Routing (SPDR) which was proposed in [25] is generally considered as the conventional OBS deflection scheme. We however place a restriction on the number of deflections by a single burst to three. This is to prevent endless looping in the OBS network which otherwise may lead to more contentions as well as other network performance degradations.

(ii) The Reinforcement Learning-Based Alternative approach (RLAR) which is analysed in [25] is basically based on Q–reinforcement learning NN nodes at each core node, and it allows data bursts to be routed on any available link rather than on a shortest path link from source to intended destination. It resolves contention by merely discarding one of the contending data bursts.

(iii) The Reinforced Learning-Based Deflection Routing Scheme (RLDRS) proposed in [26] is a proactive-based algorithm that capitalizes on what a NN set has learnt from the network by choosing a link with maximum Q in the deflection routing table to forward an incident contending burst.

Our blocking probability ranges are from $P_b = 1 \times 10^{-6}$ to about $P_b = 1 \times 10^{-3}$ and so our training data set is generated according to these desired ranges. Similarly, the same was done for the end-to-end delays which should be within the desired QoS ranges.

In our simulation, arrival processes of data bursts to a node are assumed to be Poisson-distributed, and hence service times are exponentially distributed. Listed in Table 1 are some of the key parameters for the simulation as well as training model. The data sets used are provided in Table 2.

During the actual training phase, the NN set was learned as outlined in IV; that is, initially we set the residual error value $e_0 = p$. We also set the number of hidden nodes to zero, i.e., $n = 0$. Further, we define the termination condition e. We then progressively increase the number of hidden nodes n as long as the following conditions are not violated: $n \leq n_{\max}$ and $((\varepsilon_{n-1} - \varepsilon_n)/\varepsilon_{n-1}) > 0$. Further, we go on to add an additional $g_n(.)$ hidden to the network where, (a_n, b_n) are randomly generated. We then compute the new weight β_n.

$$\beta_n = \frac{e_{n-1}^T e_n}{\|g_n\|_2^2}, \tag{29}$$

and further,

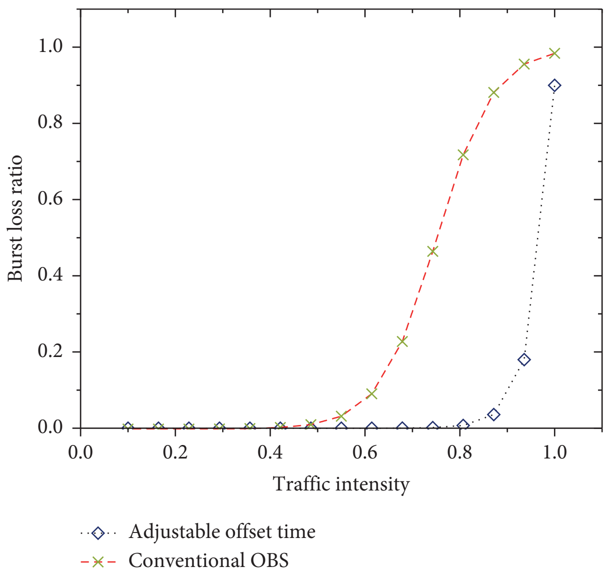


Figure 10: Burst loss ratio due to contentions.

Table 1: Input parameters to the NN set.

Parameter	Definition
N_λ	Capacity of link in, i.e., number of wavelengths
$\overline{\rho}_{s,d}$	Average normalized traffic load in s, d pair
$\Delta\overline{\rho}_{d,s}$	Difference between maximum and minimum traffic loads on a s, d pair and is uniformly distributed
$\overline{h}$	Mean path length
CR	Route traffic concentration
α	NN set learning rate

Table 2: Data sets.

Class	Lower bound		Upper bound	
	P_b	$t_{d,s}$ (μs)	P_b	$t_{d,s}$ (μs)
I	1×10^{-6}	750	1×10^{-4}	1000
II	1×10^{-4}	1001	5×10^{-5}	2000
III	5×10^{-2}	2001	0.25	5000

$$e_n = e_{n-1} - \beta_n^{g_n}. \qquad (30)$$

Once training is accomplished, simulations of various OBS network QoS metrics are obtained. In all cases, several random assignments of node destination pairs are assigned and then results obtained are averaged. Figure 11 shows the overall mean end-to-end delays experienced by the data bursts as they traverse the network. As can be observed in this figure, the SPDR outperforms all other algorithms since it is solely based on establishing shortest path deflection routes to an intended destination. The rDr is outperformed by the other three since it does not necessarily utilize the shortest paths and the burstification approach does regulate the magnitude of the offset times.

The end-to-end delay is acceptable for medium-sized networks as well as for most application types that do not impose stringent delay requirements. Because unresolved contentions may lead to burst losses, we proceed to compare the four schemes' performance with regard to blocking.

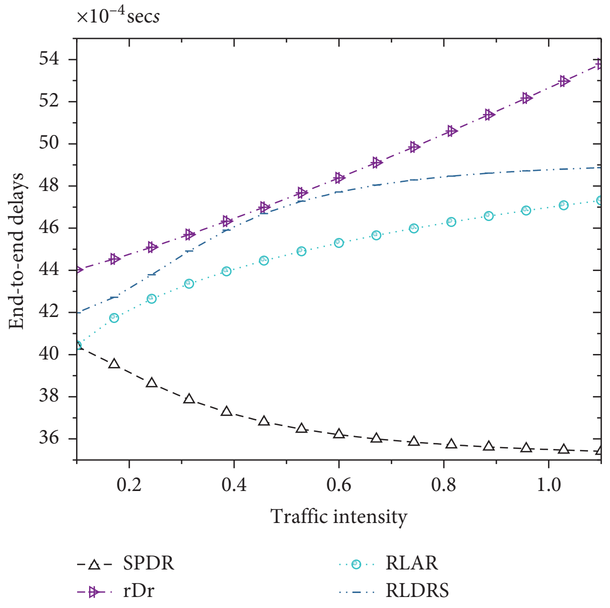


Figure 11: Average end-to-end burst delays.

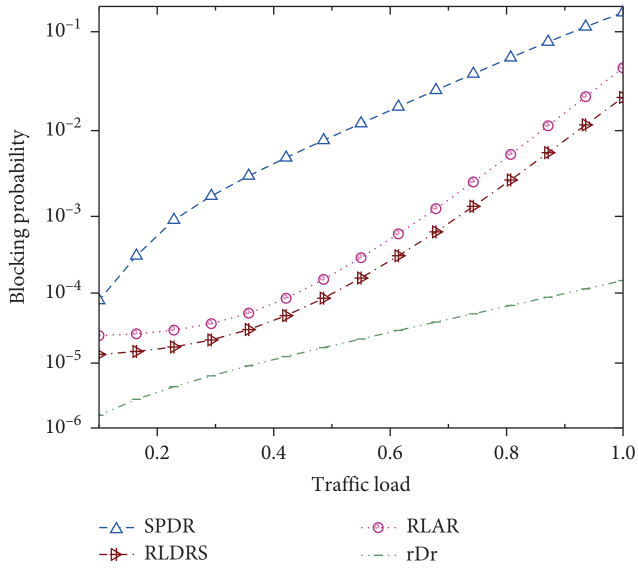


Figure 12: Blocking probability versus traffic load.

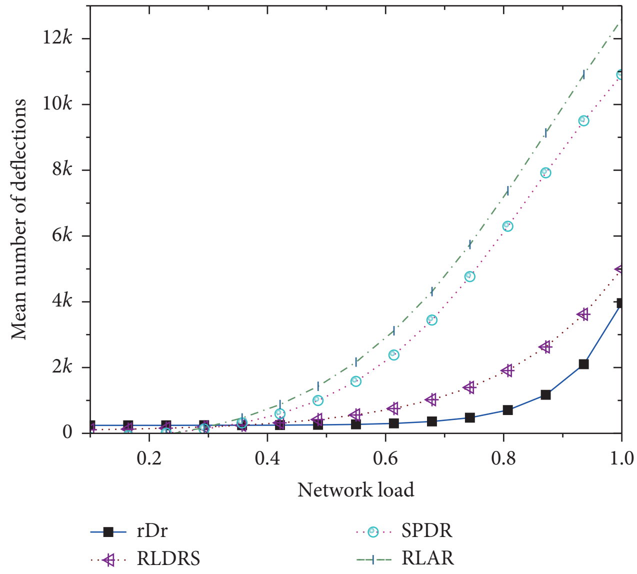


Figure 13: Mean number of deflections versus traffic load.

Shown in Figure 12 are the results obtained when the traffic load was varied from low to moderate and eventually to high (100% loading). It can be deduced from the graph that the rDr appears to be outperforming the other three algorithms at all levels of traffic in terms of burst blocking. This is partly attributed to the fact that the deflection paths in rDr are chosen from among the best two candidate routes and this still leaves the rest of the network unaffected by deflection traffic. The careful choice of deflection routes in rDr implies that any adverse effects of deflection routing are minimized. This can be easily observed in Figure 13 where we plot the mean number of deflections as the network load is increased gradually to maximum.

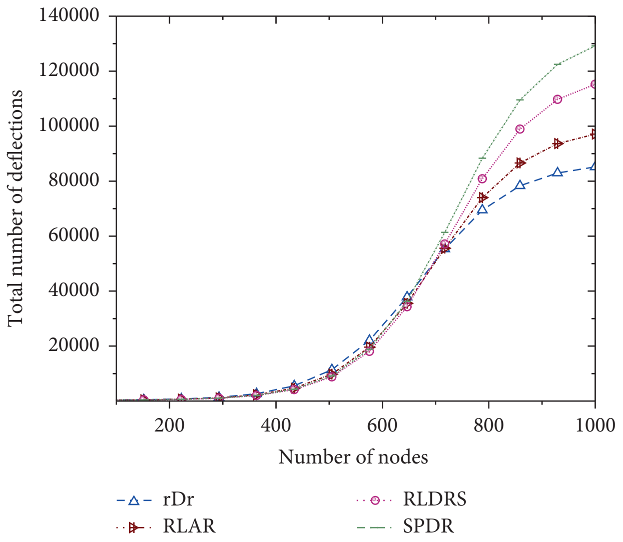


Figure 14: Total number of deflections versus number of nodes.

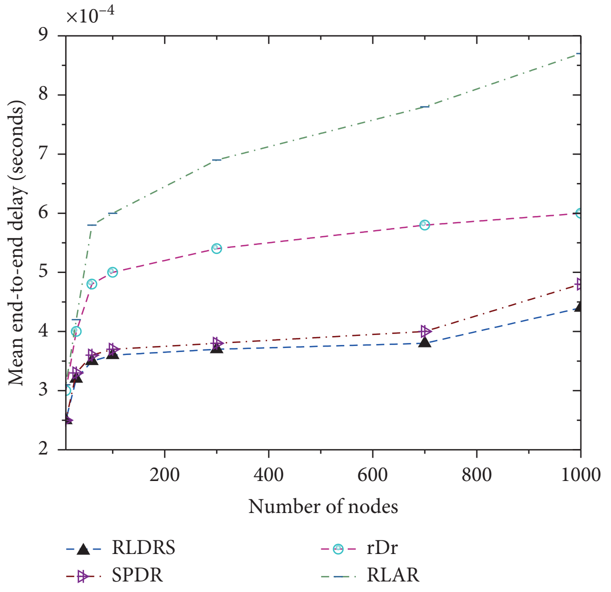


Figure 15: Mean end-to-end delays as function of network size.

It is noted that for the proposed scheme, the number of deflections only starts to surge when the network load exceeds 0.77.

The BRITE network topology generator [27] is further used to generate network topologies with varying numbers of nodes up to a maximum of 1000. With this generator, an edge that connects nodes v and u exists with probability:

$$\text{Prob}\left(\{v, u\}\right) = \beta e^{-(d(v,u)/L\delta)}. \tag{31}$$

In (28), $d(u, v)$ is the distance between nodes v and u; L is the maximum link distance between two nodes, whereas β and δ are parameters that take values in the range $[0, 1]$.

Figure 14 shows the performance of the 4 algorithms in terms of deflections. The rDr outperforms the rest as the

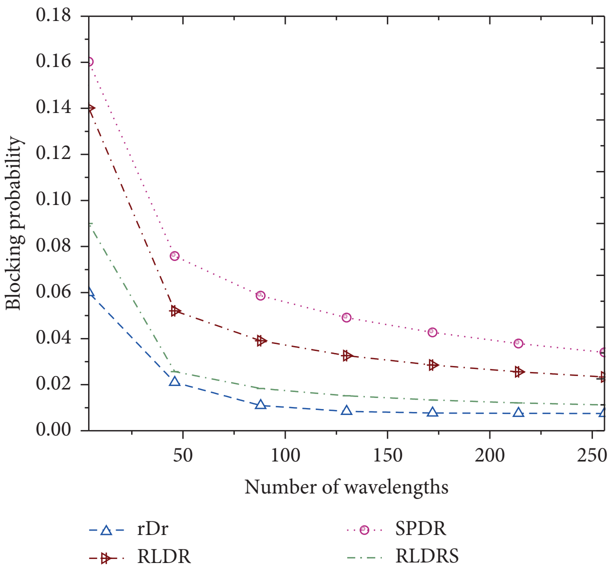


Figure 16: Blocking probabilities as a function of number of wavelengths.

network is expanded to a maximum of 1000 nodes. However, even though the blocking probabilities of the rDr algorithm is lower since the bursts are deflected less frequently, this is at the expense of having to incur larger end-to-end delays as the algorithm tends to select longer paths as indicated by the results plotted in Figure 15. RLDRS and the SPDR algorithms always opt for the shortest paths when deflecting contending bursts. However, the probability of further encountering both contentions as well as congestion is relatively higher because the majority of current day routing protocols always choose to route data via shorter routes.

Note that even though reduced burst deflection reduces the burst-loss probability, it still introduces excess traffic load in the network sections previously underloaded. We further extended our study to investigating the effects of increasing the number of wavelengths on the end-to-end delays. We had to fix the traffic load to about 60% and varied the traffic number of active wavelengths on each link from 4 to about 25.

It is observed from Figure 16 that rDr, RLDRS, and RLAR algorithms reduced blocking probabilities as the number of wavelengths is increased. However, this leads to increases in end-to-end delays.

7. Conclusions

Deflection routing being a key contention resolution scheme in present day OBS network motivated us into proposing and analysing a reinforcement learning-based regulated deflection routing (rDr) scheme. For periodic updating of the entire network resources, we propose an artificial neural network signalling algorithm (ANN-S) that utilizes a single hidden layer as well as an associative learning algorithm to periodically update routing tables so that nodes can make better routing decisions. Furthermore, we proposed coupling the use of the adjustable offset timing approach with segmented burst assembling to enhance QoS performance of the overall network, in particular, with regard to accommodating delay-sensitive applications. The reinforced learning is implemented in the NS-3 platform available online. We tested reinforcement learning-based regulated deflection routing algorithm on the same platform. We also utilized the National Science Foundation (NSF) network topology and random graphs that consisted in varying the number of nodes up to a maximum of 1000 for further simulations. Ultimately, we compared the rDr scheme to existing similar OBS deflection schemes such as Shortest Path Deflection Routing (SPDR), the Reinforcement Learning-Based Alternative Approach (RLAR) [19], and the Reinforced Learning-Based Deflection Routing Scheme (RLDRS) proposed in [20]. Key notable QoS metrics of interest during the simulations were burst loss probabilities, end-to-end delays, and the number and frequency of deflections.

Our simulation results showed that overall the rDr scheme by comparison significantly and effectively reduces loss probability even though at a cost of negligible increases in the end-to-end mean delays. Our further investigation will include increasing the number of traffic classes in an attempt to balance between the frequencies of contentions that ultimately contribute to burst loss probabilities versus end-to-end latencies especially during peak traffic periods. In [28], the authors propose and discuss methods related to QoS evaluation with regard to selecting an energy efficient network. The metrics taken into account also include energy cost, network cost, end-to-end delays (latency), and bandwidth. In our future work, we will incorporate the energy cost metric when selecting a "least cost route" so as to guarantee an energy-efficient network design [29]. With regard to proper dimensioning of the available network resources, we will further take into account the effect of access network traffic behaviours.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

[1] T. M. Hirata and T. Takine, "Dynamic burst discarding scheme for deflection routing in optical burst switching networks," *Optical Switching and Networking*, vol. 4, no. 2, pp. 106–120, 2007.

[2] L. Huo, D. Jiang, X. Zhu, Y. Wang, Z. Lv, and S. Singh, "Ä SDN based fine grained measurement and modelling approach to vehicular communication network traffic," *International Journal of Communication Systems*, 2019.

[3] L. Nie, D. Jiang, and L. Guo, "A comprehensive sensing based approach to end to end network traffic reconstruction utilizing partial measured origin-desination flows," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 18, pp. 1108–1117, 2015.

[4] S. S. Chawathe, "Analysis of burst header packets in optical burst switching networks," in *Proceedings of the 2018 IEEE*

17th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, November 2018.

[5] M. Nandi, A. Turuk, D. Puthal, and S. Dutta, "Best fit void filling algorithm in optical burst switching networks," in *Proceedings of the Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09*, pp. 609–614, Nagpur, Maharashtra, India, 2009.

[6] T. Bonald, R. Indre, and S. Oueslati, "Adaptive optical burst switching," in *Proceedings of the 23rd International Teletraffic Congress*, pp. 150–157, San Francisco, CA, USA, September 2011.

[7] C. Papazoglou, P. G. Sarigiannidis, G. I. Papadimitriou, A. S. Pomportsis, S. Andreas, and S. Pomportsis, "The use of a triangular estimator to improve scheduling in optical burst switched networks," *International Journal of Communication Systems*, vol. 23, no. 2, pp. 187–203, 2010.

[8] B. Komatireddy, D. Chandran, and V. M. Vokkarane, "TCP-aware load-balanced routing in optical burst-switched (OBS) networks," in *Proceedings of the Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference, OSA Technical Digest Series (CD) (Optical Society of America, 2007), paper OWC6*, New York, NT, USA, 2007.

[9] K. Ramantas, T. R. Varga, J. C. Guerri, and K. Vlachos, "A preemptive scheduling scheme for flexible QoS provisioning in OBS networks," in *Proceedings of the IEEE Sixth International Conference on Broadband Communications, Networks, and System (BROADNETS 2009)*, pp. 1–6, Faro, Portugal, 2009.

[10] A. H. Belbekkouche and M. Gendreau, "A reinforcement learning-based deflection routing scheme for buffer-less OBS networks," in *Proceedings of the IEEE Global Telecommunications Conference, (IEEE GLOBECOM'08)*, pp. 1–6, New Orleans, LO, USA, 2008.

[11] G. Gurel, O. Alparslan, and E. Karasan, "nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS networks," *Annals of Telecommunications*, vol. 62, no. 5-6, pp. 618–632, 2007.

[12] M. De Leenheer, J. Buysse, C. Develder, B. Dhoedt, and P. Demeester, "Deflection routing in anycast-based OBS Grids," in *Proceedings of the IEEE International Workshop on Optical Workshop on Optical Burst/Packet Switching*, Nagpur, Maharashtra, India, 2009.

[13] C. Xiaoyuan, X. H. W. Jian, and L. Jintong, "An adaptive offset time scheme in obs network," in *Proceedings of the 2008 7th International Conference on Optical Internet*, Tokyo, Japan, 2008.

[14] T. Mikoshi and T. Takenaka, "Improvement of burst transmission delay using offset time for burst assembly in optical burt switching," in *Proceedings of the7th Asia-Pacific Symposium on Information and Telecommunication Technologies*, pp. 13–18, Maldives, South Asia, 2008.

[15] H. Sibanda and B. Nleya, "Model analysis of dynamic priority segmented burstification and deflection routing," in *Proceedings of the 2017 IEEE AFRICON*, Cape Town, South Africa, September 2017.

[16] A. F. L. Calderon, E. J. V. Porras, and O. J. S. Parra, "Predicting traffic through artificial neural networks," *Contemporary Engineering Sciences*, vol. 10, no. 24, pp. 1195–1209, 2017.

[17] J. Mata, I. Miguel, R. J. Durán et al., "Artificial intelligence (AI) methods in optical networks: a comprehensive survey," *Optical Switching and Networking*, vol. 28, pp. 43–57, 2018.

[18] C.-S. Leung, W. Y. Wan, and R. Feng, "A regularizer approach for RBF networks under the concurrent weight failure situation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1360–1372, 2017.

[19] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.

[20] P. Z. Khumalo and B. Nleya, "A controllable deflection routing and wavelength assignment algorithm in OBS networks," in *Proceedings of the International Conference on Intelligent & Innovative Computing Applications (ICONIC)*, Plaine Magnien, Mauritius, December, 2018.

[21] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[22] S. Haerei and L. Trajikovic, "Intelligent deflection routing in bufferless networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 316–327, 2015.

[23] NS-Simulator, http://www.isi.edu/nsnam/ns.

[24] S. Gowda, R. K. Shenai, K. M. Sivalingam, and H. C. Cankaya, "Performance evaluation of TCP over optical burst-switched (OBS) WDM networks," in *Proceedings of IEEE International Conference on Communications, 2003. ICC '03*, vol. 2, pp. 1433–1437, Anchorage, AK, USA, May 2003.

[25] T. Coutelen, H. Elbiaze, B. Jaumard, and A. Metnani, "Impact of network topology and traffic load balancing on deflection routing efficiency," in *Proceedings of IASTED OCSN*, Banff, Canada, 2005.

[26] A. H. M. G. Belbekkouche, "A reinforcement learning- based deflection routing scheme for buffer-less OBS networks," in *Proceedings of IEEE GLOBECOM*, New Orleans, LO, USA, 2008.

[27] BRITE, 2014, http://www.cs.bu.edu/brite/.

[28] D. Jing, L. Huo, Z. Lv, H. Song, and W. Qin, "A joint multi criteria utility based network selection approach for vehicle to infrastructure networking," *IEEE Transactions on Intelligent Transporattion Systems*, vol. 19, no. 10, pp. 3305–3319, 2018.

[29] D. Jiang, L. Huo, and Y. Li, "Fine-granularity inference and estimations to network traffic for SDN," *PLoS One*, vol. 13, no. 5, Article ID e0194302, 2018.