IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 October 2022, accepted 4 December 2022, date of publication 9 December 2022,
date of current version 19 December 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3228170*

RESEARCH ARTICLE

# Intrusion Prevention Scheme Against Rank Attacks for Software-Defined Low Power IoT Networks

**CHRISTIAN MIRANDA[1], (Member, IEEE), GEORGES KADDOUM[1], (Member, IEEE), AMINE BOUKHTOUTA[2], (Member, IEEE), TAOUS MADI[2], (Member, IEEE), AND HYAME ASSEM ALAMEDDINE[2], (Member, IEEE)**

[1]École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[2]Ericsson Canada Inc., Mississauga, ON L4W 5E3, Canada

Corresponding authors: Christian Miranda (ch.miranda.m@hotmail.com) and Georges Kaddoum (georges.kaddoum@etsmtl.ca)

**ABSTRACT** The 6LoWPAN (IPv6 over low-power wireless personal area networks) standard enables resource-constrained devices to connect to the IPv6 network, blending an IPv6 header compression protocol. For this network technology, a new routing protocol called Routing Protocol for Low Power Lossy network (RPL) has been designed. The latter is a lightweight protocol that determines the route across the nodes based on rank values. This protocol is known to be non-resilient against Rank attacks, which aim at creating non-optimized routes for packet forwarding, hence overwhelming the constrained 6LoWPAN. With 5G, Software-Defined Networks (SDNs) have been developed to facilitate simple programmable control plane, Quality of Service (QoS) provisioning, and route configuration services for 6LoWPAN. However, there is still a lack of optimization mechanisms to protect 6LoWPAN against Rank attacks in SDN-based deployment. To this end, in this paper, a Reinforcement-Learning (RL) agent is leveraged to assist and complement an SDN controller in achieving cost-efficient route optimization, and QoS provisioning packet forwarding to prevent rank attacks. Experimental results confirm that our approach effectively prevents Rank attacks while providing an adequate delay and radio duty cycle. Meanwhile, it maximizes the packet delivery ratio, facilitating practical implementations in software-defined Low Power Internet of Things (IoT) networks.

**INDEX TERMS** Reinforcement learning, SDN networks, 6LoWPAN networks, RPL protocol, rank attacks.

## I. INTRODUCTION

Wireless sensor networks (WSNs) are considered one of the most important applications of the Internet of Things (IoT) [1]. In general, WSNs can be considered as Low Power and Lossy Networks (LLNs), presenting some constraints on their deployment, especially in critical and large-scale scenarios (e.g., massively distributed, and heterogeneous networks). The resource-constrained limitations prevent the deployment of WSNs in scenarios where the operation is subject to strict reliability and performance requirements. At the same time, the lack of flexibility stems from the rigidity of WSNs towards policy changes, making these networks difficult to adapt. Internet Protocol (IP) considerably brings direct and bidirectional access to devices reducing the mentioned difficulties, but some issues emerge concerning interconnections' complexity.

In WSNs, IP networks aim to provide end-to-end communication, which allows devices to be accessed without the necessity for gateways to use adaptation techniques to boost the efficiency and quality of wireless transmissions [2]. In this context, the 6LoWPAN standard uses IPv6 addresses eliminating adaptation techniques [3]. Moreover, 6LoWPAN is a network standard that defines header compression mechanisms and encapsulation rather than being an IoT application protocol technology (e.g., Bluetooth, ZigBee [4]). Nevertheless, due to common factors, such as node failure, limited bandwidth, etc., the wireless links in multihop

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

6LoWPAN are unstable, and therefore not reliable. These difficulties can severely impact the performance of the entire network [5], [6]. IP-based networks adopt distributed protocols (eg., Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Routing Information Protocol (RIP) for routing decisions and preserving topology while decreasing the overhead in the entire network [7]. Since low-power devices reduce the radio range compared to when all nodes communicate with a single base station, a multihop grid allows systems to extend over a larger area. Consequently, from the introduction of multiple hops, the link uncertainty is aggravated along the hop distance and may increase the possibility of dropped packets on the way. Specifically, RPL is a protocol based on rank values that rely on an Objective Function (OF) to determine the route across the nodes [8]. An OF defines how an RPL node selects and optimizes routes to build a Destination Oriented Directed Acyclic Graph (DODAG) rooted at the network's border router. Further, the OF defines how the nodes should consider the metrics and constraints in the rank value, which is roughly the node's distance to the DODAG root. Even though the rank values in RPL help for multiple objectives, including route discovery and distribution, loops prevention, and control overhead management, this protocol is exposed to a wide variety of routing attacks (i.e., Sinkhole attacks, Wormhole attacks, Rank attacks [9]). These attacks can significantly impact resource utilization and the network performance [9]. Precisely, in rank attacks, malicious nodes broadcast messages to advertise lower ranks than their original ones to corrupt routing cost values, which forces neighboring nodes to choose them as a preferred parent and change their rank accordingly. Thus, Rank attacks create non-optimal routes and introduce loops that overwhelm the network resources and increase resource consumption [10].

With the arrival of 5G, Software-Defined Networks (SDNs) have been developed to introduce scalability and programmability to accomplish QoS provisioning and fast routing configuration services over the 6LoWPAN. It has shown promising advances in network configurability, virtual network functions plugin, and reduction in capital expenditure [5]. In this context, a Software-Defined 6LoWPAN wireless sensor network (SD6WSN) is proposed. This architecture aims to manage data plane forwarding in 6LoWPAN according to the SDN approach [11]. SD6LoWPAN has several positive aspects, including a centralized SDN architecture that allows flexibility and scalability, presenting further opportunities to move beyond the traditional notions of low-power IoT, driving from small to various networks connected across a network backbone and protocols such as 6LoWPAN, to dynamically serve multiple applications, such as data collection, actuation, and monitoring with varying QoS requirements. However, SD6LoWPAN faces considerable challenges such as the non-negligible overhead introduced by SDN devices caused by the continuous exchange of messages and the vast distances between the data plane and the controller and is likely to suffer from Single Point of Failure (SPoF). It is valid to note that the problem of SPoF is out of the scope of this paper; however, we will deploy our solution in a distributed architecture in future work, which helps mitigate the harmful effects of SPoF in SD6LoWPAN. Hence, a lightweight SDN controller is leveraged in the border router to promote northbound and southbound communication with the data plane and applications correspondingly and reduce the non-negligible overhead introduced by SD6LoWPAN [12]. Furthermore, the incorporation of RPL in the routing layer for network discovery and the lack of routing optimization procedures to optimize the routes defined by the RPL make SD6LoWPAN susceptible to Rank attacks. Hence, to tackle this concern, in this paper, we propose an RL approach for routing optimization to prevent Rank attacks in SD6LoWPAN.

### A. MOTIVATION

The motivation behind this work is the computational complexity of managing security solutions and the sample complexity of finding the right approach for routing optimization to prevent Rank attacks in SD6LoWPAN. In this vein, the centralization of security controls in SD6LoWPAN facilitates the network (re)-configurability and network slicing which allow resource sharing and the adoption of complex solutions in a multitenant environment where a single instance of an application and its supporting resources serve multiple providers [5]. However, the programmable nature of SDNs increases the network's vulnerability to attacks [6], as applications can be easily installed.

Accordingly, in SD6LoWPAN, authentication and intrusion detection mechanisms are mainly implemented on the IoT nodes [13], [14], while RPL can be performed at the controller or application-level [15]. Moreover, the massive deployment of RPL-based low-powered IoT devices makes SD6LoWPAN more vulnerable to rank attacks. Hence, the RPL is vulnerable to internal Rank attacks taking advantage of the vulnerable rank property defined by non-optimal routes established by the OF. Consequently, these attacks jeopardize the network performance, topology, and traffic [16]. Illustratively, an attacker can accomplish this attack by misusing the rank property and infringing the routing protocol. Based on the vulnerability analysis related to the rank property, Rank attacks create non-optimal paths for all packets, which pass through malicious nodes and overwhelm the restricted SD6LoWPAN [17].

Meta-heuristic algorithms, such as Ant Colony Optimization (ACO), Swarm optimization, and artificial bee colony, are practical and widely used approaches to find solutions to combinatorial optimization problems [18], [19]. However, they are limited by the high sample complexity required to reach a reasonable solution. The sample complexity represents the number of training samples that an algorithm needs to learn a target function successfully [20]. Also, much work has been done in the field of machine learning for routing optimization, but these methods can require an unreasonably large number of samples before a good policy is obtained. Precisely, the lack of exploration in these methods leads to an unreasonably large sample complexity, which is unrealistic for dynamic environments [21]. In this context, RL seems to be a more promising and realistic solution compared to traditional machine learning approaches as it relies on an RL agent that explores and interacts with its environment to generate its own training data. To this end, in this paper, we incorporate an RL approach in the lightweight SDN controller design to achieve routing optimization and QoS provisioning packet forwarding to address the vulnerable rank value and the RPL objective functions' weaknesses while minimizing the overhead and management complexity introduced by SD6LoWPAN.

### B. RELATED WORK

A comparison between some current research works and the proposed Software-Defined Reinforcement Learning (SDRL) scheme is presented in TABLE 1. Hence, some research works have looked into management complexity, overhead reduction, and security solutions to address Rank attacks in resource-constrained 6LoWPANs. Precisely, in [22], a software-defined networking framework for IoT based on 6LoWPAN is presented to reduce the management complexity in IoT networks. Further, in [12], a lightweight SDN framework for Contiki OS is introduced to reduce the control overhead to practical levels. Moreover, in [23], a QoS-aware Adaptive Routing (QAR) based on RL with a QoS-aware reward function is introduced for multi-layer hierarchical SDNs achieving time-efficient, adaptive, and QoS-provisioning packet forwarding. These approaches reduced the management complexity in 6LoWPAN but at the cost of increasing the overhead introduced by the software-defined approach. In [24], the authors present a combination of the IoT with a heuristic framework to enhance logistics while reducing the overhead in the agri-food supply chain. In [25], the authors propose an improved objective function that relies on an RL-based link quality estimation strategy for RPL to minimize the overhead caused by active probing operations. These approaches reduced the overhead in 6LoWPAN but at the cost of increasing the management complexity by incorporating heuristic and RL approaches in the resource-constrained 6LoWPAN. In [26], the authors propose a hash chaining using a random number chosen by the root node to avoid the RPL from publishing an illegitimate reduced rank. Moreover, in [27], a challenge-response scheme is proposed to validate the nodes' authenticity within a DODAG, in [8], a cost-efficient protocol for route optimization is introduced, where the authors include steps for reliable route optimization and mutual authentication. Further, an enhanced RPL protocol is proposed in [28], where a rank threshold approach and the hash chain authentication technique are proposed to deal with RPL-based attacks. Although these approaches address the prevention of Rank attacks, they also introduce management complexity and a considerable increase in the overhead of 6LoWPAN.

**TABLE 1.** Related works comparison.

| Solution | OR | LMC | IPS |
|---|---|---|---|
| [12],[22],[23] | ✓ | ✗ | ✓ |
| [25] | ✓ | ✗ | ✓ |
| [26] | ✗ | ✗ | ✗ |
| [27] | ✗ | ✓ | ✓ |
| [28] | ✓ | ✗ | ✓ |
| [29],[30], [31] | ✓ | ✓ | ✗ |
| [32],[33] | ✗ | ✗ | ✗ |
| SDRL | ✓ | ✓ | ✓ |

Moreover, some research works have looked into reducing the overhead and management complexity of the RPL objective function in resource-constrained 6LowPANs. In particular, [29], emphasizes the quality of service differentiation by exploiting multi-topology routing feature of the RPL standard. A novel Policy Gradient-based Actor-Critic Learning (PGACL) algorithm to optimize the policy gradient for optimal rate allocation, minimize power, and guarantee a solution for Ultra-reliable and Low-latency Communications (URLLC) scheduling is proposed in [30]. Furthermore, in [31], the authors propose a Generative Adversarial Network and Deep Distribution Q Network (GAN-DDQN) to enhances smart packets by reducing the distance between the estimated and target action-value particles.

In addition, some works propose run-time verification mechanisms to detect unexpected behavior in IoT system nodes. These mechanisms monitor the real-time events coming from the IoT system elements and trigger self-healing actions if unexpected behavior is detected at an IoT device. For instance, in [32], the use of complex event processing techniques for detecting failures in the system is proposed by monitoring the run-time event occurrences with regard to the system constraints denoted by event calculus. In [33], a run-time monitoring approach for IoT systems is presented where the event relations expressed in terms of the sequential interaction messaging model of Constrained Application Protocol (CoAP) are explored. Nevertheless, while this technique helps in detecting IoT nodes' misbehavior; it also introduces an overhead due to the recurrent monitoring system installed at each DODAG node. Furthermore, this technique does not prevent 6LoWPAN from being compromised by a Rank Attack as our solution does. This is because a rank attack alters the assigned rank value but does not change the node's behavior, hence overloading the resource-constrained network. Indeed, the attacker's main objective is to overload the network using the behavioral patterns of the nodes in an RPL network. Although essential works have been proposed in the literature to target management complexity, overhead reduction, and Rank attacks in 6LoWPAN, all these deployments are not satisfactory to simultaneously guaranteeing an efficient Intrusion Prevention System (IPS), Low Management Complexity (LMC), and considerable Overhead Reduction (OR) in 6LoWPAN [34].

### C. CONTRIBUTION

In this paper, a security scheme for preventing Rank attacks in SD6LoWPAN is designed as shown in Fig. 1. The novelty of the proposed work lies in devising and

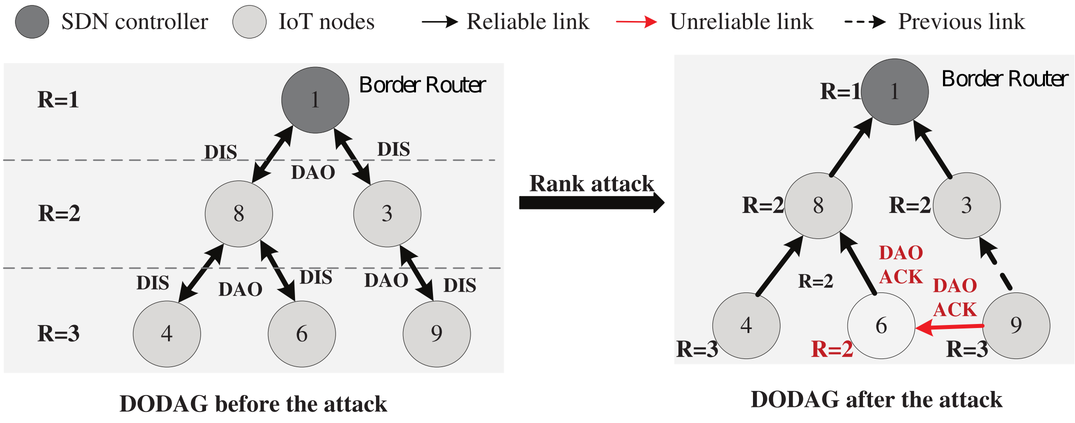


**FIGURE 1.** DODAG instance before and after Rank attack.

evaluating an intrusion prevention scheme that amalgamates SDN applications in the control plane, achieving efficient topology discovery, flow control management, and route optimization in SD6LoWPAN. The RPL-based topology discovery service is deployed to segment the SD6LoWPAN and create the route tables used for the topology optimization service. Subsequently, a coordinator flow control application is developed to coordinate the communication between the application, control, and data planes. Further, an RL-enabled topology optimization, achieving route optimization in SD6LoWPAN, is designed. It is worth mentioning that nodes' authentication and integrity are out of the proposed work scope. Thus, the main contributions of this work are summarized as follows:

1) A lightweight SDN controller is leveraged in the border router to reduce the non-negligible overhead introduced by SD6LoWPAN.
2) A coordinator flow control application is integrated into the SDN controller to handle the interaction between the layers of SD6LoWPAN.
3) In the SDN controller, northbound and southbound interfaces are enhanced to facilitate the communication between the SDN controller and the data plane and applications.
4) An RPL-based topology discovery application is employed for network discovery from the IoT nodes towards the SDN controller.
5) An RL approach is developed in the SDN controller to optimize the RPL routing paths to prevent Ranking attacks' harmful effects in SD6LoWPAN.
6) Moreover, analysis of the duty cycle and computational complexity are provided, while emulations showing the effectiveness of the proposed scheme are executed by leveraging the Contiki Cooja tool.

The remainder of this paper is organized as follows: Section II introduces the background and network model. In section III, we describe the RL-based intrusion prevention scheme, which falls into: the stack scheme, intrusion prevention algorithm and RL agent modeling and inner workings. In section IV, the emulation setup and the experimental results are conducted. Finally, the paper is concluded in section V, where future endeavors are also put forward.

## II. BACKGROUND AND NETWORK MODEL

In this section, we provide a brief background on RPL, Rank attacks, and RL. Further, we present the impact of Rank attacks on SD6LoWPAN and the considered network model. The basic concepts underlying the proposed scheme are detailed in what follows.

### A. RPL OPERATION

RPL is an IPv6 routing protocol designed and standardized by the Internet Engineering Task Force (IETF) [27]. To build the network topology, RPL employs Directed Acyclic Graphs (DAGs), which can be segregated by one or more DODAGs, where each DODAG has a root node. Multiple root nodes are integrated within a backbone network that consists of border routers that connect them to the internet. RPL is a routing protocol for wireless systems with low power consumption that starts to find routes based on the OF established in a setup stage. The OF is utilized to deliver traffic to different routes according to traffic requirements. The OF encoded these traffic requirements to be used by the RPL during routing operations. RPL applies three types of control messages, i.e., DODAG Information Object (DIO), DODAG Information Solicitation (DIS), and DODAG Advertisement Object (DAO), as shown in Fig. 1. The root node multicasts DIO messages at regular periods defined by a trickle algorithm [35]. The DIO message gives the IoT nodes information to explore the DODAGs, acquire the setting parameters, and select the favored parent set. To choose the parent set, RPL applies the OF, which contains some routing metrics [36]. A DODAG uses DIS message to request the DIO from its neighboring node to join the DODAG. DAO messages are disseminated by the IoT nodes to the root node to update the DODAG. Thus the composition of the DODAG topology is supported by the root node. The RPL operations include topology discovery, DAG construction, route generation, data path validation, and loop detection based on rank values [37].

A rank value defines the relative position of a node within the DODAG. The 6LoWPAN has unique characteristics that require the specification of new routing metrics and constraints [38], which can be used by the RPL in the path computation. These metrics/constraints can be categorized into two basic types:

- Node metrics and related constraints (e.g., hop counts, energy state.),
- Link metrics and related constraints (e.g., throughput, latency, packet loss).

### B. RANK ATTACKS

The OF is an essential factor in the parent's selection, along with the rank. Once a node receives a valid rank, the OF's setting based on the routing metrics must be determined before modifying the selected parent node. For instance, if the routing metric relies on the Expected Transmission Count (ETX), the OF is determined to hold the routing path with the lowest ETX value, and a node will receive both the rank and ETX for the chosen parent node. Mainly, to successfully originate a Rank attack, the attacking node must alter the routing metric advertised by the parent node so that the of the neighboring nodes is exposed to be attacked.

In this regard, Rank attacks have raised serious concerns about the weakness of the objective function of the RPL. This protocol usually implements two objective functions: the Minimum Rank with Hysteresis Objective Function (MRHOF) and the Objective Function Zero (OF0). The OF0 constructs a DAG with the lowest number of hops [39], while MRHOF creates a DAG considering the lowest ETX to select the best path [40]. Since the existing OFs take into account only one [38] or two metrics [41], the DODAGs cannot fully satisfy some recent applications which require several QoS constraints such as packet loss, duty cycle, and end-to-end delay [40]. For example, OF0 chooses the shortest path; however, it does not necessarily ensure the end-to-end delay requirement, which is an essential constraint for interactive applications [39]. Furthermore, in the MRHOF, the objective function aims to minimize the expected total number of packet transmissions required to deliver a packet to the ultimate destination successfully [40].

It is worth mentioning that a DODAG only uses one OF for its formation and maintenance. For instance, to illustrate a Rank attack, in this paper, we consider the ETX as the principal routing metric for a network topology creation. We account for an attacker node with a legitimate rank $R_l$. In addition, we consider $R_n$ to be the minimum rank between the neighbors. In this example, the attacker node will promote a rank value of less than $R_n$ to launch the attack. Consequently, the attacker alters his rank to become less than $R_n$, where $R_a < R_n$ is the rank announced by the attacker $R_a$. Thus, the attacker's neighbors will drop the rank value if the announced rank $R_a$ is too low because the RPL recommends that the rank setting is within a threshold. Otherwise, the unexpected rank can induce unstable network topology. Accordingly, in Rank attacks, the attacker advertises a rank with the ratio $R_p < R_a < R_n$, where $R_p$ is the attacker's preferred parent node rank.

In this vein, the updated rank advertised by the attacker is smaller than most neighboring nodes [42]. Also, to boost the severity of the attack, the ETX advertised in the DIO message is diminished compared to the minimum observed between neighbors. In real 6LoWPAN, routing metrics are subject to more variations than the rank; therefore, RPL does not propose any measures to control the routing metric values. As depicted in Fig. 1, the neighboring nodes of the attacker (compromised) node six select the latter as their new preferred parent because it changes its rank from R=3 to R=2 and the ETX announced in the DIO message is lower than the minimum perceived between neighbors. As a result of such ranking misuse, new non-optimal links are considered (depicted through red lines in Fig. 1), which impacts the network performance implicitly.

### C. REINFORCEMENT LEARNING

RL is an area of machine learning that allows an agent to learn in an interactive environment by trial and error using feedback from its actions and experiences [23]. Specifically, it addresses how an agent/decision-maker tries to learn the dynamic system's behavior through interactions with the environment. The agent receives the current state and the reward from the dynamic system at each iteration and takes an action that increases the long-term revenue. The agent obtains the state and the system's reward values, whereas the system captures the action as an input from the agent [43]. RL can increase automation or optimize sophisticated systems' operational efficiency, e.g., networking, robotics, manufacturing, and supply chain logistics [44]. However, in RL's practical implementations generally, we do not have information on the subjacent model. In such a scenario, model-free learning algorithms are more suitable. The most widely used approaches in this area are Monte Carlo (MC) and Temporal Difference (TD) learning. While MC learns directly from episodes of experience without any previous knowledge of Markovian decision Process (MDP) transitions, TD learns by bootstrapping from the current estimate of the value function [44].

### D. RANK ATTACK IMPACT

SD6LoWPAN defines a controller that communicates with the data plane through a Software Defined 6LoWPAN Wireless Sensor Network Protocol (SD6WSNP), that employs IPv6 and RPL at the routing layer, UDP at the communication layer, and CoAP at the application layer [5]. SD6WSNP uses CoAP messages to send rules dictated by SDN applications, such as wireless link quality, geolocation, and power transmission level, to the nodes. Consequently, RPL creates DODAGs of different sizes (hops) stored in flow tables for forwarding data plane packets. Therefore, when a Rank attack is performed, the DODAGs communicated by RPL with the SDN controller contains a non-optimal set of paths. As a consequence, these non-optimized paths impact the routing messages between the nodes and the SDN controller. They also affect the routing rules of the messages exchanged between the nodes in the data plane; thus, they overwhelm the SD6LoWPAN. It is worth mentioning that this work focuses on the messages between the nodes and the SDN controller in the experimental results.

### E. NETWORK MODEL

As depicted in Fig. 2, the network model in the proposed scheme is a typical SDN-based network architecture where, in the data plane, multi-hop low-power IoT nodes, connected by IPv6 to the Internet through a gateway (or border router), are deployed. These nodes are characterized by low power, low data rate, short radio range, and low cost. The control plane then consists of a lightweight SDN controller at the border router that makes decisions about where traffic is sent from the underlying data plane to selected destinations with a coordinator flow control. Precisely, a lightweight SDN controller is used to minimize the signaling delay in traditional SDNs. Finally, at the top of the architecture, the application plane is designed to discover the network and optimize the topology in Low Power IoT Networks. The proposed stack scheme is presented in the following section.

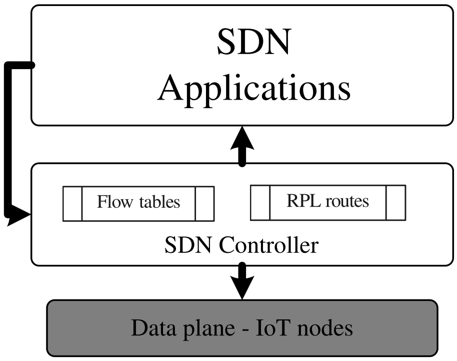


**FIGURE 2.** Network Model.

## III. PROPOSED SCHEME

We propose a scheme that creates an enhanced version of the architecture concepts proposed in [12] while incorporating architectural, protocol, memory, and controller optimization to mitigate control overload and improve scalability. Precisely, it takes advantage of a northbound Application Programming Interface (API) to facilitate the control plane's communication with the SDN core applications and a southbound API to facilitate the control's communication with the data plane. The interaction between the northbound and southbound APIs is handled by the coordinator flow control, located in the control plane, as shown in Fig. 3.

It is worth mentioning that the proposed approach does not detect or eliminate the attacker node nor be an RPL replacement. Instead, this work devises an RL-based intrusion prevention system against RPL Rank attacks' harmful effects through route optimization for low-power IoT networks. In summary, the proposed stack scheme incorporates three layers as follows. At the bottom of the stack, typical IoT nodes in the data plane combine the following communication functions: data plane forwarding, border routing, and sensing applications. In the middle, the control layer, the coordinator flow control is executed. At the top of the application layer, the SDN core applications (topology discovery and topology optimization) are integrated. This architecture is presented in Fig. 3 and is fully integrated with the IEEE 802.15.4-2012 protocol stack.

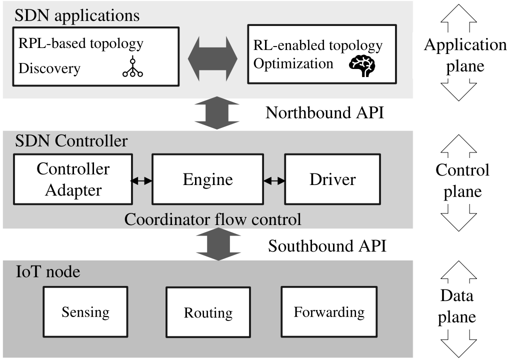


**FIGURE 3.** Network's stack scheme.

### A. SDN DATA PLANE

In the data plane, the low-power IoT network is executed. Due to the small packet size and low bandwidth, the SDN data plane requires resource-saving to maximize the lifetime of IoT nodes. The low-power IoT node has at least three components: the data plane forwarding, routing agent, and sensing components, which use large arrays of sensors to collect data from a particular environment. The IoT network interacts with the control layer through control messages. The main functions of the SDN IoT nodes are:

- Send information to the control plane;
- Examine data plane packet headers;
- Send or deny data plane packets according to matching entries in the flow table;
- Send packet-in notifications to the control plane when there is no matching entry.

Moreover, the SDN data plane modifies packet forwarding at the operating system as follows. The routing agent inspects packet headers and checks if the incoming packet is a control message. If the control message requests an RPL discovery, it is routed through the control plane following the 6LoWPAN-RPL routing standard. If the control message contains a flow table instruction, it is routed through the data plane. Otherwise, the packet is delivered to the local sensing application to perform the data plane forwarding function [45].

### B. SDN CONTROL PLANE

To provide a platform for SDN experimentation in low-power IoT networks, we have implemented a lightweight SDN control plane introduced in [12]. The lightweight SDN controller provides a coordinator flow control and enhanced southbound and northbound APIs, detailed as follows.

#### 1) COORDINATOR FLOW CONTROL

In the control plane, a coordinator flow control is developed to facilitate the communication in the proposed SD6LoWPAN architecture. In this context, the coordinator flow control integrates three SDN functions that rely on [12] to handle specific requirements of the SDN implementation. A brief explanation of the SDN functions is detailed below.

##### *a: SDN CONTROLLER ADAPTER*

The controller adapter exposes a controller interface to the SD6LoWPAN architecture, allowing the control plane to implement third party interfaces.

##### *b: SDN DRIVER*

The SDN driver determines how to manage the flow table. It provides high-level functions to accomplish particular tasks by setting up flow table entries, such as aggregating or removing flows, setting routing paths through the network, and creating security policy entries. It also handles flow table actions and determines how and when nodes communicate to the controller with specific rules.

##### *c: SDN ENGINE*

The SDN engine defines the northbound and southbound communication (application plane with the controller and the control plane with the data plane, respectively) for both incoming and outgoing messages to the controller.

#### 2) SOUTHBOUND API

The coordinator flow control utilizes a southbound API to ensure that packets are transported through the User Datagram Protocol (UDP) to enable a secure DTLS (Datagram Transport Layer Security) and provide better communication between the data plane and the controller. Also, this API ensures that each node's information is continuously sent to the controller. To this end, the API employs control messages.

#### 3) NORTHBOUND API

The coordinator flow control uses a northbound API to allow the communication of the SDN controller with the application plane. To this end, the northbound API employs control messages that are encapsulated in TCP packets. This API continuously updates the routing table's contents with the routes built by RPL and registers the optimized routes in the SDN flow table with the RL agent's decisions. The control messages implementation, which dictates how the data and application planes handle controller communication, is explained as follows.

#### 4) CONTROL MESSAGES IMPLEMENTATION

In the control plane, the coordinator flow control determines four control messages, i.e., *node-mod*, *info-get*, *flow-mod*, and *packet-in*. Accordingly, the control messages are categorized depending on the process with which they are associated. *Node-mod* and *info-get* are utilized for topology discovery and optimization applications, while *flow-mod* and *packet-in* are employed for flow control. These messages operate depending on the SDN core applications' demands. Initially, as shown in Fig. 4, the northbound API initiates a node-mod message from the RPL-based topology discovery application to the control plane to determine the network topology, requesting a notification every time a new node is identified. Once the notification is received, the control plane transmits info-get through the southbound API to obtain the discovered node's neighbors and the respective wireless links' quality.

Subsequently, the northbound API records the RPL routes in a routing table. After that, the RL-enabled topology optimization is executed, optimizing the routes based on the data collected from the topology discovery-based RPL application. Consequently, the northbound API registers the optimized routes in the SDN flow table. Afterward, the coordinator flow control sends an Info-get message to the data plane to instruct the nodes to send back a notification when they receive packets that do not match any entry in the flow table.

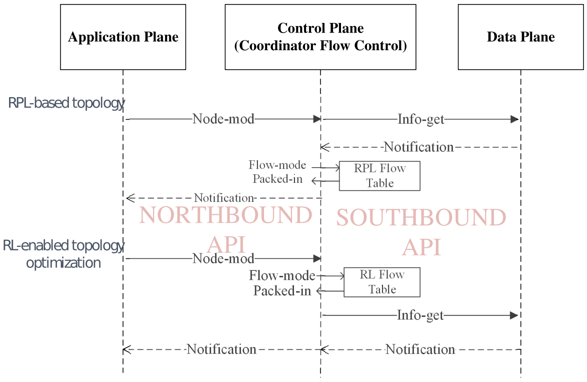


**FIGURE 4.** Control message sequence diagram.

It is worth mentioning that the Flow-mod message is used to insert and remove entries from flow tables, to establish flows according to the SDN application's purpose. The SDN core applications can also request the controller to send an info-get whenever the applications need information from the IoT nodes. The control message structure is summarized in TABLE 2. Accordingly, our approach's flow tables are composed of two fields, i.e., match and action. The match field records the incoming packet header's features distinguishing the corresponding flow, whereas the action field records the operation for a matching packet, as illustrated in Table 3 [5]. Moreover, a set of attributes related to the standard SDN flow table entries is included in the match field.

### *C. SDN APPLICATION LAYER*

In the SDN application layer, the core applications are performed. The SDN engine performs the integration with the SDN controller.

#### 1) RPL-BASED TOPOLOGY DISCOVERY

Since, the SDN controller needs to have a unified view of the SD6LoWPAN and the neighbors that each node sees,

**TABLE 2.** Control messages.

| Message | Method | Observe | Process |
|---|---|---|---|
| node-mod | GET | Required | Topology discovery, Topology optimization |
| info-get | GET | Optional | Topology discovery, Topology optimization |
| flow-mod | PUT | Required | Flow control |
| packet-in | GET | Required | Flow control |

including the quality of the wireless links connecting them, a network discovery protocol is mandatory. In this context, this scheme employs the RPL protocol in non-storing mode. In this mode, the routing table entries are maintained only on the controller to ensure that the IoT nodes always attempt to find a path to the controller.

#### 2) RL-ENABLED TOPOLOGY OPTIMIZATION

In this work, we consider a route optimization approach that aims to find an adaptive QoS-aware forwarding policy by applying an RL technique to allow each node to learn the proper forwarding rate to cooperate in the routes optimization process. This application's main objective is to minimize energy consumption, packet delivery ratio, and end-to-end delay caused by the Rank attack, thus preventing the latter from overwhelming the SD6LoWPAN.

**TABLE 3.** Flow table entry match fields.

| Match | Length(Bits) | Description |
|---|---|---|
| ipv6src | 128 | IPv6 source address |
| srcmask | 8 | Mac source address) |
| ipv6dst | 128 | IPv6 destination address |
| dstmask | 8 | Mask destination address |
| srcport | 16 | Source port |
| dstport | 16 | Destination port |
| ipproto | 8 | IP protocol (UDP, TCP, ICMPv6) |

### *D. RL MODEL*

The RL model consists of two main entities: the agent and the environment, as shown in Fig. 5. The agent is a quick learner who can make decisions according to its learning experiences and the environment is an anonymous entity that affects the performance of the agents. In the proposed solution, the agent lacks knowledge of the environment. Therefore, a model-free like State Action Reward State Action (SARSA), a well-known temporal difference (TD) algorithm, is adopted [23]. SARSA is an iterative dynamic programming algorithm to find the optimal solution based on a limited environment. It is worth mentioning that SARSA has a faster convergence rate than Q-learning and is less computationally complex than other RL algorithms [46]. Also, since our environment is resource-constrained and limited by the number of nodes per DODAG (30), different deep reinforcement learning algorithms such as Deep Q-Learning (DQL) and Deep Deterministic Policy Gradient (DDPG) are not considered in this paper, where we leave their integration in our scheme and test in a real IoT testbed for future work.

In particular, in the proposed scheme, the state is the current node, and the action is the link to follow to reach a neighboring node. Specifically, at each node, following the link to each neighbor, the agent has to exploit past actions with great rewards and simultaneously explore the system for better unknown actions. In this context, there are three components for the RL agent's design: the action policy, the quality function, and the reward function. These components are detailed as follows:

#### 1) ACTION SELECTION POLICY

The action selection policy defines an agent's action selection, which correlates an action to a state. This function evaluates the trade-off between action exploitation and exploration to maximize the reward value. Accordingly, the agent explores the state space in an unknown environment. To this end, in our proposed routing model, we consider the Boltzmann softmax policy [47], where the probability $\pi_t(s_t, s_a)$ of choosing an action $a_t$ given the current state $s_t$ is given by

$$\pi_t(s_t, s_a) = \frac{\exp(Q_t(s_t, a_t)/\tau_n}{\sum_{b=1}^{n} \exp(Q_t(s_t, b_t)/\tau_n)}, \tag{1}$$

where $n$ is the number of possible actions, $Q_t(s_t, a_t)$ is the corresponding quality function, and $\tau_n$ is a temperature control. The temperature control measures the trade-off between exploration and exploitation. As a result, if this parameter obtains high values, all actions are reasonably probable (i.e., exploration). In contrast, low values sustain the action with the maximum quality (i.e., exploitation), which causes the policy to tend to a greedy one. Therefore, in highly dynamic environments $\tau_n$ should be set to a high value while it should decrease to a low value in static environments. In this context, to guarantee a learning convergence in a limited time, temperature control is set to a linear function of the time and is expressed by

$$\tau_n = -\frac{(\tau_0 - \tau_T)n}{T} + \tau_0 \quad n \leq T, \tag{2}$$

where $T$ denotes the time to reach the convergence, and $\tau_0$ and $\tau_T$ are the initial and last value at time $T$ of the temperature control, respectively.

#### 2) QUALITY FUNCTION

The quality function estimates the quality that can be achieved by the possible next system state, which can be determined by the agent based on the state and action. Significantly, in this paper, the quality function $Q_{t+1}(s_t, a_t)$ relies on SARSA, as mentioned above, where the agent at time $t+1$ applies the action and the state to update the quality value. Indeed, SARSA uses the expected quality value, taking into account how likely each action is under the current policy, which indicates that the agent can utilize the future reward earned, rather than considering the optimal action with the highest reward [48] as follows:

$$\begin{aligned} &Q_{t+1}(s_t, a_t) \\ &\quad = Q_t(s_t, a_t) + \alpha[R_t + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)], \end{aligned} \tag{3}$$

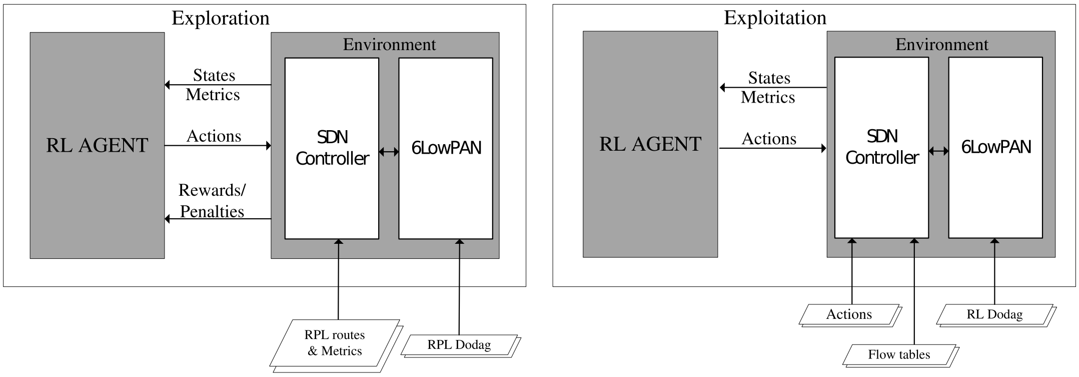


**FIGURE 5.** RL model.

where $\gamma \in [0, 1]$ is the discount factor that defines the purpose of future rewards, $\alpha \in [0, 1]$ is the learning rate that represents the override measure of the recently acquired information to the past one, and $R_t$ is the reward at time $t$. As a consequence, in Eq. (3), the agent updates the quality value based on the maximum potential quality value among the actions. Concretely, the agent selects and takes action for the current state $s_t$ through the action selection policy. Accordingly, the agent observes $R_t$ and the state $s_{t+1}$ and updates the $Q$ function.

### 3) REWARD FUNCTION

In this section, we recommend a reward function based on the network QoS requirements that are linked with the design of our route optimization approach. Specifically, the RL agent discovers the routing path with the highest QoS-aware reward based on the types of traffic and user applications.

Precisely, TABLE 4 summarizes the QoS requirements and traffic type of several applications [49]. For example, classic and real-time traffic adapts the packet transmission rate and has significant QoS awareness. For this purpose, the reward function is evaluated as

$$R_t = -g(a_t) + \beta_1(\text{ delay}_{i,j} + \text{queue}_{i,j}) + \beta_2 \text{PLR} \qquad (4)$$

This indicates that the system at state $s_t$, using an action $a_t$, forwards packets from node $i$ to node $j$. In Eq. (4), $g(a_t)$ indicates the cost to take action at time $t$, and $\beta_1, \beta_2 \in [0, 1]$ are the weights values determined by the QoS requirements of the packet flow. Unfortunately, one of the significant concerns with RL algorithms is that, as the agent iterative estimates the action values, the initial stages' learning process is extensively random exploration, which might affect the network performance.

Therefore, since this work's primary purpose is to prevent Rank attacks from overwhelming the performance of SD6LoWPANs, we introduce an exploration strategy that incorporates QoS aware functions in the action selection process to guide the learner agent, especially in the initial stages of the learning process [50], avoiding excessive consumption of resources.

Since the impact of doing an action mainly relies on the QoS aware functions, the cost $g$ is equal to a constant value over all the actions. The QoS provisioning functions are defined as

**TABLE 4.** Traffic types and applications for QoS requirements [49].

| Traffic type | Application | QoS |
|---|---|---|
| Classic | Telnet, FTP<br>HTTP, FTP<br>STMP,POP3,IMAP<br>TELNET | Delay, losses<br>Delay, Throughput<br>Losses<br>Losses |
| Real-time | Multimedia<br>Control messages | Delay, Throughput<br>Delay |

$$\text{delay}_{i,j} = \frac{2}{\pi} \arctan \left[ d^l_{i,j} - \frac{\sum_{k=1}^{A(i)} d^l_{i,k}}{A(i)} \right] \qquad (5a)$$

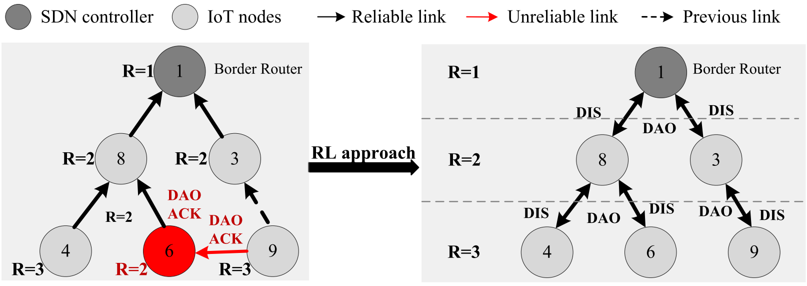
$$\text{queue}_{i,j} = \frac{2}{\pi} \arctan \left[ d^q_{i,j} - \frac{\sum_{k=1}^{A(i)} d^q_{i,k}}{A(i)} \right] \qquad (5b)$$

$$\text{PLR} = (100 - \text{PDR}) \qquad (5c)$$

where $d^l_{i,j}$ and $d^q_{i,j}$ are the link transmission and packet queueing delays from node $i$ to node $j$, respectively. $A(i)$ is node $i$'s number of neighbors in the DODAG, and *PLR* characterizes the packet loss from node $i$ to the controller. Eq. (5a) estimates the link delay of link $i - j$ compared to other possible next hops, Eq. (5b) includes the queueing delay while accounting for the average delay over the DODAG, and Eq. (5c) represents the Packet Loss Ratio (PLR), which shows the performance of the protocol in terms of percentage of Packets Delivery Ratio [PDR], i.e., the packets successfully delivered to the controller [51].

## *E. INTRUSION PREVENTION ALGORITHM*

As shown in Fig. 6.a, we consider a DODAG which consists of several IoT nodes connected to a border router that plays the role of a lightweight SDN controller. In this reference frame, the SDN controller gathers the routing paths and the global state of the network with the aid of RPL. Consequently, we assume that a Rank attack is performed over an existing node in the network, affecting node 6, which alters its rank from R=3 to R=2, and the ETX announced in the DIO message that is lower than the minimum perceived between neighbors. Hence, node 9 selects node 6 as its



a.) DODAG before our RL approach

b.) DODAG after our RL approach

**FIGURE 6.** DODAG instance before and after the proposed RL approach.

parent instead of node 3, affecting the entire network's performance. Subsequently, the SDN controller is in charge of path computation based on the network state received for each incoming route path. After that, a new DODAG is created with an optimized path based on the continuously received control messages, where node 3 is selected instead of node 6 as the parent node to node 9. Hence, through our RL approach, the controller dynamically optimizes the best data flow routes according to the QoS requirements and dynamic traffic patterns, and sets up the routing tables of the border router along the optimal path via the SDN controller, thus enabling high security while providing efficient data transmissions and superior link utilization [52], [53], [54].

It is worth mentioning that the resulting optimized paths could be different from those of DODAG without the Rank attack before applying our RL approach. In Fig. 6.b, we assume a representation of the possible optimized paths recovered by our RL approach. However, in the experimental results section, we validated the optimization of DODAG paths based on some performance metrics analysis. The intrusion prevention algorithm is summarized in the following steps.

1) Flow $f$ arrives to the controller $C_f$;
2) Set of paths and $NS$ are in introduced in $C_f$;
3) QoS requirements are configured in $C_f$;
4) The QoS functions are calculated in $C_f$;
5) $C_f$ executes the RL agent procedure;
6) Optimized paths are stored in the flow table;
7) The flow is forwarded following the flow table in $C_f$;

First, when a flow ($f$) appears at controller $C_f$, it demands the forwarding path, and the controller refreshes the current Network State (NS). Accordingly, the QoS requirements are configured in $C_f$ and the QoS provisioning functions are computed. Subsequently, $C_f$ exploits the RL agent procedure to select a possible path with regard to the QoS requirements of the flow. Consequently, $C_f$ stores the forwarding tables of the IoT nodes along with the optimized path in the flow tables. The RL agent procedure is summarized as follows:

1) Initialize $Q_0(S_0, a_0) = 0$ and $R_0$ from Eq. 3;
2) At time $t$:
3) Choose next-hop using softmax in Eq. 1;
4) Observe $R_t$ and $s_{t+1}$;
5) Update $Q_{t+1}$ function using Eq. 4;
6) update t = t+1;
7) Continue from step 3 to choose next-hop;
8) Exit;

## IV. EMULATION SETUP AND EXPERIMENTAL RESULTS

In this section, we start by presenting the scenarios, network parameters, assumptions, and metrics used in the evaluation. Afterward, we go through the experimental results and the corresponding analysis.

### A. EMULATION SETUP

We assume the composition of one DODAG with multiple sets of paths, one SDN controller, one control channel, and a real-time network state. To measure the impacts of rank attackers on the Operating System (OS), we choose an open-source OS, namely Contiki OS, designed for resource-constrained devices [55]. The benign nodes were placed at different locations and circled around a malicious node as shown in Fig. 6. Here node 1 is the SDN controller/border router in SD6LoWPAN. For the creation of DODAGs, we use RPL using the objective function MRH0F. Although MRH0F does not consider the number of hops for DODAG's design, since 6LoWPAN is a multi-hop network, we use the number of hops as a performance metric in our set of results. The network parameters used in the emulations are listed in TABLE 5, and node characteristics are mirrored according to the EXP5438 platform with TI MSP430F5438 CPU and CC2420 radio.

We measure the results of the experiment that are aligned with the data analytics of the Cooja emulation tool. The performance metrics are as follows.

- Average Packet Delivery Ratio (PDR): This is the ratio between the number of packets sent to the destination and the number of packets received by the destination.
- Average end-to-end Delay (Delay): This refers to the time to transmit a packet over the network from the source to the destination.
- Radio Duty Cycle (RDC): This is the energy consumed by an IoT node considering the time it spends in the listen, receive (Rx), and transmit (Tx) states. In other words, it is the ratio between the time spent by a node in those states and in wake-up state.

**TABLE 5.** Network Parameters.

| Parameter | Value |
|---|---|
| Emulation runtime | 3600s |
| MAC layer | ContikiMAC |
| Objective function | MRHOF, ETX |
| Number of IoT nodes | 30 |
| Transmitting nodes | All |
| Receiving node | Root/Controller |
| Link quality | 90% |
| Radio medium | UDGM |
| RPL mode | Non-Storing |
| Sending rate | 1 packet every 10 sec |
| Number of attacking nodes | 1 |
| TX range | 100 m |
| Interference range | 0-30m |
| Packet size | 50 Bytes |
| SDN update period | 180s |
| SDN flow table lifetime | 10min |
| Initial latency | 60 ms |
| Maximum number of hops | 5 |

### B. EXPERIMENTAL RESULTS

As part of this section, we analyze the results obtained from the experiments conducted using the Cooja emulator for Contiki OS which mimics the behavior of real IoT devices [56]. To this end, we consider a low-power wireless network composed of 30 IoT nodes where 29 are benign and 1 is deemed to be malicious. The network is deployed in an emulated outdoor area. Since our centralized approach's primary goal is to prevent rank attacks from overwhelming the network, given the fact that the network size consists of 30 nodes, one attacker is enough to demonstrate such a premise. It is worth mentioning that the results are obtained considering a static scenario in which there are no mobile nodes. However, we emphasize that our emulated scenario is deployed in a dynamic wireless environment. Thus, our testbed's radio channel conditions are susceptible to changes due to interference (e.g., from other 802.15.4 and 802.11 radios), where this interference is time-varying. Further, the underlying MAC protocol is ContikiMAC [57].

Moreover, it is essential to mention that the idea behind using an SDN controller in 6LoWPAN is to introduce intelligence and programmability to the root nodes, which eventually will be integrated into other DODAGs in the network. In other words, under a global IoT network scheme, there will be multiple DODAGs with multiple controllers communicating with each other. For experimental purposes, in this scenario, we use a single DODAG. In future work, we plan to analyze multiple DODAGs in a hierarchical SD6LoWPAN environment. To better illustrate the proposed solution's performance better, we first demonstrate the performance evaluation of our RL approach; afterward, a comparative scenario analysis is presented.

#### 1) PERFORMANCE ANALYSIS OF OUR RL APPROACH

Initially, we analyze our RL approach's performance to determine the best configuration settings to minimize the Delay and RDC while maximizing the PDR. To this end, we present the training phase as follows.

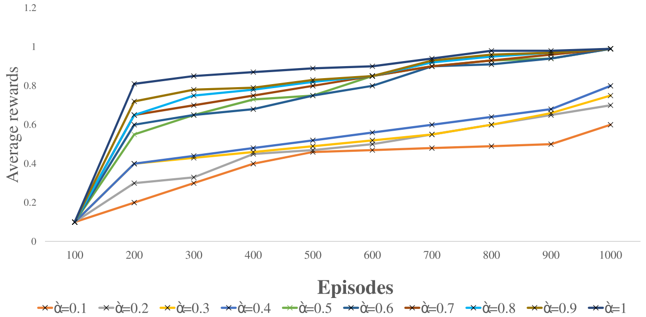


**FIGURE 7.** Learning process with respect to the number of episodes vs. the average reward in the proposed approach.

##### *a: TRAINING PHASE*

It is essential to mention that the RL model's training is made offline to record the first flow table; after that, the model delivers the optimized routes online. Since this work aims to prevent Rank attacks' harmful effects based on network QoS metrics continuously received by the control plane, to train the model, we select the step size parameters $(\alpha,\gamma)$. These parameters govern the RL agent's performance, as defined in Eq. (3). Precisely, $\alpha$ adjusts the error in the $Q$ value update; $\gamma \in [0, 1)$ takes a value of zero if the routing estimates the current reward and acts like a greedy algorithm, and a value close to one if the routing takes the long-term revenue. As we consider the long-term revenue to be significant, for this experiment, we set the value of $\gamma$ to 1. Moreover, we set the number of episodes to 1000, and each episode contains 100 steps.

The results shown in Fig. 7 demonstrate that our RL approach learns to reach a reward of 99% when the step size converges between 0.5 and 1 with a number of episodes of 1000. Hence, to analyze the Delay, RDC, and PDR, we vary the learning rate $\alpha$ from 0.5 to 1 with 1000 episodes. Additionally, $g(a_t)$ is set to 0.5 and the QoS provisioning values $\beta_1 = 1$ and $\beta_2 = 0.5$. The QoS provisioning values indicate that a longer convergence time is required when considering the end-to-end link and queue delay in the experiments. Fig. 8 shows the number of hops of a suitable path through our RL agent for a given $(\alpha, \gamma)$. It shows that there exists a trade-off between algorithm convergence and end-to-end delay. Therefore, the Delay increases with the increase of the value of $\alpha$. Additionally, the results demonstrate that when $0.5 \leq \alpha \leq 0.7$, with a number of paths between one and three hops, the Delay is higher than for other values of $\alpha$. On the contrary, when $0.7 \leq \alpha \leq 1$ with paths with a maximum of 2 hops, the Delay is lower than for other values of $\alpha$. This means that the smaller the network's size, the lower the latency when the value of $\alpha$ is greater than 0.7.

Moreover, in Fig 9, the results reveal that the PDR exponentially increases when $\alpha$ increases, reflecting a significant decrement when $\alpha$ takes a value of 0.6 or 0.5. In Fig 10, the results illustrate that high values of $\alpha$ are associated with higher RX and lower TX in the network. Consequently, the energy consumption exponentially increases with the increase of $\alpha$. Since IoT nodes often have a small battery,

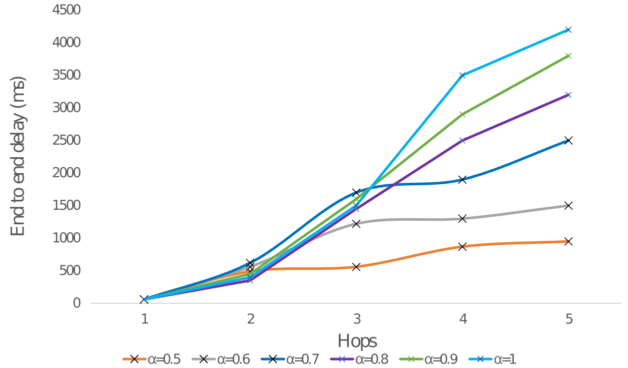


**FIGURE 8.** Delay-An illustrative comparison of our approach by using $\alpha \in (0{,}5\text{-}1)$.

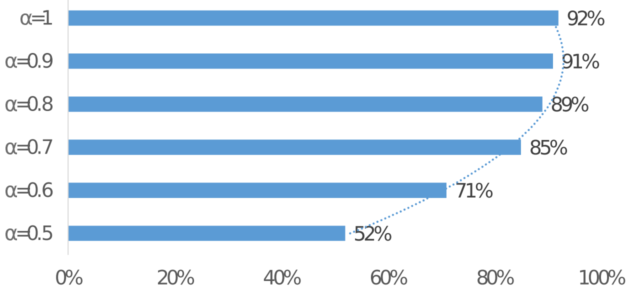


**FIGURE 9.** PDR-An illustrative comparison of our approach by using $\alpha \in (0{,}5\text{-}1)$.

measured in millivolts, extra power consumption can reduce the device's battery life by forcing the node to change its state to off. Although, the condition $0.7 < \alpha \leq 1$ introduces the lowest Delay in small DODAGs, it is not suitable for our scenario because we consider a network with a maximum of 5 hops. Moreover, for $0.5 \leq \alpha < 7$ with paths created with more than 2 hops, the Delay is lower than for other values of $\alpha$. However, there is a meaningful decrement in the PDR. Accordingly, to ensure a suitable analysis in terms of Delay, PDR, and RDC, we set the value of $\alpha$ to 0.7 in subsequent performance comparisons.

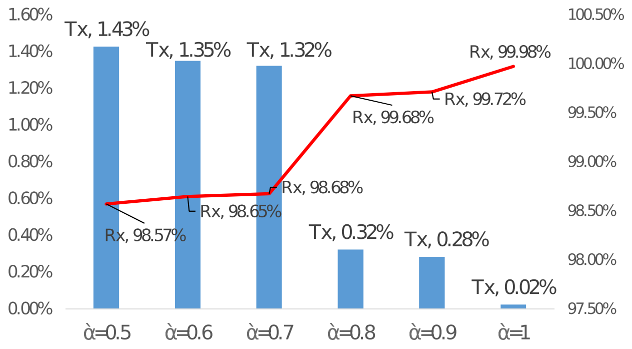


**FIGURE 10.** RDC-An illustrative comparison of our approach by using $\alpha \in (0{,}5\text{-}1)$.

### 2) PERFORMANCE COMPARISON

In what follows, we compare the Delay, PDR, and RDC with the following four scenarios.

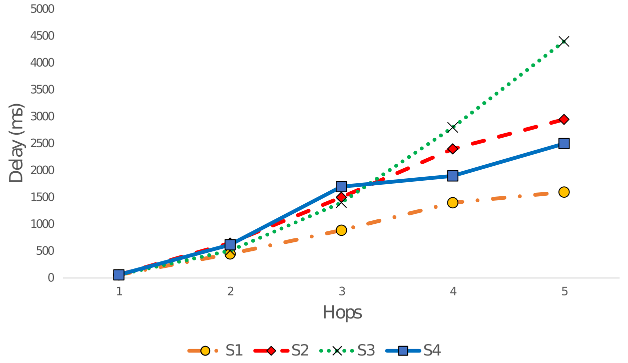


**FIGURE 11.** Delay-An illustrative comparison between S1, S2, S3, and S4.

1) S1: An RPL scenario with no SDN implementation under Rank attack.
2) S2: An RPL scenario with SDN implementation without Rank attack.
3) S3: An RPL scenario with SDN implementation under Rank attack.
4) S4: An RPL scenario with our RL-based SDN approach under Rank attack (our approach).

#### a: DELAY

In what follows, we analyze the delay of the four scenarios considering the path with the maximum number of hops. As a result, Fig. 11 demonstrates that in S1, the delay reaches 1600 milliseconds. Further, in S2, the latency in SD6LoWPAN reaches 2950 milliseconds. As a consequence, the latency is 45.72% higher than in S1. This is because an additional overhead is introduced due to the messages exchanged from the controller to the data plane. In S3, the latency reaches 4400 millisecond, which is 63.63% and 32.95% higher than S1 and S2, respectively. This is due the Rank attack, requiring the data plane to navigate downwards along the RPL topology across multiple non-optimized paths. Furthermore, in S4, the results show that the latency reaches 2500 milliseconds, which is 56.81% lower than S3 and 15.25% lower than S2.

Although S4 is 36% higher than the scenario where the SDN implementation is not used (S1), the proposed solution restores and even optimizes the typical behavior in SD6LoWPAN. This is because the number of SDN messages is decreased since the optimized paths are only delivered once the RL approach's exploration process is finished, rather than not every time the RPL collects data from the data plane. It is worth mentioning that our solution obtains the best results with DODAGs created with more than 3 hops.

#### b: PDR

In what follows, we analyze the four scenarios' PDR. To this end, we consider the path with the maximum number of hops and an average of 360 control packets delivered from the data plane to the controller. Consequently, the results illustrated in Fig 12, demonstrate that 151 packets were successfully delivered to the border router in S1. This means that this scenario reaches a PDR of 48%. Further, in S2, the average delivery variation reaches 270 packets per second, reaching

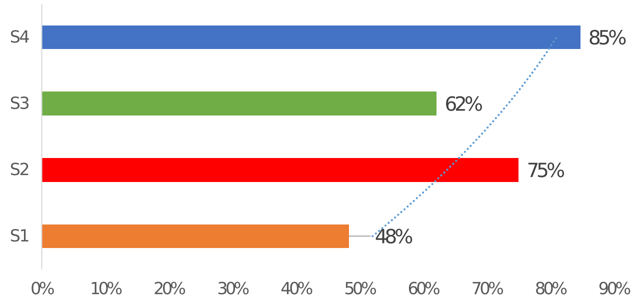


**FIGURE 12.** PDR-An illustrative comparison between S1, S2, S3, and S4.

a PDR of 75%. This is 27% more than S1 because the SDN approach helps speed up the packets delivery. Subsequently, a SD6LoWPAN under Rank attack introduces a packet loss ratio of 38%. Thus, this scenario reaches a PDR of 62% which is 13% lower than S2 and 14% higher than S1. Finally, in our method, the results demonstrate that the PDR reaches 85%, which is 23%, 10%, and 37% more efficient than S3, S2, and S1, respectively. This is because the RL optimization algorithm optimizes the network routes in SD6LoWPAN.

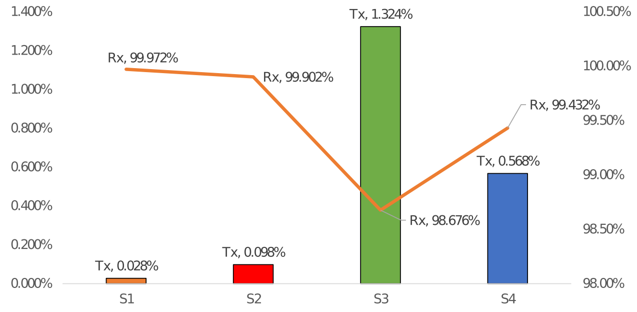


**FIGURE 13.** RDC-An illustrative comparison between S1, S2, S3, and S4.

### *c: RDC*

In S1, as illustrated in Fig. 13, the Rx reaches 99.972% and Tx 0.028%. Meanwhile, in S2, the Rx reaches 99.902% and Tx 0.098%. As a result, this scenario consumes less energy than S1 because the centralized SDN architecture optimizes the power consumption by not overloading the data plane with continuous execution of the RPL. Subsequently, in S3, the Rx reaches 98.676%, and Tx is 1.324%. Therefore, this scenario introduces a higher duty cycle than S2 due to the Rank attack execution. Conclusively, in S4, the Rx is 99.432%, and Tx is 0.568%. Consequently, this scenario consumes less energy than the third scenario restoring the excessive energy consumption introduced by the Rank attack. Although our approach introduces more latency than the other scenarios where the route consists of a maximum of 3 hops, the latency is decreased in the fourth and fifth hop due to the exploration of the RL agent's environment. Moreover, the proposed scheme provides better performance in packet delivery than S1 and S2 and restores the Rank attack's energy consumption in S3. It is worth mentioning that since the results obtained demonstrate that our approach provides network performance efficiency, thus preventing rank attacks from overwhelming the constrained SD6LoWPAN, we did not create more test scenarios, including more malicious nodes. To the best of our knowledge, the concept of a unified SDN-based intrusion prevention stack scheme, integrating RPL for fast network discovery and RL for route optimization to avoid ranking attacks, has never been attempted in any previous research works.

## V. CONCLUSION

The core of our solution is the elaboration of a security preventive control that takes advantage of the programmability of SDN in 6LoWPAN to build a self-learning agent that captures states through flow tables and metrics collected from the control plane. The learning consists of optimizing RPL routing based on QoS metrics like delays and packet loss rate. The control plane and the application plane stack can be used in a wireless border router supporting 6LoWPAN, introducing therefore a QoS awareness intelligence and avoiding RPL rank attacks sensitivity. Such a solution can support 5G agnosticism with respect to different wireless networks like 6LoWPAN networks. To analyze the performance of the proposed scheme, we leverage Contiki Cooja. The results demonstrate that the proposed scheme satisfies the requirements of SD6LoWPAN by providing low management complexity, delay reduction, and considerably preventing ranking attacks, thanks to the introduction of the learning agent reinforcing the route optimization approach.

Future work will include implementing the proposed security scheme on an IoT testbed. Moreover, our research will explore the use of network slicing to tailor our approach for heterogeneous networks with the help of hierarchical SDN drivers distributed between the cloud and the edge. Such a deployment will promote decentralized decision-making and introduces our solution in large-scale scenarios.

## REFERENCES

[1] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards Internet of Things," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–6.

[2] D. Bharadia, J.-I. Choi, M. Jain, S. Katti, T. M. Kim, and P. Levis, "Adaptive techniques for full duplex communications," U.S. Patent 10 230 419, Mar. 12, 2019.

[3] H. Al-Kashoash, *Congestion Control for 6LoWPAN Wireless Sensor Networks: Toward the Internet of Things*. Springer, 2019.

[4] L. Fisser, H. Ipach, A. Timm-Giel, and C. Becker, "Evaluation of LTE based communication for fast state estimation in low voltage grids," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Nov. 2020, pp. 1–7.

[5] M. Miguel, E. Jamhour, M. Pellenz, and M. Penna, "SDN architecture for 6LoWPAN wireless sensor networks," *Sensors*, vol. 18, no. 11, p. 3738, Nov. 2018.

[6] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[7] I. Nedyalkov, "Studying of a modeled IP—Based network using different dynamic routing protocols," in *Proc. X Nat. Conf. Int. Participation (ELECTRONICA)*, May 2019, pp. 1–4.

[8] D. Shin, V. Sharma, J. Kim, S. Kwon, and I. You, "Secure and efficient protocol for route optimization in PMIPv6-based smart home IoT networks," *IEEE Access*, vol. 5, pp. 11100–11117, 2017.

[9] A. Kamble, V. S. Malemath, and D. Patil, "Security attacks and secure routing protocols in RPL-based Internet of Things: Survey," in *Proc. Int. Conf. Emerg. Trends Innov. ICT (ICEI)*, Feb. 2017, pp. 33–39.

[10] K. K. Rai and K. Asawa, "Impact analysis of rank attack with spoofed IP on routing in 6LoWPAN network," in *Proc. 10th Int. Conf. Contemp. Comput. (IC)*, Aug. 2017, pp. 1–5.

[11] M. Charfi, A. Mouradian, and V. Veque, "Networking functions for wireless sensor network applications: An SDN-based approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[12] M. Baddeley, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, "Evolving SDN for low-power IoT networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 71–79.

[13] A. Verma and V. Ranga, "Security of RPL based 6LoWPAN networks in the Internet of Things: A review," *IEEE Sensors J.*, vol. 20, no. 11, pp. 5666–5690, Jun. 2020.

[14] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the Internet of Things in the age of machine learning and software-defined networking," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4829–4842, Jun. 2018.

[15] S. Otieno Ooko, J. Kadammanja, M. Grace Uwizeye, and D. Lemma, "Security issues in IPv6 over low-power wireless personal area networks (6LoWPAN): A review," in *Proc. 21st Int. Arab Conf. Inf. Technol. (ACIT)*, Nov. 2020, pp. 1–5.

[16] M. Preda and V.-V. Patriciu, "Simulating RPL attacks in 6LoWPAN for detection purposes," in *Proc. 13th Int. Conf. Commun. (COMM)*, Jun. 2020, pp. 239–245.

[17] R. Sahay, G. Geethakumari, and K. Modugu, "Attack graph—Based vulnerability assessment of rank property in RPL-6LoWPAN in IoT," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 308–313.

[18] G. Rajesh, X. M. Raajini, R. A. Rajan, M. Gokuldhev, and C. Swetha, "A multi-objective routing optimization using swarm intelligence in IoT networks," in *Intelligent Computing and Innovation on Data Science*. Springer, 2020, pp. 603–613.

[19] R. Salem, M. A. Salam, H. Abdelkader, and A. Awad Mohamed, "An artificial bee colony algorithm for data replication optimization in cloud environments," *IEEE Access*, vol. 8, pp. 51841–51852, 2020.

[20] Y. Chen and U. Vaidya, "Sample complexity for nonlinear stochastic dynamics," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 3526–3531.

[21] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7559–7566.

[22] F. F. J. Lasso, K. Clarke, and A. Nirmalathas, "A software-defined networking framework for IoT based on 6LoWPAN," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2018, pp. 1–7.

[23] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 25–33.

[24] D. Raba, A. A. Juan, J. Panadero, C. Bayliss, and A. Estrada-Moreno, "Combining the Internet of Things with simulation-based optimization to enhance logistics in an agri-food supply chain," in *Proc. Winter Simul. Conf. (WSC)*, 2019, pp. 1894–1905.

[25] E. Ancillotti, C. Vallati, R. Bruno, and E. Mingozzi, "A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management," *Comput. Commun.*, vol. 112, no. 1, pp. 1–13, Nov. 2017.

[26] A. Dvir, T. Holczer, and L. Buttyan, "VeRA—Version number and rank authentication in RPL," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2011, pp. 709–714.

[27] I. Wadhaj, B. Ghaleb, C. Thomson, A. Al-Dubai, and W. J. Buchanan, "Mitigation mechanisms against the DAO attack on the routing protocol for low power and lossy networks (RPL)," *IEEE Access*, vol. 8, pp. 43665–43675, 2020.

[28] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, and L. T. Jung, "SMTrust: Proposing trust-based secure routing protocol for RPL attacks for IoT applications," in *Proc. Int. Conf. Comput. Intell. (ICCI)*, Oct. 2020, pp. 305–310.

[29] K. S. Bhandari, I.-H. Ra, and G. Cho, "Multi-topology based QoS-differentiation in RPL for Internet of Things applications," *IEEE Access*, vol. 8, pp. 96686–96705, 2020.

[30] A. Salh, L. Audah, K. S. Kim, S. H. Alsamhi, M. A. Alhartomi, Q. Abdullah, F. A. Almalki, and H. Algethami, "Refiner GAN algorithmically enabled deep-RL for guaranteed traffic packets in real-time URLLC B5G communication systems," *IEEE Access*, vol. 10, pp. 50662–50676, 2022.

[31] A. Salh, L. Audah, M. A. Alhartomi, K. S. Kim, S. H. Alsamhi, F. A. Almalki, Q. Abdullah, A. Saif, and H. Algethami, "Smart packet transmission scheduling in cognitive IoT systems: DDQN based approach," *IEEE Access*, vol. 10, pp. 50023–50036, 2022.

[32] K. Incki and I. Ari, "A novel runtime verification solution for IoT systems," *IEEE Access*, vol. 6, pp. 13501–13512, 2018.

[33] K. Incki, I. Ari, and H. Sozer, "Runtime verification of IoT systems using complex event processing," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 625–630.

[34] S. Vohra and R. Srivastava, "A survey on techniques for securing 6LoWPAN," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 643–647.

[35] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, Internet Engineering Task Force, document RFC6206, 2011.

[36] Z. Magubane, P. Tarwireyi, A. M. Abu-Mahfouz, and M. O. Adigun, "RPL-based on load balancing routing objective functions for IoTs in distributed networks," in *Proc. Int. Multidisciplinary Inf. Technol. Eng. Conf. (IMITEC)*, Nov. 2019, pp. 1–6.

[37] M. B. Yassein, A. Flefil, D. Krstic, Y. Khamayseh, W. Mardini, and M. Shatnawi, "Performance evaluation of RPL in high density networks for Internet of Things (IoT)," in *Proc. 8th Int. Conf. Softw. Inf. Eng.*, Apr. 2019, pp. 183–187.

[38] W. Khallef, M. Molnar, A. Benslimane, and S. Durand, "Multiple constrained QoS routing with RPL," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[39] B. Safaei, A. M. H. Monazzah, and A. Ejlali, "ELITE: An elaborated cross-layer RPL objective function to achieve energy efficiency in Internet-of-Things devices," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1169–1182, Jan. 2021.

[40] N. Pradeska, W. Najib, and S. S. Kusumawardani, "Performance analysis of objective function MRHOF and OF0 in routing protocol RPL IPV6 over low power wireless personal area networks (6LoWPAN)," in *Proc. 8th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2016, pp. 1–6.

[41] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "New trust metric for the RPL routing protocol," in *Proc. 8th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2017, pp. 328–335.

[42] A. Rehman, M. M. Khan, M. A. Lodhi, and F. B. Hussain, "Rank attack using objective function in RPL for low power and lossy networks," in *Proc. Int. Conf. Ind. Informat. Comput. Syst. (CIICS)*, Mar. 2016, pp. 1–5.

[43] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 2, no. 1, pp. 253–279, 2019.

[44] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[45] G. Tanganelli, A. Virdis, and E. Mingozzi, "Enabling multi-hop forwarding in 6LoWPANs through software-defined networking," in *Proc. IEEE 20th Int. Symp. World Wireless, Mobile Multimedia Networks (WoWMoM)*, Jun. 2019, pp. 1–9.

[46] A. Habib, M. I. Khan, and J. Uddin, "Optimal route selection in complex multi-stage supply chain networks using SARSA (λ)," in *Proc. 19th IEEE Int. Conf. Comput. Inf. Technolog*, Dec. 2016, pp. 170–175.

[47] K. Iwata, "Extending the peak bandwidth of parameters for softmax selection in reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1865–1877, Aug. 2017.

[48] H. Erdol, S. Gormus, and M. C. Aydogdu, "A novel energy aware routing function for Internet of Things networks," in *Proc. 10th Int. Conf. Elect. Electron. Eng. (ELECO)*, 2017, pp. 1314–1318.

[49] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, and C. Yang, "Traffic engineering in software-defined networking: Measurement and management," *IEEE Access*, vol. 4, pp. 3246–3256, 2016.

[50] V. A. Tatsis and K. E. Parsopoulos, "Reinforced online parameter adaptation method for population-based metaheuristics," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2020, pp. 360–367.

[51] A. Musaddiq, Y. B. Zikria, and S. W. Kim, "Routing protocol for low-power and lossy networks for heterogeneous traffic network," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, p. 21, Dec. 2020.

[52] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Flexible traffic splitting in OpenFlow networks," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 407–420, Sep. 2016.

[53] C. M. Moreira, G. Kaddoum, and E. Bou-Harb, "Cross-layer authentication protocol design for ultra-dense 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[54] C. Miranda, G. Kaddoum, E. Bou-Harb, S. Garg, and K. Kaur, "A collaborative security framework for software-defined wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2602–2615, 2020.

[55] G. Tanganelli, A. Virdis, and E. Mingozzi, "Implementation of software-defined 6LoWPANs in contiki OS," in *Proc. IEEE 20th Int. Symp. World Wireless, Mobile Multimedia Networks (WoWMoM)*, Jun. 2019, pp. 1–6.

[56] M. Tutunovic and P. Wuttidittachotti, "Discovery of suitable node number for wireless sensor networks based on energy consumption using cooja," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 168–172.

[57] O. Ali, M. K. Ishak, M. A. Md Zawawi, M. T. A. Seman, M. K. L. Bhatti, and Z. Y. M. Yusoff, "A MAC protocol for energy efficient wireless communication leveraging wake-up estimations on sender data," in *Proc. 17th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Jun. 2020, pp. 45–50.

**CHRISTIAN MIRANDA** (Member, IEEE) received the bachelor's degree in computer and electrical engineering and the M.S. degree in applied computing security from the Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil, Ecuador, in 2010. He is currently pursuing the Ph.D. degree in electrical engineering with the École de Technologie Supérieure (ÉTS), Université du Québec à Montréal, Montreal, Canada. Since 2008, he has been a Technical Consultant in the field of cybersecurity and network architecture design in recognized industries around the world. His research findings are published in many prestigious venues, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), IEEE ICC, and IEEE WCNC. His recent research interests include analysis and deployment of intrusion detection and mitigation mechanisms using reinforcement-learning algorithms for software-defined cloud radio networks (SDCRANs).

**GEORGES KADDOUM** (Member, IEEE) received the bachelor's degree in electrical engineering from the École nationale supérieure de techniques avancées Bretagne (ENSTA Bretagne), Brest, France, the joint M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from the Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009. He is currently a Professor and the Tier 2 Canada Research Chair with the École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, Canada. Since 2010, he has been a Scientific Consultant in space and wireless telecommunications for several USA and Canadian companies. He has published more than 300 journals, conference papers, two chapters in books, and eight pending patents. His recent research interests include wireless communication networks, tactical communications, resource allocations, and security. He was a recipient of the best papers awards from the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications (WIMOB) with three coauthors and the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC) with four coauthors. He was also a recipient of the IEEE TRANSACTIONS ON COMMUNICATIONS Exemplary Reviewer Award, in 2015, 2017, and 2019; the Research Excellence Award from the Université du Québec in 2018; the Research Excellence Award from ÉTS in recognition of his outstanding research outcomes in 2019. In 2014, he was the ÉTS Research Chair in physical-layer security for wireless networks. He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING and an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE COMMUNICATIONS LETTERS.

**AMINE BOUKHTOUTA** (Member, IEEE) received the Engineering degree in computer science from the University of Science and Technology Houari Boumediene, Algeria, in 2005, and the M.A.Sc. degree in information systems security and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, Canada, in 2009 and 2016, respectively. In 2016, he joined a postdoctoral industrial program, where he worked on finding malicious indicators in evolving delivery networks by applying big data analytics and machine learning. He is an Experienced Researcher at the Ericsson Security Research Group. He was part of Cyber-Forensics Training Alliance Canada, doing research on the generation of cyber-threat intelligence based on malware and network traces. He has published five journal articles and 11 conference papers in peer-reviewed venues. His current research interests include prevention, detection of cyber threats by applying machine learning, and artificial intelligence He was a recipient of the OCTAS Prize from University Competition, in 2009, the FQRNT Doctoral Scholarship, from 2010 to 2011, the Best Paper Award, and the MITACS and PROMPT Postdoctoral Fellowships, from 2016 to 2017.

**TAOUS MADI** (Member, IEEE) received the Ph.D. degree in information systems engineering from Concordia University, Montreal. She is currently an Experienced Researcher with Ericsson Canada Inc., Canada. She has coauthored a book and several conferences and journal articles at reputable cybersecurity venues. Her research interests include network function virtualization security, software-defined network security, the Internet of Things security, security metrics, machine learning, and formal verification.

**HYAME ASSEM ALAMEDDINE** (Member, IEEE) received the M.Sc. degree from the Conservatoire national des arts et des Métiers (CNAM), Paris, France, in 2015, and the Ph.D. degree from the Concordia Institute of Information Systems Engineering, Concordia University, Montreal, Canada, in 2019. She is currently an Experienced Researcher with Ericsson, Montreal, Canada. She is also the Co-Founder of the Montreal Operations Research Student Chapter, where she held the positions of the Vice President and the Academic Events Director. Her research interests include cloud computing, network function virtualization, software-defined networks, edge computing, the Internet of Things, 5G, and network optimization and security. She was a recipient of several awards, including the NSERC PERSWADE Award in 2018 and the MITACS Elevate Postdoctoral Fellowship, in 2019.