Chapter 15
Deflection Routing in Complex Networks

Soroush Haeri and Ljiljana Trajković

Abstract Deflection routing is a viable contention resolution scheme in buffer-less network architectures where contention is the main source of information loss. In recent years, various reinforcement learning-based deflection routing algorithms have been proposed. However, performance of these algorithms has not been evaluated in larger networks that resemble the autonomous system-level view of the Internet. In this Chapter, we compare performance of three reinforcement learning-based deflection routing algorithms by using the National Science Foundation network topology and topologies generated using Waxman and Barabási-Albert algorithms. We examine the scalability of these deflection routing algorithms by increasing the network size while keeping the network load constant.

15.1 Introduction

The Internet is an example of a complex network that has been extensively studied. Optical networks are envisioned to be part of the Internet infrastructure intended to carry high bandwidth backbone traffic. It is expected that optical networks will carry the majority of TCP/IP traffic in the future Internet. Optical burst switching [50] combines the optical circuit switching and the optical packet switching paradigms. In optical burst-switched (OBS) networks, data are optically switched. Optical burst switching offers the reliability of the circuit switching technology and the statistical multiplexing provided by packet switching networks. Statistical multiplexing of bursty traffic enhances the network utilization. Various signaling protocols that have been proposed enable statistical resource sharing of a light-path among multiple traffic flows [6, 19].

In optical burst switching networks, packets are assembled into bursts that are transmitted over optical fibers. Optical burst switching is a buffer-less architecture that has been introduced to eliminate the optical/electrical/optical signal con-

Sections of this Chapter appeared in conference proceedings and a journal publication: [27–30].

versions in high-performance optical networks. The optical/electrical conversion occurs when a data packet arrives at a router and the source and destination addresses required for routing need to be extracted from the packet header. The electrical/optical conversion is then required to forward the packet through the appropriate outgoing link. Eliminating the conversions enables high capacity switching with simpler switching architecture and lower power consumption [66]. Furthermore, optical burst switching enables the optical fiber resources to be shared unlike other optical switching technologies such as Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) that reserve the entire lightpath from a source to a destination [48]. High-speed optical links are often used to connect the Internet Autonomous Systems and, hence, the optical burst switching may be used for inter-autonomous system communications.

Deflection routing is a viable contention resolution scheme that may be employed in buffer-less networks such as networks on chips or OBS networks. It was first introduced as "hot-potato" routing [8] because packets arriving at a node should be immediately forwarded [2,41]. Contention occurs when according to a routing table, multiple arriving traffic flows at a node need to be routed through a single outgoing link. In this case, only one flow is routed through the optimal link defined by the routing table. In the absence of a contention resolution scheme, the remaining flows are discarded because the optical node possesses no buffers. Instead of buffering or discarding packets of a flow, deflection routing helps to temporarily deflect them away from the path that is prescribed by the routing table. While other methods have also been proposed to resolve contention such as wavelength conversion [39], fiber delay lines [60], and control packet buffering [1], deflection routing has attracted significant attention [15, 37, 38] as a viable method to resolve contention in buffer-less networks. This is because deflection routing requires only software modifications in the routers [67] while other schemes require deployment of special hardware modules.

Deflection routing algorithms proposed in the literature have only been tested on small-size networks such as the National Science Foundation network or general torus topologies that do not resemble the current Internet topologies [11, 29, 30, 34, 57]. Recent insights emanating from the discovery of power-law distribution of nodes degree [23] and scale-free properties of communication networks [4, 7] have influenced the design of routing protocols [40, 59]. Various empirical results confirm the presence of power-laws in the Internet's inter-autonomous system-level topologies [44, 52, 58]. Waxman [63] and Barabási-Albert [7] algorithms have been widely used to generate Internet-like graphs.

In this Chapter, we compare performance of the Q-learning-based Node Degree Dependent (Q-NDD) [30] deflection routing algorithm, the Predictive Q-learning Deflection Routing (PQDR) [29] algorithm, and the Reinforcement Learning-based Deflection Routing Scheme (RLDRS) [11] by using the National Science Foundation (NSF), randomly generated Waxman [63], and scale-free Barabási-Albert [7] network topologies. Based on these deflection routing algorithms, network nodes learn to deflect packets optimally using reinforcement learning algorithms.

The remainder of this Chapter is organized as follows. In Section 15.2, we provide a brief survey of to deflection routing algorithms. We also describe algorithms for generating network topologies and briefly present the Internet traffic characteristics. The PQDR algorithm and RLDRS are described in Section 15.3. The design of Q-NDD deflection routing algorithm follows in Section 15.4. Performance evaluation and simulation scenarios are presented in Section 15.5. We conclude with Section 15.6.

15.2 Related Work

Various routing algorithms that employ reinforcement learning for generating routing policies were proposed in the early days of the Internet development [14, 21, 45, 49]. Routing in communication networks is a process of selecting a path that logically connects two end-points for packet transmission. A common approach is to map the network topology to a weighted graph and set the weight of each edge according to metrics such as number of hops to destination, congestion, latency, link failure, or the business relationships between the edge nodes. The path with the minimum cost is then selected for end-to-end communications.

An agent that learns how to interact with a dynamic environment through trial-and-error may use reinforcement learning techniques for decision-making [33]. 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 a 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 [61] 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:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \times \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) \right], \qquad (15.1)$$

where $0 < \alpha \leq 1$ is the learning rate and $0 \leq \gamma < 1$ is the discount factor. Q-learning has been considered as an approach for generating routing policies. The Q-routing algorithm [14] requires that nodes make their routing decisions locally. Each node learns a local deterministic routing policy using the Q-learning algorithm. Generating the routing policies locally is computationally less intensive. However, the Q-routing algorithm does not generate an optimal routing policy in networks with low loads nor does it learn new optimal policies in cases when network load decreases. Predictive Q-routing [21] addresses these shortcomings by recording the best experiences learned, which may then be reused to predict traffic behavior. The distributed gradient ascent policy search [49], where reinforcement signals are transmitted when a packet is successfully delivered to its destination, has also been proposed for generating optimal routing policies.

Packet routing algorithms in large networks such as the Internet should also consider the business relationships between Internet service providers. Therefore, randomness is not a desired property for a routing algorithm to be deployed in such environment. Consequently, reinforcement learning-based routing algorithms were not widely used for routing in the Internet because of their inherent random structure.

15.2.1 Deflection Routing

Deflection routing is proposed as a viable contention resolution scheme in bufferless networks such as OBS networks [50]. Two routing protocols operate simultaneously in such networks: an underlying routing protocol such as the Open Shortest Path First (OSPF) that primarily routes packets and a deflection routing algorithm that only deflects packets in case of a contention. Contention occurs when according to the routing table, multiple arriving traffic flows at a node need to be routed through a single outgoing link. In this case, only one flow is routed through the optimal outgoing link defined by the routing table. If no contention resolution scheme is employed, the remaining flows are discarded because the node possesses no buffers. In these cases, deflection routing helps temporarily misroute packets instead of buffering or discarding them.

Slotted and unslotted deflection schemes were compared [13, 20] and performance of a simple random deflection algorithm and loss rates of deflected data were analyzed [12, 26, 64, 71]. Integrations of deflection routing with wavelength conversion and fiber delay lines were also proposed [69, 72]. Deflection routing algorithms generate deflection decisions based on a deflection set, which includes all alternate links available for deflection. Several algorithms have been proposed to populate large deflection sets while ensuring no routing loops [31, 68].

Performance analysis of deflection routing based on random decisions shows that random deflection may effectively reduce blocking probability and jitter in

networks with light traffic loads [56]. Deflection protocols were recently further enhanced by enabling neighboring nodes to exchange traffic information. Hence, each node generates its deflection decisions based on better understanding of its surrounding [24, 47, 57]. Heuristic approaches may also be used to process the information gathered from the neighboring nodes.

Deflection routing may benefit from the random nature of reinforcement learning algorithms. A deflection routing algorithm coexists in the network with an underlying routing protocol that usually generates a significant number of control signals. Therefore, it is desired that deflection routing protocols generate few control signals. Reinforcement learning algorithms enable a deflection routing protocol to generate viable deflection decisions by adding a degree of randomness to the decision making process.

Reinforcement learning techniques have been recently employed to generate deflection decisions. The Q-learning path selection algorithm [34] calculates a priori set of candidate paths $P = \{p_1, \ldots, p_m\}$ for tuples (s_i, s_j), where $s_i, s_j \in S$ and $S = \{s_1, \ldots, s_n\}$ denotes the set of all edge nodes in the network. The i^{th} edge maintains a Q-table that contains a quality value (Q-value) for every tuple (s_j, p_k), where $s_j \in S \backslash \{s_i\}$ and $p_k \in P$. The sets S and P are 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 a signaling method or a procedure for handling feedback signals.

RLDRS [11] employs the Q-learning algorithm for deflection routing. The advantages of RLDRS are its precise signaling and rewarding procedures. Routing algorithms that are based on Q-learning inherit its drawbacks.

The PQDR [29] algorithm employs the predictive Q-routing (PQR) [21] that addresses the shortcomings of Q-learning-based routing by recording the best experiences learned, which may be reused to predict traffic behavior.

A drawback of the Q-learning path selection algorithm, RLDRS, and PQDR is their complexity, which depends on the size of the network. Hence, they are not easily scalable. The recently proposed Q-NDD algorithm [30] employs Q-learning for deflection routing. It scales well in larger networks because its complexity depends on the node degree rather than the network size. In the case of RLDRS and PQDR, nodes receive feedback signals for every packet that they deflect while in the case of Q-NDD, feedback signals are received only if the deflected packet is discarded by another node.

15.2.2 Network Topologies

Many natural and engineering systems have been modeled by random graphs where nodes and edges are generated by random processes. They are referred to as Erdös and Rényi models [22]. Waxman [63] algorithm is commonly used to synthetically generate such random network topologies. In a Waxman graph, an edge that connects nodes u and v exists with a probability:

$$\Pr\big(\{u,v\}\big) = \eta \exp\Big(\frac{-d(u,v)}{L\delta}\Big), \tag{15.2}$$

where $d(u,v)$ is the distance between nodes u and v, L is the maximum distance between the two nodes, and η and δ are parameters in the range $(0,1]$. Graphs generated with larger η and smaller δ values contain larger number of short edges. These graphs have longer hop diameter, shorter length diameter, and larger number of bicomponents [74]. Graphs generated using Waxman algorithm do not resemble the backbone and hierarchal structure of the current Internet. Furthermore, the algorithm does not guarantee a connected network [17].

Small-world graphs where nodes and edges are generated so that most of the nodes are connected by a small number of nodes in between were introduced rather recently to model social interactions [62]. A small-world graph may be created from a connected graph that has a high diameter by randomly adding a small number of edges. (The graph diameter is the largest number of vertices that should be traversed in order to travel from one vertex to another.) This construction drastically decreases the graph diameter. Generated networks are also known to have "six degrees of separation." It has been observed in social network that any two persons are linked by approximately six connections.

Most computer networks may be modeled by scale-free graphs where node degree distribution follows power-laws. Nodes are ranked in descending order based on their degrees. Relationships between node degree and node rank that follow various power-laws have been associated with various network properties. Eigenvalues vs. the order index as well as number of nodes within a number of hops vs. number of hops also follow various power-laws that have been associated with Internet graph properties [18, 23, 44, 51, 52]. The power-law exponents are calculated from the linear regression lines $10^{(a)}x^{(b)}$, with segment a and slope b when plotted on a log-log scale. The model implies that well-connected network nodes will get even more connected as Internet evolves. This is commonly referred as the "rich get richer" model [7]. Analysis of complex networks also involves discovery of spectral properties of graphs by constructing matrices describing the network connectivity.

Barabási-Albert [7] algorithm generates scale-free graphs that possess power-law distribution of node degrees. It suggests that incremental growth and preferential connectivity are possible causes for the power-law distribution. The algorithm begins with a connected network of n nodes. A new node i that is added to the network connects to an existing node j with probability:

$$\Pr(i,j) = \frac{d_j}{\sum_{k\in N} d_k}, \tag{15.3}$$

where d_j denotes the degree of the node j, N is the set of all nodes in the network, and $\sum_{k\in N} d_k$ is the sum of all node degrees.

The Internet is often viewed as a network of Autonomous Systems. Groups of networks sharing the same routing policy are identified by Autonomous System Numbers [5]. The Internet topology on Autonomous System-level is the arrangement of autonomous systems and their interconnections. Analyzing the Internet

topology and finding properties of associated graphs rely on mining data and capturing information about the Autonomous Systems. It has been established that Internet graphs on the Autonomous System-level exhibit the power-law distribution properties of scale-free graphs [23, 44, 52]. Barabási-Albert algorithm has been used to generate viable Internet-like graphs.

In 1985, NSF envisioned creating a research network across the United States to connect the recently established supercomputer centers, major universities, and large research laboratories. The NSF network was established in 1986 and operated at 56 kbps. The connections were upgraded to 1.5 Mbps and 45 Mbps in 1988 and 1991, respectively [54]. In 1989, two Federal Internet Exchanges (FIXes) were connected to the NSF network: FIX West at NASA Ames Research Center in Mountain View, California and FIX East at the University of Maryland [42]. The topology of the NSF network after the 1989 transition is shown in Fig. 15.1.

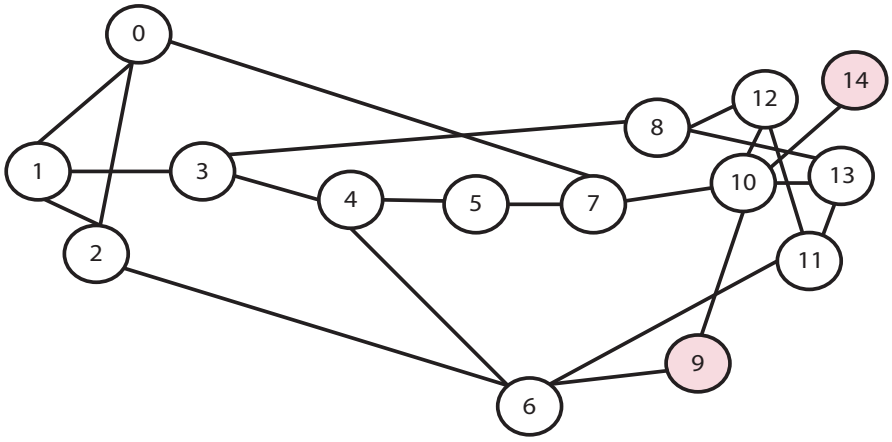

Fig. 15.1 Topology of the NSF network after the 1989 transition. Node 9 and node 14 were added in 1990.

15.2.3 Burst Traffic

Simulation of computer networks requires adequate models of network topologies as well as traffic patterns. Traffic measurements help characterize network traffic and are basis for developing traffic models. They are also used to evaluate performance of network protocols and applications. Traffic analysis provides information about the network usage and helps understand the behavior of network users. Furthermore, traffic prediction is important to assess future network capacity requirements used to plan future network developments.

It has been widely accepted that Poisson traffic model that was historically used to model traffic in telephone networks is inadequate to capture qualitative properties of modern packet networks that carry voice, data, image, and video applications [46]. Statistical processes emanating from traffic data collected from various applications indicate that traffic carried by the Internet is self-similar in nature [36]. Self-similarity implies a "fractal-like" behavior and that data on various time scales have similar patterns. Implications of such behavior are: no natural length of bursts, bursts exist across many time scales, traffic does not become "smoother" when aggregated, and traffic becomes more bursty and more self-similar as the traffic volume increases. This behavior is unlike Poisson traffic models where aggregating many traffic flows leads to a white noise effect.

A traffic burst consists of a number of aggregated packets addressed to the same destination. Assembling multiple packets into bursts may result in different statistical characteristics compared to the input packet traffic. Short-range burst traffic characteristics include distribution of burst size and burst inter-arrival time. Two types of burst assembly algorithms may be deployed in OBS networks: time-based and burst length-based. In time-based algorithms, burst inter-arrival times are constant and predefined. In this case, it has been observed that the distribution of burst lengths approaches a Gamma distribution that reaches a Gaussian distribution when the number of packets in a burst is large. With a burst length-based assembly algorithms, the packet size and burst length are predetermined and the burst inter-arrival time is Gaussian distributed. Long-range traffic characteristics deal with correlation structures of traffic over large time scales. It has been reported that long-range dependency of incoming traffic will not change after packets are assembled into bursts, irrespective of the traffic load [70].

A Poisson-Pareto burst process has been proposed [3, 75] to model the Internet traffic in optical networks. It may be used to predict performance of optical networks [30]. The burst arrivals are Poisson processes where inter-arrival times between adjacent bursts are exponentially distributed while the burst durations are assumed to be independent and identically distributed Pareto random variables. Pareto distributed burst durations capture the long-range dependent traffic characteristics. Poisson-Pareto burst process has been used to fit the mean, variance, and the Hurst parameter of measured traffic data and thus match the first order and second order statistics.

Traffic modeling affects evaluation of OBS network performance. The effect of the arrival traffic statistics depends on the time scale. Short time scales greatly influence the behavior of buffer-less high-speed networks. However, self-similarity is negligible when calculating blocking probability even if the offered traffic is long-range dependent over large time scales. Poisson approximation of the burst arrivals provides an upper bound for blocking probability [32]. Hence, we may assume that arrival processes are Poisson. Furthermore, assuming Poisson processes introduces errors that are shown to be within acceptable limits [72].

15.3 PQDR Algorithm and RLDRS

In this Section, we present details of the predictive Q-learning deflection routing algorithm (PQDR) [29]. PQDR determines an optimal output link to deflect traffic flows when contention occurs. The algorithm combines the predictive Q-routing (PQR) algorithm [21] and RLDRS [11] to optimally deflect contending flows. When deflecting a traffic flow, the PQDR algorithm stores in a Q-table the accumulated reward for every deflection decision. It also recovers and reselects decisions that are not well rewarded and have not been used over a period of time.

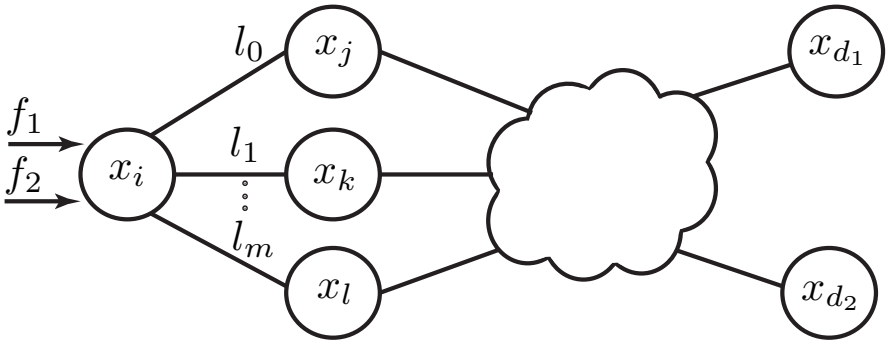

Fig. 15.2 A network with n buffer-less nodes.

An arbitrary buffer-less network is shown in Fig. 15.2. Let $\mathscr{N} = \{x_1, x_2, \ldots, x_n\}$ denote the set of all network nodes. Assume that each node possesses a shortest path routing table and a module that implements the PQDR algorithm to generate deflection decisions. Consider an arbitrary node x_i that is connected to its m neighbors through a set of outgoing links $\mathscr{L} = \{l_0, l_1, \ldots, l_m\}$. Node x_i routes the incoming traffic flows f_1 and f_2 to the destination nodes x_{d_1} and x_{d_2}, respectively. According to the shortest path routing table stored in x_i, both flows f_1 and f_2 should be forwarded to node x_j via the outgoing link l_0. In this case, node x_i forwards flow f_1 through l_0 to the destination x_{d1}. However, flow f_2 is deflected because node x_i is unable to buffer it. Hence, node x_i employs the PQDR algorithm to select an alternate outgoing link from the set $\mathscr{L}\backslash\{l_0\}$ to deflect flow f_2. It maintains five tables that are used by PQDR to generate deflection decisions. Four of these tables store statistics for every destination $x \in \mathscr{N}\backslash\{x_i\}$ and outgoing link $l \in \mathscr{L}$:

1. $Q_{x_i}(x,l)$ stores the accumulated rewards that x_i receives for deflecting packets to destinations x via outgoing links l.
2. $B_{x_i}(x,l)$ stores the minimum Q-values that x_i has calculated for deflecting packets to destinations x via outgoing links l.
3. $R_{x_i}(x,l)$ stores recovery rates for decisions to deflect packets to destinations x via outgoing links l.

4. $U_{x_i}(x,l)$ stores the time instant when x_i last updated the (x,l) entry of its Q-table after receiving a reward.

The size of each table is $m \times (n-1)$, where m and n are the number of elements in the sets $\mathscr{L}$ and $\mathscr{N}$, respectively. The fifth table $P_{x_i}(l)$ records the blocking probabilities of the outgoing links connected to the node x_i. A time window τ is defined for each node. Within each window, the node counts the successfully transmitted packets λ_{l_i} and the discarded packets ω_{l_i} on every outgoing link $l_i \in \mathscr{L}$. When a window expires, node x_i updates entries in its P_{x_i} table as:

$$P_{x_i}(l_i) = \begin{cases} \dfrac{\omega_{l_i}}{\lambda_{l_i}+\omega_{l_i}} & \lambda_{l_i}+\omega_{l_i} > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{15.4}$$

The PQDR algorithm needs to know the destination node x_{d_2} of the flow f_2 in order to generate a deflection decision. For every outgoing link $l_i \in \mathscr{L}$, the algorithm first calculates a Δt value as:

$$\Delta t = t_c - U_{x_i}(x_{d_2}, l_i), \tag{15.5}$$

where t_c represents the current time and $U_{x_i}(x_{d_2}, l_i)$ is the last time instant when x_i had received a feedback signal as a result of selecting the outgoing link l_i for deflecting a traffic flow that is destined for node x_{d_2}. The algorithm then calculates $Q'_{x_i}(x_{d_2}, l_i)$ as:

$$Q'_{x_i}(x_{d_2}, l_i) \quad = \quad \max\Big(Q_{x_i}(x_{d_2}, l_i) \;+\; \Delta t \;\times\; R_{x_i}(x_{d_2}, l_i), B_{x_i}(x_{d_2}, l_i)\Big). \tag{15.6}$$

$Qx_i(x_{d_2}, l_i)$ is then used to generate the deflection decision (action) ζ:

$$\zeta \leftarrow \arg\min_{l_i \in \mathscr{L}} \{Q'_{x_i}(x_{d_2}, l_i)\}. \tag{15.7}$$

The deflection decision ζ is the index of the outgoing link of node x_i that may be used to deflect the flow f_2. Let us assume that $\zeta = l_1$ and, therefore, node x_i deflects the traffic flow f_2 via l_1 to its neighbor x_k. When the neighboring node x_k receives the deflected flow f_2, it either uses its routing table or the PQDR algorithm to forward the flow to its destination through one of its neighbors (x_l). Node x_k then calculates a feedback value ν and sends it back to node x_i that had initiated the deflection:

$$\nu = Q_{x_k}(x_{d_2}, l_{kl}) \times D(x_k, x_l, x_{d_2}), \tag{15.8}$$

where l_{kl} is the link that connects x_k and x_l, $Q_{x_k}(x_{d_2}, l_{kl})$ is the (x_{d_2}, l_{kl}) entry in x_k's Q-table, and $D(x_k, x_l, x_{d_2})$ is the number of hops from x_k to the destination x_{d_2} through the node x_l. Node x_i receives the feedback ν for its action ζ from its neighbor x_k and then calculates the reward r:

$$r = \frac{\nu \times (1 - P_{x_i}(\zeta))}{D(x_i, x_k, x_{d_2})}, \tag{15.9}$$

where $D(x_i, x_k, x_{d_2})$ is the number of hops from x_i to the destination x_{d_2} through x_k while $P_{x_i}(\zeta)$ is the entry in the x_i's link blocking probability table P_{x_i} that corresponds to the outgoing link ζ (l_1). The reward r is then used by the x_i's PQDR module to update the (x_{d_2}, ζ) entries in the Q_{x_i}, B_{x_i}, and R_{x_i} tables. The PQDR algorithm first calculates the difference ϕ between the reward r and $Q_{x_i}(x_{d_2}, \zeta)$:

$$\phi = r - Q_{x_i}(x_{d_2}, \zeta). \tag{15.10}$$

The Q-table is then updated using ϕ as:

$$Q_{x_i}(x_{d_2}, \zeta) = Q_{x_i}(x_{d_2}, \zeta) + \alpha \times \phi, \tag{15.11}$$

where $0 < \alpha \leq 1$ is the learning rate. Table B_{x_i} keeps the minimum Q-values and, hence, its (x_{d_2}, ζ) entry is updated as:

$$B_{x_i}(x_{d_2}, \zeta) = \min(B_{x_i}(x_{d_2}, \zeta), Q_{x_i}(x_{d_2}, \zeta)). \tag{15.12}$$

Table R_{x_i} is updated as:

$$R_{x_i}(x_{d_2}, \zeta) \quad = \quad \begin{cases} R_{x_i}(x_{d_2}, \zeta) + \beta \dfrac{\phi}{t_c - U_{x_i}(x_{d_2}, \zeta)} & \phi < 0 \\ \gamma R_{x_i}(x_{d_2}, \zeta) & \text{otherwise} \end{cases}, \tag{15.13}$$

where t_c denotes the current time and $0 < \beta \leq 1$ and $0 < \gamma \leq 1$ are recovery learning and decay rates, respectively. Finally, the PQDR algorithm updates table U_{x_i} with current time t_c as:

$$U_{x_i}(x_{d_2}, \zeta) = t_c. \tag{15.14}$$

Signaling algorithms implemented in RLDRS and PQDR are similar. Their main difference is in the learning algorithm. RLDRS uses the Q-learning algorithm and, therefore, it only stores a Q-table $Q_{x_i}(x, l)$ that records the accumulated rewards that the node x_i receives for deflecting packets to destinations x via outgoing links l. As a result, a deflection decision ζ is generated using only the Q-table. Hence, instead of 15.6 and 15.7, RLDRS generates a deflection decision using:

$$\zeta \leftarrow \arg\max_{l_i \in \mathscr{L}} \{Q_{x_i}(x_{d_2}, l_i)\}. \tag{15.15}$$

15.4 Q-NDD Deflection Routing Algorithm

We describe here the newly proposed NDD signaling algorithm [30] 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 infrastruc-

ture 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. 15.3. 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. It 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.

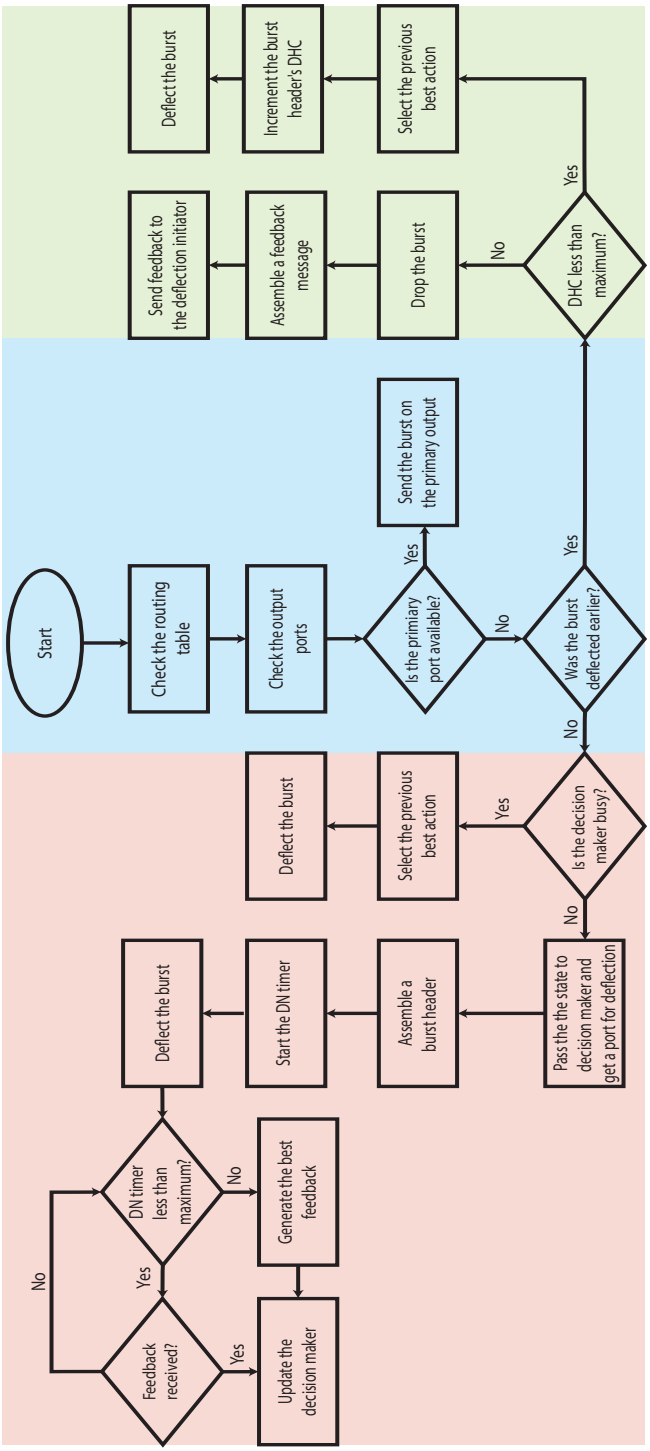

Fig. 15.3 The flowchart of the proposed signaling algorithm. The DN timer denotes the drop notification timer. Nodes wait for feedback signals until this timer reaches DHC_{max}. *DHC* denotes the deflection hop counter. This counter is a field in the burst header that is incremented by one each time the burst is deflected. DHC_{max} is set in order to control the volume of deflected traffic. A burst is discarded when its *DHC* value reaches the maximum.

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:

$$TTT = DrT - DfT. \tag{15.16}$$

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:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r - Q(s,a)\big). \tag{15.17}$$

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.

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.

15.5 Simulation Results

In order to evaluate and compare performance of Q-NDD, PQDR, and RLDRS, we implement these algorithms in the ns-3 network simulator [55] using the iDef framework [27]. ns-3 [55] is a discrete-event network simulator that is publicly distributed under the GNU GPLv2 [25] license. iDef [27] is designed to facilitate development of reinforcement learning-based deflection routing protocols by using its mapping, decision-making, and signaling modules. iDef is designed to minimize the dependency among its modules. Its components are shown in Fig. 15.4.

We compare the algorithms based on burst loss probability, number of deflections, average end-to-end delay, and average number of hops traveled by bursts. We first use the National Science Foundation (NSF) network topology shown in Fig. 15.1, which has been extensively used to evaluate performance of OBS networks [9–11, 34, 38, 64, 65]. We also use the Boston University Representative Internet Topology Generator (BRITE) [16] to generate autonomous system-level topologies that consist of 10, 20, 50, 100, 200, 500, and 1,000 nodes. These topologies are generated using the Waxman and Barabási-Albert algorithms. In all simulation

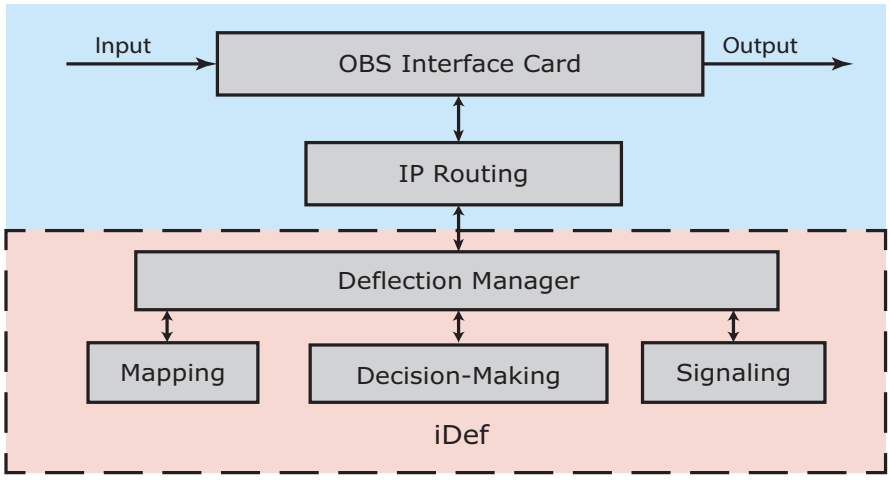

Fig. 15.4 iDef building blocks: The iDef is composed of deflection manager, mapping, signaling, and decision-making modules. The deflection manager module coordinates the communication between modules. Its purpose is to remove dependency among modules.

scenarios, we allow up to two deflections per burst ($DHC_{max} = 2$). The burst header processing time is set to 0.1 ms.

15.5.1 NSF Network Simulation Scenario

The NSF network topology was generated by extracting the geodetic coordinates [35] of the NSF network nodes from the Google Earth [53]. They are then transformed to the Cartesian coordinates. We assume the buffer-less OBS architecture for data transmission where nodes are connected using bi-directional 1 Gbps fiber links with 8 or 64 wavelengths.

Multiple Poisson traffic flows with a data rate of 1 Gbps are transmitted randomly across the network. Each Poisson flow is 50 bursts long with each burst containing 12.5 kB of payload. While the burst arrival process depends on the aggregation algorithm [43] deployed in a node, the Poisson process has been widely used for performance analysis of OBS networks because it is mathematically tractable [70, 73]. Each simulation scenario is repeated five times with various random assignments of nodes as sources and destinations. Simulation results are averaged over five runs.

The duration of the sliding window that the nodes employ to calculate burst loss probability on each of their interfaces is set to $\tau = 50$ ms. The learning, learning recovery, and recovery decay rates are $\alpha = 0.1$, $\beta = 0.7$, and $\gamma = 0.9$, respectively [21].

The burst loss probability as a function of the number of Poisson flows for 8 and 64 wavelengths is shown in Fig. 15.5. In all scenarios, the PQDR algorithm performs better than Q-NDD and RLDRS in terms of burst loss probability. The

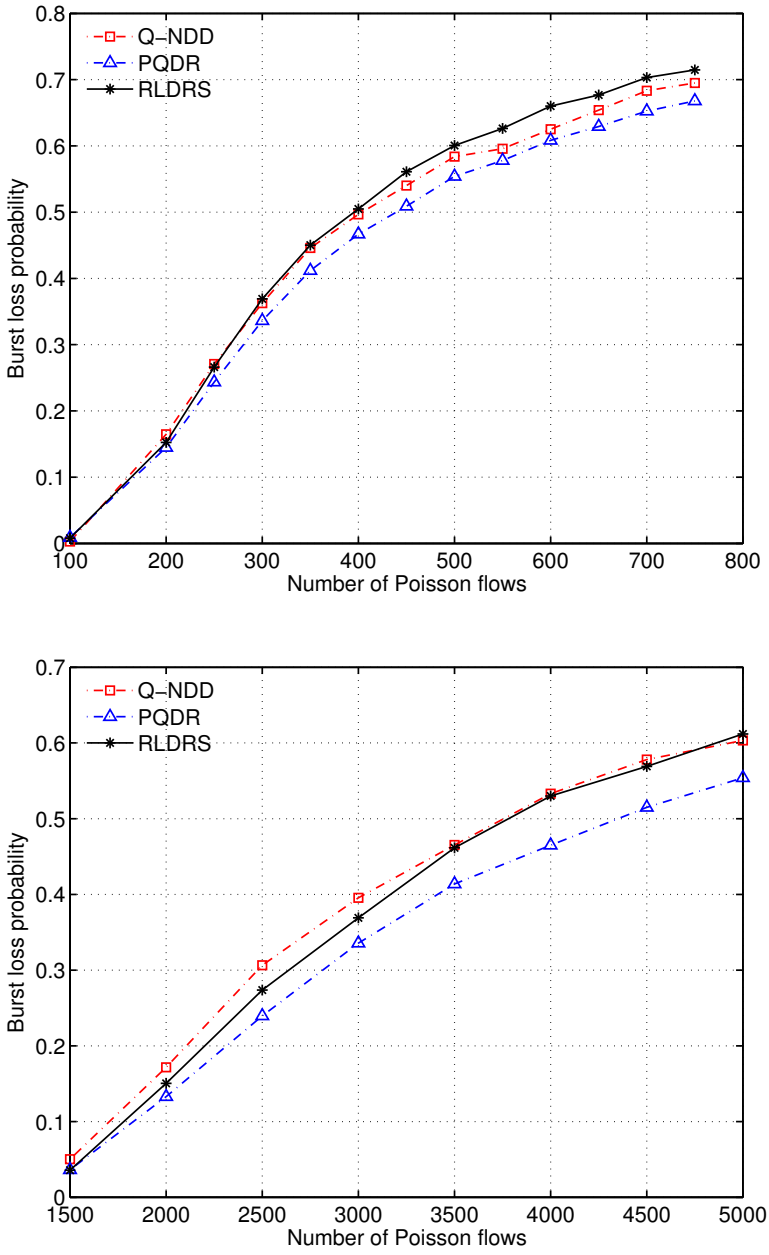

Fig. 15.5 Burst loss probability as a function of the number of Poisson flows in the NSF network simulation scenario with 8 wavelengths *(top)* and 64 wavelengths *(bottom)*.

results show that PQDR scales better as the number of wavelengths increases. For example, PQDR performs on average 7.0% better than RLDRS in case of 8 wavelengths and 800 traffic flows, as shown in Fig. 15.5*(top)*. Simulation results shown in Fig. 15.5*(bottom)* illustrate that PQDR performs on average 10% better than RLDRS in case of 64 wavelengths and 5,000 traffic flows.

Although burst deflection reduces the burst loss probability, it introduces excess traffic load to the network. This behavior is undesired from the traffic engineering point of view. Therefore, the volume of the deflected traffic should also be considered as a performance measure for deflection routing algorithms. We use two metrics to compare the Q-NDD algorithm, PQDR algorithm, and RLDRS in terms of the volume of the deflected traffic: deflection ratio and average number of deflections. We define deflection ratio as the number of deflected bursts divided by the number of transmitted bursts:

$$\text{Deflection ratio} = \frac{\text{Number of deflected bursts}}{\text{Number of transmitted bursts}}. \tag{15.18}$$

Deflection ratio and average number of deflections as a function of the number of traffic flows are shown in Fig. 15.6 and Fig. 15.7, respectively. PQDR deflects on average 20% fewer bursts than Q-NDD and RLDRS in the 64-wavelength scenario, as shown in Fig. 15.7.

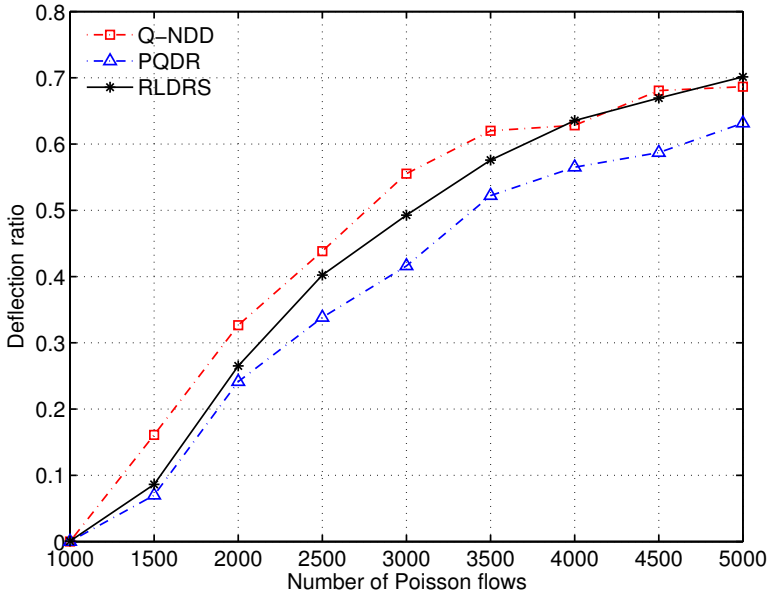

Fig. 15.6 Deflections ratio as a function of the number of traffic flows in the NSF network scenario with 64 wavelengths. PQDR deflects fewer packets.

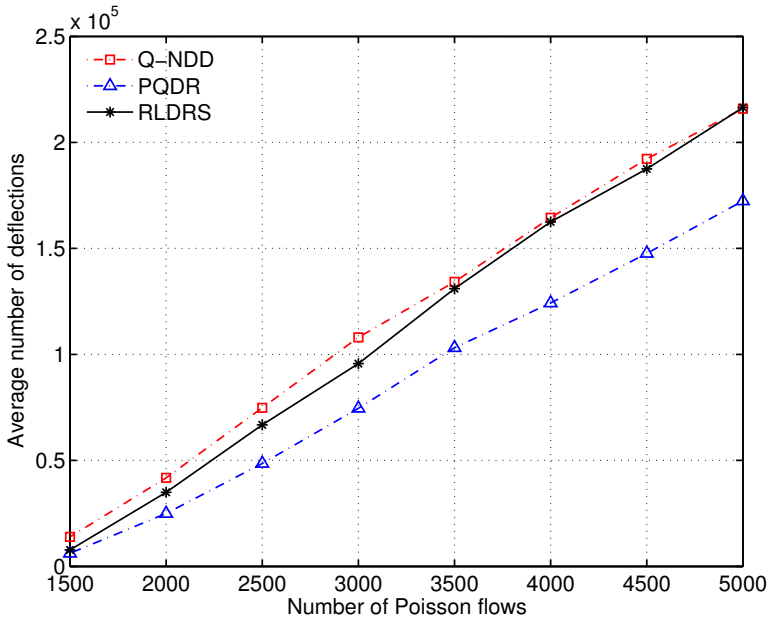

Fig. 15.7 Average number of deflections as a function of the number of traffic flows in the NSF network scenario with 64 wavelengths. PQDR deflects fewer packets.

Performance of the PQDR algorithm, the Q-NDD algorithm, and RLDRS in terms of average number of hops for the 64-wavelength scenario is shown in Fig. 15.8. When generating the reward signals that are used to update Q-values, the PQDR algorithm and RLDRS consider the number of hops to the destination. Hence, they perform better than Q-NDD in terms of the average number of hops traveled by bursts. In addition to Q-values, PQDR utilizes other variables to generate deflection decisions, which may result in selection of longer paths. Therefore, bursts may travel on average through additional hops.

Performance of the PQDR algorithm, the Q-NDD algorithm, and RLDRS in terms of average end-to-end delay for the 64-wavelength scenario is shown in Fig. 15.9. Bursts experience smaller end-to-end delays in case of RLDRS because the scheme maintains only one table (Q-table) and, therefore, the link selection process and table updates are faster than in the case of PQDR.

15.5.2 Waxman and Barabási-Albert Topologies

We consider a buffer-less OBS architecture for data transmission where nodes are connected using bi-directional 1 Gbps single wavelength fiber links. The nodes are randomly placed and each node is connected to three other nodes. Subsets of nodes

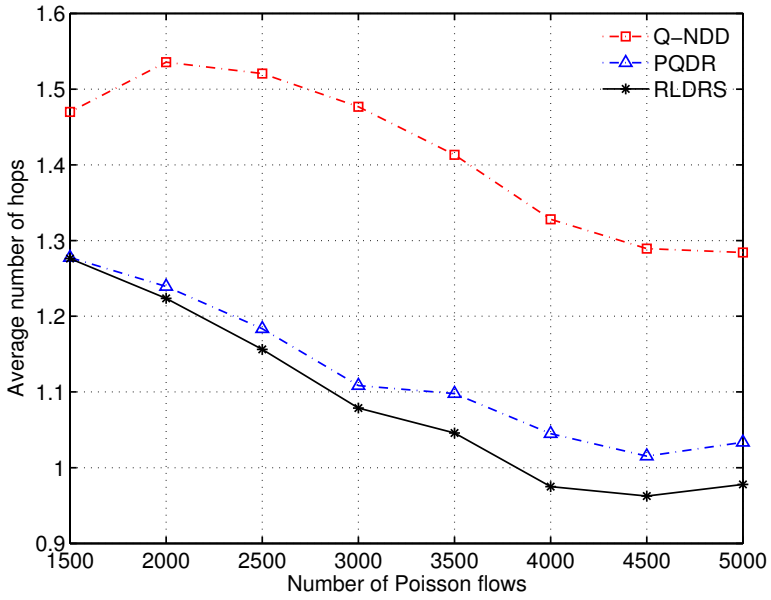

Fig. 15.8 Average number of hops traveled by bursts as a function of number of flows in the NSF network scenario with 64 wavelengths. In the case of RLDRS, the bursts travel the least number of hops.

are randomly selected as sources and destinations of Poisson traffic flows. Multiple Poisson flows with a data rate of 0.5 Gbps are transmitted across the network. Each Poisson flow is 50 bursts long with each burst containing 12.5 kB of payload. Each simulation scenario is repeated two times with various random assignments of nodes as sources and destinations. Simulation results are averaged over two simulation runs. For each scenario, the network load is maintained at 20%. Hence, scenarios with 10, 20, 50, 100, 200, 500, and 1,000 nodes have 24, 48, 120, 240, 480, 1,200, and 2,400 Poisson traffic flows, respectively. Simulations were performed on a Dell Optiplex-790 with 16 GB memory and the Intel Core i7 2600 processor. Simulation scenarios are shown in Table 15.1.

15.5.3 Burst-Loss Probability

Performance of Q-NDD, PQDR, and RLDRS in terms of burst-loss probability as a function of number of nodes for Waxman and Barabási-Albert network topologies is shown in Fig. 15.10. Burst-loss probability has a logarithmic trend. It is slightly

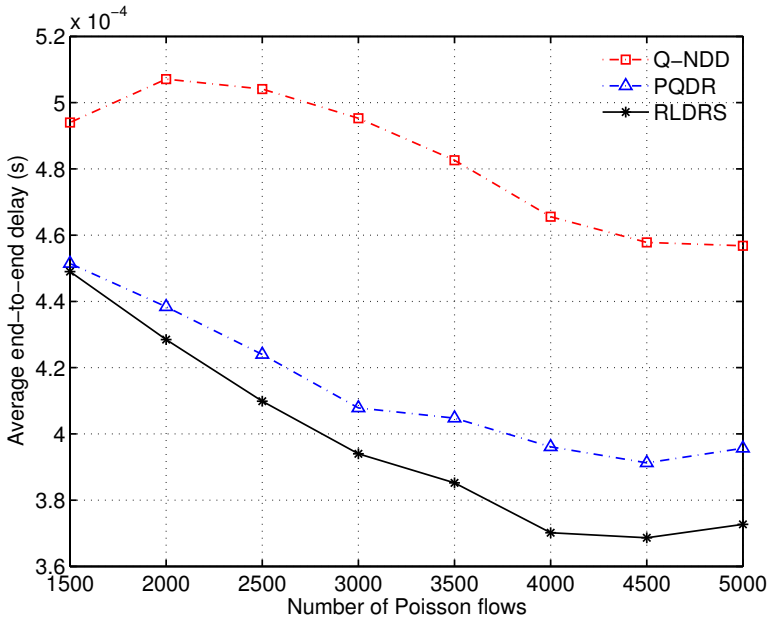

Fig. 15.9 Average end-to-end delay as a function of number of flows in the NSF network scenario with 64 wavelengths. In the case of RLDRS, the bursts experience the smallest end-to-end delay.

Table 15.1 Summary of Simulation Scenarios.

Topology generating algorithm	Deflection routing algorithm	Number of nodes	Number of links	Number of flows
Waxman Barabási-Albert	Q-NDD PQDR RLDRS	10	30	24
		20	60	48
		50	150	120
		100	300	240
		200	600	480
		500	1,500	1,200
		1,000	3,000	2,400

higher in Barabási-Albert networks. Q-NDD scales better than PQDR and RLDRS as the size of the network grows.

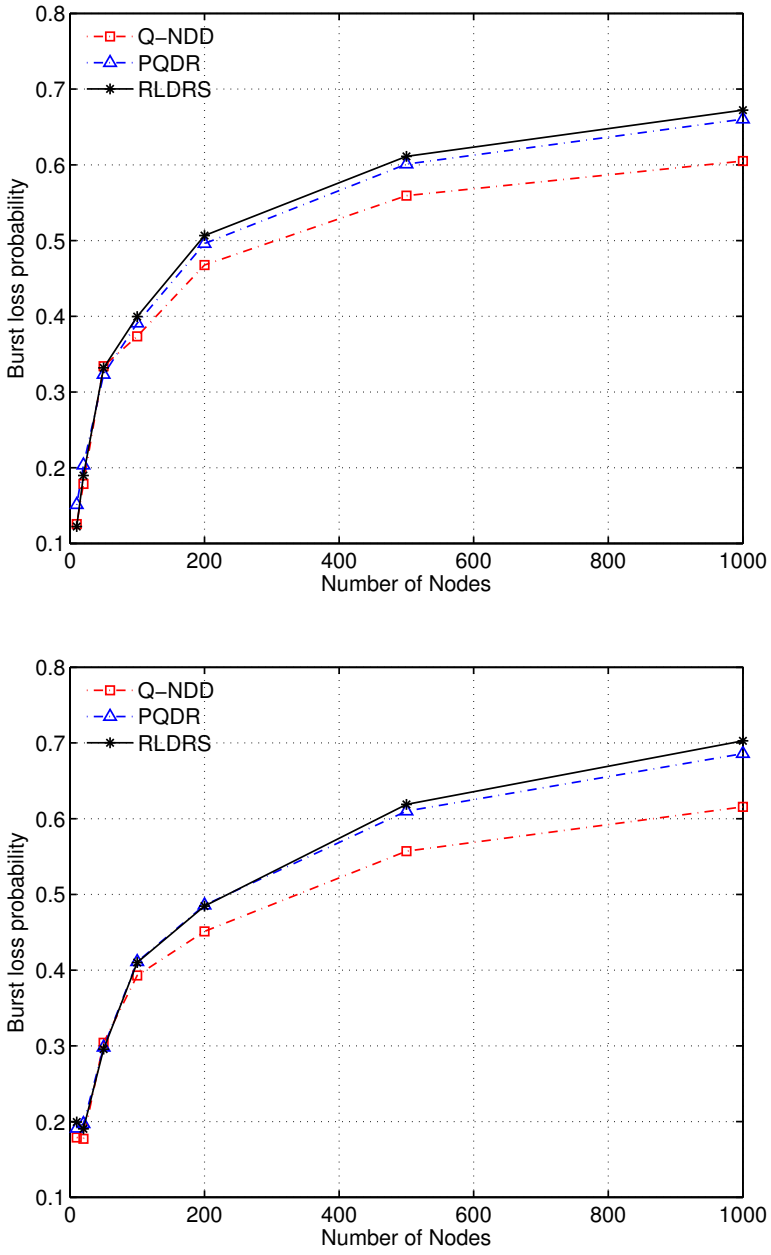

Fig. 15.10 Burst loss probability as a function of the number of nodes in Waxman (top) and Barabási-Albert (bottom) networks.

15.5.4 Number of Deflections

Although burst deflection reduces the burst-loss probability, it introduces excess traffic load to the network. Number of deflections as a function of number of nodes for Waxman and Barabási-Albert network topologies is shown in Fig. 15.11. Q-NDD deflects fewer number of bursts compared to PQDR and RLDRS. Comparison of Waxman and Barabási-Albert network topologies shows an insignificant variation in the number of deflections.

15.5.5 Average Number of Hops

Average number of hops traveled by bursts as a function of the number of network nodes for Waxman and Barabási-Albert topologies is shown in Fig. 15.12. In case of Q-NDD, bursts travel through additional hops compared to PQDR and RLDRS. When deflecting a burst, PQDR and RLDRS consider the number of hops to destination. Simulation results show that the underlying topology and nodes connectivity have an impact on the number of hops traveled by bursts. Bursts travel fewer hops in case of Barabási-Albert networks. Consider the case of 1,000 nodes shown in Fig. 15.12. In the case of Waxman topology shown in Fig. 15.12 (top), the Q-NDD algorithm causes bursts to travel 3.5 hops on average while in the case of Barabási-Albert topology shown in Fig. 15.12 (bottom), number of traveled hops is only 2.75.

15.6 Conclusion

In this Chapter, we compared performance of the Q-learning-based Node Degree Dependent (Q-NDD) deflection routing algorithm, the Predictive Q-learning-based Deflection Routing (PQDR) algorithm, and the Reinforcement Learning Based Deflection Routing Scheme (RLDRS). Simulations were performed using complex network topologies that were generated by Waxman and Barabási-Albert algorithms.

The Q-NDD burst-loss probability is smaller and bursts are deflected less frequently than PQDR and RLDRS. However, bursts travel through additional hops and thus experience longer end-to-end delays. Therefore, smaller burst-loss probability and smaller number of deflections come at the cost of selecting longer paths, which are less likely to be congested. PQDR and RLDRS consider the number of hops to destination when deflecting bursts. This, in turn, causes the bursts to travel through shorter paths. However, the probability of congestion along shorter paths is usually higher because the majority of the routing protocols tend to route data through such paths. As a result, burst-loss probability and probability of defecting bursts is higher along the paths that PQDR and RLDRS select for deflection.

While the simulation results indicate that number of deflections does not significantly depend on the underlying topology, the bursts travel through fewer hops in

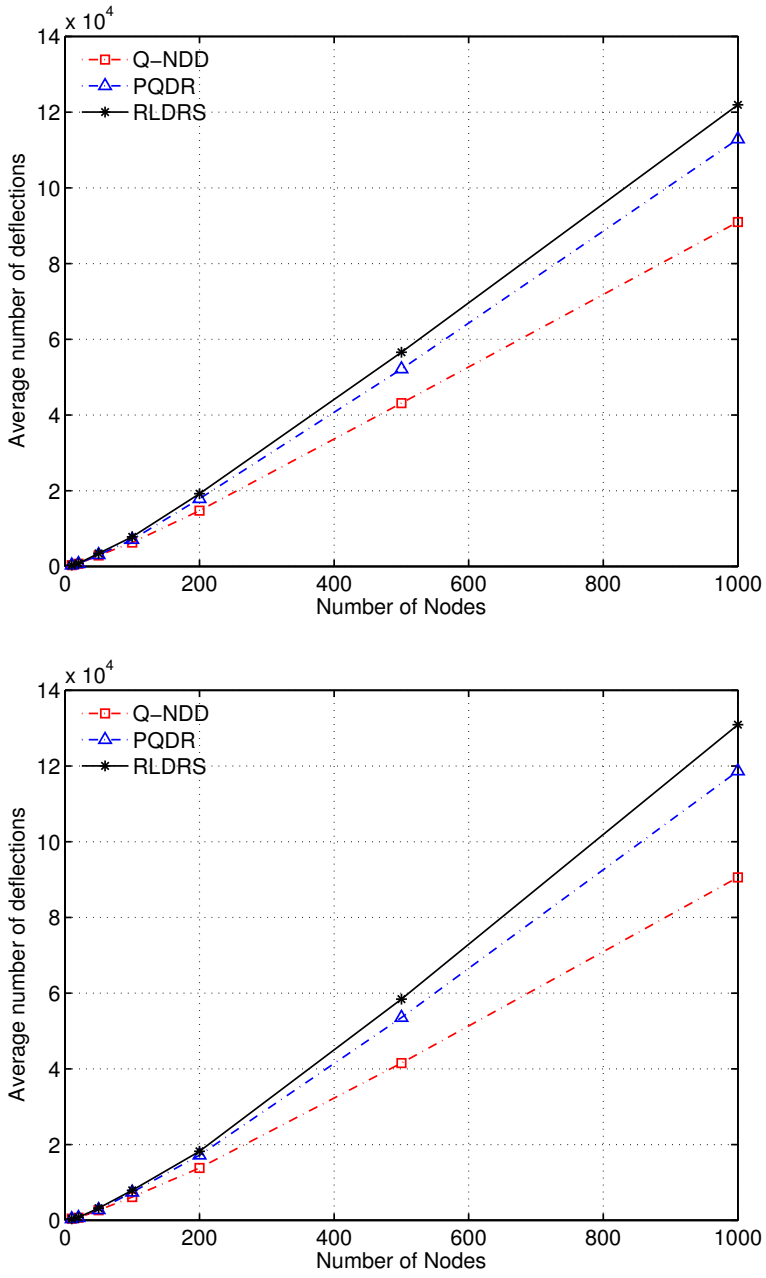

Fig. 15.11 Number of deflections as a function of the number of nodes in Waxman (top) and Barabási-Albert (bottom) networks.

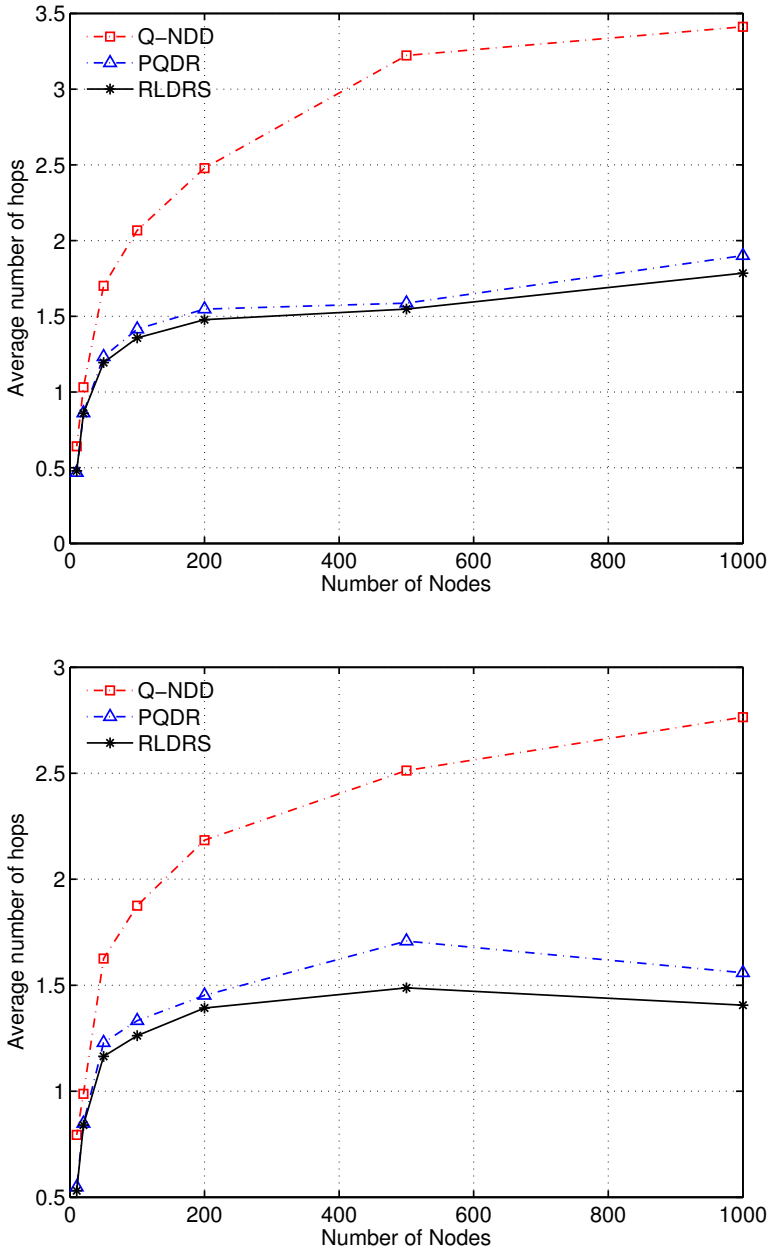

Fig. 15.12 Average number of hops traveled by bursts as a function of the number of nodes in Waxman (top) and Barabási-Albert (bottom) networks.

Barabási-Albert networks. Q-NDD outperforms PQDR and RLDRS by exhibiting smaller burst-loss probability and smaller number of deflections. However, in the case of Q-NDD, the bursts travel through additional hops. The improved performance of Q-NDD becomes more evident as the size of the network increases.

References

1. A. I. Abd El-Rahman, S. I. Rabia, and H. M. H. Shalaby, "MAC-layer performance enhancement using control packet buffering in optical burst-switched networks," *J. Lightwave Technol.*, vol. 30, no. 11, pp. 1578–1586, June 2012.
2. A. S. Acampora and S. I. A. Shah, "Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing" in *Proc. IEEE INFOCOM*, Bal Harbour, FL, USA, Apr. 1991, vol. 1, pp. 10–19.
3. R. G. Addie, T. D. Neame. and M. Zukerman, "Perfomance evaluation of a queue fed by a Poisson Pareto burst process," *Comput. Netw.*, vol. 40, no. 3, pp. 377–397, Oct. 2002.
4. R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47-97, Jan. 2002.
5. (2014, Dec.) Autonomous System Numbers [Online]. Available: http://www.iana.org/assignments/as-numbers/.
6. 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.
7. A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509-512, Oct. 1999.
8. P. Baran, On distributed communications networks, *IEEE Trans. Commun. Syst.*, vol. CS–12, no. 1, pp. 1–9, Mar. 1964.
9. B. G. Bathula and V. M. Vokkarane, "QoS-based manycasting over optical burst-switched (OBS) networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 271–283, Feb. 2010.
10. A. Belbekkouche, A. Hafid, M. Tagmouti, and M. Gendreau, "Topology-aware wavelength partitioning for DWDM OBS networks: a novel approach for absolute QoS provisioning," *Computer Networks*, vol. 54, no. 18, pp. 3264–3279, Dec. 2010.
11. 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.
12. F. Borgonovo, "Deflection routing," in *Routing in Communications Networks*. NJ, USA: Prentice-Hall, 1995, pp. 263–306.
13. F. Borgonovo, L. Fratta, and J. Bannister, "Unslotted deflection routing in all-optical networks," in *Proc. IEEE GLOBECOM*, Houston, TX, USA, Dec. 1993, pp. 119–125.
14. J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," in *Advances in Neural Inform. Process. Syst.*, J. Jack, D. Cowan, G. Tesauro, and J. Alspector, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1994, vol. 6, pp. 671–678.
15. S. Bregni, A. Caruso, and A. Pattavina, "Buffering-deflection tradeoffs in optical burst switching," *Photon. Netw. Commun.*, vol. 20, no. 2, pp. 193–200, Aug. 2010.
16. (2014, Dec.) BRITE [Online]. Available: http://www.cs.bu.edu/brite.
17. K. L. Calvert, M. B. Dora, and E. W. Zegura, "Modeling Internet topology," *IEEE Commun. Mag.*, vol. 35, no. 6, pp. 160–163, June 1997.
18. Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The origin of power laws in Internet topologies revisited," in *Proc. INFOCOM*, New York, NY, USA, Apr. 2002, pp. 608–617.

19. 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.
20. 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.
21. 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.*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. Cambridge, MA, USA: The MIT Press, 1996, vol. 8, pp. 945–951.
22. P. Erdös and A. Reńyi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, pp. 17–61, 1960.
23. M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 251–262, Aug. 1999.
24. 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.
25. (2014, Dec.) GNU General Public License [Online]. Available: http://www.gnu.org/copyleft/gpl.html.
26. A. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, no. 6, pp. 1070–1081, June 1992.
27. S. Haeri and Lj. Trajković, "Intelligent deflection routing in buffer-less networks," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 316–327, Feb. 2015.
28. S. Haeri and Lj. Trajković, "Deflection routing in complex networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Melbourne, Australia, June 2014, pp. 2217–2220.
29. S. Haeri, M. Arianezhad, and Lj. Trajković, "A predictive Q-learning-based algorithm for deflection routing in buffer-less networks," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Manchester, UK, Oct. 2013, pp. 764–769.
30. S. Haeri, W. W-K. Thong, G. Chen, and Lj. Trajković, "A reinforcement learning-based algorithm for deflection routing in optical burst-switched networks," in *Proc. IEEE Int. Conf. Inf. Reuse and Integration*, San Francisco, CA, USA, Aug. 2013, pp. 474–481.
31. S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proc. IEEE INFOCOM*, Stanford, CA, USA, Mar. 2003, vol. 1, pp. 406–416.
32. M. Izal and J. Aracil, "On the influence of self-similarity on optical burst switching traffic," in *Proc. IEEE GLOBECOM*, Taipei, Taiwan, Nov. 2002, vol. 3, pp. 2308–2312.
33. 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.
34. 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.
35. 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, NB, Canada, 1971.
36. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.
37. M. Levesque, H. Elbiaze, and W. Aly, "Adaptive threshold-based decision for efficient hybrid deflection and retransmission scheme in OBS networks," in *Proc. 13th Int. Conf. Optical Network Design and Modeling*, Braunschweig, Germany, Feb. 2009, pp. 55–60.
38. S. Li, M. Wang, E. W. M. Wong, V. Abramov, and M. Zukerman, "Bounds of the overflow priority classification for blocking probability approximation in OBS networks," *J. Opt. Commun. Netw.*, vol. 5, no. 4, pp. 378–393, Apr. 2013.
39. H.-L. Liu, B. Zhang, and S.-L. Shi, "A novel contention resolution scheme of hybrid shared wavelength conversion for optical packet switching," *J. Lightwave Technol.*, vol. 30, no. 2, pp. 222–228, Jan. 2012.

40. J. Lü, G. Chen, M. Ogorzalek, and Lj. Trajković, "Theories and applications of complex networks: advances and challenges," in *Proc. IEEE Int. Symp. Circuits and Systems*, Beijing, China, May 2013, pp. 2291–2294.
41. N. F. Maxemchuk, "Comparison of deflection and store and forward techniuques in the Manhattan street and shuffle exchange networks," in *Proc. IEEE INFOCOM*, Ottawa, ON, Canada, Apr. 1989, vol. 3, pp. 800–809.
42. Merit/NSFNET Information Services, "The Technology Timetable," *Link Letter*, vol. 7, no. 1, pp. 8–11, July 1994.
43. X. Mountrouidou and H. Perros, "On the departure process of the burst aggregation algorithms in optical burst switching," *J. Comput. Netw.*, vol. 53, no. 3, pp. 247–264, Feb. 2009.
44. M. Najiminaini, L. Subedi, and Lj. Trajković, "Analysis of Internet topologies: a historical view," in *Proc. IEEE Int. Symp. Circuits and Systems*, Taipei, Taiwan, May 2009, pp. 1697–1700.
45. 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.
46. V. Paxson and S. Floyd, "Wide-area traffic: The faliure of Poisson modeling," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, June 1995.
47. 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.
48. H. G. Perros, *Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks*. Chichester, UK: John Wiley & Sons, 2005.
49. 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.
50. 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.
51. G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power-laws and the AS-level Internet topology," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 514–524, Aug. 2003
52. L. Subedi and Lj. Trajković, "Spectral analysis of Internet topology graphs," in *Proc. IEEE Int. Symp. Circuits and Systems*, Paris, France, June 2010, pp. 1803–1806.
53. (2014, Dec.) The Google Earth. [Online]. Available: http://www.google.com/earth/index.html/.
54. (2014, Dec.) A special report: a brief history of NSF and the Internet [Online]. Available: http://www.nsf.gov/news/special_reports/cyber/internet.jsp.
55. (2014, Dec.) The ns-3 network simulator [Online]. Available: http://www.nsnam.org/.
56. W. W.-K. Thong and G. Chen, "Jittering performance of random deflection routing in packet networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 616–624, Mar. 2013.
57. W. W.-K. Thong, G. Chen, and Lj. Trajković, "RED-f routing protocol for complex networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Seoul, Korea, May 2012, pp. 1644–1647.
58. Lj. Trajković, "Analysis of Internet topologies," *IEEE Circuits and Systems Mag.*, vol. 10, no. 3, pp. 48–54, Third Quarter 2010.
59. W.-X. Wang, C.-Y. Yin, G. Yan, and B.-H. Wang, "Integrating local static and dynamic information for routing traffic," *Phys. Rev. E*, vol. 74, p. 016101, July 2006.
60. X. Wang, X. Jiang, and A. Pattavina, "Efficient designs of optical LIFO buffer with switches and fiber delay lines," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3430–3439, Dec. 2011.
61. C. J. C. H. Watkins and P. Dayan, "Technical note, Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.
62. D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440–442, June 1998.
63. B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

64. E. W. M. Wong, J. Baliga, M. Zukerman, A. Zalesky, and G. Raskutti, "A new method for blocking probability wvaluation in OBS/OPS networks with deflection routing," *J. Lightwave Technol.*, vol. 27, no. 23, pp. 5335–5347 Dec. 2009.
65. G. Wu, W. Dai, X. Li, and J. Chen, "A maximum-efficiency-first multi-path route selection strategy for optical burst switching networks," *Optik—Int. J. Light and Electron Optics*, vol. 125 no. 10, pp. 2229-2233, May 2014.
66. 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.
67. 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.
68. X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *Proc. ACM SIGCOMM*, New York, NY, USA, Oct. 2006, pp. 159–170.
69. M. Yoo, C. Qiao, and S. Dixit, "Comparative study of contention resolution policies in optical burst-switched WDM networks," in *Proc. SPIE*, Boston, MA, USA, Oct. 2000, vol. 4213, pp. 124–135.
70. 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.
71. 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.
72. A. Zalesky, H. Vu, Z. Rosberg, E. Wong, and M. Zukerman, "OBS contention resolution performance," *Perform. Eval.*, vol. 64, no. 4, pp. 357–373, May 2007.
73. 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.
74. E. W. Zegura, K. L. Calvert, and M. J. Donahoo "A quantitative comparison of graph-based models for Internet topology," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 770–783, Dec. 1997.
75. M. Zukerman, T. D. Neame, and R. G. Addie, "Internet traffic modeling and future technology implications," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar. 2003, vol. 1, pp. 587–596.