Learning-Based Resource Management for Low-Power and Lossy IoT Networks

Arslan Musaddiq, Rashid Ali, *Member, IEEE*, Sung Won Kim, and Dong-Seong Kim, *Senior Member, IEEE*

***Abstract*—Internet of Things (IoT) networks are key to the realization of modern industries and societies. A key application of IoT is in smart-grid communications. Smart-grid networks are resource constrained in terms of computing power and energy capacity. Similarly, the wireless links between devices are typically associated with high packet-loss rates, low throughput, and instability. To provide a sustainable communication mechanism, an IoT network stack is proposed for these devices. However, each network stack layer has its own constraints. For example, to facilitate the operation of these low-power and lossy network (LLN) devices, the international engineering task force (IETF) standardized a network-layer protocol called a routing protocol for low-power and lossy networks (RPLs). RPL often creates an inefficient network in densely deployed and varying traffic load conditions. Future dense IoT-based networks are expected to automatically optimize the reliability and efficiency of communication by inferring the diverse features of both the environments and actions of the devices. Machine learning (ML) provides a promising framework for such a dense network environment. In this study, we examine the underlying perspective of ML for such systems. We utilize the multiarmed bandit (MAB)-based expected energy count (BEEX) technique, which provides nodes the ability to effectively optimize their operation. Using the proposed mechanism, nodes can intelligently adapt their network-layer behavior. The performance of the proposed (BEEX) algorithm is evaluated through a Contiki 3.0 Cooja simulation. The proposed method improves the energy consumption and packet delivery ratio and produces a lower control overhead than other state-of-the-art mechanisms.**

***Index Terms*—Energy consumption, Internet of Things (IoT), multiarmed bandit (MAB), reinforcement learning, RPL.**

I. Introduction

THE Internet of Things (IoT) has been applied to various fields of application, owing to its potential effect on our daily lives. IoT is an extensive system of interconnected sensing devices having unique characteristics, such as limited memory, limited battery capacity, and limited processing power [1]. In a resource-constrained environment, enabling sustainable smart communication for future IoT networks is one of the most critical aspects of an IoT network design. IoT has several applications in smart sustainable cities [2], smart healthcare [3], smart industries [4], smart grids [5], smart homes, and smart agriculture [6]. The demand for IoT applications is continuously increasing, and billions of IoT devices are already connected. According to a Statista report, 75 billion IoT devices are expected to be used by 2025, and IoT technology is key to the realization of modern industry and society. With the development of the fourth industrial revolution (Industry 4.0), in particular, IoT has penetrated industrial systems, which has been referred to as the Industrial IoT (IIoT) [7].

One of the key IoT applications in an industrial setting is the smart grid. A smart grid is referred to as a communication network of smart sensors that sits on top of the electricity grid to sense, analyze, and transmit data from various power grid components [8]. The smart-grid infrastructure aims to provide power management to the electricity grid system. One such example of a smart grid is the advanced metering infrastructure (AMI). AMI develops a bidirectional communication network between smart meters and utility control centers to automatically measure electric usage, collect voltage and current data, identify theft and tampering, and connect and disconnect services. AMI is a key component of a smart-grid system and applies other services, including adaptive pricing, improving energy efficiency, reliability, and power system control and monitoring [9].

Smart meters are usually embedded resource-constrained devices with restricted computing power and storage and generate a large amount of data daily. For such a large amount of data sharing, providing an appropriate communication technology is one of the core issues in IoT-based smart-grid systems. Owing to the lossy nature of a wireless environment, links between devices are typically associated with high packet-loss rates, low throughput, and instability. The use of link-layer technologies normally includes IEEE 802.15.4g, IEEE 802.15.4e, IEEE 1901.2, and IEEE 802.11 standards. These types of networks are described as low-power and lossy networks (LLNs). In an LLN environment, these IoT nodes must intelligently handle data processing, energy consumption, and communication mechanisms. Energy is particularly important for the nodes in an LLN setting. The nodes consume

Manuscript received 8 December 2021; accepted 7 February 2022. Date of publication 18 February 2022; date of current version 24 August 2022. This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Grand Information Technology Research Center Support Program Supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2022-2020-0-01612, and in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under Grant 2018R1A6A1A03024003. *(Corresponding author: Dong-Seong Kim.)*

Arslan Musaddiq and Dong-Seong Kim are with the ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: arslan@kumoh.ac.kr; dskim@kumoh.ac.kr).

Rashid Ali is with the School of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, South Korea (e-mail: rashidali@sejong.ac.kr).

Sung Won Kim is with Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 8541, South Korea (e-mail: swon@yu.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3152929

most of the energy during communication and, thus, reliable connectivity in an LLN environment is particularly important for sharing real-time data [9]. Facilitating an interconnected IoT-based smart-grid system for sustainable communication requires an efficient and intelligent networking mechanism.

Constrained resources raise numerous challenges for LLN devices. To deal with such challenges, both hardware and software-based approaches have been proposed for these devices. However, hardware-based mechanisms incur high system costs due to additional hardware installation. By using software-based methods, efficient networking protocols can be designed. Moreover, lightweight IoT operating systems, such as Contiki, TinyOS, and RIOT have been developed to efficiently manage limited resources [10]. To facilitate the operation of these LLN devices, a modified network stack for IoT communications is designed. For example, the routing over lossy and low-power networks (RoLLs) working group of the international engineering task force (IETF) standardized a network-layer protocol, called a routing protocol for low-power and lossy networks (RPLs) [9].

RPL is designed to provide Internet protocol version 6 (IPv6) connectivity to LLN devices. It creates a tree-like routing topology called a destination-oriented directed acyclic graph (DODAG). The DODAG is constructed using a specific objective function (OF). The IETF RoLL working group standardized the minimum rank with hysteresis OF (MHROF), which is based on the expected transmission count (ETX) as a default metric defined by RFC 6719 [12]. Similarly, OF zero (OF0) is based on the hop count as a routing metric [13]. The OF aims to optimize specific network parameters, such as the energy, delay, or throughput. The ETX-based rank provides an approximation of the number of required transmissions to successfully deliver a packet to the sink node based on their link qualities toward the destination node. In MRHOF, the link quality is assessed by broadcasting probe packets at regular intervals. The probe packets are rebroadcast by the receiving node. This continuous link evaluation process depletes the resources of the node.

A dynamic and lossy environment drastically affects the IoT communication mechanism. RPL is designed to meet a wide range of LLN applications, including large-scale AMI systems. Although smart meters are static, the link between two communicating meters is generally unstable, owing to wireless fading and interference. In addition, the standardized RPL mechanism is unable to resolve unbalanced load and energy distribution problems, particularly for nodes that are closer to the sink. Managing the resources in a lossy environment is essential for next-generation IoT-based AMI networks. The self-sustainability of the AMI network infrastructure with a lower control overhead, lower energy consumption, and higher throughput is required for the next-generation AMI infrastructure. To sustain RPL-based LLN nodes, it is desirable to manage the energy resources and link quality and consequently construct a DODAG that improves the network lifetime and packet delivery ratio (PDR). However, utilizing energy information to create a DODAG based on a simple rule-based scenario is also ineffective in sustaining a large-scale AMI infrastructure.

For managing the performance in a lossy environment, next-generation learning-based techniques are more intelligent and self-sustaining. Future dense IoT-based LLNs are expected to automatically optimize the reliability and efficiency of communication by inferring the diverse features of both the environments and the actions of the devices. Machine learning (ML), which is one of the most widely used artificial intelligence applications, provides a promising framework for such dense LLN environments [14]. We can foresee an advanced IoT system that can efficiently manage its sources with the help of ML. An intelligent IoT device monitors and learns to perform a specific action to maintain a specific output metric. For example, one of the recently proposed resource allocation protocols called the intelligent collision probability learning mechanism (*i*CPLA), optimizes the network-layer operation using the Q-learning technique [15]. However, a Q-learning algorithm functions inadequately in numerous dynamics such as in an IoT system model where reward probabilities may change over time owing to a change in network conditions. The *i*CPLA technique is also unable to change reward weightage during node learning phases. Furthermore, the proposed Q-learning mechanism also takes a 2-D array of states and actions and, thus, it has a complexity of $O(n)(a)$, where n and a represent states and actions, respectively [15]. Q-learning also incurs higher delays due to the complex learning process. In a nonstationary reward distribution setting, it is more practical to assign a weight to recent rewards than to previous rewards. This can be achieved by employing a recency-weighted average bandit scheme. To cope with the aforementioned Q-learning problems, it is observed through extensive Contiki OS cooja-based simulation, the resource allocation issue of IoT devices can be solved with the simplest RL technique such as multiarmed bandit (MAB) [16].

In this study, we examine the underlying perspective of ML for AMI systems. Furthermore, we formulate the learning problem as a MAB task to choose the best strategy at each sensing epoch. MAB is an ML problem, in which a player attempts to obtain the maximum reward from a number of slot machines. The large-scale network link-layer operation can be optimized by integrating the MAB mechanism using energy as a learning metric. The proposed mechanism is integrated with each node that recursively examines its energy resources and link conditions. It learns and tunes the OF for RPL-based LLN nodes and updates the routing table entries by utilizing less control overhead. In this article, we propose a bandit-learning-based expected energy count (BEEX) mechanism to optimize a large-scale network link-layer operation.

The main contributions of this study are summarized as follows.

1) We first introduce the bandit model and system formulation for the IEEE 802.15.4 RPL-based network. We incorporate idiosyncrasies related to LLNs in our system model. This includes the effect of the MAC layer probing mechanism on node resources and the RPL-based load-imbalance issue of the ranking mechanism.
2) To construct the routing table entries, we introduce an expected energy count (EEX)-based ranking mechanism.

The EEX information is embedded in a control packet called a DODAG information object (DIO). The transmission of the DIO packet is based on the trickle-timer mechanism (RFC 6206).

3) We developed a bandit framework that utilizes learned Q-values to intelligently construct routing table entries. Our analysis accounts for both high-and low-density networks. We also analyzed the network in a fluctuating traffic environment.
4) The proposed mechanism investigates the AMI wireless environment by employing an exponential recency weighted average (ERWA) bandit scheme in which reward probabilities may change over time owing to a change in network conditions. The network condition can change owing to a fluctuating traffic load or a change in network size with a node failure or node addition. Under such a nonstationary reward distribution setting, it is more practical to assign a weight to recent rewards than to long previous rewards.
5) We improve the trickle-timer operations of the nodes by suppressing the DIO control packet transmissions during the exploitation phase. The proposed method maintains the stability in the network to establish a DODAG either by exploring the environment using embedded EEX information in the DIO packet or by exploiting the environment using the estimated Q-value.
6) The standard and extended performance assessment metrics (i.e., PDR, control overhead, and energy consumption) are comprehensively utilized to evaluate the performance of our proposed mechanism.

The remainder of this article is organized as follows. The background and related studies are presented in Section II. The proposed MAB-learning-enabled mechanism is described in Section III. Section IV provides the experimental results and discussion, followed by some concluding remarks in Section V. The acronyms used in this study are listed in Table I.

TABLE I
LIST OF ACRONYMS USED IN THIS ARTICLE

Acronyms	Description
IoT	Internet of things
IIoT	Industrial IoT
LLNs	Low-power and lossy networks
AMI	Advanced metering infrastructure
RPL	Routing protocol for low power and lossy networks
DODAG	Destination oriented directed acyclic graph
ETX	Expected transmission counts
MRHOF	Minimum rank with hysteresis objective function
OF0	Objective function zero
IPv6	Internet protocol version 6
RoLL	Routing over Lossy and Low-Power Networks
IETF	Internet engineering task force
MAB	Multi-armed bandit
ML	Machine learning
ERWA	exponential, recency-weighted average
ICMPv6	Internet control messages protocol version 6
DIO	DODAG information object
DAO	Destination advertisement object
DIS	DODAG information solicitation
RL	Reinforcement learning
QU	Queue utilization
CSMA/CA	carrier-sense multiple access with collision avoidance
BEEX	bandit-learning-based expected energy count
PDR	Packet delivery ratio
E2E	End-to-end delay

II. BACKGROUND AND RELATED RESEARCH

A. RPL Routing

IoT devices face environmental challenges and resource management problems. A dynamic and lossy environment drastically affects the IoT communication mechanism. Managing resources in a lossy environment is essential for next-generation IoT-based networks. The IETF standardized the RPL protocol for LLN devices. The RPL protocol uses a certain OF to construct a routing topology called the DODAG. OF defines the use of a particular metric for rank calculation. For example, the standardized OF0 finds the shortest link distance to the sink node, irrespective of the link condition. It selects a parent node that has a minimum rank in terms of the distance from the sink node. In contrast, ETX-based OF uses a path that requires a minimum number of retransmissions to deliver a packet to the DODAG root node. The rank values increase monotonically from the root node toward the child nodes. The rank of the child node is always more extensive than that of the parent node to avoid the routing loop. The root node is ranked 1, the next-hop node is ranked 2, and so on. The ranks were calculated as follows:

$$\text{Rank}(c) = h + \text{Rank}(p) + \text{rank}_{\text{increase}} \tag{1}$$

where Rank(c) is the child node rank, and h represents the one-hop distance [12], [17]. In addition, Rank(p) represents the parent node rank, and $\text{rank}_{\text{increase}}$ is the variation factor between the ranks of the parent and child nodes. In the standardized protocol, $\text{rank}_{\text{increase}}$ is the ETX and hop count metric for MRHOF and OF0, respectively. The ETX consumes 16 bits in the control overhead object field. The 16-bit ETX value was rounded to the nearest whole number. For example, if ETX $= 2.3$, then the object field value is $2.3 \times 128 = 294$.

The RPL is standardized for a lossy network environment. Each node in the DODAG maintains a list of candidate parent nodes set for fault-tolerance purposes. The construction of DODAG involves the exchange of Internet control message protocol version 6 (ICMPv6) control messages called a DIO. Other control messages include DODAG advertisement object (DAO), and DODAG information solicitation (DIS) [11]. The DODAG constructions begin as follows: first, the sink node transmits DIO messages that contain its rank calculated from an OF. Next, the receiving node measures its rank and broadcasts the information to neighboring nodes. The neighbor nodes continue this process until all DAG nodes receive the DIO message. The child node sends the destination information to the parent node through a DAO message. Any node that does not receive a DIO message can request

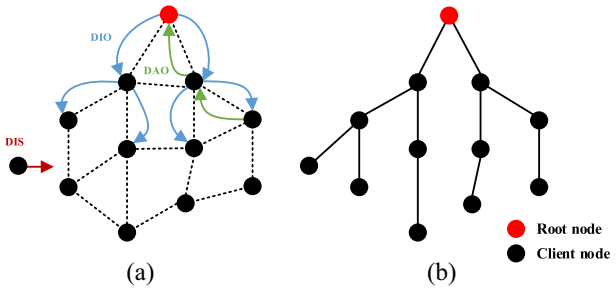

Fig. 1. RPL DAG with (a) control message sequence and (b) DODAG construction.

joining the DODAG using the DIS message. Fig. 1 shows the transmission direction of an RPL node control message. The red and black colored nodes indicate a root and client node, respectively.

The IoT nodes contain limited computational and energy resources. It is highly desirable to limit the control overheads. The DIO control message transmission is governed by a trickle-timer mechanism [18], [19]. The trickle timer manages the energy consumption by controlling the rate at which DIO messages are transmitted. If the network is consistent, fewer DIOs are transmitted. It doubles the transmission period each time the network is found to be inconsistent until the timer reaches the maximum period. When an inconsistency is detected, for example, a new node joins the DODAG or link disruption, it resets the timer to the minimum value to quickly update the DODAG. The trickle algorithm consists of three variables, i.e., a message counter c, trickle interval length I, and a random interval length t. It also has three configuration parameters, i.e., redundancy constant k_c, minimum interval length $I_{\min}$, and maximum interval length $I_{\max}$. In the start, trickle set I to a value between $[I_{\min},\ I_{\max}]$. It sets the counter c to 0. The trickle then selects a transmission interval t from the range $[I = 2,\ I]$. Whenever there is consistent transmission trickle increment the counter by 1. The DIO is transmitted if the counter is less than the k_c value; otherwise, transmission is suppressed. When I expires, trickle double the interval length until it reaches $I_{\max}$. If inconsistent transmission receives the timer is reset to an initial value. The RPL control message structure is shown in Fig. 2, and a detailed description of the RPL protocol features is provided in Table II.

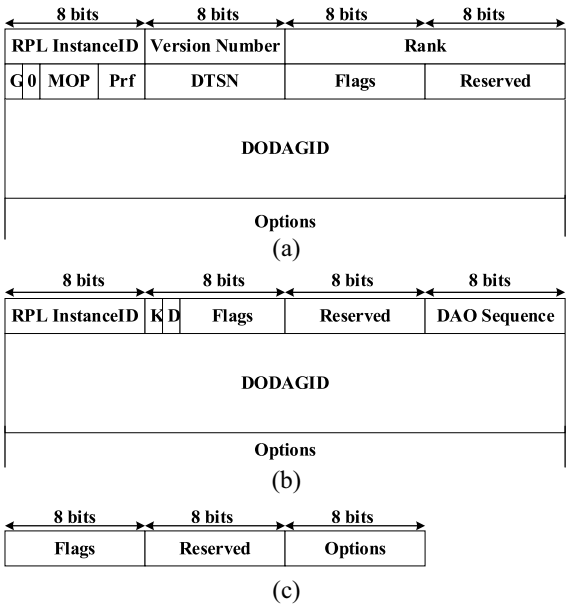

Fig. 2. RPL control message structure. (a) DIO message object. (b) DAO message object. (c) DIS message object.

TABLE II
RPL FEATURES

Features	Description
Design Principle	It supports multiple instances and requires bidirectional links.
Objective Functions	The standardized OFs are MRHOF and OF0
Loop detection	Child node rank is always greater than the parent node ranks to avoid creating loops, i.e., $Rank(c_j) = 128 + Rank(p_i) + rank_{increase}$
Route Repair	RPL support two repairs: local and global repairs
ICMPv6 RPL Control Message	ICMPv6 header-based DIS, DIO, DAO and DAO-ACK
Mode of operation (MOP)	Four MOPs: MOP (0), MOP (1), MOP (2), MOP (3)
Modes	Storing and non-storing
Control messages timer	The control messages are controlled using trickle timer following parameters c Maintains the record of received consistent messages k_c Redundancy constant, a natural number greater than 0 I_{min} A minimum interval size I_{max} A maximum interval size

B. Related Research

In LLNs, improving the network reliability along with the lifetime is of utmost importance. In recent years, numerous variations in the RPL protocols have been presented. For example, the *i*CPLA mechanism is based on cross-layer optimization using the *Q*-learning technique [15]. This method uses MAC layer collision information to perform network-layer decisions. In *i*CPLA, a complex Bellman's equation determines the optimal policy. Similarly, queue-utilization-based RPL (QU-RPL) [17] enhances the RPL performance by utilizing the queue factor in the RPL OF. This method balances the routing tree using the queue utilization factor, ETX, and hop counts. QU-RPL improves the congestion and load balancing in the RPL network. However, the proposed OF does not consider the energy consumption of the node, which may lead to low-energy node selection. Another technique, CoA-OF, uses the QU, residual energy, and ETX for parent selection [20]. The PDR, throughput, and energy consumption are improved, but they incur frequent parent switching in high traffic networks, which also causes high control overhead. Taghizadeh *et al.* [21] proposed a QU for energy efficiency and packet-loss issues in heavy traffic load networks. The proposed mechanism shows improved results in terms of energy and packet loss, but the control overhead is increased.

The enhanced RPL (E-RPL) [22] mechanism improves the energy consumption by limiting the control overhead transmission. Limiting the control overhead to update the rank information increases the network convergence time. Another technique based on energy and the ETX metric is based on fuzzy logic. Although the mechanism improves the throughput, the energy consumption increases. Similarly, Nassar *et al.* [23] proposed a multiobjective function to enhance the Quality of Service (QoS) of a smart grid. The routing metric is based on the ETX, delay, and power states. The proposed mechanism assigns weightage to the routes based on the power state. This mechanism improves the E2E delay, PDR, and network lifetime. However, this mechanism is unable to deal with changes in network dynamics. Lamaazi and Benamar [24] proposed a solution for energy efficiency and data reliability. The path selection metric is based on the ETX, energy consumption, and forwarding delay. Although the proposed solution improves the delivery ratio and energy consumption, the control overhead is increased. The genetic-based mechanism improves the end-to-end delay, packet transmission delay, and energy utilization. This method uses multiple metrics, such as the residual energy, ETX, weighted queue length, hop count, and delay. However, control overhead was also not considered in this mechanism. Another technique based on energy and the ETX metric is based on the fuzzy logic technique [25]. Although the mechanism improves the throughput, the energy consumption increases. Ghaleb *et al.* [26] provided a load-balancing solution by maintaining the list of child nodes. The child node list was updated using a fast propagation timer. This improves the network reliability in terms of the PRR, and increases the network convergence time. A stability-aware load-balancing (SL-RPL) [27] mechanism is proposed to avoid load balancing and frequent parent-switching mechanisms. SL-RPL utilizes the ETX and packet transmission rate as a routing metric.

Ancillotti *et al.* [28] proposed a link quality estimation (LQE) strategy for RPL. The LQE employs a received signal strength indicator (RSSI) and the ETX metric to enhance the link repair procedure (RL-Probe). While applying RL-Probe, the control overhead is increased. Aziz *et al.* [29] utilized a MAB-based clustering mechanism for ETX probing. However, communication with the cluster head incurs additional control overhead.

All of these solutions either incur a high energy consumption, high control overhead, or an inability to handle changes in network dynamics. For next-generation networking mechanisms, such as AMI, the network should have learning-based protocols instead of rule-based methods.

III. Proposed MAB-Learning-Enabled Energy Conservation for Low-Power and Lossy IoT Networks

A. Machine Learning and MAB-Learning Model

ML, a subcategory of artificial intelligence, is divided into three groups: supervised learning, unsupervised learning, and reinforcement learning [30]. In supervised learning, an agent uses input values to predict the output values [31]. In this way, the agent can predict the outcome of future events using regression and classification methods. In contrast, the unsupervised learning technique is used to reduce the features in the data set by finding symmetries in the data, for example, k-means clustering [32], an independent component analysis [33], and a principal component analysis [34].

In contrast, RL trains the agent to take action in an environment to optimize the reward. The RL mechanism aims to find an optimal method to achieve a specific goal. In the case of sensor networks, the agent is a sensor node that learns and makes decisions after learning an unknown environment. The sensor node learns an action by adapting to fluctuating network conditions to pursue its goal. RL decision-making problems could be bandit problems or Markov decision process (MDP)-based problems [30]. A Markov process consists of states; for each state s, there is a set of actions a. The agent takes action a in state s and transitions to the next state s'. A reward is given during the state transition process. During this process, a discount factor and learning rate were used to optimize the learning estimate. If the learning rate α is high, the node learns new values more, and if the learning rate is low, the node tends to learn new values less. Thus, if the learning rate is high, the learning estimate varies because, in each episode, the node tends to give more consideration to new rewards without considering previous experience. As the learning rate decreases, the learning estimate becomes stable by considering both the current value and the previous experience.

If the RL problem does not satisfy the Markov property, the problem can be solved using the bandit technique [16]. For example, in a case in which there is only one state or the states are static, then the process reduces to only a set of actions and rewards depending on the action taken and a discount factor. Under such scenarios, the bandit technique can be used to solve the process. The bandit is a classical RL decision-making problem-solving technique in which an agent performs a sequence of trials to find a strategy that maximizes the total payoff. If there is one possible action, it is called a single-armed bandit, also known as a slot machine. If the agent has to choose an action from multiple actions, it is known as a MAB problem.

The MAB is a classic RL decision-making technique. The MAB is suitable in situations where agents have to choose from a set of actions in a sequence of trials to maximize the total payoff in the future. The agent has different options and must choose one of the options during each iteration. In MAB, there is a slot machine with $\boldsymbol{K}$ arms called bandits. Each arm has its own probability of success. Pulling each arm gives a positive reward ($\boldsymbol{r}^+$) for success and a negative reward ($\boldsymbol{r}^-$) for failure. The objective is to pull the arm in a sequence of trials to maximize the total reward in the long run. The probability distribution of each arm was learned using trial-and-error and a value estimation. With each action, there is an action value $\boldsymbol{Q}(\boldsymbol{a})$ that helps the agent improve its future action decisions. With each iteration or episode, the agent updates the value of $\boldsymbol{Q}(\boldsymbol{a})$.

The agent either explores the actions or exploits the previous experience. Exploitation is the process of making the best decision given the available date, whereas exploration refers

to the process of acquiring additional data. In learning techniques, there must be a balanced exploration and exploitation. Exploring too much may yield a negative reward, and if the exploitation is high, it may prevent an optimal long-term reward. The exploration–exploitation tradeoff can be achieved using the epsilon-greedy (ϵ-greedy) technique. In ϵ-greedy, the ϵ value was between 0 and 1. For example, if ϵ is 0.5, the algorithm will apply 50% exploration and 50% exploitation. Similarly, if $\epsilon = 0.7$, the exploration rate is 70%, and the exploitation occurs 30% of the time. The ϵ-greedy can add randomness to ML problems [34].

The devices conduct the actions in two ways: first, it selects an optimal action based on $Q(a)$ as a reference, and second, it performs a random action and updates $Q(a)$. If an agent maintains the estimates of the action values, then at any time step, one of the estimated values of the actions is the highest. Selecting the highest estimated action value is called a greedy action. Nongreedy actions help improve the estimation of the action values. Whether it is better to explore or exploit depends on the value of the estimates, learning convergence fluctuations, and number of remaining steps. Balancing the exploration and exploitation of K-armed bandits and related problems can be achieved using many sophisticated models. Based on averaging the rewards, $Q_n(a)$ is updated by taking the sum of rewards of action a prior to the nth step divided by the number $n-1$ times that action a is taken

$$Q_n(a) = \frac{1}{n-1}(r_1 + r_2 + \cdots + r_{n-1}) \tag{2}$$

where $n-1$ indicates the number of times action a was chosen in the past. The rewards $r_1+r_2+\cdots+r_n$ represent the stochastic rewards for each time action a is chosen. If the denominator is 0, then $Q_n(a)$ is defined with a default value such as zeros. If the denominator is infinity according to the law of large numbers, $Q_n(a)$ converges to $Q(a^*)$. This method is also known as the sample-average method because the estimate of $Q_n(a)$ is the average of the samples of relevant rewards. The obvious rule is to select the action that provides the best $Q_n(a)$ estimate, that is, arg max $Q_n(a)$. This is a greedy approach and exploits the current knowledge to choose the best estimate.

The value of $Q_n(a)$ can also be updated using the ϵ-greedy-algorithm, which means that it behaves greedily most of the time while intermittently selecting randomly from among all of the actions

$$A \leftarrow \begin{cases} \text{arg max } Q_n(a), & \text{with probability } 1-\epsilon \\ \text{random action with probability } \epsilon. \end{cases} \tag{3}$$

The bandit technique deals with the problem in which there are multiple options to choose from, and not much information about the options is available. We can also refer to these problems as being part of stochastic scheduling. The origin of these problems is based on the strategies used to play a slot machine. In a slot-machine problem, the player pulls the lever to receive a reward. The reward distribution of each machine is changed over time. The player's objective is to make more money by finding the highest payoff pattern. Finding the right machine and right combination is an extremely important factor for players to win big in the future. The notations used in this study are listed in Table III.

TABLE III
LIST OF NOTATIONS

Symbol	Definition
A	Set of actions
R	Set of rewards
R^+	Positive reward
R^-	Negative reward
M	Set of arms
θ	Probability of reward
α	Step-size parameter (alpha)
$Q_n(a)$	Q-value of action a during n$^{\text{th}}$ iteration
$\in$	Epsilon
EEX_t	Expected energy count

B. Link Quality Assessment Mechanism

The standardized MRHOF-based key metric for calculating the quality of the links is the ETX. The ETX mechanism is a link-layer function that calculates the link quality using a probing mechanism. With this method, the node transmits a probe packet and measures the number of retransmissions required to transmit the probe packet successfully. ETX is based on the callbacks of the CSMA protocol, which provide information on a number of retransmission attempts [36]. Under a wireless transmission scenario, when a transmitting node sends a packet to the destination node, the destination node sends an acknowledgment (ACK) packet to the transmitting node. When an ACK is received during a specific time duration, the transmission of a packet is considered a success; otherwise, the sending node retransmits the packet. The IoT node measures the number of data frame transmissions and the number of received ACK frames. The ETX is estimated by measuring the probability of frame loss ratio at the link l to each neighbor in the forward direction as d_f and in the reverse direction as d_r. Each attempt to transmit a frame can be considered a Bernoulli trial. The probability p that frame transmission from node $\boldsymbol{x}$ to $\boldsymbol{y}$ is unsuccessful is

$$p = 1 - \left(1 - p_f\right) \times (1 - p_r). \tag{4}$$

The ETX for the successful delivery of frame after k attempts in a single hop is measured as

$$\text{ETX}_l = \sum_{k=1}^{\infty} k \times p^k \times (1-p)^{k-1} = \frac{1}{1-p}. \tag{5}$$

The ETX in terms of forward delivery ratio d_f, i.e., $(1-p_f)$ and reverse delivery ratio d_r, i.e., $(1-p_r)$ is measured as

$$\text{ETX}_l = \frac{1}{\left(d_f \times d_r\right)}. \tag{6}$$

The delivery ratios d_f and d_r are measured using a link probe packet. Alternatively, ETX represents the link reliability as

$$\text{ETX}_l = \frac{1}{\text{reliability } (l)}. \tag{7}$$

The value of ETX indicates only the link quality between the two nodes. The cumulative ETX considers all expected numbers of transmissions to the sink node; for example, the cumulative ETX is the sum of each hop ETX. In an RPL-based

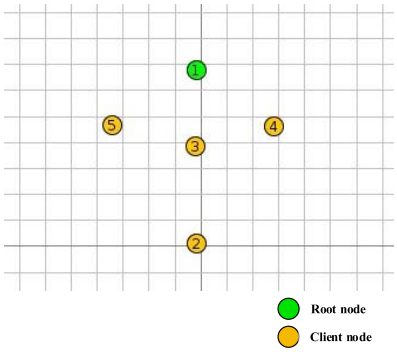

Fig. 3. Contiki OS Cooja simulation of four client nodes as example scenario to illustrate cost Tx (mJ).

TABLE IV
SAMPLE CONTIKI *Energest* VALUES OUTPUT FOR THREE NODES

Energy	*Node 3*	*Node 4*	*Node 5*
$Energest\_Tx$	711.64	779.32	1090.28
Tx (μJ)	113.3	124.1	174.0

network, the nodes are ranked based on their position in the DAG and the link quality metric. As specified in (1), $\text{rank}_{\text{increase}}$ is the ETX value.

C. Energy Assessment Mechanism

A node generates packets at random transmission rates. With each packet transmission, the node consumes valuable energy resources. The nodes utilize most of the energy in the electronic circuits during the communication phases. The nodes at a longer distance to their parent nodes need to utilize a higher radio power, and as a result, their onboard battery power is depleted early. The transmission energy is the amount of energy required to transmit k-bit of packets to a node at a distance d [37], [38]. This is defined as follows:

$$E_T(k, d) = E_{\text{ele}} \times k + E_{\text{amp}} \times k \times d^p \tag{8}$$

where E_{ele} is the energy required to run a transmitter or circuit, E_{amp} is the transmit amplifier to achieve an acceptable E_b/N_o, and p is the path loss index. The typical values of these parameters are $E_{\text{ele}} = 50$ nJ/bit, $E_{\text{amp}} = 100$ pJ/bit/m^2, and $2 \leq p \leq 4$., respectively. In Contiki OS [39], the *Energest()* function is used to estimate the power consumption of the node. The *Energest* module implemented in Contiki OS provides the accumulated time the sensor node spends in different communication modes. For example, the transmission energy consumption is expressed as follows:

$$Tx\ (mJ) = (Energest\_Tx \times 17.4 \times 3)/32768. \tag{9}$$

The *Energest()* model provides the value of *E_Tx*, where 17.4 mA is the current consumption required to run Zolertia Z1 mote, and consumes 3 V [40]. Similarly, 32 768 is the tick-per-second value of the Z1 mote. We obtained *Energest* values at 10 s intervals using different nodes as a parent node; for example, in Fig. 3, if node 2 selects nodes 3, 4, or 5 as a parent node, the corresponding *Energest* and *Tx* values are as given in Table IV. These values were obtained using the Contiki Cooja simulation environment.

TABLE V
OPTIMIZATION OF LLN NODES USING BANDIT METHOD

MAB framework	Optimization of LLN nodes
Player	Sensor nodes (child nodes)
Slot machines	Potential candidate parents
Reward	EEX
Objective	Resource management

D. Proposed Bandit-Learning-Based Expected Energy Count RPL Mechanism

In this article, a BEEX RPL mechanism is proposed. In the IoT network, each node has an $M \in (1, \ldots, m)$ set of neighboring nodes. A node selects one of the neighboring nodes for the path forwarding decision. The ETX of each neighboring node might be different owing to the different link conditions. The ETX parameter is based on the number of expected retransmission attempts, and each retransmission attempt consumes the energy resources. We can also define the ETX value in terms of the energy unit as follows:

$$\text{EEX}_t = \text{ETX} \times Tx. \tag{10}$$

To represent ETX in terms of the energy metric unit, we utilized (10) as $\text{rank}_{\text{increase}}$ in (1). The energy metric also represents the reward function for the bandit protocol Q-value generation. Each action results in either a positive or negative reward. The increment or decrement in EEX_t represents a reward of $+1$ or -1, respectively. The reward function R is computed as a function of EEX. The reward for transmission from the child to parent nodes is defined as a piecewise function

$$R = \begin{cases} -1, & \text{if EEX increases} \\ +1, & \text{otherwise.} \end{cases} \tag{11}$$

The bandit framework of the proposed study is summarized in Table V. In the bandit algorithm, the selection of a particular forwarding node is an action. The reward or reinforcement signal describes whether the action is favorable. The averaging method estimates the reward if the reward probability distribution does not change over time. In the case of a wireless environment, the reward probabilities may change over time, owing to a change in the network conditions, for example, a change in network size owing to a node failure. In such cases, the bandit problem can be solved by assigning more weight to recent rewards than to long-past rewards. In such a nonstationary reward distribution setting, the update rule (2) changes to

$$Q_{n+1} = Q_n + \alpha(r_n - Q_n). \tag{12}$$

The value of α is a constant step-size parameter, and its range is between 0 and 1. The value of $(r_n - Q_n)$ is an error in the estimate (the target minus the old estimate). Estimating $Q_n(a)$, it requires a record of all rewards obtained $n-1$ times. This requires complex memory and computational requirements. With each reward, additional memory is required. An incremental formula can be derived to reduce the computational and memory requirements. The node generates a Q_{n+1}, a value corresponding to each action using an incremental formula, as follows:

$$Q_{n+1} = \alpha r_n + (1 - \alpha)Q_n$$

Algorithm 1 BEEX Framework *(Updating Reward and Q-Value)*

Initialization: //The nodes initialize the reward and Q-value globally and then instant reward and cumulative reward for all actions are saved.
while the device is on **do**
set maximum retry limits = 3
set maximum back-off stages = 5
set $CW_{min} = 0$, $CW_{max} = 31$
set current reward = 0, $Q_n(a) = 0$, $Q_{n+1}(a) = 0$
Calculate ETX using *neighbor_link_callback ()*
Calculate *EEX* using *new_ETX* in (10)
if $(EEX_{(current)} = EEX_{(Previous)})$, **then**
  reward = r^+
else if $(EEX_{(current)} > EEX_{(Previous)})$, **then**
reward = r^-
update reward table for action a
update Q-values table according to (13)
end if
end **while**

$$
\begin{aligned}
Q_{n+1} &= \alpha r_n + (1-\alpha)\alpha r_{n-1} + (1-\alpha)Q_{n-1} \\
Q_{n+1} &= \alpha r_n + (1-\alpha)\alpha r_{n-1} + (1-\alpha)^2 Q_{n-1} \\
Q_{n+1} &= \alpha r_n + (1-\alpha)\alpha r_{n-1} + (1-\alpha)^2 \alpha r_{n-2} \\
&\quad + \cdots + (1-\alpha)^{n-1}\alpha r_1 + (1-\alpha)^n Q_1 \\
Q_{n+1} &= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} r_i. \qquad (13)
\end{aligned}
$$

Here, Q_{n+1} is based on the weighted average of past rewards, as well as the initial Q_1 estimate. The weight $\alpha(1-\alpha)^{n-i}$ is given to the reward r_i. In addition, $(1-\alpha)^n$ indicates that weight is given to reward r_i and decreases as the number of rewards increases. Algorithm 1 illustrates the flow of the bandit-based learning mechanism to update reward and Q-values.

The weights are given to the reward decay exponentially as the number of rewards increases. The value of α is the step size or the learning rate. If $\alpha = 1$, the nodes assign weightage to only the recent reward. This mechanism is known as the ERWA. The node creates a Q-table of actions and rewards. With each iteration or episode, the node updates the Q-table. The episode is a 10-s duration of the transmission interval. The Q-table is updated by exploring or exploiting the environment. In the exploration phase, the node utilizes the proposed EEX_t-based ranking mechanism, whereas during the exploitation phase, a node uses the generated Q-value using (13) for routing table entries.

The nodes use a trickle-timer-based DIO transmission mechanism to update the rank information. The DIO control packet transmission also consumes energy and computational resources. The purpose of bandit-based ranking allows nodes to select the forwarding parent with a minimum number of control overheads. During the exploitation phase, the node uses the learned Q-value to select a forwarding path. Thus, the node significantly suppresses the DIO transmission during the exploitation phase by reducing the control overhead without compromising the network performance. Algorithm 2 illustrates the node procedure to perform the action.

Algorithm 2 BEEX Framework (*Performing Action)*

Initialize trickle parameters:
set $I = I_{min}$, *counter* $c = 0$,
New interval:
set $I = I \times 2$
if $(I_{max} \leq I)$, **then**
$I = I_{max}$
end if
if (exploitation), then
find $minQ_{n+1}(a)$ IP address
suppress DIO
end if
if (exploration), then
  if (node = root node), **then**
    root rank = 1
  end if
  if (parent = null), **then**
    rank = max path cost
  end if
  if (parent ! = null), **then**
    $Rank(c_j) = h + Rank(p_i) + rank_{increase}$
    $rank_{increase} = EEX$
  end if
  if $(Rank(p_i) = 0)$, **then**
    rank = base rank (128)
  end if
  return MIN $(Base\ rank + rank_{increase})$
  embed $rank_{increase}$ in DIO
  $t_{timer} = random[I/2, I]$
  if (consistent network), **then**
    counter + +
  else,
    $I = I_{min}$
  if (t_{timer} expires), **then**
  broadcast DIO
  end if
end if

IV. Experimental Results and Discussion

The proposed system was evaluated in a simulation environment using the Contiki OS Cooja simulator [39]. The proposed BEEX mechanism is compared with the current literature [15] along with other state-of-the-art (SOTA) techniques as depicted in Figs. 4–8.

The network contains one root node and several client nodes (i.e., a DAG size of 20–100). Each node generates traffic with variable transmission rates, for example, one packet per second, one packet every 2 s, one packet every 6 s, and one packet every 60 s. We utilized system parameters according to IEEE 802.15.4 standard and Z1 mote specifications. For example, IEEE 802.15.4 PHYs layer only supports frames of up to 127 bytes. Similarly uIP, which is an open-source implementation of TCP/IP network protocol stack for microcontrollers

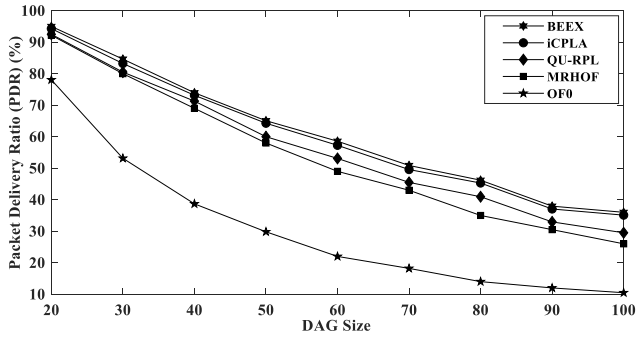

Fig. 4. PDR (%) in different DAG-sized networks.

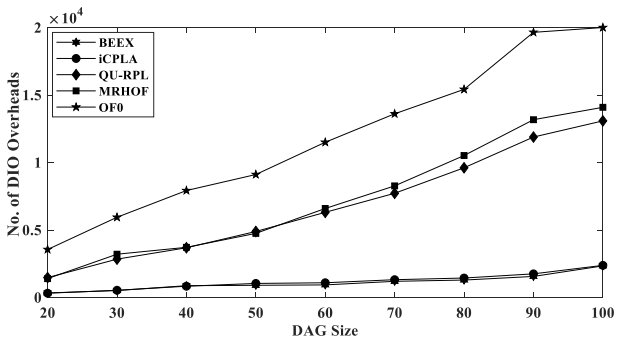

Fig. 5. Number of DIO control overheads in different DAG-sized networks.

TABLE VI
SIMULATION PARAMETERS

Parameter type	Value
DAG size	20–100
Run time	3600s
Buffer occupancy	4 packets
Packet size	127 bytes
PHY & MAC	802.15.4
Uip payload size	140 bytes
Mote	Z1
Maximum retry limits	3
Maximum backoff stages	5
I_{min}	10
I_{max}	8 doublings
RPL_min_hoprankinc (h)	128
Packets transmission	Variable
Script testlog analysis	Python 3.8

allows a maximum IP payload size of 140 bytes. In the 802.15.4 MAC layer, the initial value of the backoff exponent is 3 and it can reach a maximum of five backoff stages. The Z1 mote uses the MSP430 low-power microcontroller, with 8-kB RAM and a 92-kB flash memory, and runs the IEEE 802.15.4-compliant CC2420 transceiver. The energy parameters are also according to Z1 mote specifications. All the system parameters are according to standardized protocols and Z1 mote specifications. Table VI explains the details of the MAC and PHY layer parameters utilized during the implementation of the proposed mechanism.

Fig. 4 shows that the proposed mechanism improves the PDR, indicating that the bandit-based mechanism effectively improves the network performance. The PDR defines the ratio of packets successfully received by the sink node. Packet collisions, congestion, and other environmental factors significantly affect the PDR. We compared the PDR of the proposed mechanism with those of the SOTA, i.e., MRHOF and OF0 and QU-RPL and *i*CPLA mechanism. In addition, QU-RPL and MRHOF offer a better PDR than OF0. Here, OF0 is based only on the hop counts to forward the packets and often selects a low-performing forwarding path, whereas MRHOF uses only an ETX-based link assessment mechanism, which fails to significantly improve the network performance. Similarly, QU-RPL uses queue information along with the ETX-based link assessment and develops a slightly more reliable network than MRHOF. The BEEX is more effective than the *i*CPLA technique due to faster learning process. The proposed BEEX technique uses the bandit method to intelligently learn and predict the best choice for path forwarding while also maintaining a low DIO control overhead, as shown in Fig. 5. The proposed mechanism incurs less data loss owing to the faster learning-based selection of a forwarding path. The MRHOF and QU-RPL also offer a higher PDR than OF0. Low PDR, as in a case of OF0, creates an unstable network with high retransmissions producing higher DIOs. Estimating the DODAG construction using the BEEX mechanism has a significant impact on the network performance.

The nodes update the rank value with the transmission of the DIO control overhead. The control overhead is transmitted iteratively according to the trickle-timer mechanism. The transmission frequency of the DIO packets increases or decreases depending on the network conditions. In the *i*CPLA and proposed mechanism, the child node generates a Q-value for each potential parent node. The Q-value represents the quality of the parents in terms of the EEX metric in the proposed technique. During the proposed mechanism exploitation phase, the nodes select the best Q-value parent node using (13). The DIO packets are suppressed when the node exploits the environment. In this way, the control overhead is reduced compared to other mechanisms without degrading the performance. The proposed protocol has less DIO control overhead as compared to QU-RPL, MRHOF and OF0 (Fig. 5). The *i*CPLA and BEEX mechanisms incur almost similar control overheads while BEEX maintains lower delay that is one of the critical aspects of IoT communication. The OF0 has the highest number of DIO control packets owing to the poor selection of the forwarding path by utilizing only hop count information. Here, OF0 develops a DODAG based on the hop count. In a large-scale IoT network, such as AMI scenarios, load balancing and energy exhaustion are some of the main problems of the network. The nodes close to the root node face a high relay burden, which causes their energy resources to deplete faster, thus leading to a node failure.

Since energy is one of the main problems in IoT devices, we also compared CPU energy consumption of the proposed mechanism compared to other protocols (Fig. 6). The CPU energy consumption using the BEEX technique is lower than *i*CPLA due to less complexity. Using the BEEX mechanism, the scarce-resourced IoT devices can learn the wireless environment faster than devices in the *i*CPLA mechanism. The EEX-based OF for routing table generation, along with the

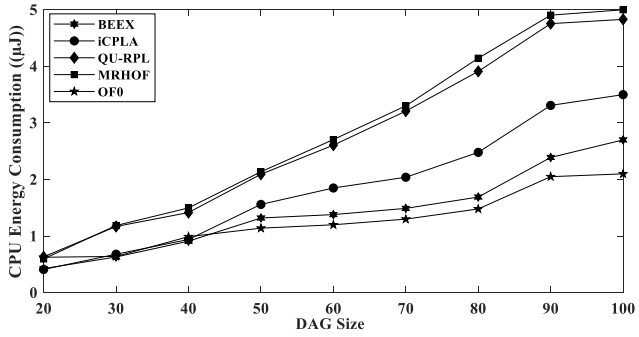

Fig. 6. Total CPU energy consumption in different DAG-sized networks.

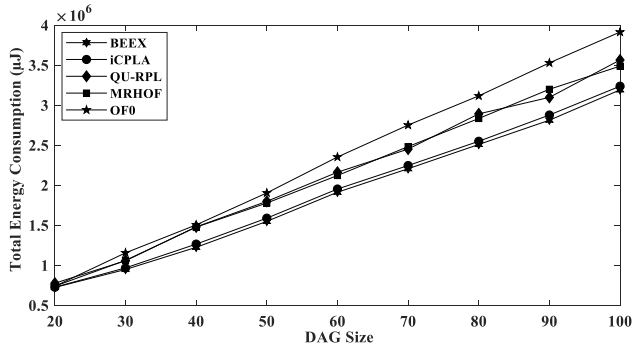

Fig. 7. Total energy consumption in different DAG-sized networks.

TABLE VII
SAMPLE OUTPUT OF A NODE CONTIKI ENERGEST VALUES

Energest_CPU	*Energest_LPM*	*Energest_Tx*	*Energest_Rx*
436563	10016916	60627	9967375
478296	10139035	88397	10169106
488167	10325771	90321	10363783
491771	10518777	90732	10559980

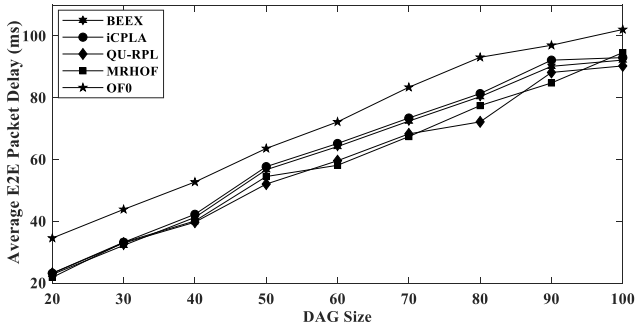

Fig. 8. Average end-to-end (E2E) packet delay (ms) in different DAG-sized networks.

simplest RL mechanism helps to reduce complexity. The complexity using MAB is $O(K)$, where K is a set of arms or slot machines (i.e., potential parents in our proposed technique). Whereas the previously proposed Q-learning mechanism had the complexity of $O(n)(a)$ (where n and a represent states and actions). It is more complex and resource-intensive to utilize Q-learning.

The reduction in the control overhead and improvement in PDR also impacts the total energy consumption of the node. The node consumes most of its energy during the communication. A valuable energy resource is consumed during all four stages of communication, that is, *LPM*, *CPU*, *Tx*, and *Rx*. To obtain tick value in each stage of Contiki OS, we used the *Energest* feature (energest flush()). The energy and power consumption of each stage are measured as follows:

$$\text{Energy } (J) = (\text{Energest} \times \text{current} \times \text{voltage})/32768. \quad (14)$$

Here, *Energest* provides the count number of rtimer ticks in each state. The sample outputs of the node *Energest* values for all four stages are depicted in Table VII; for example, the energest_CPU mote current consumption during *LPM*, *CPUI*, *Tx*, and *Rx* are 20 μA, 42.6 μA, 17.4 mA, and 18.8 mA, respectively.

Here, OF0 consumes the highest amount of energy (as shown in Fig. 7) owing to the increased control overhead. Compared to OF0, MRHOF, QU-RPL, and *i*CPLA, the proposed method reduces the total energy consumption. In the proposed method, the nodes learn, evaluate, and predict actions faster and intelligently to achieve optimal performance. If there are more collisions and congestion, the PDR is reduced, and the network becomes unstable, which leads to more DIO transmissions. Thus, OF0 incurs the highest number of control packets. The QU-RPL incurs less overhead compared to MRHOF owing to the queue information utilization, which helps balance the load. The proposed mechanism has a high PDR and low control overhead, and consumes most of the energy in the data packet transmissions. Energy-saving using the bandit framework depicts the effectiveness of the bandit-based learning approach in enhancing the network lifetime.

The average end-to-end packet delivery delay (E2E) is the amount of time the packet takes to reach the destination node from the source node. Fig. 8 shows the performance in terms of the E2E delay (in ms) of the proposed BEEX mechanism as compared to the other protocols. The E2E delay in OF0 is the highest because it relies only on the hop counts without considering the link quality. MRHOF and QU-RPL incur almost similar E2E delays. The delay in the proposed mechanism is lower compared to the *i*CPLA technique owing to the faster learning mechanism. The simulation results show an overall enhanced performance using the BEEX technique. Using the MAB mechanism in the RPL, the PDR, control overhead, and energy consumption can be improved. The results indicate that the proposed BEEX mechanism has the potential to enhance future IoT network communication.

V. CONCLUSION

The future IoT network is expected to be deployed in a densely deployed scenario where the environmental conditions are lossy and dynamic. In particular, the proliferation of the AMI infrastructure has led to an increased number of connected devices. The computational and energy capacities of sensor nodes are severely limited and as the network size and network traffic increase, the allocation of resources is becoming a challenging task. To handle such challenges, a modified network stack for IoT communication was designed. However, each IoT network stack layer was constrained. For example, the RPL routing protocol is designated for network-layer operations to support LLN characteristics. This study evaluated the bandit-learning applicability of low-power and

lossy IoT devices. In this article, the EEX was proposed as a network-layer metric and MAB-based approach for constructing a DODAG with minimum control overhead. This study presented the MAB mechanism under a dense IoT network scenario with variable data generation rates. The proposed mechanism addresses the challenge of enhancing the network performance in a dynamic and lossy environment. The proposed BEEX mechanism was evaluated using the Contiki 3.0 Cooja simulation. The simulation results indicated that the proposed mechanism improves the LLN device performance in terms of the PDR, control overhead, and energy consumption. In the future, we plan to further analyze the network performance using the upper confidence bound-based technique.

REFERENCES

[1] I. Yaqoob *et al.*, "Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 10–16, Jun. 2017.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[3] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The future of healthcare Internet of Things: A survey of emerging technologies," *Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1121–1167, 2nd Quart., 2020.

[4] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[5] V. C. Gungor *et al.*, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[6] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of Things for the future of smart agriculture: A comprehensive survey of emerging technologies," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 4, pp. 718–752, Apr. 2021.

[7] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4322–4334, Jun. 2021.

[8] Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, "Internet of Things-aided smart grid: Technologies, architectures, applications, prototypes, and future research directions," *IEEE Access*, vol. 7, pp. 62962–63003, Apr. 2019.

[9] A. Musaddiq, Y. B. Zikria, Zulqarnain, and S. W. Kim, "Routing protocol for low-power and lossy networks for heterogeneous traffic network," *EURASIP J. Wireless Commun. Netw.*, vol. 21, no. 1, pp. 1–23, Jan. 2020. [Online]. Available: https://doi.org/10.1186/s13638-020-1645-4

[10] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir, and S. W. Kim, "A survey on resource management in IoT operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, Feb. 2018.

[11] T. Winter *et al.*, "RPL: IPv6 routing protocol for low power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 6550, 2012.

[12] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," Internet Eng. Task Force, Fremont, CA, USA, RFC 6719, Sep. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6719.txt (accessed May 14, 2021).

[13] P. Thubert, "Objective function zero for RPL," Internet Eng. Task Force, Fremont, CA, USA, RFC 6552, 2012.

[14] E. Alpaydm, *Introduction to Machine Learning*, 3rd ed. Cambridge, MA, USA: MIT Press, 2014.

[15] A. Musaddiq, Zulqarnain, Y. A. Qadri, R. Ali, and S. W. Kim, "Reinforcement learning-enabled cross-layer optimization for low-power and lossy networks under heterogeneous traffic patterns," *Sensors*, vol. 20, no. 15, p. 4158, Jul. 2020.

[16] A. Slivkins, "Introduction to multi-armed bandits," 2019, *arXiv:1904.07272*.

[17] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 964–979, Apr. 2017.

[18] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," Internet Eng. Task Force, Fremont, CA, USA, RFC 6206, 2011.

[19] A. Musaddiq, Y. B. Zikria, and S. W. Kim, "Energy-aware adaptive trickle timer algorithm for RPL-based routing in the Internet of Things," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Sydney, NSW, Australia, 2018, pp. 1–6.

[20] K. S. Bhandari, A. Hosen, and G. Cho, "CoAR: Congestion-aware routing protocol for low power and lossy networks for IoT applications," *Sensors*, vol. 18, no. 11, p. 3838, Nov. 2018.

[21] S. Taghizadeh, H. Bobarshad, and H. Elbiaze, "CLRPL: Context-aware and load balancing RPL for IoT networks under heavy and highly dynamic load," *IEEE Access*, vol. 6, pp. 23277–23291, Apr. 2018.

[22] A. Zier, A. Abouaissa, and P. Lorenz, "E-RPL: A routing protocol for IoT networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[23] J. Nassar, N. Gouvy, and N. Mitton, "Towards multi-instances QoS efficient RPL for smart grids," in *Proc. 14th ACM Symp. Perform. Eval. Wireless Ad Hoc Sensor Ubiquitous Netw.*, Miami, FL, USA Nov. 2017, pp. 85–92.

[24] H. Lamaazi and N. Benamar, "OF-EC: A novel energy consumption aware objective function for RPL based on fuzzy logic," *J. Netw. Comput. Appl.*, vol. 117, pp. 42–58, Sep. 2018.

[25] P. Fabian, A. Rachedi, C. Gueguen, and S. Lohier, "Fuzzy-based objective function for routing protocol in the Internet of Things," in *Proc. IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, 2018. pp. 1–6.

[26] B. Ghaleb, A. Al-Dubai, E. Ekonomou, W. Gharib, L. Mackenzi, and M. B. Khala, "A new load-balancing aware objective function for RPL's IoT networks," in *Proc. IEEE 20th Int. Conf. High-Perform. Comput. Commun. IEEE 16th Int. Conf. Smart City IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Exeter, U.K., 2019, pp. 909–914.

[27] F. Wang, E. Babulak, and Y. Tang, "SL-RPL: Stability-aware load balancing for RPL," *Trans. Mach. Learn Data Min.*, vol. 13, no. 1, pp. 27–39, 2020.

[28] 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, pp. 1–13, Nov. 2017.

[29] M. Aziz, "On multi-armed bandits theory and applications," Ph.D. dissertation, Khoury College Comput. Sci., Northeastern Univ., Boston, MA, USA, 2019.

[30] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," 2nd ed. Cambridge, MA, USA: MIT Press, 1998.

[31] D. Bzdok, M. Krzywinski, and N. Altman, "Points of significance: Machine learning: Supervised methods," *Nat. Methods*, vol. 15, no. 1, pp. 5–6, 2018.

[32] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *Proc. 4th Comput. Intel. Commun. Netw. (CICN)*, 2012, pp. 140–144.

[33] Z. Uddin, A. Ahmad, M. Iqbal, and M. Naeem, "Applications of independent component analysis in wireless communication systems," *Wireless Pers. Commun.*, vol. 83, no. 4, pp. 2711–2737, 2015.

[34] Y.-A. L. Borgne, S. Raybaud, and G. Bontempi, "Distributed principal component analysis for wireless sensor networks," *Sensors*, vol. 8, no. 8, pp. 4821–4850, Aug. 2008.

[35] A. S. Mignona and R. L. A. Rochaa, "An adaptive implementation of ε-greedy in reinforcement learning," *Procedia Comput. Sci.*, vol. 109, pp. 1146–1151, 2017. [Online]. Available: https://doi.org/10.1016/j.procs.2017.05.431

[36] D. S. J. D. Couto, D. Aguayu, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, San Diego, CA, USA, Sep. 2003, pp. 134–146.

[37] J. Banerjee, S. K. Mitra, and M. K. Naskar, "Comparative study of radio models for data gathering in wireless sensor network," *Int. J. Comput. Appl.*, vol. 27, no. 4. pp. 49–57, 2008.

[38] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, p. 10.

[39] Contiki, Anaheim, CA, USA, "Contiki: The Open Source Operating System for the Internet of Things," 2015. [Online]. Available: http://www.contiki-os.org/ (accessed May 14 2021).

[40] Zolertia, Barcelona, Spain, "Z1 Datasheet," 2010. [Online]. Available: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf (accessed May 14 2021).