Sustainable Cities and Society 59 (2020) 102234

Contents lists available at ScienceDirect

Sustainable Cities and Society

journal homepage: www.elsevier.com/locate/scs

Intelligent learning automata-based objective function in RPL for IoT

Ahsan Saleem[a], Muhammad Khalil Afzal[a], Muhammad Ateeq[a], Sung Won Kim[b], Yousaf Bin Zikria[b,*]

[a] Department of Computer Science, COMSATS University Islamabad, Wah Campus 47040, Pakistan
[b] Department of Information and Communication Engineering, Yeungnam University, 280 Daehak-Ro, Gyeongsan, Gyeongbuk 38541, Republic of Korea

ARTICLE INFO

Keywords:
Internet of Things
RPL
Objective function (OF)
Learning automata
Smart grid (SG)
Link estimation
Intelligent routing

ABSTRACT

Sustainable cities are widely adopting the standards of the Internet of Things (IoT) in almost every domain, e.g., smart grids (SG) to provide services to a sustainable community. It enables two-way communication to manage the energy resources, where routing protocol has a significant role in communication. The diversification of IoT networks arises many challenges for the routing protocol for low power and lossy networks (RPL). The dynamic and lossy environment is one of the key challenges in various IoT networks, specifically SG. RPL does not able to adjust its link metric efficiently against the dynamic and lossy environment, which have a great impact on the performance metrics. To address this issue, we have introduced cognition in RPL by integrating learning automata with the objective function (LA-OF). Learning automata (LA) is applied to expected transmission count (ETX) to tune it according to the environment. LA learns through interacting with the environment and yields the best ETX values, afterwards the environment is monitored to trace down the instability in the environment. The proposed LA-OF is compared with standardized techniques MRHOF and OF0. The simulation results show a significant improvement with overall 7.04% in PRR, 17.52% in energy consumption, and 18.72% in overhead.

1. Introduction

With the growing population, the utilization of resources has become a critical factor to be considered for the world. To overcome this factor, the concept of sustainable cities has been introduced, which has three main pillars economic, social, and environment development (ten Have & Gordijn, 2020). The urban cities are transforming into smart sustainable cities, where the concept of machine-to-machine (M2M) communication and tactile networks are widely adapted (Bibri & Krogstie, 2017). The integration of M2M communication in every aspect of life has been seen from the past few decades that affect human life unimaginably because it helps to improve efficiency and manageability of systems without any human interaction.

The M2M networks may consist of millions of devices and mostly integrate compact devices due to the economic constraints of a sustainable community, where each devices have to connect with the network. To address these global requirements, the researchers come up with the concept of the IoT (Al-Turjman, 2020; Gubbi, Buyya, Marusic, & Palaniswami, 2013; Sethi & Sarangi, 2017). IoT is like a big umbrella that can enable each entity in the world to connect and communicate over the Internet without human interaction. IoT networks are based on Internet Protocol version 6 (IPv6), so they can able to assign a unique identification (ID) to each device, which permits devices to transmit data over the Internet. There are a variety of sustainable cities applications that are adapting the IoT standards, e.g., smart grids (SG), smart cities, industry, agriculture, hospital, transportation, etc. (Al-Turjman & Malekloo, 2019; Chen, Xu, Liu, Hu, & Wang, 2014; Lee & Lee, 2015; Sailaja & Rohitha, 2018; Schulz et al., 2017). According to stats of Intel around 200 billion devices will be connected to IoT by 2020 (Intel, 2018).

Smart sustainable cities strongly rely on wireless sensor networks (WSNs) (Abujubbeh, Al-Turjman, & Fahrioglu, 2019; Bibri, 2018; Yick, Mukherjee, & Ghosal, 2008) for the environmental and economic development of the cities. WSNs plays a vital role in sustainability, as it makes the physical system manageable without any human interaction, flexible, reliable, and redundant. The IoT based sensors are compact devices that are embedded with a small processing unit, battery-powered, and few kilobytes of memory to make them compact, portable, and affordable. These hardware constraints raise many challenges for the low power and lossy networks (LLNs). To deal with them, a modified network stack has been presented in RFC 8352 for IoT, as shown in Fig. 1. Furthermore, lightweight operating systems (Javed, Afzal, Sharif, & Kim, 2018) like Contiki, RIOT, TinyOS, etc. have been proposed for IoT devices to manage the limited resources efficiently and

* Corresponding author.
E-mail address: yousafbinzikria@ynu.ac.kr (Y.B. Zikria).

https://doi.org/10.1016/j.scs.2020.102234
Received 7 October 2019; Received in revised form 15 April 2020; Accepted 26 April 2020
Available online 04 May 2020

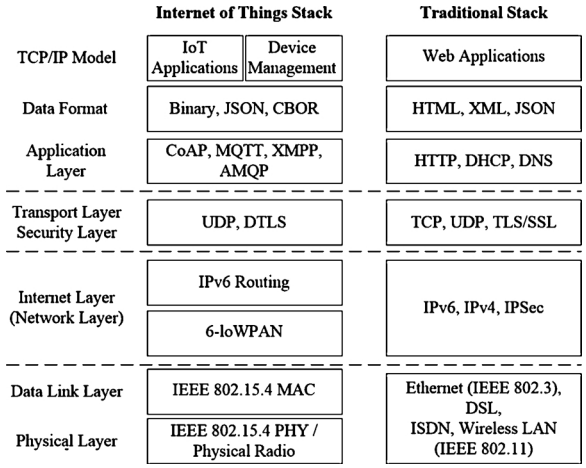

Fig. 1. Comparison of Internet of Things vs traditional network stack.

effectively. Each layer of IoT stack has its constraint for the LLNs.

IoT based WSNs generate the bulk of data collecting from different resources of sustainable cities, each associated node with the IoT network has to dissipate their data to a sink or root node. The Internet layer has the responsibility to define the path to nodes for the transmission of data, where routing protocols define the rules and regulations for the creation and selection of a path from nodes towards the root node. Taking the requirements of LLNs into the consideration, Internet Engineering Task Force (IETF) presented the IPv6 based routing protocol for low power and lossy network (RPL) (Hui & Vasseur, 2012). Routing protocols make decisions on some parameters or functions. RPL creates logical tree topology based on ranks, where rules for routing and calculation of rank are defined in objective function (OF) (see the section for more details). OF negotiate among link and/or node metrics for the calculation of rank, and the selection of preferred parent nodes (it is used for routing). The standardized OF performs these tasks on the bases of link metrics.

1.1. Motivation

IoT is rapidly converging in every field of a smart sustainable community. Each of the IoT based sustainable city application has its challenges and constraints, while many of them have to encounter with environmental factors. Specifically, smart grids where the dynamic and lossy environment dramatically affects the IoT based WSNs. According to the phenomena of physics, the environment has a direct impact on wireless communication. Moreover, next-generation technologies are more intelligent and self-sustaining as they adjust their performance according to the environment. The integration of the new generation in real-world applications is essential to make them self-sustainable. So, by integrating learning algorithm in RPL make them self-sustainable according to the environment.

1.2. Contribution

In the IoT based WSNs, RPL protocol is used for routing, which extensively takes the decision (defined in OF) based on the link metric. The standardized OF is unable to perform efficiently and effectively due to effect of environmental constraint on the link metric (ETX), which in result degrade the performance of the network in term of packet delivery ratio, energy consumption, and control overhead. The self-sustainability of link metrics, according to the environment, is one of the open issues of IoT. To deal with this issue, we propose a new learning automata-based OF (LA-OF). In LA-OF, LA is integrated with each node, to individually and recursively examines the environment. It learns and tunes the link metric (ETX) through interacting with the environment because the routing decision of RPL is based on the link metric. Each node in a network learns the environment and yield the best link metric, and update the preferred parent table accordingly. Then the routing strategies are made on the bases of tuned link metrics. Moreover, in offline learning mode, nodes continuously learn the environment without updating the link metric until the instability in the environment has been traced.

The rest of the article is comprised of the following sections. Section 2 provides a brief introduction about RPL and then discusses the related work. Section 3 presents the problem statement, motivation for this research, and proposes the solution. Section 4 discusses the performance evaluation and analysis of the proposed OF with standardized OFs. Finally, Section 5 concludes the paper. Whereas, Table 1 presents the list of abbreviations used in this article.

Table 1
List of abbreviations.

Symbol	Description
ACK	Acknowledgement
DAO	Destination advertisement object
DAO-ACK	DAO acknowledgement
DIO	DODAG information object
DIS	DODAG information solicitation
DODAG	Destination oriented directed acyclic graph
EC	Energy consumption
ETX	Expected transmission count
HC	Hop count
IETF	Internet Engineering Task Force
ID	Identification
IoT	Internet of Things
IPv6	Internet Protocol Version 6
LA	Learning automata
LA-OF	Learning automata based objective function
LLNs	Low power and lossy networks
LQE	Link quality estimation
M2M	Machine-to-machine
MAB	Multi-arm bandit
MRHOF	Minimum rank hysteresis objective function
NAN	Neighborhood area network
OF	Objective function
OF0	Objective function zero
RL	Reinforcement learning
PDR	Packet delivery ratio
PL	Packet loss
PRR	Packet reception ratio
QU	Queue utilization
QL	Queue length
RE	Remaining energy
RPL	Routing protocol for low power and lossy networks
RSSI	Received signal strength indicator
SG	Smart grid
WSNs	Wireless sensor networks
UDGM	Unit disk graph medium

2. Related work

2.1. Background

This section briefly discusses the RPL: types of control messages used in RPL, objective functions, and RPL topology.

2.1.1. RPL control messages

There are four types of control messages, as shown in Fig. 2 are exchanged between nodes to build and maintain the topology:

1. DODAG information solicitation (DIS) exchanges when a node wants to join the network by probing its neighbor for the nearest DODAG.
2. DODAG information object (DIO) messages are exchanged to build and maintain the topology, and it is generated by parent nodes

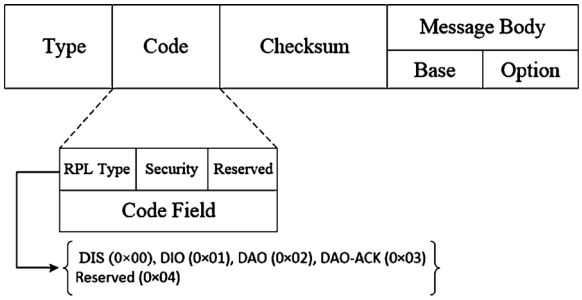

Fig. 2. Control messages of RPL.

containing its rank, DODAG version number, RPL instance ID, etc., while Trickle Timer (Levis, Clausen, Hui, Gnawali, & Ko, 2011) is used to transmit DIO messages periodically.

3. Destination advertisement object (DAO) is used to transmit destination information upward. In storing mode DAO send to preferred parent, while in non-storing mode, DAO sends towards the root.
4. DAO acknowledgment (DAO-ACK) is the response of the DAO message sent by DAO recipients.

2.1.2. Objective function (OF)

An objective function is an essential attribute in RPL. Routing decisions of RPL are based on OF, and it defines the rules for rank calculation and parent selection in RPL topology. The routing in RPL is done by using rank and parent selection, and each node assigns a rank based on the rules defined in OF and node with low-rank act as a parent and higher ranked nodes as a child. Child nodes maintain the parent table, and selection of the preferred parent is made according to the rules defined in OF. There are two standardized OFs objective function zero (OF0) and minimum rank with hysteresis objective function (MRHOF).

OF0 (Thubert, 2012) calculates the rank on Hop Counts by adding constant value in rank at each hop according to Eq. (1). OF0 just consider the relative distance from the root node for rank calculation rather than considering link condition, latency, or any other parameter, and the node will select the parent having lowest hop distance.

$$\text{Rank} = \text{parent\_r ank} + \text{rank\_i ncrease}, \tag{1}$$

where rank_i ncrease is the constant value.

MRHOF (Gnawali & Levis, 2012) considers the link metric (e.g., expected transmission count (ETX) or latency) for rank computation. It calculates the rank by using Eq. (2), which adds ETX value in the parent rank. It is also a default OF used in RPL.

$$\text{Rank} = \text{parent\_r ank} + \text{path\_c ost}, \tag{2}$$

where path_c ost is the measurement of link metric.

2.1.3. RPL topology

RPL (Hui & Vasseur, 2012) is the distance vector routing algorithm that creates logical typologies based on destination-oriented directed acyclic graphs (DODAGs). DODAG has a tree-like structure that has one or more root nodes that act as a sink or gateway node, and the remaining topology comprises the pairs of parent and the child nodes. The nodes with lower rank act as a parent, while the higher-ranked nodes are associated with children. The child nodes have information about all possible parents while the parent has no information about associated child's. The root node starts the DODAG construction by broadcasting control message (DIO) containing rank, instance ID, DODAG version, etc. Root node had always had *rank 0* so that traffic should be routed toward that node. Every node which receives DIO message update the rank field using MHROF and broadcast the updated DIO. This process continues until every single node receives the DIO. Each node maintains its parent table, which contains the entry of directly connected nodes with lower ranks than its own rank. The node will select the preferred parent, and selection is based on rules defined in objective function MRHOF. RPL can support multiple DODAG's and instances in a single network. Each instance has a unique ID and allows to implement multiple OF's within a network. Nodes with the same instance ID share common OF, and the nodes can belong to multiple instances. RPL has two types of mechanisms for topology management which are; local repair and the global repair.

1. Local repair is called by child node when there is inconsistency in the network such as link failure, loop detection, etc.
2. Global repair is the rebuilding of the whole topology triggered by the root node by increasing the version of DODAG when the local repair will not able to solve the issue.

There are two modes of operation in RPL for packet forwarding; storing mode and non-storing mode:

1. Storing mode in which the parent stores the DAO and regenerate the DAO by combining previous DAO and route information.
2. Non-storing mode parent will not store the DAO and only add their route information.

RPL can support three types of traffic: multipoint-2-point (MP2P) mostly adapted type of traffic used for transmitting traffic upward from child nodes toward the root, point-2-multiPoint (P2MP) for transmitting downward traffic toward child nodes from the root node and point-2-point (P2P) which is used for the communication between two nodes.

2.2. Existing enhancements in RPL

The rules for routing in RPL are defined in OF; however, it is not restricted to only use the standardized OF. Therefore, RPL provided the room for researchers to enhance or develop the OF according to the requirement. Different routing approaches (Kamgueu, Nataf, & Ndie, 2018) have been proposed in order to improve the performance of the network by optimizing different performance parameters, e.g., packet delivery ratio, throughput, power consumption, overhead, etc.

In the wireless network, the exact measure of link quality is an important factor for communication. To improve measuring procedure Ancillotti et al. proposed a reinforcement learning (RL) based link quality estimation (LQE) strategy for RPL (RL-Probe) (Ancillotti, Vallati, Bruno, & Mingozzi, 2017). RL-Probe used both asynchronous and synchronous LQE. In asynchronous LQE, it has both proactive and reactive phases; in a proactive phase, it measures the trend of the received signal strength indicator (RSSI) alongside ETX, while in the reactive phase, it performs an immediate local repair of RPL. In synchronous LQE, it classified nodes into clusters and apply multi-arm bandit (MAB) (Aziz, 2019) for probing procedure to prioritize specific groups to improve probing. In RL-probe author optimized the probing procedure; however, the tuning of the link metric has not been considered. Moreover, clustering increased the control overhead.

In context-aware and load balancing RPL (CLRPL) for IoT networks under heavy and highly dynamic load (Taghizadeh, Bobarshad, & Elbiaze, 2018), authors present new OF named as context-aware objective function (CAOF). It computes the rank based on the remaining energy, ETX, and rank of the parent; moreover, they also address the thundering herd problem. They also proposed context-aware routing metric (CARF) for load balancing of routes as well as power balancing in parent's chain. CARF focused on the utilization of the queue and the remaining energy of the parent chain rather than a single parent. They also overcome the problem of equality illusion by selecting the best parent based on CARF. The proposed scheme improves energy consumption and decreases the packet loss ratio but increases the DODAG information object (DIO) overhead.

In congestion-aware RPL (CoAR) (Bhandari, Hosen, & Cho, 2018),

authors address the issue of congestion at parent nodes because of the non-symmetric distribution of child nodes. They introduced a new OF named congestion aware objective function (CoA-OF) based on multi-criteria decision making (MCDM) to overcome the congestion. CoA-OF is based on the technique for order preference by similarity to ideal solution (TOPSIS) (Papathanasiou & Ploskas, 2018), which considers three metrics (ETX, queue utilization (QU) and RE) for parent selection. CoAF also presents an adaptive threshold technique for congestion detection by measuring the buffer occupancy based on the past and present traffic. CoAR improves packet delivery ratio, throughput, and energy consumption while on the other hand, it increases control overhead and frequent parent changes in high traffic scenarios.

Author's in (Nassar, Gouvy, & Mitton, 2017) proposed quality of service (QoS) based multi-objective (OFQS) OF to meet the requirements of SG. OFQS automatically adapts the multiple instances according to the requirement of SG. The routing decisions of OFQS are based on three metrics: ETX, delay, and power state. They classify the nodes into three power states based on their remaining energy. On the bases of those three metrics, OFQS allocates the weight to each route. Additionally, they categorized the traffic into three types: critical, non-critical, and periodic. Critical traffic chooses the route, which has a minimal delay (between 1s to 30s) and reliability of greater than 99.5%. While non-critical traffic chooses route having the maximum delay and reliability of 98%. Periodic traffic follows the path which has a moderate delay (about 5min to 4hours) and having a reliability of 98%. The proposed solution improves end-to-end delay, PDR, and network lifetime. However, their tuning parameters are fixed and cannot deal with the dynamic network. It can be solved by employing machine learning algorithms.

In (Lamaazi, El Ahmadi, Benamar, & Jara, 2019), the authors are intended to provide a solution for applications that required data reliability, efficient energy consumption, and guaranteed delivery of real time data. The proposed flexible OF, where forwarder is selected through the combined result of three metrics: ETX, EC and forwarding delay. It calculates the composite additive metric, where each metric has defined weight. The ratio of weight depends on the application or type of traffic. The proposed technique is applied to the parent table. The composite additive metric is calculated against each entry of the parent table and reconstructs the parent table based on the calculated metric. The proposed technique improves the PDR and EC, while it increases the overhead.

Chaotic genetic algorithm (CCG) (Cao & Wu, 2018) is an improved version of RPL. The main objective of this algorithm is to enhance parent selection mechanism using chaos and genetic algorithm. With the help of the ergodicity of a chaotic algorithm, CGA increases the search by using global search quality of genetic algorithm to find the optimal solution. They create a composite metric (CM) by utilizing queue length (QL), end-to-end delay, residual energy ratio, hop count and ETX, and assign a weight to each metric. Weighting factors in CM are optimized by a chaotic genetic algorithm to select the best parent. This algorithm improves the average success ratio, end-to-end delay, and residual energy. However, the network overhead has not been considered in this research.

In Lamaazi and Benamar (2018), new objective function named as OF-EC is proposed. OF-EC employs a fuzzy logic technique to make routing decisions. OF-EC use ETX and energy consumption (EC) as a combined metric. The proposed OF can reduce energy consumption as well as packet loss by selecting the best parent. However, it increases the frequently parents change ratio.

In Bahramlou and Javidan (2018), the author integrates the aggregation technique to efficiently utilize limited resources. Firstly, the authors designed the appropriate aggregation technique, which merges the correlated data to reduce resource consumption by limiting the data packets. They also introduced the trigger function to observe the environment and select the less congested parent. Furthermore, the number of children is also used for rank calculation. This approach improves the DIO overhead, packet retransmission ratio, PDR, and energy consumption, but it increases the congestion at the parent in dense traffic scenarios.

E-RPL has been presented in Zier, Abouaissa, and Lorenz (2018) to meet the QoS routing criteria and decrease the control overhead in the network. To control the DIO overhead for ETX, they limit the nodes to wait for DIO. Sink node generates the DIO packet at first instance rather than waiting for DIS or DAO packet. On the other hand, nodes wait for DIO from their neighbors rather than sending DIS, if they did not receive it then nodes generate DIS for the DIO packet. They also proposed a new OF named as multi-constraint OF. E-RPL uses energy and delays with random weighted to calculate the node rank. Their contribution improves energy consumption as well as an end-to-end delay. However, it increases the network convergence time.

In Fabian, Rachedi, Gueguen, and Lohier (2018), authors present the new OF using the fuzzy rules to dynamically adapts the environment. They apply fuzzy logic by using ETX and energy consumption. For rank calculation, they define three cases. First, when the battery level is above the threshold value, ETX is used for rank computation. Second, when the battery level is less than the threshold, then the combined metric of ETX and the remaining energy is used for rank computation. Third, when the battery is empty, then this node is eliminated. The proposed objective function shows improvement in PDR and throughput, but it increases energy consumption.

In Ghaleb et al. (2018), a load balancing mechanism is used to select the parent and improves network reliability. They used the number of children along with the ETX metric for parent selection. To monitor the load, each node generates a child list (CHlist) by analyzing the data packet. The expiration of Trickle timer will cause a frequent change in load balancing information. To avoid this, they introduced fast propagation timer along with Trickle timer to update the CHlist. The rank is calculated by standardized metric (e.g., ETX or HC depends on application) and if the node has more than one parent with the same rank. The CHlist will be checked, and a node having fewer children is selected as a preferred parent. Balancing Timer is introduced to avoid the frequent changing of parents. Their contribution improves the PDR and energy consumption, but it increases network convergence time, and fast propagation may cause looping in the network.

Table 2 provides a summary of related work.

To provide the QoS for applications that are delay-sensitive and require reliability, the author presents new opportunistic fuzzy logic-based objective function (OOP-OF) (Kechiche et al., 2019). OOP-OF considered three metrics for the parent node: ETX, HC, and children nodes. The fuzzy set takes three metrics (HC, EX, and CN) as an input, which are evaluated based on defined rules and aggregated. The defuzzification is performed to get the output from the aggregated set, and then the parent table is reconstructed on those outputs. It improves the PDR and delay, while it will increase energy consumption.

3. The proposed work

In the proposed work, an application of smart sustainable cities is considered, specifically, IoT based WSNs for Neighbor Area Network (NAN) in Smart Grid (SG), as shown in Fig. 3. SG (Yan, Qian, Sharif, & Tipper, 2012) system is a new generation of the electrical grid to facilitate the sustainable community. It enables the monitoring and manageability of intelligent electrical systems remotely. SG architecture comprises of home area networks (HANs), neighborhood area networks (NANs), wide area networks (WANs) and SG management system. HANs is the end network consists of intelligent devices, sensors, actuators, etc. which collect the data from different appliances (e.g., entertainment, lights, air-container, water management, security system, door locks, and heat) or devices. The collected data have to be forward to the SG management system for processing. NANs are the second layer of SG, as shown in Fig. 3, which comprise of smart devices belonging to multiple HANs. NANs support communication between

Table 2
Existing enhancement in routing protocol for low power and lossy network.

Ref.	Objective of research	Metrics	Method	Improvement	Shortcomes	Simulation tool
Ancillotti et al. (2017)	To predict link quality	ETX, RSSI	Multi-Arm Bandit (MAB)	Decrease in packet delivery ratio	Increases control overhead	Cooja and IoT testbed
Taghizadeh et al. (2018)	To improve the lifetime and packet loss at high-speed	ETX, parent's rank, remaining power	CAOF and CARF OF's	Load balancing and improves networks lifetime	DIO overhead is increased	Cooja
Bhandari et al. (2018)	To overcome the congestion problem	ETX, QU, NI	TOPSIS	Improves PRR, end-to-enddelay, PL and EC	Frequent parent changes at high traffic, overhead	Cooja with Contiki 2.7
Nassar et al. (2017)	To provide quality of service for smart grid	ETX, Delay, Power Status	mOFQS	PDR, end-to-end delay, network lifetime	Fixed tuning parameters, route redundancy issue	FIT-IoT lab
Lamaazi et al. (2019)	To develop a protocol for reliability, efficient energy consumption and real data	ETX, EC Forwarding delay	weighted metric	PDR and energy consumption	Increase packet overhead	Cooja with Contiki 3.0
Cao and Wu (2018)	Use weighted distribution theory to select best parent	QL, HC, ETX, RE, end-to-end delay	Chaotic genetic algorithm	End-to-end delay, throughput, residual energy	Network overhead increased	-
Lamaazi and Benamar (2018)	To select the best route using combine metrics	ETX, hope count, EC	Fuzzy Logic	Increase PDR, lifetime and reduced overhead	Combine the metrics but didn't tuned them.	Cooja with Contiki 2.7
Bahramlou and Javidan (2018)	To efficiently utilize the resources	No. of child's, ETX,	Aggregation	DIO overhead, retransmission, PDR	Congestion in dense environment	Cooja
Zier et al. (2018)	To decrease the control overhead	Energy, Delay, ETX	Multi-constraint OF	Energy consumption, end-to-end delay and routing overhead	Network convergence time might increase	Cooja
Fabian et al. (2018)	To dynamically adapts the environment	ETX, HC, RE	Fuzzy Logic	Throughput, PDR	Energy consumption increased.	Cooja
Ghaleb et al. (2018)	To improve network reliability thorough load balancing	HC, ETX, No. of child's	Load balancing	PDR, load balancing and EC	Increase convergence time and may subject to looping	Cooja
Kechiche, Bousnina, and Samet (2019)	To provide QoS for delay sensitive applications	ETX, HC, CN	Fuzzy Logic	Delay, PDR	Energy consumption increased.	Cooja

different smart meters (e.g., these are the devices that collect the data from HANs and aggregate it), which forwards and disseminates the data to the SG management system. The data are forwarded periodically to the gateway to monitor and manage the devices. To effectively and efficiently manage those devices, packet reception ratio, latency, and energy consumption in the NANs are important performance parameters to be considered.

In RPL, the link metric (ETX) has a direct impact on these performance parameters, because IoT adopts a connection-less transport protocol. To ensure the delivery of packets, RPL gets feedback from the MAC layer by counting the expected number of transmission (ETX) required for the successful delivery of the packet. The exact estimation of ETX increases the ratio of successful transmission, which in result improves the PRR and avoids the redundant transmission (unwanted retransmission and control packets), which may consume extra energy. In NANs, the environment is not very much stationary, as many factors affect radio transmission (e.g., weather, electromagnetic devices, and temporary events which may be an obstacle, etc.). These factors greatly affect the link metric, where RPL couldn't be able to measure the exact estimation of ETX, so tuning the ETX according to the environment will help the nodes to have the exact ETX. To tune the desire ETX values according to the environment, learning automata is integrated with OF because of its adaptive nature and lightweight algorithm in terms of resource utilization.

3.1. Learning automata

Learning automata (LA) (Narendra & Thathachar, 2012) is known for its adaptive nature, as it adapts the changes according to the environment. LA is generally described as a learning model that interacts with the random environment repeatedly and selects the optimal action for the system based on reward and penalty mechanisms. The learning process consists of two stages; firstly, the chosen action is tested in a random environment that generates the reinforcement signal indicating, whether the selected action is favorable or not. After that, the agent (LA) on the base of the reinforcement signal updates its internal parameters as a probability vector. This learning cycle continues until the termination condition occurs, where the optimal value is selected, which yield the highest probability. Table 3 lists the notations used in the equations.

Formally (Narendra & Thathachar, 1974; Unsal, 1998), the random environment is represented as $E = \alpha, \beta, c$ where $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$ is the set of output for automaton and input for the environment, $c = c_1, c_2, \ldots, c_n$ is the probability vector also called penalty probabilities, and $\beta = \beta_1, \beta_2, \ldots, \beta_n$ is the set of input for automaton and output for the environment. The random environment is classified into three models; P-model, where the reinforcement signal is either 0 or 1, in Q-model reinforcement signal, is from interval {0, 1} having finite values, and the third is an S-model where the reinforcement signal is from the interval {0, 1} having infinite values.

LA updates the probabilities of action based on the reinforcement signal received form the environment. Some different norms or practices are used for updating the probabilities. Evaluate the behavior on receiving an average penalty of LA:

$$M(n) = E\{x(n)|p(n)\},$$
$$p(n) = \sum_{i=1}^{r} p_i(n)c_i, \tag{3}$$

where $p(n)$ is the average penalty condition of action α_i at stage n having a probability p_i.

LA has two types of working structures. Fixed-structured LA: in which transition probabilities are constant. Variable structured LA: where transition probabilities are updated at each iteration. Variable-structured LA is used in this research that can be represented as:

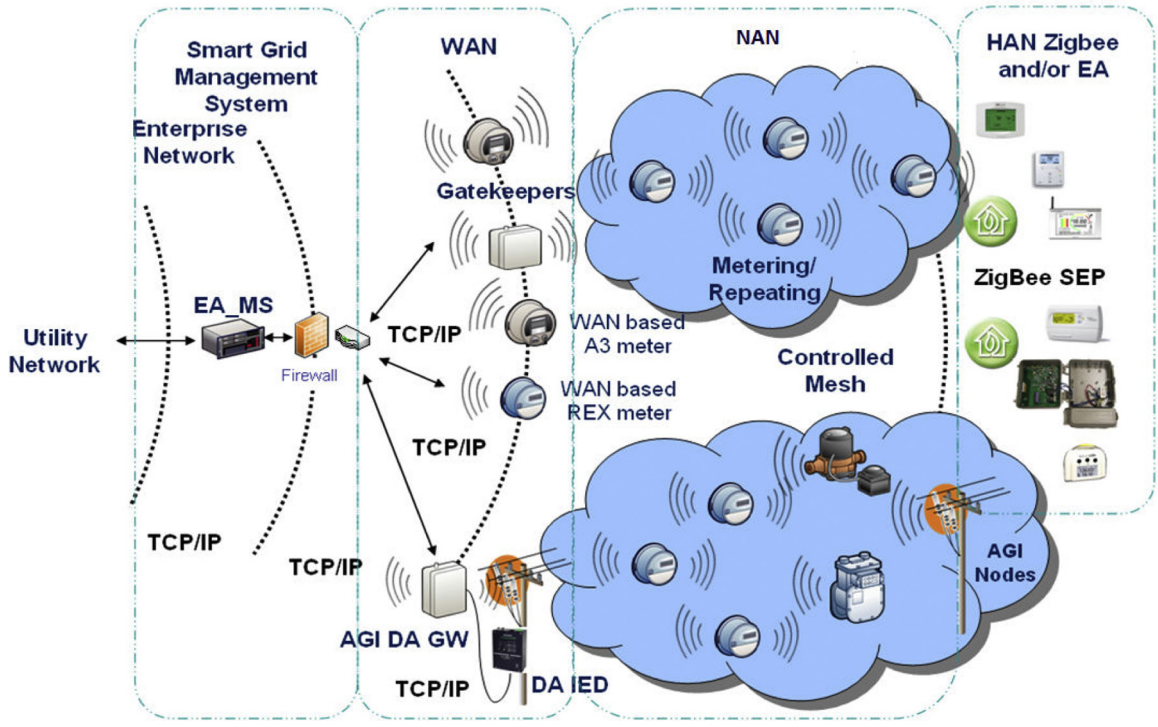

Fig. 3. IoT based network architecture of smart grid.

Table 3
List of notations.

Symbol	Definition
α	Set of inputs for automaton
β	Set of outputs from automaton
a	Reward factor
b	Penalty factor
c	Penalty Probability
E	Random environment
$p(n)$	Average penalty condition
r	Number of actions
T	Mapping Function

$$p(n+1) = T[p(n), \alpha(n), \beta(n)], \tag{4}$$

where T is the mapping function, α is the action, β is the reinforcement signal from the environment, and $p(n)$ is the probability vector. Moreover, the reinforcement scheme is said to be linear if $p(n+1)$ is a linear function of $p(n)$ otherwise nonlinear.

The reinforcement learning is categorized into three learning schemes; linear, non-linear, and hybrid. The linear reinforcement learning scheme is applied in this research, where at each iteration, the probabilities are updated at a constant rate. It has three learning functions: linear reward-penalty (LR-P), where getting a reward or penalty on a given action, will result in an equal penalty or reward on other actions. linear reward-inaction (LR-I) was getting a reward or penalty on a given action, which will result in no change in the probabilities of other actions. Linear Reward-Penalty (LRεP) with $\varepsilon < 1$ where getting a reward or penalty on a given action will result in penalty or reward on other actions but a small amount.

3.2. Proposed LA-OF

Every node in a network measures ETX periodically, so the tuning of the link metric should be done independently. In this case, leaning automata is the most suitable type of learning approach because of being a lightweight algorithm in terms of resource utilization and its adaptive nature. Furthermore, it does not necessarily require a definite starting state or point and runs on any configuration. It learns through interacting with the environment at run-time and tunes the parameter according to the environment under less processing overhead.

LA is integrated into the nodes to fine-tune the ETX to provide the exact estimation of transmission. The rationale behind this scheme is the independent tuning of ETX takes to redundant estimation, especially in an open environment. The LA-based system has two types of parameters; controllable parameters and observable parameters.

1. Controllable parameters are the internal parameters that are input to the network and can be changeable according to requirement.
2. Observable parameters are the external parameters that are measured or the output of the system.

In LA-OF, learning automata takes ETX as a controllable parameter and Packet Loss (PL) as an observable parameter. Learning automata tune the controllable parameter by getting a reinforcement signal from the environment on the bases of observable parameters. At the start of the learning process, probabilities are distributed uniformly among all actions. Every action of the controllable parameter has an equal probability. In other words, at the start, every action has an equal chance to be chosen. At each round, the reinforcement signal updates the probability vector. This property helps to fasten up the local optima though it may result in increased overhead in the algorithms execution time. There are two types of learning phases; offline and online learning phase:

1. Online learning phase updates the probability vector based on a reinforcement signal at each iteration for N iterations.
2. Offline learning phase doesn't update the probability vector until the unlikely condition has occurred.

LA-OF adopts both learning phases, and the online learning phase has been employed in the environment learning phase until the termination condition has occurred. The offline learning phase has triggered after the tuning of ETX, to trace the changes in the environment. Moreover, it

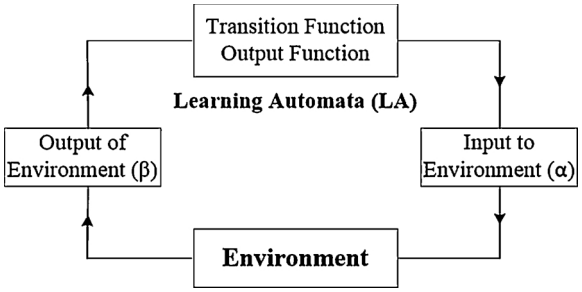

Fig. 4. Learning cycle of learning automata (α is the action; β is the reinforcement signal).

avoids the sudden changes that happen due to unimportant event or temporary environmental changes.

Every node in the network assigns learning automata that work parallel and runtime. The following configuration has been made on each learning automata:

Set of action (α): set of finite possible actions of controllable parameters (ETX) in which the environment is observed.

Output (β): is the reinforcement signal from the environment which is based on the observable parameter (Packet Loss).

Model: define the type of reinforcement signal, where a P-model is used in this research which can have two types of signals 0 or 1 (unfavorable or favorable).

Learning Function (T): defines the rate of reward and penalty, Linear reward penalty is used in the proposed work.

At each iteration, as shown in Fig. 4 learning automata select an action from α and observe the environment. Then update the probability vector using learning function according to reinforcement signal β. After the $\max_{th}$ (N_I terations), the action having the highest reward will be chosen. Learning automata runs on every node parallel to learn the expected transmission count. Learning automata learn through interacting with the environment by getting a reinforcement signal. It repeatedly updates the probability vector until the termination condition occurs. A maximum threshold is defined $\max_{th}$, the learning phase has continued until the threshold $\max_{th}$ (line 10, Algorithm 1) condition occurs. After the learning period (N_I terations) has stopped the best value is selected which yields the highest probability (line 18, Algorithm 1). Therefore, learning automata will consume some extra memory to store the probability vector to improve network performance.

Algorithm 1. LA-OF (main function)

```
1:  function neighbor_l ink_c allback rpl_p arent_t*p, status, numtx
2:    recorded_e tx = nbr − > link_m etric
3:    if status = =MAC_T X_N OACK then
4:      packet_e tx = (10*RPL_D AG_M C_E TX_D IVISOR)
5:      if Iteration > Threshold && N_I tration = =Negative_T hreshold then
6:        Reset the probability vector
7:        Reset the iterations
8:      end if
9:    end if
10:   if Iteration < = Threshold then
11:     if packet_e tx = =(10*RPL_D AG_M C_E TX_D IVISOR) then
12:       learning_p enalty
13:     else
14:       learning_r eward
15:     end if
16:   end if
17:   if Iteration > = Threshold then
18:     highest_p robability
19:   end if
20: end function
```

Algorithm 2. LA-OF (learning automata)

```
Require:
  E_i (Set of ETX values)
  P_i (Probability vector)
1:  function learning_r ewardETX_v alues*learn_a uto, pack_e tx
2:    a = 0.1
3:    for all k ∈ E_i do
4:      if k = =pack_e tx then
5:        Increase the probability of current value with factor a
6:      else
7:        Decrease the probabilities of other values with factor a
8:      end if
9:    end for
10: end function
11: function learning_p enaltyETX_v alues*learn_a uto, packet_e tx
12:   b = 0.1
13:   for all k ∈ E_i do
14:     if k = =packet_e tx then
15:       Decrease the probability of current value with factor b
16:     else
17:       Increase the probabilities of other values with factor b
18:     end if
19:   end for
20: end function
21: function highest_p robabilityETX_v alues*learn_a uto
22:   choose the best value having highest probability
23: end function
```

The probabilities are updated on the bases of the linear reward penalty function. On receiving a reward for the current action probability vector is updated by using Eq. (3). It increases the probability of current action by a factor a (line 5, Algorithm 2) and decreases the probabilities of other actions by a factor a (line 7, Algorithm 2):

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)], \\ p_j(n+1) &= p_j(n) - \mathrm{ap}_j(n) \quad \forall j. \end{aligned} \tag{5}$$

While on receiving a penalty for the current action probability vector is updated by using Eq. (6). It decreases the probability of action by a factor b (line 15, Algorithm 2) and increases the probabilities of other actions (line 17, Algorithm 2) by a factor b.

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n), \\ p_j(n+1) &= \frac{b}{r-1} + [(1-b)p_i(n)] \ \forall j. \end{aligned} \tag{6}$$

wherein both Eqs. (3) and (6), $p_i(n)$ is the probability of current action, $p_j(n)$ is the probability of other actions in probability vector and r is the total number of actions.

After finding the best action, the learning process will stop. It may have a negative impact on the network, for instance, when the learning automata process is stopped, and the network is converged to a stable state later, the environmental condition changes which cannot be traced back to the previous stable state. To avoid this negative impact, an offline learning phase has been triggered. It analyzes whether the change is due to the unimportant events or not. If it is not a temporary change, then the learning process is restarted (line 5–7, Algorithm 1).

At each iteration status of the sent packet is checked whether ACK is received or not. If NOACK status is showing for the packet, then penalize the current ETX using Eq. (6) (lines 3, 4, 11, 12 Algorithm 1 and lines 11–19 Algorithm 2). For example, if the probability of current action is 0.11 after getting a penalty, it reduces to 0.099, and probabilities of other actions are increased with factor 0.1. If ACK is shown in the status, then reward the current ETX and penalize the other actions using Eq. (3) (lines 14, Algorithm 1 and lines 1–9, Algorithm 2). For example, if the probability of current action is 0.11 after getting a reward, its increase to 1.99 and probabilities of other actions are reduced by factor 0.1. After the termination condition, the best ETX value is selected, which yields the highest probability (lines 17, 18, Algorithm 1, and lines 21, 22, Algorithm 2).

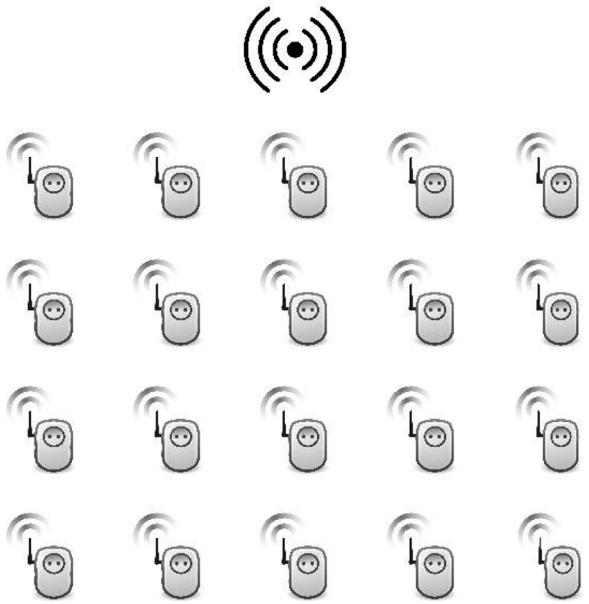

Fig. 5. Example topology of network (1 sink node and 20 data nodes).

4. Simulation and results

4.1. Simulation environment

The evaluation of the proposed scheme is based on simulation results that are carried on the Cooja simulator with operating system Contiki 3.0 (Dunkels, Gronvall, & Voigt, 2004). Cooja is a widely adopted simulator by the IoT developer to simulate their work. The proposed work is designed for IoT based NANs in smart grids, where mostly multi-hop mesh topology has been adopted. Therefore, for the performance evaluation of the proposed work, the multi-hop grid topology is considered, as shown in Fig. 5. The network consists of 1 sink and 20 client nodes. Each node generates a data packet every 2 seconds. Cooja provides unit disk graph medium (UDGM), which adds looseness into the wireless medium, which has used to get more realistic results. It added looseness in the medium according to the relative distances between the devices in the radio medium and made the environment non-stationary. In UDGM, as the distance between the sender and receiver is increases, the packet reception ratio decreases, while the reception rate increases as the distance decrease. In our scenario, we set 100 meters transmission range as well as interference ranges for the nodes, respectively. The nodes are linearly distributed in an area of 300×300 m in a grid topology, while the sink is placed on the top center of topology. The motes that are used in the simulation are Tmote Skye, which has microcontroller MSP430 with 2.4 GHz wireless transceiver Chipcon CC2420 and having 8MHz processing power, 48k of ROM and 10k of RAM. The motes run Contiki 3.0 operating system and compliance with communication protocol IEEE 802.15.4.

The learning automata has its configuration parameters. Learning reward-penalty (LR-P) function is being used, which yields the equal ratio of reward and penalty at each iteration. If the system gets a reward or penalty on a given action, it will result in an equal penalty or reward on other actions. The values of reward (a) and penalty (b) is set to 0.1 ($a = b = 0.1$) because at 0.1 WSN's gets the highest PRR as shown in Fig. 6. The number of iterations (online learning period) is set to 25 because, after 25 iterations, the network is not getting a significant improvement in PRR, as shown in Fig. 7. The number of negative iterations (offline learning) was 4 because if an inconsistency is observed due to unimportant event, the node will get back to the previous state within 3–4 unsuccessful attempts. The set of ETX contains the range of values from 1 to 9.

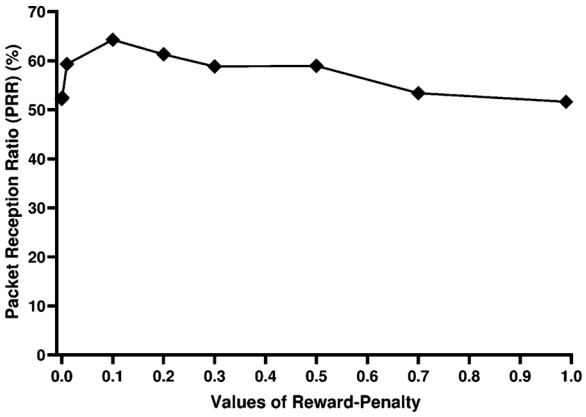

Fig. 6. The effect of different reward-penalty values on PRR.

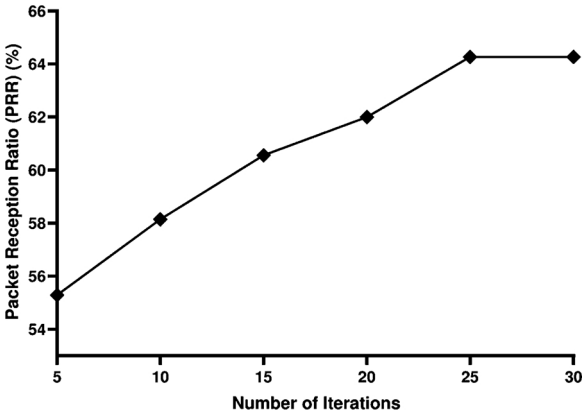

Fig. 7. The effect of different learning periods on PRR.

Energy consumption is measured through the power-traces by using javascript. Energy consumption is the sum of energy consumed in transmission, receiving, in processing (CPU), and the LPM mode. Energest module is used to calculate the energy consumption. It calculates the number of ticks (time) spends at each stage TX, RX, CPU, and LPM. Eq. (7) is used to convert the number of ticks into Joules:

$$\text{Energy Consumption} = \frac{\text{Energest\_v alue} \times \text{Current} \times \text{Voltage}}{\text{RTIMER\_S ECONDS}} \quad (7)$$

where Energest_v alue is the number of ticks spent at each stage Tx, Rx, CPU, and LPM. Current is the value of the current consumed at each stage. As Tmote Sky is used, which consumes 21,800 μAh in Transmission, 19,500 μAh in Receiving, 1800 μAh in CPU and 5.1 μAh in LPM mode. Voltage is the battery voltage which is 3 V and RTIMER_S ECOND is the number of ticks per second which is 32,768/s.

The packet size is set to 200 bytes, which sent every 2 seconds. The value of the DIO interval is set to be a default of which the minimum value is 12, and the doubling value is 8. The simulation is run for 300 s 10 times. Table 4 contains simulation and environmental parameters.

4.2. Results and analysis

4.2.1. Data reception

The packet reception ratio is being defined as the ratio of packets received at the sink node over the packet generated from the source node. The environmental factors greatly affect the link parameters at the time of measuring due to unimportant events of the environment that may lead to the wrong estimation of ETX. As link metric (ETX) has a significant impact in PRR, fail to measure desired ETX value will cause packet loss. The transmission ratio is set to be 50% to get more realistic values bringing looseness in the environment. Due to the high

Table 4
Simulation parameters.

Parameters	Values
Simulator	Cooja Contiki 3.0
Simulation area	300 × 300 m
Traffic type	CBR
Number of nodes	20
Transmission mode	Storing
Transmission range	100 m
Interference range	100 m
Packet format	IPv6
Mote type	Tmote Sky
Microcontroller	MSP430
Initial energy	20 J
Reward value	0.1
Penalty value	0.1
Energy module	ENERGEST
TX energy consumption	21,800 μAh
RX energy consumption	19,500 μAh
CPU energy consumption	1800 μAh
LPM energy consumption	5.1 μAh
DIO MIN	12
DIO doubling	8
MAC	IEEE 802.15.4
Simulation time	300 s
Radio model	UDGM

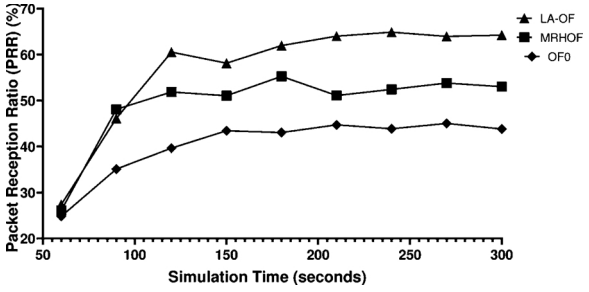

Fig. 8. Data reception at root node over time, PPR at different time intervals.

rate of data traffic from source nodes, queue overflow may also affect packet loss. Fig. 8 represents the comparative analysis of proposed work with standardized objective function MRHOF and OF0. PRR has been measured at every 30s-time interval. At the start, LA-OF and MRHOF have almost identical behavior because LA-OF is in the learning phase. At each iteration, the LA interacts with the environment to tune the ETX accordingly. After tuning the ETX according to the environment, the PRR of LA-OF is a significant increase as compared to MRHOF and OF0. The LA-OF based network has experienced less packet loss due to the exact estimation of transmission and tuning of the ETX according to the environment. While MRHOF is unable to measure exact ETX because of the non-stationary environment, which causes loss of packets.

Network throughput is defined as the ratio of data bytes received over the bytes sent. Fig. 9 present the result of network throughput. The comparison of the proposed work is made with MRHOF and OF0. The

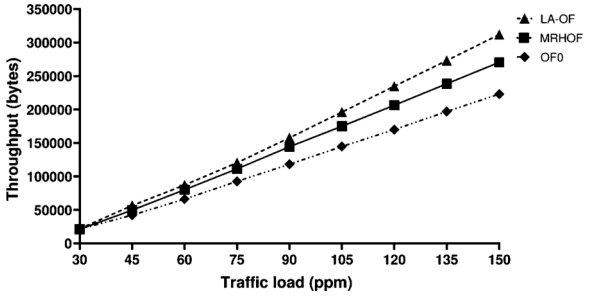

Fig. 9. Network throughput under different traffic load.

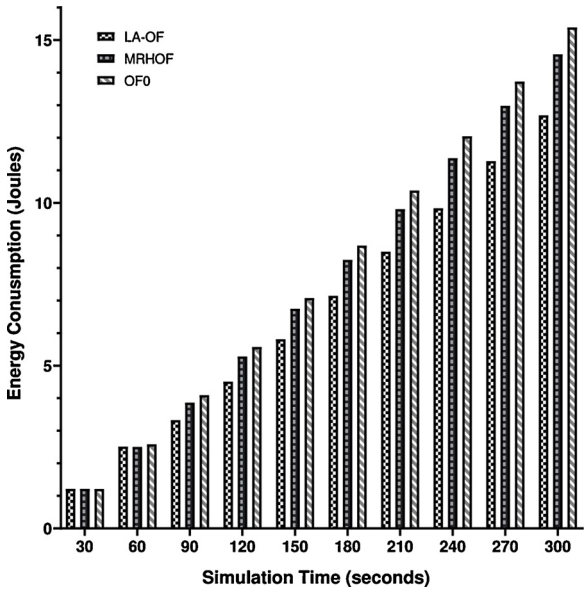

Fig. 10. Network energy consumption with respect to time.

results are taken at different data rates. The graph depicts that the throughput of LA-OF increases with the increase in data rate. The network throughput of the proposed OF is high as compared to MRHOF and OF0 because they are unable to measure desired ETX due to a random environment. For that reason, the less data loss in LA-OF due to the stable and tuned ETX according to the environment and non-stationary environment lead other approaches towards the wrong estimation. Furthermore, the increase of data rate the LA-OF achieves higher throughput as compared to other OF's, and It indicates that controlling link metrics has a great impact on network throughput. The graphs also depict that at 160 packets, LA-OF achieves 7.04% higher throughput as compared to MRHOF.

4.2.2. *Energy consumption*

Network energy consumption is the measure of the average consumption of energy over time of all nodes in the network. The comparison of proposed LA-OF with MRHOF and OF0 is presented in Fig. 10. The network EC is measured in the time intervals of 30 s. At the start, the environment is observed where LA-OF integrated nodes are learning the environment under different uncertainties. The LA-OF tunes the ETX according to the environment considering the uncertainty, with that the energy consumption of the network is observed less as compared to other OFs. The reason for being less consumption is the fine-tuning of ETX, so nodes attempt an exact number of re-transmissions rather than the extra unnecessary transmissions. The other reason is the control packets, which have the extra burden on the network. Due to the stable link metric, fewer control packets are being sent, as shown in Fig. 13. This will limit the radio to consumes extra energy by preventing redundant transmission and reception. The other objective functions (MRHOF and OF0) consume more energy because they are unable to trace down the non-stability of the environment. Overall, LA-OF made 17.52% improvement as compared to MRHOF and OF0.

The network lifetime is defined as; for how long nodes can be alive without charges or replacement of the battery. The initial energy of each node is set to be 20 J, and the remaining energy is measured every 30 s. While the nodes with having energy less than 0.025 J is considered to be dead. The comparative analysis of LA-OF with MRHOF and OF0 has been shown in Fig. 11. It is observed that the network lifetime of the proposed OF is greater as compared to other OF's. This is because of the improvement in the estimation of the link metric according to the environment. LA-OF conserve the energy by limiting the transmissions

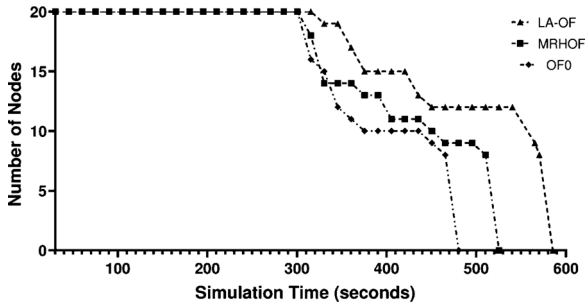

Fig. 11. Number of nodes dead over time.

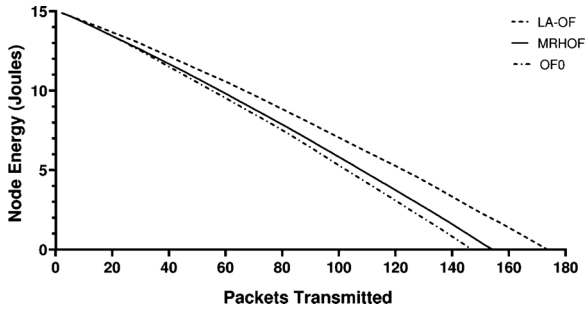

Fig. 12. Total packet transmitted without charging a battery.

through, adjusting the exact number of transmissions, and having fewer control packets. Therefore, the conservation of energy will allow the nodes to be active for a longer time. Therefore, this will improve a lifetime of the LA-OF based network as compared to MRHOF and OF0.

The number of packets a node can transmit before being dead is defined as node lifetime. The only successful transmitted packets have been considered. Fig. 12 presented the comparison of LA-OF with MRHOF and OF0. The node with the proposed OF has a higher number of transmissions as compared to other OF0's. The reason for that is the fine-tuning of ETX helps the nodes to conserve energy by avoiding unnecessary transmissions. The conservation of energy in LA-OF allows the node to transmit more packets as compare to MRHOF and OF0.

4.2.3. DIO overhead

DIO overhead is caused by the packets used to create and maintains the network topology. DIO is the control packet used in RPL for the construction and maintaining of topology. Every node broadcasts the DIO packet periodically using the trickle algorithm to keep other nodes updated about its status. If the link parameters are stable, the frequency of the DIO packets will be reduced through increasing the DIO interval with factor 8.

The proposed OF is compared with MRHOF and OF0, as shown in Fig. 13. The results depict that the proposed technique has less DIO overhead as compared to the other objective functions. This is because the LA-OF learns and tunes the ETX according to the environment,

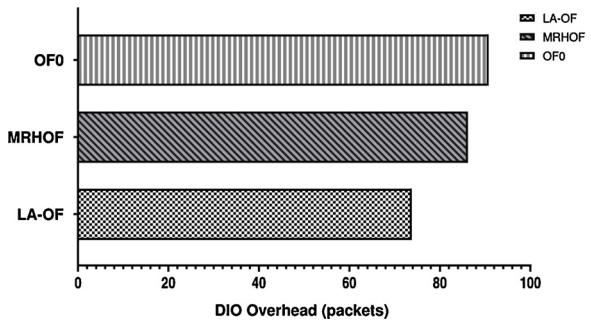

Fig. 13. The total number of DIO packets sent (300 s).

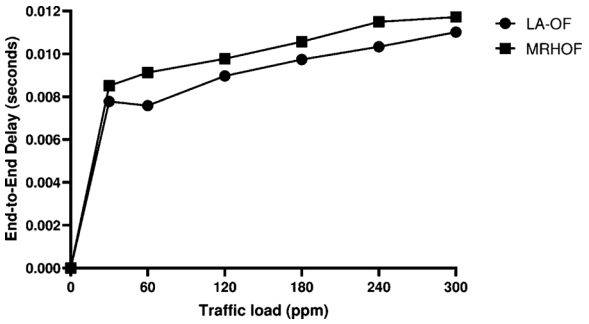

Fig. 14. The average end-to-end delay of packet over traffic load.

which stabilizes the link parameter. As it was discussed, the stable link metric increases the tickle timer, which reduces the frequency of the DIO packets. For that reason, LA-OF made an 18.72% improvement in DIO overhead as compared to MRHOF and OF0.

4.2.4. Packet delay

To measure the packet delay, we have considered two delay metrics; network delay and End-to-end delay. End-to-end delay is the average packet delay measured from a specific source node to the sink. While the network delay is the average packet delay observed from source to destination in the whole network. The delay is measured from the packet generated by the node to a packet received successfully at the root node. Figs. 14 and 15 presents the evaluation of End-to-End and Network delay. The results demonstrate that LA-OF perform little better than MRHOF.

In Fig. 14, the delay is measured at a different packet rate, where the proposed OF performs marginally better than MRHOF. The LA-OF measures the exact estimation of the link metric, which helps to reduce the transmission delay. On the other hand, in Fig. 15 is the measure of delay over the simulation time of 300 seconds. In the learning phase, both LA-OF and MRHOF have the same response. However, when the LA-OF tunes the ETX, it helps to reduce transmission delay; consequently, the LA-OF performs marginally better. The reason for not having significant improvement in delay, because the node metrics (i.e., queuing delay) have a great impact on the packet.

5. Conclusion

IoT standards are integrated with sustainable cities to offer services to sustainable communities. This research address the challenge of the dynamic and lossy environment on the routing protocol in the neighborhood area network of smart grids, where standardized objective functions of RPL were not able to deal with it. To overcome this issue, we integrated learning automata with OF to tune the link metric (ETX) through learning the environment. It learns through interacting with the environment for specific iterations in order to measure the best ETX. LA adopts the learning reward-penalty mechanism, which updates the

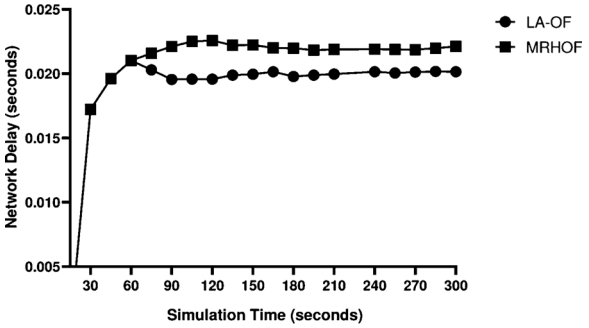

Fig. 15. The average packet delay with respect to time.

probability vector at each iteration and yields the best ETX by repeatedly interacting with the environment. The ETX table is then fed with the best ETX value for each neighbor node, and the preferred parent table is updated as well. After the convergence of LA, the monitoring of the environment is continued in the background to trace down the instability in a network. The proposed LA-OF tune ETX according to the environment, which improves the performance metrics, i.e., packet reception ratio, energy consumption, and control overhead, which are key metrics to consider for smart grids. We simulated our proposed work in Cooja with Contiki 3.0. The results show that our work outperformed the standardized objective function MRHOF and OF0. The proposed technique also has significantly improved packet reception ratio and throughput. Moreover, it reduces energy consumption and DIO control overhead.

In the future, this scheme can be enhanced further if the learning automata applies to the multiple network metrics, and decisions are taken on the basis of their utility function.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1A09082266) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00313) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

Abujubbeh, M., Al-Turjman, F., & Fahrioglu, M. (2019). Software-defined wireless sensor networks in smart grids: An overview. *Sustainable Cities and Society,* 101754.

Al-Turjman, F., & Malekloo, A. (2019). Smart parking in iot-enabled cities: A survey. *Sustainable Cities and Society,* 101608.

Al-Turjman, F. (2020). Intelligence and security in big 5g-oriented iont: An overview. *Future Generation Computer Systems, 102*, 357–368.

Ancillotti, E., Vallati, C., Bruno, R., & Mingozzi, E. (2017). A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management. *Computer Communications, 112*, 1–13.

Aziz, M. (2019). *On multi-armed bandits theory and applications*Northeastern University (Ph.D. thesis).

Bahramlou, A., & Javidan, R. (2018). Adaptive timing model for improving routing and data aggregation in internet of things networks using RPL. *IET Networks, 7*(5), 306–312.

Bhandari, K., Hosen, A. S. M. S., & Cho, G. (2018). Coar: Congestion-aware routing protocol for low power and lossy networks for iot applications. *Sensors, 18*(11), 3838.

Bibri, S., & Krogstie, J. (2017). On the social shaping dimensions of smart sustainable cities: A study in science, technology, and society. *Sustainable Cities and Society, 29*, 219–246.

Bibri, S. E. (2018). The iot for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustainable Cities and Society, 38*, 230–253.

Cao, Y., & Wu, M. (2018). A novel RPL algorithm based on chaotic genetic algorithm. *Sensors, 18*(11), 3647.

Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal, 1*(4), 349–359.

Dunkels, A., Gronvall, B., & Voigt, T. (2004). Contiki – A lightweight and flexible operating system for tiny networked sensors. *29th annual IEEE international conference on local computer networks,* 455–462.

Fabian, P., Rachedi, A., Gueguen, C., & Lohier, S. (2018). Fuzzy-based objective function for routing protocol in the internet of things. *2018 IEEE global communications conference (GLOBECOM),* 1–6.

Ghaleb, B., Al-Dubai, A., Ekonomou, E., Gharib, W., Mackenzi, L., & Khala, M. B. (2018). A new load-balancing aware objective function for RPL'S iot networks. *2018 IEEE 20th international conference on high performance computing and communications; IEEE 16th international conference on smart city; IEEE 4th international conference on data science and systems (HPCC/SmartCity/DSS),* 909–914.

Gnawali, O., & Levis, P. (2012). *The minimum rank with hysteresis objective function* Report 2070-1721.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660.

Hui, J., & Vasseur, J. P. (2012). *The routing protocol for low-power and lossy networks (RPL) option for carrying RPL information in data-plane datagrams* Report 2070-1721.

Intel (2018). *A guide to internet of things.* https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html.

Javed, F., Afzal, M. K., Sharif, M., & Kim, B.-S. (2018). Internet of things (iot) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials, 20*(3), 2062–2100.

Kamgueu, P. O., Nataf, E., & Ndie, T. D. (2018). Survey on RPL enhancements: A focus on topology, security and mobility. *Computer Communications, 120*, 10–21.

Kechiche, I., Bousnina, I., & Samet, A. (2019). A novel opportunistic fuzzy logic based objective function for the routing protocol for low-power and lossy networks. *2019 15th international wireless communications & mobile computing conference (IWCMC),* 698–703.

Lamaazi, H., & Benamar, N. (2018). Of-ec: A novel energy consumption aware objective function for RPL based on fuzzy logic. *Journal of Network and Computer Applications, 117*, 42–58.

Lamaazi, H., El Ahmadi, A., Benamar, N., & Jara, A. J. (2019). Of-ecf: A new optimization of the objective function for parent selection in RPL. *2019 international conference on wireless and mobile computing, networking and communications (WiMob),* 27–32.

Lee, I., & Lee, K. (2015). The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons, 58*(4), 431–440.

Levis, P., Clausen, T., Hui, J., Gnawali, O., & Ko, J. (2011). *The trickle algorithm* Report 2070-1721.

Narendra, K. S., & Thathachar, M. A. L. (1974). Learning automata – A survey. *IEEE Transactions on Systems, Man, and Cybernetics,* (4), 323–334.

Narendra, K. S., & Thathachar, M. A. L. (2012). *Learning automata: An introduction.* Courier Corporation.

Nassar, J., Gouvy, N., & Mitton, N. (2017). Towards multi-instances QOS efficient RPL for smart grids. *Proceedings of the 14th ACM symposium on performance evaluation of wireless ad hoc, sensor, & ubiquitous networks,* 85–92.

Papathanasiou, J., & Ploskas, N. (2018). *Topsis. Multiple criteria decision aid.* Springer1–30.

Sailaja, K., & Rohitha, M. (2018). Literature survey on real world applications using internet of things. *2018 IADS international conference on computing, communications & data engineering (CCODE)*.

Schulz, P., Matthe, M., Klessig, H., Simsek, M., Fettweis, G., Ansari, J., Ali Ashraf, S., Almeroth, B., Voigt, J., Riedel, I., et al. (2017). Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine, 55*(2), 70–78.

Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering,* 2017.

Taghizadeh, S., Bobarshad, H., & Elbiaze, H. (2018). Clrpl: Context-aware and load balancing RPL for iot networks under heavy and highly dynamic load. *IEEE Access, 6*, 23277–23291.

ten Have, H., & Gordijn, B. (2020). *Sustainability*.

Thubert, P. (2012). *Objective function zero for the routing protocol for low-power and lossy networks (RPL)* Report 2070-1721.

Unsal, C. (1998). *Intelligent navigation of autonomous vehicles in an automated highway system: Learning methods and interacting vehicles approach.*Virginia Tech (Ph.D. thesis).

Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE Communications Surveys & Tutorials, 15*(1), 5–20 ISSN 1553-877X.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks, 52*(12), 2292–2330.

Zier, A., Abouaissa, A., & Lorenz, P. (2018). E-rpl: A routing protocol for iot networks. *2018 IEEE global communications conference (GLOBECOM),* 1–6.