

Received 4 May 2023, accepted 23 May 2023, date of publication 2 June 2023, date of current version 14 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3282365

RESEARCH ARTICLE

An Online Learning Approach to Shortest Path and Backpressure Routing in Wireless Networks

OMER AMAR, ILANA SARFATI, AND KOBI COHEN, (Senior Member, IEEE)

School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel

Corresponding author: Kobi Cohen (yakovsec@bgu.ac.il)


This work was supported in part by the Israel Science Foundation under Grant 2640/20, and in part by the Israel Ministry of Economy (Magnet).

ABSTRACT We consider the problem of adaptive routing in wireless communication networks. The problem is investigated in the online learning context, where the link states are assumed to be random variables drawn from unknown distributions, independent and identically distributed across links and time. This setting has attracted a growing interest in recent years in cognitive radio networks and adaptive communication systems. In these networks, the devices (or nodes) are cognitive in the sense of learning the link states and updating the transmission parameters, to allow efficient utilization of the network resources. This model contrasts sharply with the vast literature on routing algorithms that assumed complete knowledge about the link state means. The objective is to develop an algorithm that learns online optimal paths to transmit the data so as to maximize the network throughput with low path cost over the network. This study makes significant contributions in terms of algorithm design, theoretical analysis with performance guarantees, and extensive numerical analysis to evaluate the algorithm's performance. To achieve this goal, we present a novel algorithm, dubbed Online Learning for Shortest-path and Backpressure (OLSB). OLSB optimizes an objective function that balances between the cost and the load over paths. Since the path states are unknown, the design is based on a novel learning strategy that allows efficient adaptive path selections in OLSB. We evaluate the theoretical performance of OLSB by computing the regret, defined as the loss between OLSB and a genie which holds full information on the link state means. We analyze the performance of OLSB rigorously, and show that it achieves a logarithmic regret with time. Finally, extensive simulations are presented to evaluate the performance of OLSB numerically as well. The numerical results support the theoretical findings and demonstrate the high efficiency of OLSB.

INDEX TERMS Adaptive routing, online learning, cognitive radio networks, shortest path, backpressure.

I. INTRODUCTION

The demand for wireless communication services has increased along with the rapid development of communication network technologies. However, spectrum scarcity remains one of the major limitations in supporting this growing demand. Therefore, being able to develop adaptive routing algorithms that utilize judiciously the spectral resources and schedule data transmissions efficiently is a main challenge in modern communication networks. Traditional algorithms assumed complete knowledge about the link state means when scheduling and transmitting user data

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du .

over selected paths. However, in a real environment, link states vary randomly, and the distributions are often unknown. Thus, these traditional algorithms have become inefficient and suffer from performance degradation. This is particularly relevant in the era of adaptive communications and dynamic cognitive networking, where the user loads are dynamic and heterogeneous, and need to be balanced in a challenging unknown environment. Therefore, designing online learning algorithms for adaptive routing in an unknown environment has attracted a growing interest in recent years in the study of dynamic networks, distributed learning, adaptive communications and cognitive radio networks [2], [3], [4], [5], [6], [7], [8], [9], [10]. Specific applications include wireless sensor networks, Internet of Things (IoT), and software-defined

networking (SDN), where learning-based routing, particularly the OLSB algorithm developed in this paper, can be used to optimize data transmission between different devices and sensors, reducing latency and improving queue stability and network reliability.

In this paper, we consider a time-slotted wireless network, where the state of each link is modeled by a random process drawn from an unknown distribution, independent and identically distributed (i.i.d.) across time and other links, as in [2], [3], [11], [12], [13], [14], and [15]. The random state of the link is typically used to represent a channel effect of the link quality caused by an external random process. Examples for such models have been studied in hierarchical cognitive radio networks, where primary users (licensed) are modeled by external random processes, or a fading channel effect in the open sharing wireless communication model [16]. As commonly done in routing and scheduling studies, the state of a path (or its cost) at each time slot t is defined as the accumulated states of all links on the path at time slot t (e.g., when summing over path rate measures, delay effects, small packet-drop probability effects [2], [15]). The source node does not know the random state of a path before transmission. Only after the transmitted packet reaches its destination, the source node observes the random state of the selected path (e.g. by ACK signal information [17]). We denote a flow $f_{(s,d)}$ in the network, as a traffic data from source node s to destination node d , which generates a random number $A_{s,d}(t)$ of packets at each time slot t with arrival rate $\lambda_{s,d}$. We denote the set of all flows in the network by \mathcal{F} . We aim at developing an online learning-based routing algorithm for flow transmissions in the network under unknown network state. It is desired to achieve efficient learning of the network state to maximize the network throughput, while preserving low sum path costs over scheduled flows in the network. An explicit formulation of the stochastic optimization problem is given in Section II.

A. ROUTING DATA FLOWS WITH COMPLETE KNOWLEDGE ON LINK STATES

Solving the shortest path problem in data networks is one of the most popular approaches in routing algorithms. When the link states are assumed to be completely known, the shortest path for each source-destination pair (s, d) is defined by the path with the minimal accumulated cost over links in a path among all possible paths for data transmissions from source node s to destination node d . For instance, the popular Open Shortest Path First (OSPF) routing protocol transmits data flows in the network over shortest paths, where the computations of shortest path are implemented by the well-known Dijkstra algorithm [17], [18]. The drawback of using shortest paths to route data in communication networks is its tendency to increase congestion on short paths, and consequently decrease performance in highly-loaded networks. Backpressure routing is an alternative approach used to overcome this issue. Using backpressure routing, the data is transmitted through paths with low congestion.

Mathematically, the algorithm maximizes the differential queue backlog between nodes when scheduling data transmissions. An important theoretical property of backpressure routing is its ability to maximize the network throughput [17]. Therefore, it has attracted a growing attention in designing routing algorithms [17], [19], [20], [21], [22], [23]. The disadvantage of backpressure routing, however, is its inefficiency when the network congestion is low. The reason is that backpressure routing tends to send packets through long paths to reduce the congestion over short paths. These pros and cons of backpressure and shortest path routing mentioned above, led to a hybrid approach that takes advantage of the strengths of both algorithms, shortest path and backpressure routing. The basic idea is to use shorter paths for data transmissions when the network congestion decreases (since using backpressure that causes large delays should be limited in these network states), and longer paths when the network congestion increases (since using shortest path routing that increases congestion should be limited in these network states) (see [24] and subsequent studies). A joint optimization of backpressure and shortest path routing was introduced in [24] that maximizes the network throughput as in backpressure, but with much smaller path costs.

B. LEARNING TO ROUTE DATA FLOWS UNDER UNKNOWN LINK STATES

In real-world communication networks, the link states are random with unknown distributions. Consequently, the mean values of the link states are unknown, and adaptive routing algorithms should be able to learn these values online. Recent studies on cognitive radio networks and adaptive communications have focused on addressing this challenge. In these studies, various transmission scheduling algorithms have been suggested that learn the mean values of the network states during time to update the transmission parameters and improve the resource allocation in the network. Transmission scheduling algorithms have been studied from different approaches and perspectives. In [25], [26], [27], [28], [29], and [30], the multi-user dynamics was analyzed based on game-theoretic optimization. In [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], and [41] the long-term reward optimization of users in the network was analyzed based on multi-armed bandit learning framework. The learning strategy has been based on various methods such as reinforcement learning and Upper Confidence Bound (UCB)-based algorithms [10], [42], [43], [44], as well as deep reinforcement learning that uses deep neural network in the reinforcement learning optimization [45], [46], [47]. These online learning methods focused on single-hop transmissions. where in this paper we perform the learning on multi-hop link states to allow efficient routing. Existing learning methods for adaptive routing in ad-hoc wireless networks were presented in [2], [3], [6], [13], [15], [48], and [49]. In [6] the focus was on developing energy-efficient routing, which is different than the focus in this paper. In [49] the focus was on reinforcement learning to scheduling routes based on the congestion

level, but without considering noisy observations of the random link states, and without theoretical guarantees on convergence. The adaptive random problem with noisy observations of the random link states was studied in [2], [3], [13], [15], and [48], where the focus was on solving the stochastic shortest path routing. The challenge in these papers is to develop an online learning algorithm that trades-off efficiently between exploring paths to learn the network state, at the price of selecting sub-optimal paths during this exploration phase, and exploiting the information the algorithm has gained to solve the shortest paths and schedule data transmissions through these paths. One approach to schedule data transmissions is to make decisions for end-to-end paths, and route packets in these selected paths. This approach was adopted in [2], [3], [13], and [48]. In [15], a different approach was used in order to make hop-by-hop decisions. This approach was shown to achieve better performance, particularly in adaptive communication systems, since it allows to adjust dynamically the inference outcome and consequently the selected routes. The algorithm that we develop in this paper allows hop-by-hop decisions as well. Those algorithms ([2], [3], [13], [15], [48]) were shown to achieve efficient learning in terms of converging to the shortest path solution. However, they all suffer from poor performance in terms of load balancing since they tend to increase the congestion over short paths. This negative effect becomes especially pronounced when the network load increases, as explained in Subsection I-A.

There exist other path finding algorithms such as metaheuristic methods like ACO [50], PSO [51], and AI approaches such as A*, RRT, BUG2 [52]. ACO and PSO are metaheuristic algorithms inspired by the behavior of social insects and bird flocking, respectively. They have been used in networking applications to solve various optimization problems, including routing tasks. However, both algorithms can suffer from slow convergence and may not be suitable for real-time routing decisions. Furthermore, our method described in this paper allows us to prove strong theoretical properties that have not been demonstrated using ACO and PSO, as we will explain later. AI approaches such as A*, RRT, BUG2 are all routing algorithms used mainly in robotics, autonomous vehicles, and other applications where path planning is required. A* is used to find the shortest path between two points in a graph. RRT is designed to quickly explore the configuration space and find a feasible path between the start and goal points. BUG2 is designed for robot navigation and uses a combination of gradient descent and heuristic methods to navigate to the goal while avoiding obstacles. These algorithms are not commonly used in packet routing in communication networks, and typically used for path planning and navigation in robotics and autonomous systems. The approach in this paper, however, is fundamentally different. We adopt the optimization problem in [24] that was shown to achieve throughput optimal performance. This paper is the first attempt to solve [24] in the online learning context, where the link states are unknown, and need to be inferred online. This problem was not solved by other AI approaches

and metaheuristic algorithms as specified above. Note that it is very difficult to estimate the theoretical speed of convergence of those algorithms. By contrast, here we directly solve the stochastic optimization problem by the proposed OLSB algorithm, based on a rigorous UCB analysis. This allows us to achieve strong theoretical convergence guarantees of OLSB, which cannot be guaranteed in general using metaheuristic algorithms, as discussed in the next subsection. We refer the interested readers to [53] and [54] that offer excellent recent overviews on routing algorithms.

C. CONTRIBUTIONS

The main contribution of this paper is the development and analysis (theoretically and numerically) of a novel online learning algorithm for adaptive routing under unknown link state means, dubbed OLSB algorithm. This is the first paper that solves the routing optimization problem introduced in [24] in the online learning context, with rigorous theoretical analysis and performance guarantees. The main contributions are described below:

1) A NOVEL ONLINE LEARNING ALGORITHM FOR ROUTING UNDER UNKNOWN LINK STATES

We tackle the problem of adaptive routing in the online learning context, where the link states are unknown, and need to be inferred online. The objective is to develop an online learning algorithm that schedules and routes data transmissions over the network such that the network throughput is maximized, but keeps the path cost small. Mathematically, we adopt the routing optimization problem introduced in [24] as described in Subsection I-A. Existing solutions and algorithms to the deterministic optimization problem in [24] (and its variations) were developed under the assumption that the path states in the network are completely known, as discussed earlier in Subsection I-A. However, solving the problem in the online learning context, without relying on this assumption remained an open research problem and represents a significant challenge addressed by this work. This is the first paper that solves this problem. In this paper, we are thus facing a stochastic optimization problem associated with an exploration versus exploitation dilemma. On the one hand, the algorithm should explore sub-optimal paths to infer the network state, since link states are random variables following unknown distributions. On the other hand, the algorithm should exploit the information gathered so far to route packets in optimal paths, with the goal of converging to the deterministic optimization in [24]. We describe the performance measure of the convergence rate rigorously in Subsection I-C2.

To achieve the objective, we develop a novel online learning algorithm for adaptive routing under unknown link state means, called Online Learning for Shortest path and Backpressure (OLSB) algorithm. By implementing OLSB, the source node computes a desired upper bound on the cost of all possible path selections for each new data flow that arrived

for transmission. This computation is done by solving a stochastic optimization that trades-off between the estimated state of a path and its load, as well as the learning efficiency of the unknown random states. In contrast to existing online learning algorithms for adaptive routing that converge to a single and fixed path which is optimal in terms of minimizing the expected cost (see e.g., [2], [15] and references therein), in this paper the optimal selection of a path is time-varying since the backpressure scheduling term in the optimization problem depends on the time-varying queue dynamics. As a result, the learning algorithm is fundamentally different in both the design and analysis. Specifically, we develop a novel UCB-type rule to select paths for data flow transmissions, named Queue UCB (QUCB), used in OLSB. The QUCB rule maps the dynamic queue state and the empirical mean of the path state into a path selection index, that dictates a cost limit accordingly. OLSB uses the QUCB rule to determine the cost limit for packet transmissions, and backpressures packets through paths that meet the QUCB's cost conditions. Intuitively, when the load is low, QUCB assigns tight constraint to prioritize transmissions via short paths. As the load increases, the cost constraint is relaxed to allow transmissions via longer back-pressured paths. A detailed description of the QUCB rule and the OLSB algorithm is given in Section III. It is worth noting that the design of OLSB is competitive with existing algorithms in terms of computational complexity as well, as detailed in Subsection III-D.

2) THEORETICAL ANALYSIS AND PERFORMANCE GUARANTEES

We provide rigorous theoretical analysis to evaluate the performance of OLSB. To analyze the performance theoretically, our benchmark for performance is a genie-aided algorithm that has complete knowledge on the link state means, and consequently solves the deterministic optimization problem in [24], which was shown to be throughput optimal [24]. We define the regret as the performance loss of OLSB (that learns online to route packets under unknown link state means) compared to the genie-aided algorithm. As a result, the regret evaluates how fast OLSB learns the side information and approaches genie's performance. We show analytically that the regret scales logarithmically with time, which indicates that OLSB converges to genie's performance with the best known rate. The theoretical analyses are described in detail in Section IV.

3) NUMERICAL ANALYSIS

Finally, we present extensive simulation results to demonstrate the efficiency of OLSB. We simulated the identical network of 64 nodes and 119 links as in [24]. An illustration of the simulated network is shown in Fig. 1. The first part of the simulations are used to validate the theoretical analysis of the regret, and indeed the simulation results support the theoretical findings for all parameter settings. In the second part, we present an algorithm comparison to demonstrate the efficiency of the OLSB algorithm, showing that

OLSB is superior to existing methods. The results were tested under various network settings of lightly-loaded, moderately-loaded, and highly-loaded networks, and performance measures of regret, queue length, end-to-end delay, and supportable rates. The numerical analyses are described in detail in Section V.

D. ORGANIZATION

The rest of this paper is organized as follows: In Section II, we present the system model and formulate the problem. In Section III, we present the proposed Online Learning for Shortest path and Backpressure (OLSB) algorithm to achieve the objective. In Section IV, we analyze the performance of OLSB rigorously theoretically, and show that it achieves a logarithmic regret with time. Detailed proofs are given in the Appendix. In Section V, we present simulation results to validate the theoretical findings, and demonstrate the efficiency of OLSB. Section VI concludes the paper.

II. DESCRIPTION OF THE SYSTEM MODEL AND PROBLEM STATEMENT

We start by describing the system model in Subsection II-A, and then formulate the problem in Subsection II-B.

A. DESCRIPTION OF THE SYSTEM MODEL

We consider a time-slotted communication network, and we denote the time slot index by t . Let V denote the set of nodes (i.e., users) in the network, and E the set of edges (i.e., communication links). The communication network is thus modeled by a directed graph $G = (V, E)$. A directed communication link between transmitter v and receiver u in the communication network is modeled by a link $(v, u) \in E$ from node $v \in V$ to node $u \in V$ in the directed graph (i.e., v, u are neighbors). Every node in the network maintains a queue for packet transmissions through the links using a certain MAC protocol (which we will describe later in detail). We consider a general model of communication network where multiple data flows need to be routed through the network simultaneously and to share the network resources. Let $f_{(s,d)}$ denote a data flow from source node $s \in V$ to destination node $d \in V$, with arrival rate $\lambda_{(s,d)}$. Let \mathcal{F} denote the set of all data flows in the communication network. To model the transmission cost over a link, at each time t , each link $e \in E$ in the directed graph is associated with a weight $w_e(t)$. The weight $w_e(t)$ is a random process drawn from an unknown distribution, with support normalized to $[0, 1]$, i.i.d. across time slots and other links, as in [2], [3], [11], [12], [13], [14], and [15]. Next, we define by $\mathcal{P}_{(v,d)}$ the set of all loop-free paths in the directed graph G from node v to destination node d . We often indicate a loop-free path $p \in \mathcal{P}_{(v,d)}$ by a sequence of links from v to d , e.g., $p = ((v, u), (u, x), \dots, (y, d))$, or either by a sequence of nodes from v to d , e.g., $p = (v, u, x, \dots, y, d)$. Let $C_p(t)$ be the state of path p (or path cost) at time t . The path cost $C_p(t)$ is defined by the normalized sum of all link weights on path p : $C_p(t) = \frac{1}{|V|} \sum_{e \in p} w_e(t)$. Note that $0 \leq C_p(t) \leq 1$.

B. PROBLEM STATEMENT

As introduced in [24] (with complete knowledge of all path state means), the objective is to maximize the network throughput, while obtaining low sum path costs over transmitted flows in the network. Specifically, when the path state means, $\mu_p = \sum_{e \in p} E(w_e(t))$, $p \in \mathcal{P}_{(s,d)}$, are completely known, the throughput-optimal solution is to solve the following deterministic optimization problem at each time t [24]:

$$\arg \min_{p \in \mathcal{B}_{(s,d)}} \left(K\mu_p + Q_{(s,d,m(\mu_p))}(t) \right), \quad (1)$$

where $\mathcal{B}_{(s,d)}$ is a barycentric spanner on the path set $\mathcal{P}_{(s,d)}$ (see Section III-A for details), and $Q_{(s,d,m(\mu_p))}(t)$ is the number of packets (i.e., queue state) in the $m(\mu_p)$ th queue of node s destined to node d by time t , where $m(\mu_p)$ is a mapping function from μ_p to a queue index stored by the node. The term K is a tuning parameter used to balance between short paths and backpressured paths. Intuitively, the solution tends to use short paths when the network congestion is light, and backpressured long paths when the network congestion increases.

Solutions to the deterministic optimization problem (1) and variations have been studied in recent years under complete knowledge of all path state means (see [24] and subsequent studies). However, solving the problem in the online learning context without assuming prior knowledge of path state means remained open. In this paper we address this problem. The objective of this paper is thus to develop an algorithm that converges (the performance measure is described later) to the solution of (1) in the online learning context under unknown path states. We are thus facing an online learning problem with the well-known exploration versus exploitation dilemma. On the one hand, the algorithm should explore all paths in order to infer their states. On the other hand, it should exploit the information gathered so far to route packets in the optimal paths (which vary over time). To evaluate the performance of online learning algorithms, it is common to define the regret, which is the loss of an algorithm as compared to the performance achieved by genie with side information on the system. Here, we wish to design an algorithm that minimizes the regret with respect to the optimal solution of (1). In Section III, we develop the OLSB algorithm to solve this problem. In Section IV, we analyze the performance of OLSB rigorously and show analytically that the regret scales logarithmically with time, which indicates that OLSB converges to genie's performance with the best known rate.

III. THE ONLINE LEARNING FOR SHORTEST PATH AND BACKPRESSURE (OLSB) ALGORITHM

We now develop the OLSB algorithm to achieve the objective. In the OLSB optimization, a cost constraint is computed for each newly arrived packet based on the dynamic system state. Then, a path for packet transmission is selected by OLSB among the set of permitted paths that meet the cost constraint. This allows to trade off between selecting paths with small costs (i.e., short paths) and selecting path with low

congestion level (i.e., backpressured paths). At the same time, the algorithm is required to efficiently learn the unknown system state (as described later) to converge to the optimal solution.

Algorithm parameters are defined as follows. Let C_0, C_1, \dots, C_M be $M + 1$ quantization thresholds, such that $0 = C_0 < C_1 < \dots < C_{M-1} < 1 < C_M$, used to quantize the path cost in the network (e.g., with uniform spacing). Every node $v \in V$ in the network maintains M queues for each destination node, determined by the quantization thresholds as described below.

Let $m(c) : [0, 1] \rightarrow \{0, \dots, M - 1\}$ be a mapping function from a cost to a quantized cost level, such that $m(c) = i$ iff $C_i \leq c < C_{i+1}$ ($0 \leq i \leq M - 1$). Consider a newly arrived packet that arrives at node v , destined to node d , which is associated with path cost constraint c , such that $C_i \leq c < C_{i+1}$ ($0 \leq i \leq M - 1$). Then, node v inserts the packet to one of its queues $0, 1, 2, \dots, m(c) = i$ (corresponding to the quantization thresholds C_0, C_1, \dots, C_i , respectively) destined to node d . The mechanism that determines the selected queue is done by solving a stochastic optimization defined by OLSB as detailed later. If node v inserts the arrived packet to the j th queue (where $0 \leq j \leq i$), then the algorithm updates the path constraint for the packet to C_j . The packets stored in queue $j = 1, \dots, i$ are transmitted to the destination by backpressure routing restricted to paths with cost lower than C_j . As a result, OLSB trades off between the congestion level by using backpressure routing and the overall path cost by using a subset of the paths (i.e., short paths) for transmission as we detail later. OLSB routes the packets stored in queue 0 (corresponding to path cost $C_0 = 0$) through the shortest path only. The queue state $Q_{(v,d,m)}(t)$ is defined by the number of packets stored in the m th queue of node v destined to node d by time t .

In the next subsections we describe in detail the three main phases implemented by the OLSB algorithm. The pseudocode of OLSB is given in Algorithm 1.

A. AN INITIALIZATION STEP

We initialize OLSB by a preprocessing step used to span the paths in the network, as commonly done in adaptive routing algorithms (see e.g., [2], [55] and subsequent studies). This step is done by exploiting dependencies between paths in the network to reduce the number of paths that the nodes need to learn by constructing a barycentric spanner [2], [55]. Consider node v and destination node d . OLSB constructs a barycentric spanner on the path set $\mathcal{P}_{(v,d)}$ to obtain a smaller barycentric path set $\mathcal{B}_{(v,d)}$ that spans $\mathcal{P}_{(v,d)}$.

Recall that the link states and path costs to destinations are random processes with unknown distributions, thus need to be inferred. Let $\bar{C}_p(t)$ be an estimate of the path cost mean for path p by time t . Using OLSB, every node in the network (say v) computes and holds $\bar{C}_p(t)$ for all $p \in \mathcal{B}_{(v,d)}$ for each destination node d . Let $T_p(t)$ be the number of times that path $p \in \mathcal{B}_{(v,d)}$ was selected for data transmission by time t .

We use this property to design an efficient learning of the path states in OLSB that allows convergence to the optimal solution in the best-known rate as described and analyzed rigorously later.

We kick off the algorithm by an exploration step, in which each destination node (say d) routes a single packet for each data flow $(s, d) \in \mathcal{F}$ through every path $p \in \mathcal{B}_{(s,d)}$. Denote path p by $p = (s, v_1, v_2, \dots, v_I, d) \in \mathcal{B}_{(s,d)}$, and a sub-path of p by $p_i = (v_i, v_{i+1}, \dots, v_I, d)$. The random cost $C_p(0)$ of path p is observed at the source node, and the random cost $C_{p_i}(0)$ of sub-path p_i is observed at node v_i (e.g., each link on the path adds its random cost to the message and consequently the cost of any sub-path is observed through the path). The estimate mean cost of sub-path p_i is initialized to $\bar{C}_{p_i}(0) = C_{p_i}(0)$ at node v_i for each node v_i on path p , and the estimate mean cost of path p is initialized by $\bar{C}_p(0) = C_p(0)$ at the source node s . We set $T_p(0) = 1, \forall p \in \mathcal{B}_{(s,d)}$.

B. SCHEDULING PACKETS AT THE SOURCE NODE

We now present the routine of OLSB. We refer to time t when describing the execution of the algorithm. The mechanism described in the previous subsection is used to estimate the network state during the routine of OLSB as well. Specifically, the learning mechanism uses ACK signals transmitted from the destination node to the source node whenever a packet (or a frame of packets) is routed through every path in the barycentric spanner.¹ The estimate mean cost $\bar{C}_p(t)$ of path p stored at the source node s , and $\bar{C}_{p_i}(t)$ of sub-path p_i stored at node $v_i \in p$ are updated by evaluating the empirical mean of each path $p \in \mathcal{B}_{(s,d)}$ for each flow $(s, d) \in \mathcal{F}$.

We next describe the stochastic optimization problem solved at the source node used to schedule a newly arrived packet. Consider flow $f_{(s,d)}$ from source node s to destination node d . Consider a newly arrived packet (or a frame of packets) of flow $f_{(s,d)}$ at source node s . In OLSB, the source node selects queue $Q_{(s,d,m)}$ ($1 \leq m \leq M$) among the M queues of flow $f_{(s,d)}$ it maintains, and stores the packet for transmission in the selected queue. The queue selection is based on the current queue states and the estimated path costs. Intuitively, it is desired to store packets in queues with low loads (to reduce the congestion in the network), as well as low cost constraint (i.e., permitting transmissions through short paths in terms of path cost).

Formally, to solve the scheduling problem under unknown link states considered here, we develop a novel UCB-type learning rule, called Queue UCB (QUCB). The Queue UCB (QUCB) rule balances between the dynamic queue states and the estimated path costs in the online scheduling decision making. Specifically, at time t , a newly arrived packet of flow $f_{(s,d)}$ is stored in the $m(\bar{C}_{p^*}(t))$ th queue, i.e., $Q_{(s,d,m(\bar{C}_{p^*}(t)))}$, where p^* solves the following QUCB's stochastic

¹Note that all paths in the network can be observed by a simple linear combination of paths in the barycentric spanner [2].

optimization problem:

$$p^* = \arg \min_{p \in \mathcal{B}_{(s,d)}} \left(K \bar{C}_p(t) + Q_{(s,d,m(\bar{C}_{p^*}(t)))}(t) - \sqrt{\frac{2 \ln t}{T_p(t)}} \right). \quad (2)$$

Here, K is a design parameter that balances between scheduling packets for transmission through short paths and through low-congested paths. In the case where the mean cost of paths (say μ_p of path p) are known, it was shown in [24] that solving the following deterministic optimization: $\arg \min_{p \in \mathcal{B}_{(s,d)}} \left(K \mu_p + Q_{(s,d,m(\mu_p))}(t) \right)$ (which was formulated in (1) in Section II) maximizes the network throughput, while obtaining low sum path costs over the flows in the network. Decreasing the path cost is done by increasing K , and consequently increasing the priority of scheduling packets in shorter paths. However, this comes at the price of increasing the network congestion and consequently the queuing delay (since packets are scheduled in queues with large backlogs). In Section IV we analyze OLSB rigorously, showing that the novel scheduler solved by QUCB's stochastic optimization converges to the optimal solution of the deterministic optimization problem in (1) (i.e., as in the case where the path state means are completely known) with a logarithmic regret order with time.

C. ROUTING PACKETS THROUGH PERMITTED PATHS

Once the packet is stored in the selected queue based on the QUCB rule, the path constraint of the queue dictates the permitted paths for routing the packet, where only paths with a smaller cost are permitted. We now describe the routing algorithm through the permitted paths. Packets are routed through the network using backpressure algorithm which transmits them to neighbor queues, such that their differential backlog is maximized. The selected queue at the neighbor node is determined by the path constraint.

Specifically, for any node v , packets stored in the m th queue (i.e., with state $Q_{(v,d,m)}(t)$) need to be send to destination node d through a path whose cost is at most C_m . Also, once transmitting the packets to a selected neighbor node (say v'), they can only be stored in queues $Q_{(v',d,m')}$ where m' is the index of the m' th queue in v' , and $C_{m'} \leq C_m$. The selection of the neighbor node is done by the configuration of the backpressure parameter in OLSB as described later. Note that the cost constraint of the packet is updated through the path. To guarantee the routing under these path constraints, the backpressure parameter in OLSB is defined as follows. At time t , we define the backpressure value between neighbor queues in nodes v and v' to destination d , with queue levels m and m' , respectively by:

$$P_{(v,d,m)}^{(v',d,m')}(t) \triangleq \begin{cases} Q_{(v,d,m)}(t) - Q_{(v',d,m')}(t), \\ \text{if } C_{m'} \leq \max(C_m - w_{(v,v')}(t), 0), \\ -\infty, & \text{otherwise,} \end{cases} \quad (3)$$

and we define the backpressure value of link $(v, v') \in E$ by:

$$P_{(v,v')}(t) \triangleq \max \left(\max_{d,m,m'} \left(P_{(v,d,m)}^{(v',d,m')}(t) \right), 0 \right). \quad (4)$$

At time t , the backpressure value $P_{(v,v')}(t)$ is computed at node v for each link $(v, v') \in E$, with neighbor node v' . Then, link (v, v^*) is selected for transmission, where node v^* solves

$$v^* = \arg \max_{v'} P_{(v,v')}(t). \quad (5)$$

The parameter values d, m, m' that solve (4) for link (v, v^*) dictate that the next transmitted packet through link (v, v^*) leaves the m th queue destined to d at node v and enters the m' th queue destined to d at node v^* . If the solution of (4) is zero, then node v does not transmit a packet through link (v, v^*) at time t . Note that in cases of half-duplex transmission systems or the existence of interference between links, then a certain MAC protocol can be readily applied to manage multi-access transmissions in the link layer, as commonly implemented by routing algorithms in wireless networks.

Finally, once packets reach the destination node d through path $p = (s, v_1, v_2, \dots, v_I, d)$ originated at source node s , the destination node d transmits back an ACK signal to source node s through path p . Then, the nodes through the path use the ACK signal to estimate the path cost, as described earlier in Subsection III-A.

D. COMPLEXITY ANALYSIS

We now analyze the computational complexity of OLSB. It was shown in [2], that by executing a barycentric spanner, the growth in the number of paths is only cubic with the number of nodes instead of an exponential growth of the path complexity by executing a naive search. Hence, the optimization in (2) used to schedule data packets at the source node has only cubic complexity with $|V|$, $O(|V|^3)$. Thus, the path complexity of OLSB is similar to the path complexity order in [2] and subsequent studies. Furthermore, when routing packets through permitted paths, each node (say node v) executes at most $O(NM|\mathcal{D}|)$ computations in (3) and (4) in order to implement the backpressure routing in OLSB. Here, N is the number of neighbors of v , M is the number of queue levels, and $|\mathcal{D}|$ is the number of destinations in the network dictated by the data flows in the network. This backpressure complexity order is similar to the one shown in [24] and subsequent studies.

IV. PERFORMANCE ANALYSIS

In this section, we provide rigorous theoretical analysis to evaluate the performance of OLSB. To analyze the performance theoretically, our benchmark for performance is a genie-aided algorithm that has complete knowledge of all path state means, $\mu_p = \sum_{e \in p} E(w_e(t))$, $p \in \mathcal{P}_{(s,d)}$, and consequently solves the deterministic optimization problem (1), defined in Section II at each time t , which was shown to be throughput optimal in [24].

The performance of online learning algorithms are commonly evaluated by the regret, defined as the loss of an

Algorithm 1 The OLSB Algorithm

Initialize: for every node $v \in V$ and every flow with destination node $d \in V$ do:

- Construct a barycentric spanner $\mathcal{B}_{(v,d)}$ that spans the path space $\mathcal{P}_{(v,d)}$.
- At time $t = 0$, transmit one packet through every path $p \in \mathcal{B}_{(v,d)}$, observe the current random path cost, and set it as the initial value of the estimate path cost $\bar{C}_p(0)$, $\forall p \in \mathcal{B}_{(v,d)}$.
- Set $T_p(0) = 1$, $\forall p \in \mathcal{B}_{(v,d)}$.
- For time slot $t \geq 1$, and each flow (say $f_{(s,d)}$) do:

Step 1 (at the source node):

- Process a newly arrived packet (or a frame of packets) at source node s for destination node d .
- Insert the packet to queue $Q_{(s,d,m(\bar{C}_{p^*}(t)))}$, where p^* solves (2).

Step 2 (at node v in the route):

- Let the packet arrive at node v in the route (and node v will then transmit the packet forward).
- Compute the backpressure parameter $P_{(v,v')}(t)$ for every link $(v, v') \in E$ using (4).
- Consider link (v, v^*) , where $v^* = \arg \max_{v'} P_{(v,v')}(t)$.
- If $P_{(v,v^*)} > 0$ and d, m, m' solves (4) for link (v, v^*) , then transmits a packet that leaves $Q_{(v,d,m)}$ and store it in $Q_{(v^*,d,m')}$ at neighbor node v^* .
- Repeat Step 2 for every node in the route until reaching the destination node.

Step 3 (at the destination node):

- Let the packet reaches the destination node d through path $p = (s, v_1, \dots, v_I, d)$.
- Update $\bar{C}_p(t)$ at node s , and $\bar{C}_{p_i}(t)$ of each sub-path $p_i = (v_i, \dots, v_I, d)$ at node v_i .
- Repeat Steps 1-3 for all packets and for all data flows.

algorithm as compared to genie with side information on the system. Here, we define the regret as the performance loss of OLSB (that learns online to route packets under unknown link state means) compared to the genie-aided algorithm, described above. As a result, the regret evaluates how fast OLSB learns the side information and approaches genie's performance. Below, we show analytically that the regret scales logarithmically with time, which indicates that OLSB converges to genie's performance with the best known rate. Specifically, to evaluate the performance measure, we condition on the same queue states for both algorithms, and define the regret R_n^{OLSB} at time n as the aggregated loss in performance attained by OLSB as compared to genie's performance over time slots $t = 1, \dots, n$:

$$R_n^{OLSB} = E \left[\sum_{t=1}^n \left(KC_{p_t} + Q_{(s,d,m(C_{p_t}(t)))} \right) \right] - \sum_{t=1}^n \min_{p \in \mathcal{P}_{(s,d)}} \left(K\mu_p + Q_{(s,d,m(\mu_p))} \right), \quad (6)$$

where p_t is the path selected by OLSB at time slot t , and $C_{p_t} = \sum_{e \in p_t} w_e(t)$ is the actual path cost incurred through the selected path p_t at time slot t .

In the following theorem we provide the upper bound on the regret R_n^{OLSB} for each flow and for all n , and show that it scales logarithmically with time.

Theorem 1: The regret R_n^{OLSB} is upper bounded by:

$$\left[8 \sum_{\substack{i=1, \dots, L \\ i: \Delta_i^{min} \neq 0}} \frac{\Psi_i \ln n}{(\Delta_i^{min})^2} \right] + (L-1) \left(1 + \frac{\pi^2}{3} \right) \sum_{i=1}^L \Psi_i, \quad (7)$$

where

$$\Psi_i \triangleq (K \mu_i + \eta_{m(\mu_i)}) - \left(K \min_{\substack{j=1, \dots, L \\ j \neq i}} \mu_j + \min_{\substack{j=1, \dots, L \\ j \neq i}} \eta_{m(\mu_j)} \right), \quad (8)$$

$$\Delta_i^{min} \triangleq \min_{\substack{j=1, \dots, L \\ j \neq i}} \left((K \mu_j + \eta_{m(\mu_j)}) - (K \mu_i + \eta_{m(\mu_i)}) \right). \quad (9)$$

Here, $\ln(\cdot)$ is the natural logarithm, L is the number of barycentric spanner paths of the flows, and $\eta_{m(\mu_i)}$ is the mean value of queue $m(\mu_i)$ at the source node.

The proof is provided in the Appendix.

A. DISCUSSION ON THE THEORETICAL RESULTS

Here, we discuss the theoretical convergence and the efficiency of the performance obtained under OLSB, as analyzed in this section.

1) CONVERGING TO THE OPTIMAL TIME-VARYING STRATEGY

Unlike existing online learning algorithms for adaptive routing that converge to a single and fixed path which is optimal in terms of minimizing the expected cost (see e.g., [2], [15] and references therein), the proposed OSLB algorithm converges to the optimal time-varying selection of paths. The reason is that the backpressure scheduling term (needed to achieve throughput optimality [24]) in the stochastic optimization problem depends on the time-varying queue dynamics. As a result, the learning mechanism in OLSB is fundamentally different in both the design and convergence analysis of the learning algorithm. This can be observed in the design of the QUCB rule that maps the dynamic queue state and the empirical mean of the path state into a path selection index, that dictates the cost limit of the paths accordingly. Then, OLSB uses the QUCB rule to determine the cost limit for packet transmissions, and backpressures packets through paths that meet the QUCB's cost conditions.

2) ACHIEVING STRONG REGRET UNDER OLSB

Differing from weak regret analysis used to simplify the design of the learning algorithm by converging to a static genie, which is restricted to select the same action over time (see e.g., [56] and subsequent studies in [2], [15], [32], [33], and [39]), here the algorithm converges to the optimal

time-varying selection of paths. The definition of genie is consistent with the optimal time-varying selection of paths, without restricting its selection rule. OLSB thus minimizes strong regret, as analyzed in Theorem 1.

V. SIMULATION RESULTS

In this section, simulation results are presented to validate the theoretical findings, and demonstrate the efficiency in performance of OLSB. We simulated a directed network alike the one in [24] with 64 nodes and 119 links. An illustration of the network is shown in Fig. 1. The additional links were inserted to model the case of different hopping transmissions (as in 5G mesh networks). We simulated a network with nine flows as shown in Table 1, such that two flows originate in the same source node, two flows are targeted to the same destination node and the last five are random flows. The packet arrivals of all flows follow a Poisson process with rate λ . We initialized the nodes with empty queues.

A. EVALUATING THE CONVERGENCE OF OLSB TO THE OPTIMAL STRATEGY

In these simulations we demonstrate the learning efficiency of OLSB as compared to the optimal solution by genie that has complete knowledge of the path state means [24].

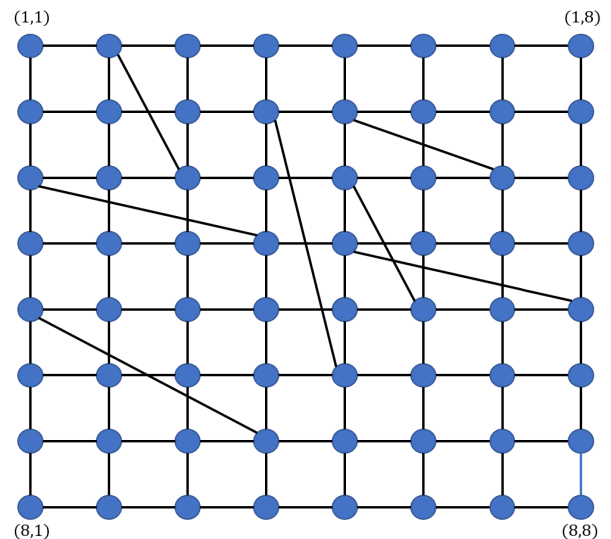


FIGURE 1. An illustration of the network used in the simulations.

We start by validating the theoretical analysis of the regret, which measures the convergence speed of OLSB to the optimal strategy. For this, we computed the regret empirically according to (6) and normalized it by $\log(t)$ (i.e., converging to a constant value validates the logarithmic order of the regret with time). In Fig. 2, we show the impact of the value of the K parameter (given in (2)) on the regret curve. We note that since the coefficient of the logarithm in the regret expression is inversely proportional to the value of K , lower values of K result in a longer convergence time. This means that it is easier to learn strategies that assign high priority for

TABLE 1. The locations of flows used in the simulations. Flows 1 and 2 both originate at node (1, 2); flows 4 and 9 both destined to reach node (8, 8).

Flow	Source	Destination
1	(1, 2)	(4, 4)
2	(1, 2)	(8, 4)
3	(2, 2)	(3, 7)
4	(2, 6)	(8, 8)
5	(3, 3)	(8, 6)
6	(3, 4)	(5, 8)
7	(4, 1)	(6, 8)
8	(5, 3)	(7, 8)
9	(5, 4)	(8, 8)

transmissions over short paths. This observation is intuitively satisfying, as the algorithm is required to learn smaller subsets of path selections. It can be seen clearly that we obtain a logarithmic regret order with time for each selection of K , which supports the theoretical results.

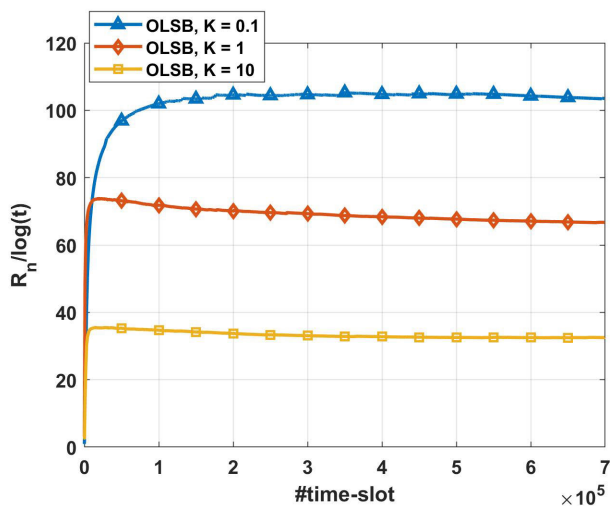


FIGURE 2. The empirical regret (normalized by $\log t$) obtained by OLSB as compared to genie’s performance, for $K = [0.1, 1, 10]$ and $\lambda = 1$.

B. EVALUATING THE LATENCY AND NETWORK CONGESTION UNDER OLSB

Next, we evaluate the latency and network congestion achieved by OLSB. For this purpose, we present the average end-to-end delay of all successful transmissions, along with the average per-node queue lengths, and supportable rates for different values of the K parameter, and for low, moderate and high loads. We set the time slot duration to $20\mu s$. We compare the results with the following routing methods: (i) The backpressure routing algorithm, that routes data in directions that maximize the differential queue backlog between nodes to reduce the congestion [57]; (ii) the Adaptive

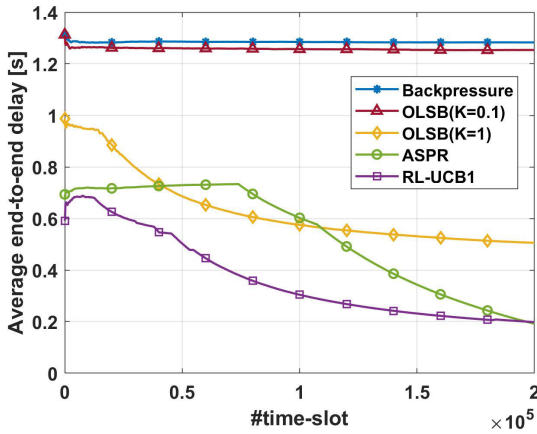
TABLE 2. Performance under lightly-loaded network.

Algorithm	Average flow rate	Average delay
OLSB ($K = 1$)	40Mbps	0.49 [s]
OLSB ($K = 0.1$)	40Mbps	1.26 [s]
Backpressure	40Mbps	1.28 [s]
RL-UCB1	40Mbps	0.2 [s]
ASPR	40Mbps	0.22 [s]

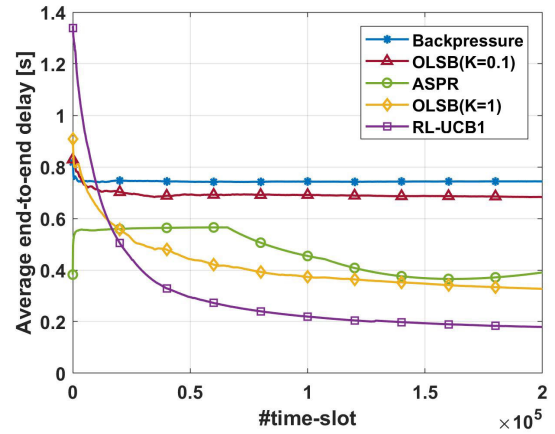
Shortest Path Routing (ASPR) algorithm, that uses adaptive strategies to learn the shortest path routing [2]; and (iii) the recently suggested reinforcement learning routing method that uses multi-armed bandit framework based on UCB1 for path learning and packet transmissions (RL-UCB1) [44].

In Fig. 3, we present simulation results of a lightly-loaded network ($\lambda = 0.6$). It is shown that setting larger K values in OLSB leads to better performance in terms of average end-to-end delay but results in higher queue loads. This is because large K values leads to more frequent selections of short paths by increasing this priority in the objective function. However, we note that this is an acceptable behaviour for low arrival rates, since the exploration of longer paths is not necessary for load balancing. Furthermore, as discussed above, the backpressure algorithm performs poorly under light loads because of extensive and unnecessary exploration of paths for network stability. Therefore, while backpressure routing remains stable over time, it does not exploit better paths, in terms of the total cost, as the OLSB algorithm. The ASPR and RL-UCB1 tends to be unstable over time, as they fail to balance the congestion in the network. In Table 2, we present the specific supportable rates in Mbps and the average delay over time for the various algorithms. It can be seen that all algorithms were able to achieve supportable flow rate of 40Mbps. In this scenario, RL-UCB1 and ASPR achieves the best delay (although more susceptible to instability due to higher backlogged queues), since they learn shorter paths to send packets. OLSB with $K = 1$ comes in third place in terms of delay, as it provides a good balance between network stability and delay. OLSB with $K = 0.1$ and backpressure achieve significantly higher delay in this case.

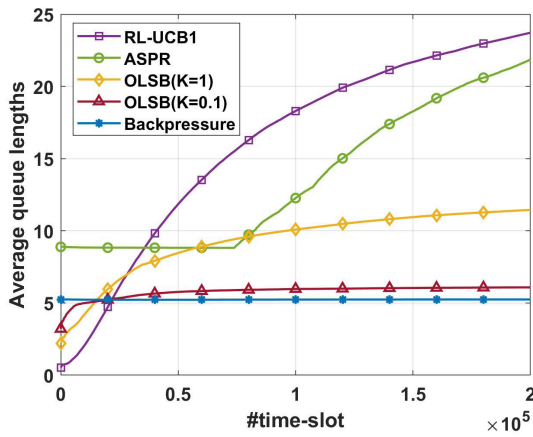
In Fig. 4, we present simulation results of a moderately-loaded network ($\lambda = 1$). We obtained a similar behaviour of OLSB as in the lightly-loaded network, as it still performs well. It can be seen that the improvement of OLSB over the backpressure algorithm increases. The RL-UCB1 performs well in this scenario as well, although its stability is limited. The ASPR tends to be unstable over time. In Table 3, we present the specific supportable rates in Mbps and the average delay over time for the various algorithms. It can be seen that OLSB, backpressure, and RL-UCB1 algorithms were able to achieve supportable flow rate of 67Mbps, where ASPR was not able to support this rate. In this scenario, RL-UCB1 achieves the best delay again (although more susceptible to instability due to higher backlogged queues). OLSB with $K = 1$ comes in second place in terms of delay, as it



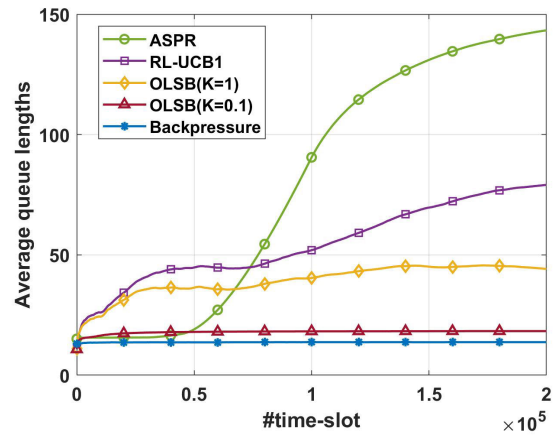
(a) Average end-to-end delay.



(a) Average end-to-end delay.



(b) Average per-node queue lengths.



(b) Average per-node queue lengths.

FIGURE 3. The average end-to-end delay, and average per-node queue length in a lightly-loaded network ($\lambda = 0.6$).

FIGURE 4. The average end-to-end delay, and average per-node queue length in a moderately-loaded network ($\lambda = 1$).

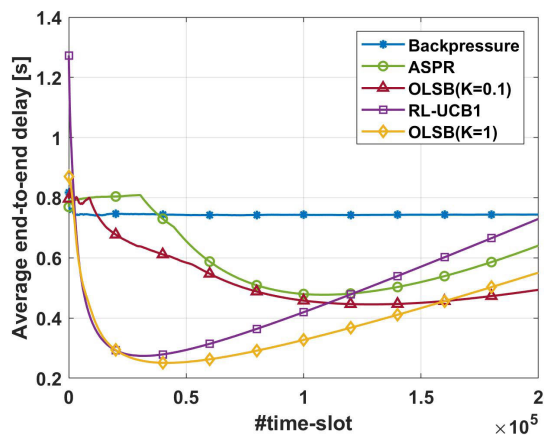
TABLE 3. Performance under moderately-loaded network.

Algorithm	Average flow rate	Average delay
OLSB ($K = 1$)	67Mbps	0.35 [s]
OLSB ($K = 0.1$)	67Mbps	0.69 [s]
Backpressure	67Mbps	0.72 [s]
RL-UCB1	67Mbps	0.21 [s]
ASPR	Non supportable	Non supportable

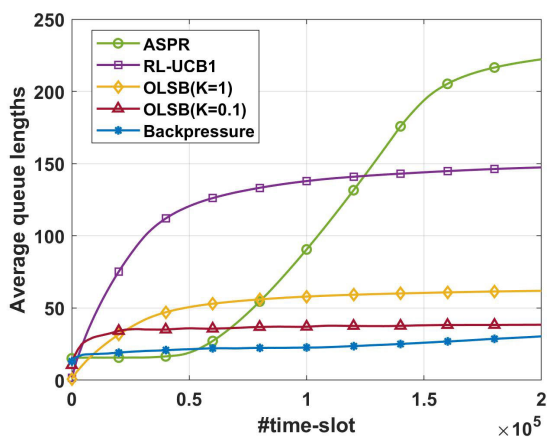
provides a good balance between network stability and delay. OLSB with $K = 0.1$ and backpressure achieve significantly higher delay in this case.

Finally, we simulated a highly-loaded network ($\lambda = 1.5$). The results are presented in Fig. 5. It can be seen that both RL-UCB1 and OLSB (with $K = 1$) learn the path cost quickly. However, it can be inferred that the shortest-path queues are filled quickly and the delay grows over time. This is an undesired behaviour since sub-optimal queues remain mostly unused. In this case, it can be seen that OLSB with $K = 0.1$ shows strong performance both in terms of end-to-end

delay as well as queue stability. This is obtained by reducing the priority of using short paths when decreasing K in the OLSB optimization. This is intuitively satisfying, as increasing the priority of backpressured transmissions together with efficient path exploration and exploitation mechanism of the OLSB optimization is desired in high loads. Finally, the pure backpressure algorithm shows balanced behaviour, as expected when all queues are utilized equally. Moreover, it can be seen that OLSB with $K = 0.1$ achieves low congestion level compared to the other algorithms. As expected, both RL-UCB1 and ASPR perform poorly under high loads in terms of average queue length since they highly prioritize transmissions through short paths rather than transmissions that achieve efficient queue balancing. Furthermore, it can be seen that OLSB outperforms the RL-UCB1 and backpressure algorithms. This is because RL-UCB1 learns a fixed set of paths across time, while OLSB balances between the minimal cost and the time-varying queue states. Also, the backpressure algorithm results in sending packets in long paths, which reduces the performance in terms of end-to-end delay.



(a) Average end-to-end delay.



(b) Average per-node queue lengths.

FIGURE 5. The average end-to-end delay, and average per-node queue lengths in a highly-loaded network ($\lambda = 1.5$).

TABLE 4. Performance under highly-loaded network.

Algorithm	Average flow rate	Average delay
OLSB ($K = 1$)	100Mbps	0.59 [s]
OLSB ($K = 0.1$)	100Mbps	0.8 [s]
Backpressure	100Mbps	0.75 [s]
RL-UCB1	Non supportable	Non supportable
ASPR	Non supportable	Non supportable

In Table 4, we present the specific supportable rates in Mbps and the average delay over time for the various algorithms. It can be seen that only OLSB and backpressure algorithms were able to achieve supportable flow rate of 100Mbps, where RL-UCB1 and ASPR were not able to support this rate. OLSB with $K = 1$ achieves the best delay, and provides an excellent balance between network stability and delay. OLSB with $K = 0.1$ and backpressure achieves higher delay in this case again.

In summary, the superior performance of OLSB compared to existing methods can be attributed to its hybrid

approach, which exploits the strengths of both backpressure and shortest-path algorithms. The underlying idea of OLSB is to use shorter paths for data transmissions during times of low network congestion, while longer paths are used during periods of high congestion. This strategy helps to limit the use of backpressure (which can cause significant delays) during periods of low congestion and restrict the use of shortest path routing (which increases congestion) during periods of high congestion. In our simulations, we observed excellent performance of OLSB when the balance parameter K was set to 1. The second reason for the strong performance of OLSB is its ability to solve the stochastic optimization problem associated with exploration versus exploitation of network states. OLSB is carefully designed to achieve the best possible regret order, as outlined in Theorem 1.

VI. CONCLUSION

We investigated in this paper the problem of efficient adaptive routing under unknown path states. We developed a novel algorithm, dubbed OLSB, to maximize the network throughput while maintaining small cost of end-to-end flows. We have analyzed OLSB theoretically and showed that it attained a logarithmic regret order as compared to genie with complete knowledge of the path state means. We presented simulation results that support the theoretical findings, and demonstrate strong performance of OLSB. Specifically, OLSB demonstrated strong and robust performance in all simulations, while other existing methods failed to present robust performance. Furthermore, OLSB has the ability of optimizing the performance depending on the network load by adjusting a simple tuning parameter that controls the balancing between using short paths and reducing the congestion level, which makes it simple for implementation in practical networks.

There are several aspects for extending this work. Firstly, it would be interesting to investigate whether the leading factor in the logarithmic regret order could be improved to enhance the learning efficiency. Secondly, it may be possible to develop techniques to dynamically optimize the parameter K , which balances the use of short paths against network congestion. Thirdly, although OLSB is proven to achieve a logarithmic regret, the learning mechanism becomes challenging as the network size increases (as in other learning-based routing algorithms). Therefore, one can extend OLSB by developing cluster-based OLSB to optimize routing decisions over smaller clusters, as done in BGP that exchanges routing data among autonomous systems (AS). Another alternative is to utilize centralized computation units, as seen in 5G deployments, which could potentially improve the learning performance. This could reduce the exploration phase required for inferring network states in the OLSB algorithm, resulting in improved network performance.

APPENDIX

In this appendix, we provide the proof of Theorem 1.

Throughout the proof we denote the source node and destination node of the flow by s, d , respectively. The selected path by OLSB at time t is denoted by p_t . and let

$$g_t \triangleq \arg \min_{p \in \mathcal{P}(s,d)} \{ \mu_p + Q_{(s,d,m(\mu_p))}(t) \}$$

be the optimal path which is selected by genie at time step t . The cumulative regret after n plays is given by:

$$R_n = E \left[\sum_{t=1}^n \left(KC_{p_t}(t) + Q_{(s,d,m(C_{p_t}(t)))}(t) \right) - \sum_{t=1}^n KC_{g_t}(t) + Q_{(s,d,m(C_{g_t}(t)))}(t) \right). \quad (10)$$

We can rewrite the first term on the RHS of (10) by summing the balanced cost over paths:

$$E \left[\sum_{t=1}^n \left(KC_{p_t}(t) + Q_{(s,d,m(C_{p_t}(t)))}(t) \right) \right] = \sum_{p \in \mathcal{P}(s,d)} (K\mu_p + \eta_{m(\mu_p)}) E[T_p(n)]. \quad (11)$$

Next, we can bound the second term on the RHS of (10) by using the linearity of expectation and summing the minimum over $K\mu_p$ plus the minimum over $\eta_{m(\mu_p)}$ at each time t :

$$E \left[\sum_{t=1}^n KC_{g_t}(t) + Q_{(s,d,m(C_{g_t}(t)))}(t) \right] \geq \left(\min_{p \in \mathcal{P}(s,d)} K\mu_p + \min_{p \in \mathcal{P}(s,d)} \eta_{m(\mu_p)} \right) n. \quad (12)$$

By substituting (11) and (12) in (10), we can upper bound the cumulative regret by:

$$R_n \leq \sum_{p \in \mathcal{P}(s,d)} (K\mu_p + \eta_{m(\mu_p)}) E[T_p(n)] - \left(\min_{p \in \mathcal{P}(s,d)} K\mu_p + \min_{p \in \mathcal{P}(s,d)} \eta_{m(\mu_p)} \right) n. \quad (13)$$

We next upper bound the expected value of the number of times that path p was selected for transmission. Let

$$c_{t,s} \triangleq \sqrt{\frac{2 \ln t}{s}}. \quad (14)$$

Then,

$$\begin{aligned} T_p(n) &=_{(a)} 1 + \sum_{t=L+1}^n \left\{ p_t = p \right\} \\ &\leq_{(b)} l + \sum_{t=L+1}^n \left\{ p_t = p, T_p(t-1) \geq l \right\} \\ &\leq_{(c)} l + \sum_{t=L+1}^n \left\{ K \underbrace{\bar{C}_{g_{t-1}}(T_{g_{t-1}}(t-1))}_{\triangleq \bar{C}_g} \right. \\ &\quad \left. + Q_{(s,d,m(\bar{C}_g))}(t-1) - c_{t-1, T_{g_{t-1}}(t-1)} \right. \\ &\quad \left. \geq K \underbrace{\bar{C}_p(T_p(t-1))}_{\triangleq \bar{C}_p} + Q_{(s,d,m(\bar{C}_p))}(t-1) \right. \\ &\quad \left. - c_{t-1, T_p(t-1)}, T_p(t-1) \geq l \right\} \\ &\leq_{(d)} l + \sum_{t=L+1}^n \left\{ \max_{0 < s_g < t} \min_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} K \bar{C}_r(s_g) \right. \\ &\quad \left. + Q_{(s,d,m(\bar{C}_r(s_g))}(s_g) - c_{t-1, s_g} \right. \\ &\quad \left. \geq \min_{l \leq s_p \leq t} K \bar{C}_p(s_p) + Q_{(s,d,m(\bar{C}_p(s_p))}(s_p) \right. \\ &\quad \left. - c_{t-1, s_p} \right\} \\ &\leq_{(e)} l + \sum_{t=L+1}^n \sum_{s_g=1}^{t-1} \sum_{s_p=l}^{t-1} \sum_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} \left\{ K \bar{C}_r(s_g) + Q_{(s,d,m(\bar{C}_r(s_g))}(s_g) - c_{t-1, s_g} \right. \\ &\quad \left. \geq K \bar{C}_p(s_p) + Q_{(s,d,m(\bar{C}_p(s_p))}(s_p) \right. \\ &\quad \left. - c_{t-1, s_p} \right\}, \quad (15) \end{aligned}$$

where $\left\{ E \right\}$ is the indicator function, which equals 1 when event E is true, and equals 0 otherwise. Below, we explain each bounding step of $T_p(n)$ in (15):

- (a) Step (a) follows since the number of times that path p was selected for transmission up to time n is given by the sum of one (due to the first initial path selection) plus the number of time-slots in which path p was selected by the algorithm, i.e. $p^*(t) = p$.
- (b) Step (b) follows since we take $l - 1$ occurrences out of the sum and condition the sum to count path p selection only after it was selected l times.
- (c) Step (c) follows since the event $p_t = p$ occurs when p solves the QUCB rule in OLSB:

$$p = \arg \min_{r \in \mathcal{P}(s,d)} K \bar{C}_r(t) + Q_{(s,d,m(\bar{C}_r(t)))}(t) + c_{t, T_r(n)}.$$

Also, note that by the definition of minimization the solution is smaller or equal than the value of the function

when the argument is path g_t which was selected by genie, which yields Step (c).

- (d) Step (d) further upper bounds the expression since if the value of path p is smaller than the value of path g_{t-1} then its minimal value from time l to the current time t is smaller than the maximum over all minimal values by other path selections up to time t . When the condition holds, we get one triplet of (r, s_g, s_p) that we count as path p selection.
- (e) Step (e) follows since we count every (r, s_g, s_p) triplet that meets the condition.

Next, note that for condition

$$K\bar{C}_r(s_g) + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) - c_{t,s_g} \geq K\bar{C}_p(s_p) + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - c_{t,s_p} \quad (16)$$

to hold, then for each $r \neq p$ at least one of the following inequalities must hold:

Inequality 1:

$$K\bar{C}_r(s_g) + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) \geq K\mu_r + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) + c_{t,s_g}. \quad (17)$$

Inequality 2:

$$K\bar{C}_p(s_p) + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) \leq K\mu_p + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - c_{t,s_p}. \quad (18)$$

Inequality 3:

$$K\mu_r + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) > K\mu_p + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - 2c_{t,s_p}. \quad (19)$$

Therefore, we get $L - 1$ sets of these three inequalities.

We prove by contradiction that by assuming that if for all $r \in \mathcal{P}(s,d)$, $r \neq p$ all inequalities are false, then:

$$\begin{aligned} & K\mu_r + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) \\ & > (a) \quad K\bar{C}_r(s_g) + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) - c_{t,s_g} \\ & \geq (b) \quad K\bar{C}_p(s_p) + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - c_{t,s_p} \\ & > (c) \quad K\mu_p + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - 2c_{t,s_p}. \end{aligned} \quad (20)$$

Below, we explain each bounding step in (20):

- (a) Step (a) follows by assuming that inequality (17) is false.
- (b) Step (b) follows by inequality (16).
- (c) Step (c) follows by assuming that inequality (18) is false.

Therefore, we get:

$$K\mu_r + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) < K\mu_p + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - 2c_{t,s_p} \quad (21)$$

which meets inequality (19), which is in contradiction to the assumption that all three inequalities are false.

Next, we apply the Chernoff-Hoeffding bound on inequalities (17) and (18), and get:

$$\begin{aligned} & Pr(K\bar{C}_r(s_g) - K\mu_r \geq c_{t,s_g}) \\ & \leq e^{-2s_g c_{t,s_g}^2} = e^{-2s_g \frac{2 \ln t}{s_g}} = t^{-4}, \end{aligned} \quad (22)$$

and

$$\begin{aligned} & Pr(K\mu_p - K\bar{C}_p(s_p) \geq c_{t,s_p}) \\ & \leq e^{-2s_p c_{t,s_p}^2} = e^{-2s_p \frac{2 \ln t}{s_p}} = t^{-4}. \end{aligned} \quad (23)$$

Also, it suffices to choose inequality (19) to be false:

$$K\mu_r + Q_{(s,d,m(\bar{C}_r(s_g)))}(s_g) \leq K\mu_p + Q_{(s,d,m(\bar{C}_p(s_p)))}(s_p) - 2c_{t,s_p}. \quad (24)$$

We take expectation and get:

$$K\mu_r + \eta_{m(\mu_r)} \leq K\mu_p + \eta_{m(\mu_p)} - 2c_{t,s_p}, \quad (25)$$

and by arranging terms we get:

$$2c_{t,s_p} \leq K\mu_p + \eta_{m(\mu_p)} - K\mu_r - \eta_{m(\mu_r)} \triangleq \Delta_{r,p}(K).$$

Also, note that

$$2c_{t,s_p} = 2\sqrt{\frac{2 \ln t}{s_p}} \leq \Delta_{r,p}(K), \quad \forall r \in \mathcal{P}(s,d), r \neq p.$$

Note that we get this inequality $L - 1$ times, for all $r \neq p$. Next, we define:

$$\begin{aligned} \Delta_p^{\min}(K) & \triangleq \min_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} \Delta_{r,p}(K) \\ & = \min_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} K\mu_p + \eta_{m(\mu_p)} - K\mu_r - \eta_{m(\mu_r)}. \end{aligned} \quad (26)$$

Now, we choose $\tilde{s}_p \in \mathbb{R}$ such that $2c_{t,\tilde{s}_p} = |\Delta_p^{\min}(K)|$ holds. Thus,

$$2\sqrt{\frac{2 \ln t}{\tilde{s}_p}} = |\Delta_p^{\min}(K)|,$$

and we get

$$\frac{8 \ln t}{\tilde{s}_p} = (\Delta_p^{\min}(K))^2,$$

and finally, we get

$$\tilde{s}_p = \frac{8 \ln t}{(\Delta_p^{\min}(K))^2}. \quad (27)$$

Next, recall that by definition $s_p \geq l$ and $l \in \mathbb{N}$. Then,

$$\left\lceil \frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right\rceil = l \leq s_p.$$

Now, we can upper bound $T_p(n)$ as follows:

$$\begin{aligned}
T_p(n) &\leq \left[\frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right] \\
&\quad + \sum_{t=1}^n \sum_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} \sum_{s_g=1}^{t-1} \sum_{s_p=1}^{t-1} \left[Pr(K\bar{C}_r(s_g) \geq K\mu_r + c_{t,s_g}) \right. \\
&\quad \left. + Pr(K\bar{C}_p(s_p) \leq K\mu_p - c_{t,s_p}) \right] \\
&\leq \left[\frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right] + \sum_{t=1}^n \sum_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} \sum_{s_g=1}^{t-1} \sum_{s_p=1}^{t-1} 2t^{-4} \\
&\leq \left[\frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right] + \sum_{t=1}^{\infty} \sum_{\substack{r \in \mathcal{P}(s,d) \\ r \neq p}} \sum_{s_g=1}^t \sum_{s_p=1}^t 2t^{-4} \\
&= \left[\frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right] + (L-1) \sum_{t=1}^{\infty} 2t^{-2} \\
&\leq \left[\frac{8 \ln t}{(\Delta_p^{\min}(K))^2} \right] + (L-1) \left(1 + \frac{\pi^2}{3}\right). \tag{28}
\end{aligned}$$

Finally, we substitute (28) in (13), which completes the proof.

ACKNOWLEDGMENT

In this journal version we include: (i) A detailed discussion on the implementation of the algorithm; (ii) a rigorous theoretical analysis of the algorithm with detailed proofs; (iii) more extensive simulation results; and (iv) a detailed discussion of the results, and comprehensive discussion and comparison with the existing literature.

An earlier version of this paper was presented at the 2021 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT45174.2021.9518237].

REFERENCES

- O. Amar and K. Cohen, "Online learning for shortest path and backpressure routing in wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 2702–2707.
- K. Liu and Q. Zhao, "Adaptive shortest-path routing under unknown and stochastically varying link states," in *Proc. 10th Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt)*, May 2012, pp. 232–237.
- P. Tehrani and Q. Zhao, "Distributed online learning of the shortest path under unknown random edge weights," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 3138–3142.
- B. Pourpeighambar, M. Dehghan, and M. Sabaei, "Joint routing and channel assignment using online learning in cognitive radio networks," *Wireless Netw.*, vol. 25, no. 5, pp. 2407–2421, Jul. 2019.
- J. Scarlett, I. Bogunovic, and V. Cevher, "Overlapping multi-bandit best arm identification," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 2544–2548.
- W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10737–10750, 2021.
- H. B. Salameh, S. Otoum, M. Aloqaily, R. Derbas, I. A. Ridhawi, and Y. Jararweh, "Intelligent jamming-aware routing in multi-hop IoT-based opportunistic cognitive radio networks," *Ad Hoc Netw.*, vol. 98, Mar. 2020, Art. no. 102035.
- R. N. Raj, A. Nayak, and M. S. Kumar, "A survey and performance evaluation of reinforcement learning based spectrum aware routing in cognitive radio ad hoc networks," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 1, pp. 144–163, Mar. 2020.
- T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated learning: A signal processing perspective," 2021, *arXiv:2103.17150*.
- Z. Huang, Y. Xu, and J. Pan, "TSOR: Thompson sampling-based opportunistic routing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7272–7285, Nov. 2021.
- A. Somekh-Baruch, S. S. Shamai, and S. Verdú, "Cooperative multiple-access encoding with states available at one transmitter," *IEEE Trans. Inf. Theory*, vol. 54, no. 10, pp. 4448–4469, Oct. 2008.
- Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. IEEE Symp. New Frontiers Dyn. Spectr. (DySPAN)*, Apr. 2010, pp. 1–9.
- T. He, D. Goeckel, R. Raghavendra, and D. Towsley, "Endhost-based shortest path routing in dynamic networks: An online learning approach," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2202–2210.
- A. Ghosh and S. Sarkar, "Secondary spectrum oligopoly market over large locations," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Jan. 2016, pp. 1–10.
- M. S. Talebi, Z. Zou, R. Combes, A. Proutiere, and M. Johansson, "Stochastic online shortest path routing: The value of feedback," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 915–930, Apr. 2018.
- Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.
- R. Srikant and L. Ying, *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- H. Gong, L. Fu, X. Fu, L. Zhao, K. Wang, and X. Wang, "Distributed multicast tree construction in wireless sensor networks," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 280–296, Jan. 2017.
- A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Syst.*, vol. 50, no. 4, pp. 401–457, Aug. 2005.
- A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.
- L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2936–2940.
- A. Sinha and E. Modiano, "Optimal control for generalized network-flow problems," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 506–519, Feb. 2018.
- C. Joo, "On the performance of back-pressure scheduling schemes with logarithmic weight," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3632–3637, Nov. 2011.
- L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun. 2011.
- I. Menache and N. Shimkin, "Rate-based equilibria in collision channels with fading," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1070–1077, Sep. 2008.
- I. Menache and A. Ozdaglar, *Network Games: Theory, Models, and Dynamics* (Synthesis Lectures on Communication Networks), vol. 4, no. 1. San Rafael, CA, USA: Morgan & Claypool, 2011, pp. 1–159.
- K. Cohen and A. Leshem, "Distributed game-theoretic optimization and management of multichannel ALOHA networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1718–1731, Jun. 2016.
- K. Cohen, A. Nedic, and R. Srikant, "Distributed learning algorithms for spectrum sharing in spatial random access wireless networks," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2854–2869, Jun. 2017.
- I. Bistritz and A. Leshem, "Game theoretic dynamic channel allocation for frequency-selective interference channels," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 330–353, Jan. 2019.
- J. Chen, Q. Wu, Y. Xu, Y. Zhang, and Y. Yang, "Distributed demand-aware channel-slot selection for multi-UAV networks: A game-theoretic learning approach," *IEEE Access*, vol. 6, pp. 14799–14811, 2018.
- C. Tekin and M. Liu, "Online learning in opportunistic spectrum access: A restless bandit approach," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2011, pp. 2462–2470.
- C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, Aug. 2012.

- [33] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multiarmed bandit with unknown dynamics," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1902–1916, Mar. 2013.
- [34] K. Cohen, Q. Zhao, and A. Scaglione, "Restless multi-armed bandits under time-varying activation constraints for dynamic spectrum access," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 1575–1578.
- [35] T. Gafni and K. Cohen, "Learning in restless multi-armed bandits using adaptive arm sequencing rules," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1206–1210.
- [36] I. Bistriz and A. Leshem, "Distributed multi-player bandits—A game of thrones approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7222–7232.
- [37] E. Turgay, C. Bulucu, and C. Tekin, "Exploiting relevance for online decision-making in high-dimensions," *IEEE Trans. Signal Process.*, vol. 69, pp. 1438–1451, 2021.
- [38] M. Yemini, A. Leshem, and A. Somekh-Baruch, "Restless hidden Markov bandit with linear rewards," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 1183–1189.
- [39] T. Gafni and K. Cohen, "Learning in restless multiarmed bandits via adaptive arm sequencing rules," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 5029–5036, Oct. 2021.
- [40] T. Gafni and K. Cohen, "Distributed learning over Markovian fading channels for stable spectrum access," 2021, *arXiv:2101.11292*.
- [41] T. Gafni, M. Yemini, and K. Cohen, "Learning in restless bandits under exogenous global Markov process," 2021, *arXiv:2112.09484*.
- [42] R. Agrawal, "Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem," *Adv. Appl. Probab.*, vol. 27, pp. 1054–1078, Dec. 1995.
- [43] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.
- [44] G. Tabei, Y. Ito, T. Kimura, and K. Hirata, "Multi-armed bandit-based routing method for in-network caching," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2021, pp. 1899–1902.
- [45] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
- [46] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.
- [47] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
- [48] W. Xia, C. Di, H. Guo, and S. Li, "Reinforcement learning based stochastic shortest path finding in wireless sensor networks," *IEEE Access*, vol. 7, pp. 157807–157817, 2019.
- [49] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek, and M. Reisslein, "QR-SDN: Towards reinforcement learning states, actions, and rewards for direct flow routing in software-defined networks," *IEEE Access*, vol. 8, pp. 174773–174791, 2020.
- [50] R. G. C. Upeksha and W. P. J. Pamarathne, "Ant colony optimization algorithms for routing in wireless sensor networks: A review," in *Recent Advances in Electrical and Electronic Engineering and Computer Science*. Malaysia: Springer, 2022, pp. 47–57.
- [51] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022.
- [52] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, p. 7898, Nov. 2021.
- [53] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019.
- [54] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook," *Ad Hoc Netw.*, vol. 130, May 2022, Art. no. 102790.
- [55] B. Awerbuch and R. Kleinberg, "Online linear optimization and adaptive routing," *J. Comput. Syst. Sci.*, vol. 74, no. 1, pp. 97–114, Feb. 2008.
- [56] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2002.
- [57] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 279–290.



OMER AMAR is currently pursuing the M.Sc. degree in electrical and computer engineering from Ben-Gurion University, Israel. He is also a ML engineer, focusing on evaluating, deploying, and monitoring DL and RL algorithms. His research interests include sequential learning, decision theory, and statistical inference and learning, with applications in large-scale systems.



ILANA SARFATI received the B.Sc. degree in computer science from the Israel Institute of Technology (Technion), in 2001, and the Executive M.B.A. degree from the Hebrew University of Jerusalem, in 2019. She is currently the Senior Engineering Manager with more than 20 years of experience in software architecture, programming, and delivering to customers.



KOBI COHEN (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in electrical engineering from Bar-Ilan University, Ramat Gan, Israel, in 2007 and 2013, respectively. He was with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, from August 2014 to July 2015, and the Department of Electrical and Computer Engineering, University of California at Davis, Davis, from November 2012 to July 2014, as a Postdoctoral Research

Associate. In October 2015, he joined the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev (BGU), Beer Sheva, Israel, where he is currently an Associate Professor. His research interests include statistical inference and learning, signal processing, communication networks, decision theory, and stochastic optimization, with applications to large-scale systems, cyber systems, and wireless and wireline networks. He is also a member of the Cyber Security Research Center and the Data Science Research Center, BGU. Since 2021, he has been serving as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. Other selected awards and honors include highlighting in top 50 popular paper list, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2019 and 2020) for paper: "Deep multi-user reinforcement learning for distributed dynamic spectrum access," highlighting in popular paper list, *IEEE Signal Processing Magazine*, in 2022, for paper: "Federated learning: A signal processing perspective," receiving the Best Paper Award in the International Symposium on Modeling and Optimization in Mobile, Ad hoc and Wireless Networks (WiOpt), in 2015, the Feder Family Award (second prize), awarded by the Advanced Communication Center, Tel Aviv University (2011), and the President Fellowship (2008–2012) and top Honor List's prizes (2006, 2010, and 2011) from Bar-Ilan University.

• • •