BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 4
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0042

A Multi-Agent Reinforcement Learning-Based Optimized Routing for QoS in IoT

T. C. Jermin Jeaunita, Sarasvathi V.

PESIT Bangalore South Campus, Bangalore, India and affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India
E-mails: jerminjeaunita@gmail.com sarsvathiv@pes.edu

***Abstract*:** *The Routing Protocol for Low power and lossy networks* (*RPL*) *is used as a routing protocol in IoT applications. In an endeavor to bring out an optimized approach for providing Quality of Service* (*QoS*) *routing for heavy volume IoT data transmissions this paper proposes a machine learning-based routing algorithm with a multi-agent environment. The overall routing process is divided into two phases: route discovery phase and route maintenance phase. The route discovery or path finding phase is performed using rank calculation and Q-routing. Q-routing is performed with Q-Learning reinforcement machine learning approach, for selecting the next hop node. The proposed routing protocol first creates a Destination Oriented Directed Acyclic Graph* (*DODAG*) *using Q-Learning. The second phase is route maintenance. In this paper, we also propose an approach for route maintenance that considerably reduces control overheads as shown by the simulation and has shown less delay in routing convergence.*

***Keywords*:** *QoS routing, multi-agent system, Internet of Things* (*IoT*), *reinforcement learning, RPL routing.*

1. Introduction

Internet of Things (IoT) has been known for its multifaceted data generating and decision actuating networking capability that is so commonly applied for various solution expected real world requirements. The cyber physical portion of network that is deployed in the real environment collects heterogeneous data that is sent to the sink for further processing. This heterogeneous data when meddled by the intelligent architectures, that needs to follow various constraints and when the protocols actuate the network, based on various metrics, a single-agent model may not be able to perform global optimization, even though greedy it is [1, 2]. An agent-based system holds a software entity that is responsible for the routing process that takes the responsibility of collecting the routing information and forwarding. A single-agent system is a centralized entity that takes responsibility of data collection and decision making all by itself. If the size of the IoT network is growing vast, a single-agent network does not facilitate and detail the global network topology, and hence in here

we need to involve a multi-agent framework that will highly help us to obtain optimized solution. In a multi-agent system, it is assumed that every sensor node holds an agent, and agents together coordinate among each other for path finding. They share control information among each other and makes the routing process confined towards themselves.

The optimized path is found for a route based on certain metrics used to express the link or route quality. Usually hop count is used as a metric to find the shortest path. The nodes considered making up an IoT network are usually away from fixed power source and so the energy usage for a node becomes a crucial constraint. In this case, node energy or remaining energy of a node is considered as one of the concerning metrics in path finding. A path with nodes of better energy will be chosen for data transmission. Routing algorithms designed should look this as a concern of consuming less energy for data transmission [3-6]. As so the lifetime of the network is going to be good, and will be prolonged. To improve the lifetime of the network, which is proportional to the node energy, the placement of nodes in the network and the distance between these sensor nodes is taken in concern [7]. To avoid retransmissions, packet loss should be minimal. Bandwidth and throughput become important factors for successful data transmission. Response requirement on timeliness called as latency varies from one application to another.

The remaining part of the paper is organized as follows: Section 2 provides the survey of similar work already available in the literature. Section 3 gives the objective of the proposed research. Section 4 explains the system model and the proposed work. Section 5 discusses the results and importance of the proposed work. Section 6 concludes the paper.

2. Related study

In-order to achieve optimization on a global scale, learning, based on the global network topology necessitates design of successful routing approaches. A study on multi-agent-based routing and reinforcement learning-based routing along with Quality of Service (QoS) issues is given below.

Research on routing for IoT suggests for the design of multi-agent systems, which is found to be better than the single-agent systems so that to learn more about the network topology in a broader scale. Efficiency here is achieved by reducing the overheads in communication without compromising the facts on dynamic nature of the network, arrival of huge volume of data and unpredictable and irresolute network topology.

2.1. Existing routing approaches

Routing Protocol for Low Power Lossy Networks (RPL) is the widely used protocol for routing path finding in IoT networks [8-10]. The RPL routing protocol creates a routing path in the form of a graph between every source node and the root node. The RPL routing approach suffers storage restrictions, and was developed for static nodes [11, 12].

Authors of [13] have proposed an intelligent-agent system to communicate with neighbours based on coverage range and distance between two nodes. Authors of [14] propose architecture of establishing relationships between the agents and devices. Here the agents operate as one of the three modules: border module, sequential module and jumping module. At border module, the agents take responsibility of storing and maintaining the load of the network and new devices. The sequential module takes responsibility of storing the sequential information of agents and seeking possibility of merging them. In jumping module, the agents maintain two tables: a predecessor table and a successor table.

Authors of [15] discuss an integrated approach solving consumption, aggregation, and routing problems in WSNs. The authors have developed a Type-2 Fuzzy ontology based multi-agent system where the agents share information among themselves, the membership value being based on residual energy of the nodes. Multi-agent-based routing protocols in concern with QoS are discussed in many papers [16-20].

A combinatorial optimization problem for solving routing and scheduling problems using multi-agent framework is proposed in [21], where the agents act with other agents based on the AMAM metaheuristic framework. In [22] a multi-agent framework has been proposed to reduce the energy consumption of sensor nodes in a military network. The agents here work as three different layers: sink driven, time driven and emergency data driven. The work [23] proposes an aggregation and routing mechanism using multi-agent system considering the network like fish-bone structure.

Reinforcement-based learning addresses problems like finding the states and actions of the agents and the optimization function responsible for the reward [24]. Authors of [25] propose a packet routing framework based on multi-agent deep reinforcement learning approach. The authors have developed a deep Q-Routing algorithm where the learning and communication process is fully distributed. Fang Wang, Feng and Chen in [26] propose a dynamic routing algorithm using delayed Q-Learning to achieve better convergence. Singh and Kaur in [27] perform link cost estimation using various machine learning algorithms and have found c4.5 decision tree machine learning algorithm which has better classification accuracy than the multilayer perceptron, radial basis function neural networks and Naive Bayes.

2.2. Motivation of the proposed research

Routing is a primary process that uses the network resources at wide extent. In this scenario if the process of routing itself consumes more power, memory, and bandwidth, then the actual data transmission through forwarding becomes a minor entity using the network resources and may at times lead to unavailability of network resources for data transmission. Hence, the routing protocols designed should play less time in the network lifetime for route generation and spend more time for data transmission. During data transmissions there might be need for network maintenance due to the mobility of a node or due to node failure. Network maintenance again should not consume more power and other resources.

The proposed research work aims at developing a protocol that reduces control overhead and improves QoS (Quality of Service). This helps the IoT network to be implemented in diverse environment even with scare resources and in locations where resources cannot be replenished or replaced. The lifetime of the network is improved, and the network is made available for longer duration and with less maintenance. Packet loss due to congestion and retransmission of packets can be reduced by the proposed work.

3. Objective of the research work

Most of the existing works mention RPL as the currently most widely used routing protocol for IoT networks. Even though RPL provides various options on parameters of concern, path instances must be created between every source and destinations. This leads to increased control packet transmissions and memory requirements. Increased transmissions and receptions are directly concerned with the lifetime of the devices used in the network. Hence, the proposed research work aims at optimizing the usage and necessity of control packets, to use efficiently the network resources for the actual sensed data transmission without compromising the efficiency of routing paths between the different sensor nodes and the gateway node.

4. System model

Table 1. List of abbreviations

RPL	The Routing Protocol for Low power and lossy networks	DODAG	Destination Oriented DAG
DIO	DODAG Information Object	MOP	Mode Of Operation
OF	Objective Function	MP2P	MultiPoint to Point
P2MP	Point to Multipoint	P2P	Point to Point
OF0	Object Function Zero	MRHOF	Minimum Rank with Hysteresis Objective Function
ETX	Expected Transmission Count	R_s	The step of rank
R_r	Stretch in Rank	R_f	Rank factor
R_n	Rank of Node	R_p	Rank of parent
etx_min	The minimum ETX metric	etx_neigh	ETX of the node to its neighbour
etx_path	Summation of the etx_neigh	$r(n, h)$	Reward function
$\delta(n, h)$	State transition function	$V^{\pi}(n)$	Value corresponding to the policy π.
γ	Decides delayed or immediate reward that is in range $0 \leq \gamma < 1$	$V^*(\delta(n, h))$	Optimal discounted value
$Q(n, h)$	Q-function	$P(n'\|n, h)$	Probability of attaining the state n' by performing the action h which is presently in state n
$\hat{Q}k(n, h)$	Revised Q-function		

The IoT network is made of dispersal of one or more sensor nodes in the monitoring environment, which all are with the capability to self-organize among themselves and can be connected to a sink node. The sink node sends the collected information to the cloud for storage and analysis. This data can be provided to the user. The user using this data can perform decision making and can actuate the

network in reverse. Thus, a two way routing of data packets from the leaf node to the sink and from the sink to the leaf node is necessitated in IoT. The Routing Protocol for Low power and lossy networks (RPL) is used as a routing protocol in IoT applications. In this paper we suggest a Q-Learning reinforcement algorithm to improve QoS in IoT. The abbreviations used in this section are listed in Table 1.

4.1. RPL routing

RPL protocol considers the monitoring nodes in the IoT network, as individual nodes in a graph and frames a directed acyclic graph called Destination Oriented DAG (DODAG).The sink node sends a DODAG Information Object (DIO) to the network nodes. This generates a downward traffic towards the leaf nodes. The DIO packet carries the following information: node rank, Mode Of Operation (MOP), Objective Function (OF) and other metrics. Rank of a node is found based on the hop distance of a node from itself to the sink node. The monitoring nodes receive DIO from multiple neighbours that are connected to the sink node in the DODAG. The neighbour that leads to a shortest path to the sink node is selected by the node using the OF. The OF varies according to the application as different applications have different QoS specific constraints. The working of RPL's OF based on hop count is called as Object Function Zero(OF0) and OF based on ETX is called as the Minimum Rank with Hysteresis Objective Function(MRHOF) are explained below.

4.1.1. OF0 of RPL

The OF0 of RPL aims to find a feasible parent that helps the node to forward its packet upwards next hop towards the sink node. The algorithm below shows the steps of OF0 in detail. Step 1 calculates the rank of a node. R_s, the step of rank, is a static metric based on the hop count. It is the amount calculated based on the link along the path that tells how much the rank can be incremented. As two or more parents from the root may be of same distance and allow a node to choose one feasible successor, R_s can be stretched to a variable limit using the variable R_r. R_f is the rank factor. For the different varieties of nodes or links in a heterogeneous network environment, the R_f is fixed with various constant values based on the network type. To obtain this equation of rank increase calculation as an instance with a metric, the entire parameter is multiplied by the variable min_hop_rank_increase. From the rank, R_n, calculated for different received DIOs, the minimal one is determined, and the corresponding parent node is the feasible successor node. Table 2 gives the algorithm of OF0.

Table 2. Algorithm for OF0

Root node initiates DIO
The following steps are performed by every node in the network:
Step 1. Compute rank
Step 1.1. Find the rank_increase (using (1))
Step 1.2. Find the rank of the node (R_n) (using (2))
Step 2. Select the feasible successor, which is the neighbour with min{R_n}

The rank of the node is found as follows:

(1) $\text{rank\_increase} = (R_f * R_s + R_r) * \text{min\_hop\_rank\_increase},$

(2) $R_n = R_p + \text{rank\_increase},$

where R_p is rank of the parent node, rank_increase is the difference between the rank of selected parent and the node itself, R_f is the rank factor, R_s is the step of rank, R_r is the stretch of rank, min_hop_rank_increase is the metric factor.

4.1.2. MRHOF

ETX is the expected transmission count, which is a dynamic metric that is based on the least number of successful transmissions from a node. Like rank calculation of the nodes, the Minimum Rank with Hysteresis Objective Function (MRHOF) calculates the ETX parameter of a node for various successor nodes if there are, and the successor that results with least ETX for the OF is considered as the preferred parent node for its data transmission. MRHOF finds the additive component, the smallest path cost, only if the path cost is smaller than the already available path by at least a given threshold called hysteresis.

In MRHOF, firstly the ETX values for the paths through which the DIOs arrived are calculated. Then the minimal ETX value is picked, and its corresponding parent node is selected as successor node for its data transmission. Lastly, the selected ETX minimal value for that path is advertised to the downward nodes.

Table 3 shows the step-by-step detailed process of ETX path finding. When DIO is initiated by the root node and broadcasted, the etx_min is the minimum ETX metric, which is advertised by the root node to the neighbours. For every received DIO, the etx_min as received from the neighbour node is considered as the minimum ETX recognized so far through that neighbour upwards to the root node.

Therefore, for every neighbour advertised the ETX, the etx_path is calculated newly that gives the min_etx from the calculating node to the root. etx_path is the summation of the etx_neigh, i.e, the ETX of the node to its neighbour through the link, and the etx_min as advertised by the neighbour. From multiple etx_path calculated by a node, the least one is selected, and the corresponding neighbour node is the preferred successor. The DODAG is extended in this way without loop formation, and this newly found etx_path is advertised through the DIO downwards to the neighbouring nodes. The process is continued until the leaf node is reached.

Table 3. Algorithm of MRHOF

Root node initiates DIO.
The following steps are performed by every node in the network:
Step 1. Compute ETX Path Metric.
Step 1.1. Initialize the *etx_min* to the minimum accepted ETX path metric.
Step 1.2. Calculate etx_path = etx_neigh + etx_min.
Step 2. Select the feasible successor feasible successor is the neighbour with min{etx_path}.
Step 3. Advertise minimum ETX etx_min = min{etx_path}
Copy etx_min in the DIO message and advertise the DIO to the neighbour nodes.

4.1.3. Reinforcement learning

Machine learning provides with algorithms that helps solution generators, to design protocols that are dynamic and scalable. In this environment of IoT, the nodes activated with agents take the responsibility of path finding using reinforcement learning. Reinforcement learning is a branch of machine learning where the agents learn their environment from positive rewards and negative penalties to proceed with what to do and what not do during situations, respectively.

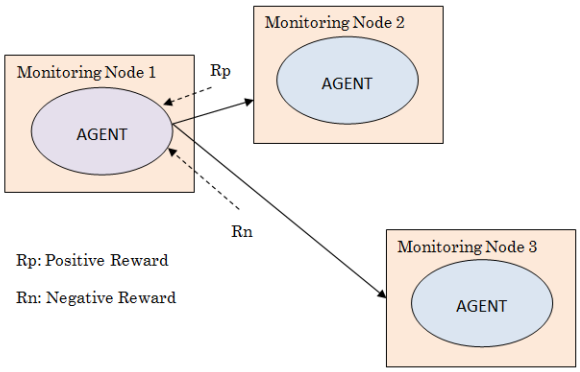


Fig. 1. Reinforcement learning in routing path finding

Traditional algorithms are static in nature and many not be generically used for varying network features. Reinforcement learning is a model free learning, which is the most suitable one to help developing routing paths in IoT environment. The agents keep communicating with the environment for changing state and every change of state that leads to the sink at the shortest path or least delay is rewarded that encourages the agents to learn the right routing path quickly. The agent process is shown in Fig. 1. Three monitoring nodes connected to each other are shown with their respective agents, and the distance between the nodes are understood as the length of the arrows. Monitoring nodes 2 and 3 are the states of the agent that are possible for monitoring node 1. Selecting one between these two states and recording that state as the next hop node in the DODAG is the action. The agent of monitoring node 1 must be rewarded based on its action of selecting the next hop node. When the agent of monitoring node 1 selects the monitoring node 2 as its parent node, the agent is given positive reward and if the node selects monitoring node 3 as its parent it is given negative reward. The sensor nodes collect the routing information immediately after the self-organization of the nodes, as well as when a path is demanded and if the path is not available. Hence, it behaves as a hybrid routing approach of path finding.

When a network is deployed, initially the nodes are unaware of each other and their locations. Until before the self-organization phase the nodes do not know their neighbours and they are not smart. They send the sensed data to the sink node. Since initially the nodes store an empty routing table and the learning algorithm is left unguided, we prefer a model free learning technique that explores from the environment and learn to pick the right neighbours to carry their data upwards. This

process is called exploitation in machine learning and the approach is termed as model-free.

The proposed approach is divided in to two phases for clarity, and for hassle free learning-based routing process:

- Route Discovery Phase
  - Reward propagation
  - DODAG formation
- Route Maintenance Phase

4.1.4. Route discovery phase

When the number of steps to converge or the number of required state changes to converge is known in advance a finite-horizon method can be followed, where the expected reward is the summation of all the rewards until the terminal state is reached. Expectation is to achieve the maximum reward obtained of moving from the start state to the terminal state. As in the IoT networks deployed for environmental monitoring, the nodes are scalable and the nodes are prone for tampering, an infinite-horizon model will best suit the discounted future rewards.

The OF calculated using hop count or ETX metrics is concerned with the static or dynamic nature of the network respectively and involves more control messages. To avoid control overheads due to frequent information sharing and DODAG updating, machine-learning algorithms can be used. Hence, we can use non-deterministic reinforcement learning approach with Q-Learning for the generation of DODAG. In this route discovery phase, reward propagation and DODAG formation is performed.

The state function consists of the neighbouring nodes upward to the root. They are represented as $s = \{n\}$ and the actions are whether to select them as next hop node or not. The actions set is $a=\{h\}$. Every agent is modelled as holding a reward function $r(n, h)$ and the hop transition function as $\delta(n, h)$ which may have probabilistic outcomes. Here n represents the parent node and h represents the hop count.

For each parent node there is a value $V^{\pi}(n)$ corresponding to the policy π. As the network is considered as the model free environment, a delayed reward approach is considered. The value γ determines producing a delayed or immediate reward. γ is chosen in the range $0 \leq \gamma < 1$. Therefore, the expected discounted cumulative reward value is given by

(3) $$V^{\pi}(n) \equiv r + \gamma\, r_{t+1} + \gamma^2\, r_{t+2} + \ldots \equiv \sum_{i=0}^{\infty} \gamma^i\, r_{t+i}.$$

An optimal discounted value is given by $V^*(\delta(n, h))$, where δ is the state transition function. Our aim is to find the action of moving to the next best hop that maximizes the sum of the immediate reward $r(n, h)$ and the discounted cumulative reward of the preferred successor, multiplied by γ. This evaluation function or Q-function is given by

(4) $$Q(n, h) \equiv r(n, h) + \gamma\ V^*(\delta(n, h)) \equiv r(n, h) + \gamma \sum_{n'} P(n'|n, h) V^*(n'),$$

where $P(n'|n, h)$ is the probability of attaining the state n' by performing the action h which is presently in state n. We can rewrite the Q-function as

(5) $$Q(n, h) \equiv r(n, h) + \gamma \sum_{n'} P(n'|n, h) \max_{h'} Q(n', h').$$

This training rule is revised to make the learning converge as

(6) $\hat{Q}_k(n, h) \leftarrow (1 - \alpha_k)\, \hat{Q}_{k-1}(n, h) + \alpha_k[r + \gamma \max_{h'} \hat{Q}_{k-1}(n', h')]$,

where, $\alpha_k = \frac{1}{1+\text{visits}k(n,\ h)}$.

Here, n and h represent the parent node and the next hop updated during the k-th iteration respectively, and visists$k(n, h)$ gives the sum of all times this node is selected as the nexthop node including the k-th iteration.

Every node maintains DODAG in the form of a *Q*-routing table obtained using this Q-Learning algorithm.

4.1.5. Route Maintenance Phase

When the Q-Learning algorithm is performed by every agent in the network, a *Q*-routing table is obtained that will have the *Q*-values to the corresponding neighbouring nodes. If any link breakage or node isolation is identified, the node that identifies this issue can find the next better *Q*-value and its corresponding neighbour node from the *Q*-routing table. Otherwise, if a link breakage is identified, control messages are generated to notify all the nodes and to regenerate the DODAG network. This will consume the network bandwidth and reduces the reliability of the network during the reformation duration of DODAG. Hence, *Q*-routing table helps in avoiding packet losses and efficient bandwidth usage. The algorithm for route maintenance is given in Table 4.

Table 4. Algorithm for route maintenance

Performed by the agent that identifies link failure and has a routing packet for transmission.
Step 1. Set the *Q*-value as 0 for the node to which the link is not available in the *Q*-routing table.
Step 2. Search the next highest non-zero *Q*-value among the connected neighbours.
Step 3. Select that node as the preferred parent.
Step 4. If the *Q*-value of all other nodes is 0, initiate RPL route-repair process.

5. Experimental results and discussion

5.1. Simulation

To analyse the proposed Q-Learning approach, for QoS attained routing in IoT networks, simulation of the environment has been performed. The IETF standardized protocol – RPL, the Routing Protocol for low power and lossy networks, is designed with the IoT features in concern. The Contiki OS is designed for IoT applications, and Cooja is its supported IoT framework. IoT devices are resource constrained in terms of memory, processing, and power. The Contiki OS is an Ubuntu OS that supports such low cost IoT resource constrained devices. The communication network stack supports a variety of protocols including IPv4, IPv6, ICMPv6, TCP, UDP, CoAP and RPL. The propagation model helps in the reduction of path loss and interference. Four different propagation models are supported by Cooja simulator: Unit Disk Graph Medium (UDGM) Distance Loss, UDGM Constant Loss, Directed Graph Radio Medium (DGRM) and Multi-path Ray-tracer Medium (MRM). Unit Disk Graph Medium (UDGM) Distance Loss is selected as the radio propagation

model to cover the motes in the 'interference' vicinity. The simulation parameters are set as shown in Table 5. To perform simulation, an area of 10000 m^2 is selected with various node densities of 10, 20, 30, 40 and 50. With the possibility of random, linear, and elliptic node positioning, random node positioning is selected so as to cover a hierarchical network. udp_sink firmware is used for the sink Sky mote and udp_sender firmware is used for the sensing Sky motes.

Table 5. Simulation parameters

Parameter	Values
Operating System	InstantContiki 2.7
Simulator	Cooja
File System	Coffee
Radio Medium	UDGM Distance Loss
Mode start-up delay	1000 ms
Random seed	1, 23, 456
Mote Type	Sky mote
Number of Motes	10, 20, 30, 40, 50
Network Area	100×100 m
Topology	Random
Objective Function	OF0, MRHOF
MAC Layer	IEEE 802.15.4
Simulation time	3600 s
Packet Transmission Interval	60 s
Transmission range	50 m
Interference range	100 m

Fig. 2 shows the DODAG graph formed by the nodes with their root node after the route discovery phase.

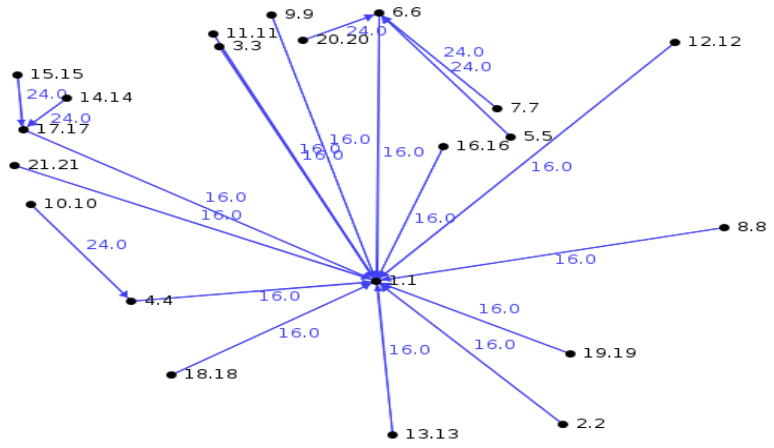


Fig. 2. DODAG formed by the Sky motes using the Cooja simulation framework

5.2. QoS parameters

The goal of the designed routing protocol is to achieve the desired QoS parameters for the IoT applications, using machine-learning approach. An analysis of the QoS parameters achieved by the objective functions of RPL protocol is studied first. When

the sensor motes are deployed, they are unaware of where to send the sensed data initially. The sensors communicate to each other through small packets of broadcast information during the neighbour discovery stage. In RPL the root node first sends a broadcast packet called DIO message to initiate a DAG formation process. Through these messages, the nodes of the entire network learn the best next hop to transmit their sensed data to the root node. This process is achieved by the objective functions OF0 and MRHOF.

The routing metrics are used to find a 16-bit integer value called rank that is updated in the ICMPv6 DIO control message. A sensor node becomes a parent node to forward data of other sensor nodes or just stays as an ordinary sensing node based on the rank it holds for itself. The rank in general represents the position of the sensing node in the DAG from its sink. Rank and the sequence number carried in a DIO detect and avoid looping formation during DAG generation.

5.3. Hop count

For a node with multiple parents, the preferred parent is selected using the node with minimum rank value. The OF0 is determined to find the rank of a node from the sink node based on the hop count. By recognizing the hop count the rank of a node towards its root node is shared among the other nodes to find their own rank and choose their parent node. Fig. 3 shows the average hop count of the nodes in the network used for simulation with various network densities.

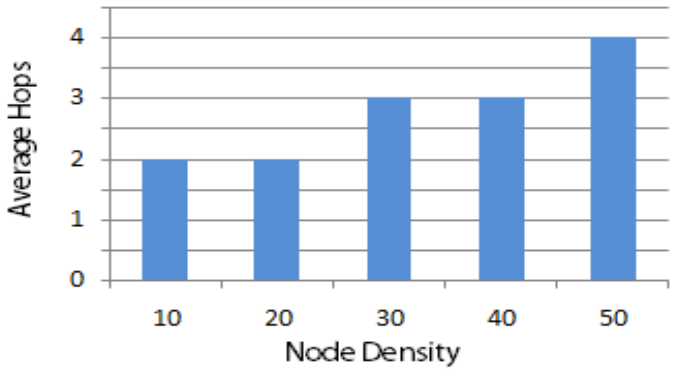


Fig. 3. Average hop count of the nodes in the network increases that the increase of the network density. DODAG is generated hop by hop starting from the root node. Higher hop count represents higher convergence time

5.4. Expected transmission count

Expected Transmission Count (ETX) is the count of successfully transmitted packets including the number of retransmissions found by the analysis of the count of probe packets sent and the acknowledgements received. It is a link-based metric. It directly affects the energy of a node, as a greater number of transmissions in a link may require more power consumption. Fig. 4 shows the average ETX of the links found in the network through the two objective functions OF0 and MRHOF.

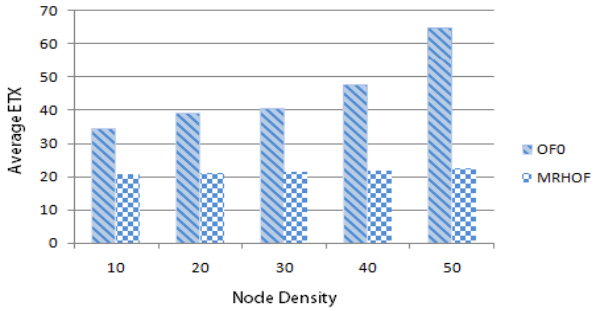


Fig. 4. Average ETX of the links found using OF0 and MRHOF

5.5. Average number of received packets

Average of packets received during the simulation time is shown in Fig. 5. This clearly depicts that if the density of nodes increases, the ratio of time required for DODAG formation is more. And if every time a link breakage occurs, this ratio of time spent for control packet transmission also gets more.

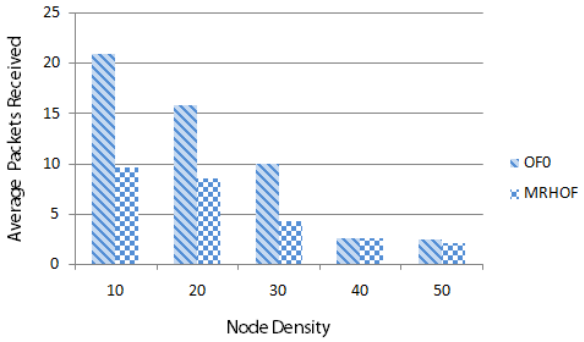
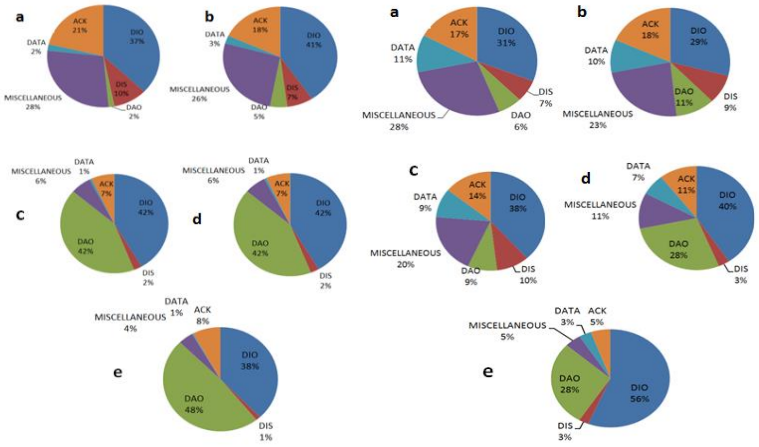


Fig. 5. Average number of packets received during the simulation time, using the two different RPL approaches OF0 and MRHOF, which seems to be reducing with the increasing node density

5.6. Control packets overhead

ICMPv6 control messages are used by the RPL protocol for the neighbour discovery and DODAG generation. The sink node initiates a DIO message. DIS, the DODAG Information Solicitation messages are sent by a newly joined node to the network to find an RPL instance for participation in the DODAG. DAO, DODAG advertisement Object is a unicast message sent by the nodes upward carrying the routing information. From the simulation performed and the results attained, we obtain the charts shown in Fig. 6 (i) and (ii) for the count of control and data messages transmitted during the simulation period.

It is found that the ICMPv6 messages are much more compared to the 6LoWPAN data packets. The transmitted packets are monitored and collected by the "Collect View" tool of the Cooja simulator. They are converted to pcap files for further analysis. Wireshark tool is used for filtering the control packets from the collected packets. The pcap files obtained during simulation with 10, 20, 30, 40 and 50 nodes were analyzed and the percentage of DIO, DIS, DAO, data packets and other miscellaneous UDP packets necessary for probing are shown in Fig. 6.



(i) (ii)

Fig. 6. (i): (a), (b), (c), (d), (e) represent the analysis made from the simulation of 10, 20, 30, 40, 50 nodes respectively that were using OF0 algorithm. They show the percentage of DIO, DIS, DAO, DATA, ACK and other miscellaneous packets generated and where in transition in the network during the one-hour simulation period. From the partitions ICMPv6 control packets are more in transition when compared to the actual data packets; (ii): (a), (b), (c), (d), (e) represent the analysis made from the simulation of 10, 20, 30, 40, 50 nodes respectively that were using MRHOF algorithm. They show the percentage of DIO, DIS, DAO, DATA, ACK and other miscellaneous packets generated and where in transition in the network during the one-hour simulation period. From the partitions it is clear that ICMPv6 control packets are more in transition when compared to the actual data packets. Compared to OF0, control packets seem to be more in MRHOF

5.7. Energy consumption

The energy consumption of a node is the average energy expended by a node during the lifetime of the network and is given by

(7) E = (Tx*19.5 mA+L*21.5 mA+CPU_p*1.8 mA+LPM*0.0545 mA)3 V)/(32768 mJ),

where, Tx represents the transmission power, the power needed for transmission, L is the power need for packet reception, CPU_p is the CPU power used for processing during full power mode and LPM is the power used even during the idle mode of the node. When more number of control messages circulates in the network, the nodes need to listen to the data transmission and energy is expended.

5.8. Multi-agents implementation

From the simulation performed it has been found that for OF0 and MRHOF, the average control packets generated during the simulation time is 79% and 86.4%, respectively. Every node is preferred to hold an agent component that communicates with other agents for “routing information” sharing. This routing agent is implemented with Q-Learning Algorithm.

5.9. QoS routing

The RPL routing protocol depends on the distance vector routing algorithm for routing that takes less execution time for the routing convergence. However, in RPL

routing approach in case of link failure route maintenance does not support back up path storage. This again leads to the generation of huge volumes of ICMPv6 control packets in the network for alternate path finding. This consumes the bandwidth of the network and until the alternate path is found the packets in-transit will be lost. Due to packet loss, retransmission is triggered at the sender node, which again leads to packet loss, as the alternate path is still not found. Fig. 7 shows a network of 6 sensing motes from mote 1 to mote 6 and one sink mote 0. From the figure, it is evident that motes 1, 2 and 3 are directly connected to the sink 0. Hence, the *Q*-value of these three motes is higher in Fig. 8. A packet transmission from mote 4, checks this *Q*-routing values and finds *Q*-value of 2 is higher than the *Q*-values of columns 3, 5 and 6. Hence, node 2 is selected for data transmission.

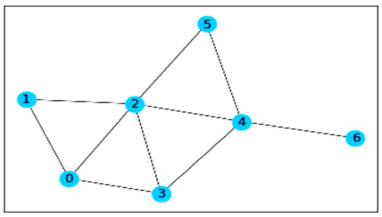


Fig. 7. A network of 7 sensing motes (motes 0 to 6) and 1 sink mote (mote 0) embedded with the proposed multi-agent approach for routing process

Nodes	0	1	2	3	4	5	6
0	100.00	89.67	89.67	87.95	0.00	0.00	0.00
1	100.00	0.00	90.00	0.00	0.00	0.00	0.00
2	100.00	89.31	0.00	89.67	80.38	80.38	0.00
3	100.00	0.00	90.00	0.00	80.38	0.00	0.00
4	0.00	0.00	89.31	87.95	0.00	80.38	72.34
5	0.00	0.00	89.67	0.00	80.38	0.00	0.00
6	0.00	0.00	0.00	0.00	80.38	0.00	0.00

Fig. 8. Image of *Q*-routing values generated for the network shown in the Fig. 7. The state change from mote 6 and mote 9 is encouraged with high reward of 99.64

If a link failure between motes 4 and 2 is identified by mote 4, then mote 4 sets the *Q*-value to be 0 and checks for the next highest value in the row. That will be 3 and mote 4 uses mote 3 to transmit its packet to mote 0. In this way alternate path finding in case of link breakage is found without again initiating for a new DODAG instance. As the packets are sent, using immediate alternate path data loss is prevented. By avoiding multiple control packets generation and transmission bandwidth of the network is not wasted. As there is a chance of finding alternate path without disturbing the network, reliability of the network is improved. Power of a node is highly consumed on listening and transmission. Because the control packets transmission is avoided during link breakage, unnecessary packets listening, processing and forwarding is avoided. This in-turn reduces power consumption.

5.10. Convergence time

Routing is the process of communication with relative nodes for path finding. Through simulations it was found that the proposed routing approach requires a very short convergence time compared to the two approaches of RPL, as shown in Fig 9.

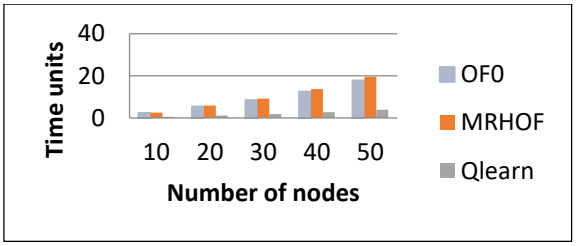


Fig. 9. Comparison of Convergence time of the OF0 based RPL routing protocol, MRHOF based RPL routing protocol, and the proposed Q-Learning based routing protocol

5.11. Route maintenance

An IoT network is dynamic in nature and prone to transmission challenges. Hence if a network node is down, to maintain reliability and robustness, an immediate alternate path finding becomes necessary. The graph in Fig. 10 Shows the time required for finding alternate path using OF0, MRHOF and the proposed Q-Learning based routing algorithm.

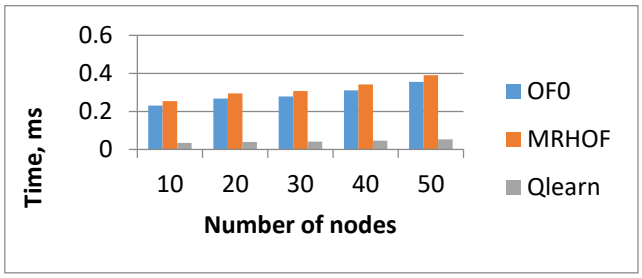


Fig. 10. Comparison of time required for finding alternate path using OF0 based RPL routing protocol, MRHOF based RPL routing protocol, and the proposed Q-Learning based routing protocol

6. Conclusion

In this paper, we have presented an optimized QoS routing approach for IoT. We have used reinforcement-based routing approach to achieve path finding using Q- Learning. This Q-Learning based routing generates *Q*-values that are considered as the rank of a node using which the preferred parent is found. The *Q*-values are stored in a *Q*-routing table representing the structure of the generated DODAG. Using this table an alternate path can be obtained during the time of link breakage. Simulation has been performed using the Contiki OS based simulator framework-Cooja. The packets are captured using the collect view tool of Cooja. The control and data packets generated using RPL are analyzed with Wireshark tool. An extensive analysis was made and found the percentage of each type of ICMPv6 control packets

and 6LoWPAN packets generated for both the OF0 and MRHOF. The extent these control packets affect the bandwidth, and the average power consumption of the network was recognized. The proposed approach provides a better path finding using multi-agents and during link breakage, initiation of new DODAG instance generation is avoided. This helps in achieving QoS routing.

References

1. Boutaba, R., M. A. Salahuddin, N. Limam et al. A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities. – J Internet Serv. Appl., Vol. **9**, 2018, No 16.
   https://doi.org/10.1186/s13174-018-0087-2
2. Liang, X., I. Balasingham, S.-S. Byun. A Multi-Agent Reinforcement Learning Based Routing Protocol for Wireless Sensor Networks. – In: Proc. of 2008 IEEE International Symposium on Wireless Communication Systems, Reykjavik, 2008, pp. 552-557. DOI: 10.1109/ISWCS.2008.4726117.
3. Sarasvathi, V., N. Ch. S. N. Iyengar, S. Saha. An Efficient Interference Aware Partially Overlapping Channel Assignment and Routing in Wireless Mesh Networks. – International Journal of Communication Networks and Information Security (IJCNIS), March 2014.
4. Sarasvathi, V., N Ch. S. N. Iyengar. Centralized Rank-Based Channel Assignment for Multi-Radio Multi-Channel Wireless Mesh Networks. – Procedia Technology, Elsevier, Vol. **4**, January 2012, pp. 182-186.
5. Sarasvathi, V., N. Ch. S. N. Iyengar, S. Saha. QoS Guaranteed Intelligent Routing Using Hybrid PSO-GA in Wireless Mesh Networks. – Cybernetics and Information Technologies, Vol. **15**, 2015, No 1, pp. 69-83.
6. Sarasvathi, V., S. Saha, N. Ch. S. N. Iyengar, M. Koti. Coefficient of Restitution Based Cross Layer Interference Aware Routing Protocol in Wireless Mesh Networks. – International Journal of Communication Networks and Information Security (IJCNIS), Vol. **7**, Novemer 2018, Issue 3.
7. JerminJeaunita, T. C., V. Sarasvathi. Fault Tolerant Sensor Node Placement for IoT Based Large Scale Automated Greenhouse System. – International Journal of Computing and Digital Systems, UoB, Vol. **8**, 2019, Issue 2.
   http://dx.doi.org/10.12785/ijcds/080210
8. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.
   https://tools.ietf.org/html/rfc6550 .
9. Thubert, P. Objective Function Zero for RPL. – RFC 6552, Vol. **33**, 2012, pp. 3-8.
10. Gnawali, O., P. Levis. The Minimum Rank with Hysteresis Objective Function. – RFC 6719 (Proposed Standard), Internet Engineering Task Force, September 2012.
    http://www.ietf.org/rfc/rfc6719.txt
11. Safaei, B., A. A. M. Salehi, A. M. H. Monazzah, A. Ejlali. Effects of RPL Objective Functions on the Primitive Characteristics of Mobile and Static IoT Infrastructures. – Microprocessors and Microsystems, Vol. **69**, 2019, pp. 79-91. ISSN 0141-9331.
    https://doi.org/10.1016/j.micpro.2019.05.010.
12. Ghaleb, B., A. Al-Dubai, E. Ekonomou, I. Wadhaj. A New Enhanced RPL Based Routing for Internet of Things. – In: Proc. of 2017 IEEE International Conference on Communications Workshops (ICC Workshops), 2017, pp. 595-600. DOI: 10.1109/ICCW.2017.7962723.
13. Mateo Sanguino, T. J., E. Navarro Lozano, M. Sánchez Alcántara. Intelligent Agent-Based Assessment of a Resilient Multi-Hop Routing Protocol for Dynamic WSN. – Wireless Pers. Commun., 2020.
    https://doi.org/10.1007/s11277-020-07136-1
14. Rocha, V., A. A. F. Brandao. A Scalable Multi-Agent Architecture for Monitoring IoT Devices. – Elsevier, Journal of Network and Computer Applications, 139, 2019, pp. 1-14.
    https://doi.org/10.1016/j.jnca.2019.04.017

15. M i t t a l, M., S. S r i n i v a s a n, M. R a n i, O. P. V y a s. Type-2 Fuzzy Ontology-Based Multi-Agents' System for Wireless Sensor Network. – In: Proc. of IEEE Region 10 Conference (TENCON'17), Penang, 2017, pp. 2864-2869. DOI: 10.1109/TENCON.2017.8228350.
16. L i a n g, X., I. B a l a s i n g h a m, S.-S. B y u n. A Multi-Agent Reinforcement Learning Based Routing Protocol for Wireless Sensor Networks. – In: Proc. of 2008 IEEE International Symposium on Wireless Communication Systems, Reykjavik, 2008, pp. 552-557. DOI: 10.1109/ISWCS.2008.4726117.
17. R u d e k, R., L. K o s z a l k a, I. P o z n i a k-K o s z a l k a. Introduction to Multi-Agent Modified Q-Learning Routing for Computer Networks. – In: Proc. of Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/e-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05), Lisbon, Portugal, 2005, pp. 408-413. DOI: 10.1109/AICT.2005.53.
18. B u s o n i u, L., R. B a b u š k a, B. d e S c h u t t e r. Multi-Agent Reinforcement Learning: An Overview. – In: D. Srinivasan, L. C. Jain, Eds. Innovations in Multi-Agent Systems and Applications – 1. Studies in Computational Intelligence. Vol. **310**. Berlin, Heidelberg, Springer, 2010, pp. 183-221.
https://doi.org/10.1007/978-3-642-14435-6_7
19. L i u, M., S. X u, S. S u n. An Agent-Assisted QoS-Based Routing Algorithm for Wireless Sensor Networks. – Journal of Network and Computer Applications, Elsevier, January 2012.
https://doi.org/10.1016/j.jnca.2011.03.031
20. B e n d j i m a, M., M. F e h a m. Multi-Agent System for a Reliable Routing in WSN. – In: Proc. of 2015 Science and Information Conference (SAI'15), London, 2015, pp. 1412-1419. DOI: 10.1109/SAI.2015.7237331.
21. S i l v a, M. A. L., S. R. de S o u z a, M. J. F. S o u z a, A. L. C. B a z z a n. A Reinforcement Learning-Based Multi-Agent Framework Applied for Solving Routing and Scheduling Problems. – Elsevier, Expert Systems with Applications, Vol. **131**, 2019, pp. 148-171.
https://doi.org/10.1016/j.eswa.2019.04.056
22. B e l a g a l i, R., A. M. A n u s h a, P. S a n g u l a g i. Energy-Efficient Secure Routing and Aggregation in Military Sensor Network Using Multi-Agent Approach. – In: Proc. of International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, 2015, pp. 286-292.
DOI: 10.1109/ICATCCT.2015.7456897.
23. S u t a g u n d a r, A. V., S. S. M a n v i. Fish Bone Structure-Based Data Aggregation and Routing in Wireless Sensor Network: Multi-Agent Based Approach. – Telecommun. Syst., Vol. **56**, 2014, pp. 493-508.
https://doi.org/10.1007/s11235-013-9769-z
24. M a m m e r i, Z. Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches. – IEEE Access, Vol. **7**, 2019, pp. 55916-55950.
DOI: 10.1109/ACCESS.2019.2913776.
25. Y o u, X., X. L i, Y. X u, H. F e n g, J. Z h a o, H. Y a n. Toward Packet Routing with Fully-Distributed Multi-Agent Deep Reinforcement Learning. – Journal of LATEX Class Files, Vol. **14**, August 2015, No 8, arXiv:1905.03494v2.
26. W a n g, F., R. F e n g, H. C h e n. Dynamic Routing Algorithm with Q-Learning for Internet of Things with Delayed Estimator. – In: IOP Conf. Series: Earth and Environmental Science. Vol. **234**. 2019. DOI:10.1088/1755-1315/234/1/012048.
27. S i n g h, K., J. K a u r. Machine Learning Based Link Cost Estimation for Routing Optimization in Wireless Sensor Networks. – Advances in Wireless and Mobile Communications, Vol. **10**, 2017, No 1, pp. 39-49. ISSN 0973-6972.

Received: 09.05.2021; Second Version: 25.10.2021; Accepted: 08.11.2021