International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 13, No.1 (Jan-2023)

http://dx.doi.org/10.12785/ijcds/130120

Reinforcement Learning based Optimized Multi-path Load Balancing for QoS Provisioning in IoT

T. C. Jermin Jeaunita[1] and Sarasvathi V[2]

[1,2] *Computer Science and Engineering, PESIT Bangalore South Campus, Bangalore, India and affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India*

Received 2 Feb. 2022, Revised 25 May. 2022, Accepted 12 Jan. 2023, Published 31 Jan. 2023

Abstract: In an IoT system, to achieve optimization, performance should be an equal concern along with satisfying the growing requirements and demand of solutions, for real time, especially time critical applications. Some of these applications are complex to accommodate current solutions that is concerned with multivariate and multiobjective performance optimizations. Hence smart learning of the system helps identify the nuances in the system that affects the performance of the system. The main goal of the protocols used in the network layer is to perform routing process and forwarding packets by recognizing and achieving best decisions to optimize network performance to achieve better Quality of Service (QoS) for the application. Prolonging the lifetime of the network keeps the network on its purpose active and achieves QoS. Hence in this paper we have proposed an algorithm for load balancing in an uncertain IoT network by choosing multi-path for data transmissions. We categorize the data into various classes that can use various levels of optimized paths. Using the reinforcement learning algorithm – Q-learning approach and the QoS parameters as the hyper parameters, the algorithm we have proposed is compared with the conventional Q-routing algorithm and proved the improvements of the proposed algorithm in network longevity and throughput.

Keywords: IoT, Load balancing, Multi-path, Optimization, Reinforcement Learning.

1. INTRODUCTION

Internet of Things (IoT) is a revolution in the information technology era, next to the technology wondered World Wide Web. Embedded processing plays a crucial role in IoT devices, building them as smart objects [1]. Being smart, these objects must sense or track the surrounding, communicate with other objects and compute data for aggregation or context aware path finding [2]. Along with performing these tasks, the challenge in the communication of these devices is interoperability. Heterogeneous device capabilities, different vendors, difference in service provisions and distributed tasks set are the major concerns towards interoperability [3].

When the size of the network grows or in case of large-scale applications, the volume of data generated by the end nodes is enormous and leads traffic congestion, buffer contention and other packet overheads in the network. Efficient load balancing in such cases that utilizes even unused nodes in the routing process is a critical requirement for IoT network performance. This will also overlook scalability, energy management, fault tolerance, diverse data involvement, delay in data transmission, resource consumption due to control overheads, packet delivery ratio, packet loss ratio, network lifetime and resilience of the network [4], [5], [6], [7], [8].

The ability of the network to shrink or expand according to the changing need of the application is called scalability. When a network is scaled the routing protocol used should be able to tolerate the changes and should not affect or disturb the network existence. The existing routing protocols should require nil or minor modifications without consuming much cost on this expected or unexpected changes.

Wireless Sensor Network (WSN) makes up a portion of IoT consisting of sensor nodes either small or huge, conventional, or smart, or battery powered, or self-power generated. In these senses the nodes are energy constrained and decides the longevity of the network life. Only until a node could successfully communicate the sensed information to the sink node, the network can be deployed on the field. Hence the routing algorithms designed should minimize the number of control packets required for path learning, as every transmission and reception of control packets has to pay a price in the form of power. By piggy backing routing information or aggregation the number of transmissions can be reduced. Routing algorithms can also be cautious about the average remaining energy of the path and use or neglect the path.

E-mail address: jerminjeaunita@gmail.com, sarsvathiv@pes.edu

Wireless networks pose another challenge of link failure during the active network usage. The routing protocol around be able to recognize this failure at the earliest and should gossip to the neighbouring nodes at the earliest. This helps in avoiding transmissions through that path, which otherwise will lead to packet loss and packet retransmissions. Packet retransmissions again will consume the resources as same as it did for the first time and waste network resources. When nodes are intimated about the path failures, the routing algorithm should be able to find an alternate better path to reach the destination. Time critical applications requires their packets to be transmitted to the sink without interruptions can use proactive routing algorithms that keeps an alternate path ready in case of path failures. This characteristic of fault tolerance improves reliability of the network.

Internet of Things needs to meddle with data transmissions from various types of devices and carry the data to the cloud for storage and decision making or analysis. As the nature of the data usually varies, the routing algorithm designed should be able to recognize and forward the data of various sizes, priorities, and confidentiality and computability requirements. Every data packet should be verified before forwarding to identify the right next hop node. Also high priority data packets should not make other nodes victims by holding the best path for only for its own transmission. The routing algorithm also should utilize the scare resources efficiently by using other unused paths for low priority or off-line data. This also helps in avoiding hotspot problem that leads to dying of network nodes around the sink node caused by usage of same nodes for data transmission repeatedly.

Latency is another factor concerned with routing algorithm development. Routing algorithms can be proactive, reactive or hybrid in path finding. Based on the requirement of the application in which the network is deployed and the availability of resources, routing algorithm can be one of these three. Proactive routing approaches finds path well ahead and a node in need of data transmission can use the available path for forwarding data. This is the preferred approach for hard real time applications which are time critical. IoT applications are also deployed in cases where the mobile nodes keep joining and leaving the network or dynamic by moving from one neighbourhood to another. The routing algorithm should be able to identify the movement of nodes and should be able to find path immediately ignoring or including such nodes based on their movements. This helps the high prioritized packets to keep unaware of the mobility of intermediate nodes and maintenance of routing path. Latency may also be caused due to buffering of data in the intermediate nodes. If only the best path is used for most of the data transmissions, there are chances of contention on the path for the medium for data transmission and the data has to be stored on the buffer. This waiting time may increase the delay in transmission. In case of reactive routing, if the algorithm is not light weight and complex, then the path finding time would be high. Such algorithms will also need much of computation and storage resources. Hence the routing algorithm should be designed based on the type of routing required, i.e. whether proactive, reactive or hybrid approach and knowing the buffering or transmission delays that may be caused during path finding and forwarding.

The control packets used for route finding and maintenance could be route request, route reply and route error. These packets can be designed such that it does not consume much bandwidth and processing time and components, by including only the necessary fields. Some algorithms design the route reply packets with all the intermediate nodes' addresses on the way. If it is a large-scale network, this will take much more bits for storing all the addresses. Solutions can be devised to carry only the addresses of cluster heads or few addresses.

The overall goal of the routing process is to device algorithm that helps efficient data transmission. Efficiency here can be viewed as successfully transmitting the data packets with less loss, less congestion, and less retransmissions. When multiple sensors transmit the same information, the data can be aggregated at a node which minimizes the number of data packet transmissions. Involving machine learning algorithms, in devising routing algorithms help the process to identify paths in uncertain and unknown environments. Reinforcement learning algorithms best suits for uncertain environments and in IoT networks the dynamic nature of the network can be predicted for path finding or go along with the changing nature of the network for path finding. As machine learning algorithms involve more iteration for the learning process, optimization helps in reducing the resource usages.

A. *Contribution of our work*

Considering the benefits of multipath routing in IoT, the main contributions of our work are as follows:

- We propose a reinforcement learning based multipath and QoS routing algorithm considering priority of the packets, residual energy of sensors and congestion on the path.
- Design of a dynamic Q-routing table.
- Increase the lifetime of the network by avoiding hot spots, network holes and nodes diminishing.
- We propose fault tolerance through multipath routing.
- We provide QoS solution by efficient load balancing.
- Simulation results on the proposed multipath load balancing algorithm have been compared with existing load balancing protocols.

Rest of the paper is organized as follows: Section 2 gives the literature review. Section 3 briefs out the problem statement.

In section 4 we have shared the motivation of the proposed work. Section 5 explains the methodology of the proposed work. Section 6 gives the simulation results and in Section 7 we conclude the paper.

2. LITERATURE REVIEW

Large scale IoT networks are used for monitoring or tracking the immediate environment, which collects enormous amount of data, and must move all these different collections upward towards the same sink. When a best path is chosen for data transmission, the resources of the same intermediate nodes very soon will be exhausted and goes down and unusable anymore. When the collected data is tactically handled, concerning the usage of resources, the network resources will be managed efficiently, and the lifetime of the network is improved.

A. Reliability through Multipath Routing

Reliability is a feature of routing algorithms which keeps the network users unaware of the faults that occur in the network, but still maintains the purpose of the network. The authors of [9] have proposed a routing scheme based on a bridge node participation mechanism. The scheme achieves real time reliability by giving the bridge node privilege to observe the transmission failures occurring on the path. Using these bridge nodes, the path can be diverted through other live nodes if there are link failures or node failures.

B. Load balancing through Multipath Routing

When the network is shared by multiple nodes or group of nodes of same application or different applications, and if the same path is used again and again concerning its optimized benefits, this may lead to the ruin of the path over a period. This criticality calls for load balancing in the networks. Due to the limitations found in the existing routing algorithms like end-to-end delay, communication cost and so forth, the authors of [10] have proposed a multipath load balancing algorithm based on Dynamic hop selection static routing protocol (DHSSRP). The protocol chooses minimal nodes for data transmission and uses Poisson method to find the rate of messages that belong to a priority class. Chinyang [11] has proposed a multipath load balancing routing for Internet of Things networks, which is based on two main designs. The first design aims to provide multiple paths for routing based on the distance to the IoT gateway node. The second design is based on find the least loaded path for data transmission. In [12], the authors have proposed a routing algorithm based on a virtual layout infrastructure in the network. the algorithm is based on clustering, where the sink node finds a merit value for every cell in the virtual grid. The merit value is based on the residual energy of the nodes inside the respective cell and the distance between the middle of the cell to the sink node. Using the merit value, the sink finds the next hop node of every cluster head.

C. Q-Learning based Multipath Routing

Authors of [13] have proposed a scheme where an agent measures the network load using the user data and the network traffic using deep belief network method to accomplish load balancing. The load prediction is performed by the Q learning algorithm. Yue Xu et al have proposed a load balancing scheme using deep learning to overcome issues caused with traffic fluctuations in ultra-dense networks [14]. A machine learning based load balancing technique in heterogeneous networks has been proposed in [15]. The authors have used four stages in the process: data pre-processing, hidden patterns discovery, supervised learning and decision-making model. Authors of [16] have contributed a Q-Learning based Load Balancing Routing for vehicular networks. They use Q-learning and find the routing policy decision. A gradient based routing protocol as an enhanced Q-learning approach has been proposed in [17]. The authors by incorporating Q-learning algorithm have found benefits like achieving the routing goals by changing only the reward function and also mentioned, the routing path varies based on the network load. Q-learning approach has been used in [18] for improving the network lifetime, for energy efficient data gathering and routing. Here the reward function is based on the number of packets successfully sent and the packets that are not sent. The forwarding node is based on the neighbour with maximum residual energy.

D. QoS based Multipath Routing

QoS requirements calls for optimized and resource aware routing algorithms, that improves the routing process and achieves efficient resource utilization. To avoid energy hole problem, an energy efficient load balancing approach has been proposed in [19]. In this paper, authors have used Grey Wolf Optimization and two fitness functions for routing and clustering. Authors of [20] have proposed Swarm intelligence-based approach for energy efficient multipath routing. This protocol works on three stages: neighbour finding through the link information, data transmission and efficient and reliable data delivery from source to destination. A high-density traffic scenario and an energy efficient optimal multipath routing have been proposed in [21]. The authors have considered lifetime, resilience, and the traffic density to find the next hop node. An optimal multipath routing based on QoS achievement has been proposed in [22], where the optimal path is found based on the optimal cost factor considering the lifetime and congestion in a node. Authors of [23] explain a bio-inspired multi-swarm optimization strategy to obtain k-disjoint multipath routes.

E. Existing Systems

In the review of literature on routing mechanisms for IoT networks, some of the limitations identified are produced below:

- Many approaches are based on static schemes, which may lead to high communication overheads and such schemes are un-optimized. This may lead to the consumption of more resources in the already resource constrained network.

- Considering alternate communications, switching between two paths may not be able to achieve required minimal end to end delay for prioritized transmissions if the alternate path is un-optimized and if the path is a high latency path.
- Choosing minimal nodes for transmissions should not choose the same nodes for every transmission. Doing so may lead to hot spots and causes network holes.
- Schemes need huge volume of user and network data for predicting the variable of interest, that involves in learning process.
- Takes longer time to learn and make decisions which may not be suitable for real time implementations.
- Dynamic nature of IoT networks is not taken into consideration.
- Some schemes do not consider Q-Learning approach with dynamic parameters, but where these parameters play a major role in achieving optimization in the routing process.
- Some are centralized schemes which are not reliable.
- Some schemes consists of consists of complex processing that consumes more computational and storage resources.
- Priority based algorithms lock the path for high priority data transmissions for a particular period. This may keep the resources of the path idle even when the other paths suffer congestion.

3. PROBLEM STATEMENT

The IoT network is modelled as a set of sensor nodes Si. i represents the sensor nodes, where $i = 1, 2, 3....n$. n is the total number of sensor nodes during deployment of the network for a large-scale application. The proposed work aims at using the network resources efficiently, to prolong the liveliness of the n nodes for a maximum expected lifetime of the network. The proposed work also aims at maintaining the optimized paths between multiple sources and destination nodes without vanishing due to the hot spot problem.

Reinforcement learning algorithm has been used as we consider the randomness involved in the IoT in form of mobility of nodes, liveliness of nodes, liveliness of links and scalability of the network. Q-learning is a reinforcement learning algorithm that computes a quality value with respect to the neighbouring nodes. This quality value called as q-value is maintained in the q-routing table. We calculate a q-value for every neighbouring node. The q-value is found from the congestion status, buffering level and distance, of the path from the sink node. Then we use the q-value to identify the optimized path.

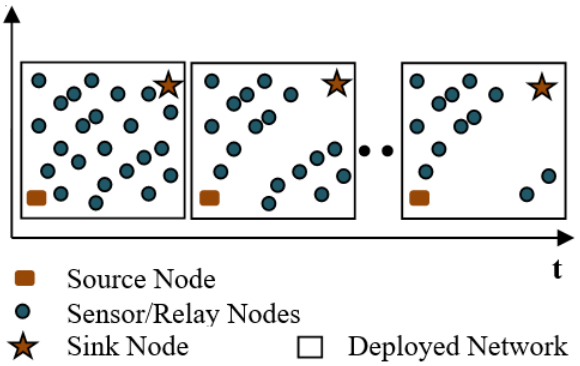

Figure 1. The network with multiple sensor or relay nodes, a source node, and a sink node at the time of deployment is shown in the first block. Second block shows the optimized path is lost after some time of deployment. The third block shows more nodes are lost on the way to destination and this leads to two nodes isolated from the functional network.

4. MOTIVATION OF THE PROPOSED WORK

Figure 1 illustrates the WSN portion of an IoT network in square blocks, at different time durations. During the time of deployment, the density of the network is more as shown in the first block of the timeline. The first block is compared with the other two blocks shown in the same figure. This representation visualizes the loss of sensor nodes in the optimized path due to the usage of same path for a huge number of transmissions and leads to the entire path loss at some point of time and even isolates nodes in the network.

Block 2 shows after the optimized best path is completely lost after some time of network deployment. Usually then the routing algorithms go in search of the optimized path with the remaining nodes. The process continues until there is no way finding a connection through intermediate nodes from the source to the destination. Hence, we propose a load balancing algorithm that aims at maintaining the density of the network approximately the same throughout the network usage. This helps in providing optimized path for real and time critical data packets to reach the destination without transmission delay. At the same time the unused paths will also be used for transmission of less priority or off-line data.

A. Q Routing

The Q routing algorithm considers the different nodes in the networks as various states of the algorithm. Q Learning comes under the category of reinforcement learning algorithms, which can be used for solution finding in uncertain networks. The algorithm starts assuming an initial state and moves ahead to find the next states and identifying the action that leads to the next state. For every state change a q value is calculated and the highest q value state in comparison is selected as the best next state. The selection of actions and next state is altogether assumed as being performed agents installed in every node.

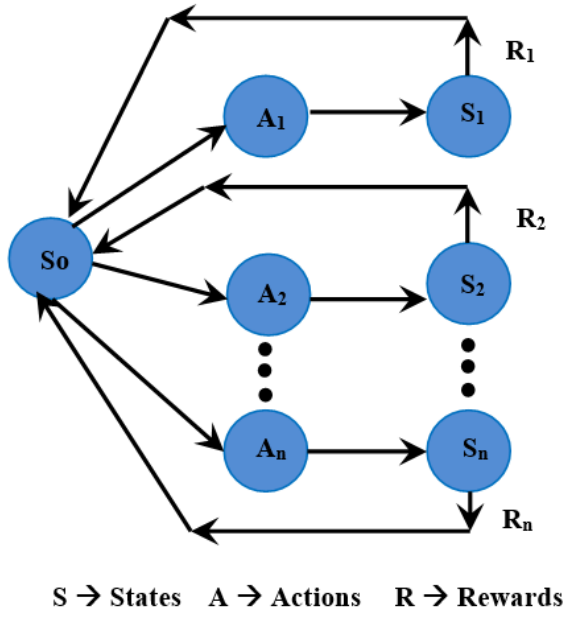

Figure 2. S0 is the initial state. On choosing one of the actions from A1 to An, the agent selects the respective state as the next state. Corresponding reward is received for the action selected.

5. Proposed Methodology

Load balancing and multipath routing in IoT helps to solve various issues like fault tolerance, network lifetime, resource utilization and QoS provisioning. The increasing IoT applications call for more research improvements in the specified area as seen in the literature. Even though various works exists for load balancing and multi-path routing as specified in the literature survey section, and proved to be better than existing, these schemes are limited to specific characteristics of IoT or they are developed only for a specific application. Being needed to be deployed for heterogeneous data communication and heterogeneous network interconnections, IoT throws challenges to the researchers which when focused will bring revolution in the IT application sector.

The entire network is assumed to consist of cooperative agents that work in synchronization to achieve path finding by sharing routing information and rewards. The main objective of these agents is to achieve receiving maximum rewards by performing iterations on hyper parameters, q-values and policy updates. It performs trial and error task in this process of learning. Every sensor node has one agent each and exchanges their rewards to each other. The agents strive to identify the next state, which is the best next hop, through a best action, in reaching the destination state. When there are multiple actions available, the agent must choose the best action that maximizes the reward. The transition probability from one state to another state depends on the type of the data packet, congestion level of the path, queuing status of the node and the distance from the root node. The ¡state, action, reward¿ scenario of the q-learning process is shown in figure 2. Every node has an agent which is responsible for selecting the best action in order to reach the best state. Whichever state returns maximum reward is considered the best state. Through multiple select and receive options, the agent learns the best action and best state.

The proposed work is explained as a sequence of steps using the flowchart of figure 3. Initially every agent is made aware of the available of the states and actions. The agent randomly chooses an action, and this action makes the agent to select a next state from the current state. When a new state is selected, that new state returns a reward to the agent which selected an action to reach the new state. This reward helps to calculate the q-value of the new state in the current state. The learning process continues to learn and update the q-values of new states selected by the current agent and its actions. Along with this a policy is involved that helps choosing an optimized path for the high priority data and choosing a path for load balancing for other data. Until a maximum q-value is reached through the maximum achievable reward, the q-value stored in the q-routing table and the policy is both updated. At the end of the learning process, the policy concludes selecting actions and the states and makes the algorithm proactive.

The goal of reinforcement learning is to maximize the reward and update the action to be taken until an optimum action is recognized. Hence the agents use Bellman Optimality Equation 5 by considering the possible future states that the selected action leads to. The state-action value, called as the q-value is given by

$$Q_{n+1}(S,A) \leftarrow \sum_{(S_{n+1})} P(S,A,S_{n+1})\,[R(S,A,S_{n+1}) + \gamma \cdot \max_{A_{n+1}} Q_k(S_{n+1},A_{n+1})\,] \quad \forall (S,A) \tag{1}$$

Where, $P(S,A,S_{n+1}) \rightarrow$ Transition probability for the transition from state S to state S_{n+1} by choosing action A.

$R(S,A,S_{n+1}) \rightarrow$ Reward received for the transition from state S to state S_{n+1} by choosing action A

$\gamma \rightarrow$ Discount rate

The optimal policy responsible for the selection of an action that leads to optimal q-value is denoted by $\pi^*(S)$. The agent accepts the optimal policy for the case when an action produces the highest q-value:

$$\pi^*(S) = \arg\min_{A} Q^*(S,A) \tag{2}$$

Routing in IoT network is modelled as a Markov Decision Process (MDP), and the dynamic nature of the network should be captured. Hence the learning process considers

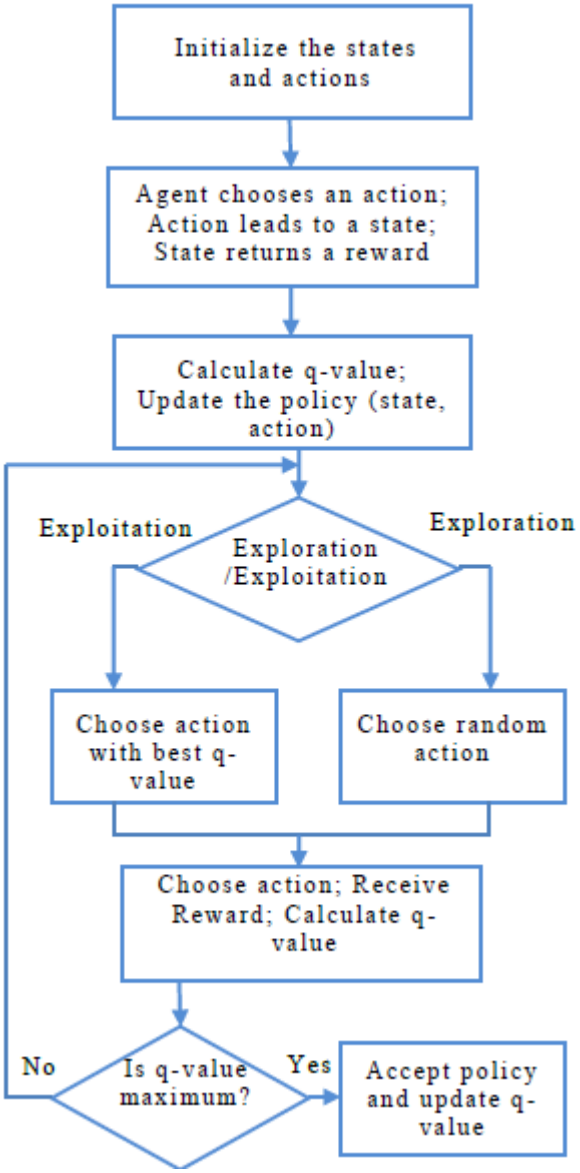

Figure 3. Flow chart describing the proposed methodology

the newly joining node and calculates the q-value of the new node and tries to choose the new node for iterating with the future rewards multiple times to converge its q-value. The MDP initially starts with the available nodes and links to identify the states and actions by the learning process and gradually learns the new states and their transition probabilities. Equation 3 is derived from 5, as initially the transition probabilities and the rewards are not known.

$$Q_{n+1}(S,A) \leftarrow (1-\alpha)Q_n(S,A) + \alpha(R+ \gamma \cdot \max_{A_{n+1}} Q_k(S_{n+1},A_{n+1}) \quad (3)$$

Where, $\alpha \leftarrow$ learning rate

$R \rightarrow$ average of the rewards on taking an action leaving a state and the corresponding expected future rewards.

As the algorithm is expected to converge to an optimal value and the learned policy is optimal, we choose the maximum of the q-value estimates of the next state.

When new states add on or leave off in the MDP, and to obtain an optimal policy learning every state, a random policy is executed by the agents. This random policy is an exploration policy called as ϵ-greedy policy.

One of the main concerns of IoT network is working with heterogeneous data transmissions. In the proposed work, to provide QoS routing for heterogeneous data transmission, the data packets are divided into two priority classes as shown below:

Class 1: Hard real time data

Class 2: Soft real time data

The ϵ-greedy policy is based on providing priority-based data transmissions. ϵ-greedy algorithm for Prioritized data transmission using the two different classes is given below:

If Class 1 data transmission needed:

Select states with probability 1-ϵ

Else if Class 2 data transmission needed:

Select states with probability ϵ

Exploration and Exploitation in figure 3 relates to ϵ-greedy policy. Exploitation makes use of the best path on more probability. Exploration allows selecting even the unused paths, so that the unused resources can be utilized as well as the load of data transmission can be shared among multiple paths.

A. *Q-Routing table*

In order to provide load balancing by choosing the actions not performed much, the eqn. 3 – the Q-learning algorithm can be considered as a function of Q-value of a state with its number of occurrences in the previous selections. Hence equation 3 is modified to 4 as:

$$Q(S,A) \leftarrow (1- \leftarrow)Q(S,A) + \alpha(R+ \gamma \cdot \max_{A_{n+1}} f(Q(S_{n+1},A_{n+1}),N(S_{n+1},A_{n+1})) \quad (4)$$

Where,

$N(S_{n+1},A_{n+1}) \rightarrow$ Number of times an action A_{n+1} is selected corresponding to S_{n+1}

$f(Q,N) \rightarrow$ Load balancing function The load balancing function is given by

$$f(Q,N) = Q + \frac{l}{1+N} \quad (5)$$

$l \rightarrow$ Load monitoring parameter

To achieve QoS routing, the algorithm explores the congestion level, buffer length and the hop count from the root node, for every next states while finding $Q(S, A)$. The load monitoring parameter is a hyper parameter that depends on the mentioned three parameters. Hence the load monitoring parameter is given by:

$$l = \frac{1}{c + b + h} \tag{6}$$

Where,

$c \rightarrow$ State of congestion through the next state S_{n+1}

$b \rightarrow$ State of buffer length of the next state S_{n+1}

$h \rightarrow$ Number of hops of S_{n+1} from the root node

Eqn 6 ensures that when the congestion or buffer length or the distance is more then such paths will be selected with less probability.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example

6. SIMULATION RESULTS

In this section we discuss about the simulation of our proposed work and analyze the performance. Simulation was performed using Python 3. The simulation has been executed with 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 sensor nodes distributed in a square area. We used the random spring topology as depicted in the Figure 4. Figure 4 is a sample network structure depicted with 50 nodes for the WSN portion of the IoT network. We perform comparison of the proposed work on load balancing and multipath routing (QQoS) with the conventional Q-learning algorithm ('QRout' meaning the Q routing algorithm, as referred for comparison). QQoS achieves optimized routing path where as QRout tries to find shortest routing path. In order to find the optimized path, QRout considers the congestion parameter, buffer length parameter and the hop count parameter. Figure 5 shows the q-values of nodes in the network after convergence. The rows and columns represent the source and destination nodes of the network. The neighbour with maximum q-value will be selected as next hop node during data forwarding. To provide load balancing, the nodes with lower q-value also are selected on exploration. Figure 6 shows the convergence of reward through multiple iterations. Only when the reward is maximized and leads to maximum q-value, the policy is freezed. The graph shows the number of iterations Vs. the reward gained.

A. *Residual Energy*

Routing algorithms generally consumes nodes' battery power for control packets transmission and reception during the routing process. Usage of control packets must be minimized to reduce the power consumption.

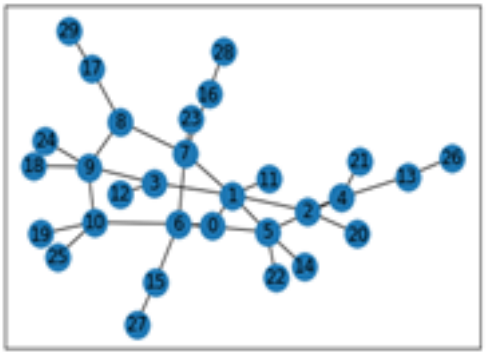

Figure 4. Spring layout structure of 30 sensor nodes

```
[
[ 0.  63.65303544  0.    0.   0.    0.    79.5662943   0.
  0.        0.      ]
 [ 61.15245099   0.   50.92242835   79.5662943   0.
  62.6771976  0.   62.6771976   0.    0.      ]
 [ 0.   3.65303544   0.        0.        50.14175808   0.
  0.        0.        0.        0.      ]
 [ 0.       63.65303544  0.        0.        0.   0.        0.
  0.        0.       100.      ]
 [ 0.       0.      50.14175808  0.        0.   63.65303544
  0.        0.        0.        0.      ]
 [ 0.     62.6771976   0.        0.      50.14175808   0.
  79.5662943   0.       0.       0.      ]
 [ 63.65303544   0.     0.     0.     0.    63.65303544   0.
  62.6771976   0.       99.45786787]
 [  0.       63.65303544   0.        0.          0.       0.
  78.34649701  0.       78.34649701  0.      ]
 [ 0.       0.        0.        0.          0.     0.          0.
  62.6771976   0.       99.45786787]
 [ 0.  0.     0.    80.       0.    0.       79.5662943   0.
  78.34649701  99.45786787]
]
```

Figure 5. q-values calculated by a routing node

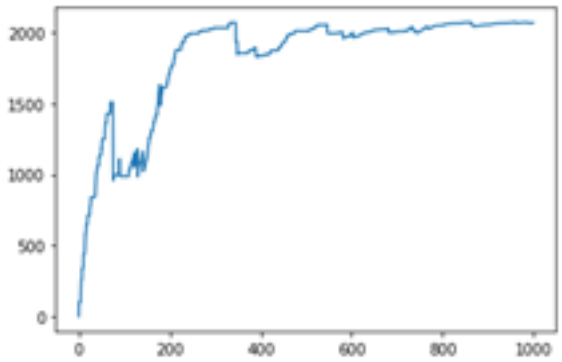

Figure 6. Convergence of reward gained by a node for learning in 1000 iterations; x axis – Number of iterations; y axis – Reward gained

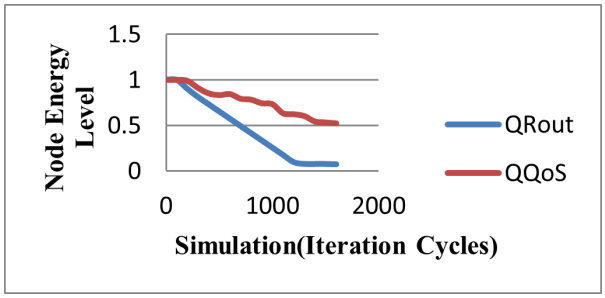

Figure 7. Simulation time based on iteration cycles Vs. Node energy level

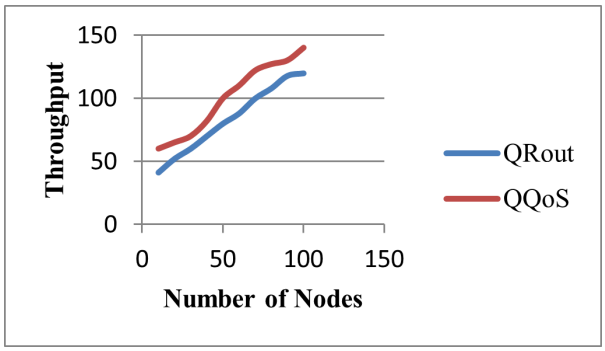

Figure 8. Number of nodes Vs. Throughput

The proposed optimized and enhanced hyper parameter-based Q algorithm is simulated and compared with the conventional Q-routing algorithm and the results are shown in figure 7. The figure illustrates the energy balancing in the IoT network using the conventional Q routing algorithm (QROUT) and the proposed Q routing algorithm based on QoS parameters (QQoS). It is found that energy utilization of the nodes is well balanced in the network by the proposed algorithm and hence even after multiple iterations the minimum energy of the living node is better compared with conventional algorithm. The duration of the time the network is alive as meant for data sharing to the sink nodes is an expected criterion as the routing algorithm plays a major role in increasing the lifetime of the network.

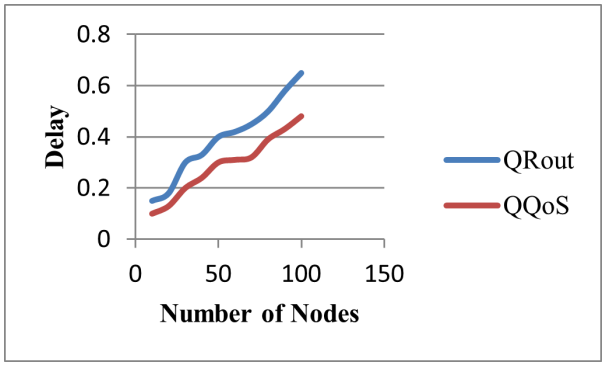

Figure 9. Number of nodes Vs. Packet transmission delay

B. Throughput

The measure of data packets reached the destination out of the packets generated at the source, gives an understanding of how successful the algorithm can support the network's goal in-spite of the heterogeneity and other issues in the network. The figure 8 illustrates the throughput achieved using QRout and QQoS.

C. Packet Transmission Delay

The time difference between when the packet is transmitted at the source and when the packet is received at the destination is called the packet transmission delay. Packet transmission delay may be due to queuing delay, processing at each intermediate node, ignoring the link latency or the medium latency. In the proposed work as the q-value is already determined based on the path congestion and queuing delay, the forwarding of data packets is proactively ignoring such delays. The figure 9 shows the packet delay and the number of nodes for QRout and QQoS.

7. CONCLUSION

In this paper, firstly an investigation of factors that affect the lifetime of a heterogeneous IoT network is discussed. The recognized factors involved like congestion, queuing delay and distance of the node from sink is then used as hyper parameters in the routing process. Next, as reinforcement learning behaves well at unknown environments, QoS requirements as hyper parameters are used to find q-values of the neighboring nodes and the Q-routing table is obtained and maintained at each node for forwarding of data packets. The simulation results show the performance improvement of the proposed work in comparison with the conventional Q-routing. As future work, heterogeneous data transmission in a highly dynamic network environment using QoS based Q learning algorithm will be implemented.

References

[1] P. J. Ryan and R. B. Watson, "Research challenges for the internet of things: what role can or play?" *Systems*, vol. 5, no. 1, p. 24, 2017.

[2] T. Alsboui, Y. Qin, R. Hill, and H. Al-Aqrabi, "Distributed intelligence in the internet of things: Challenges and opportunities," *SN Computer Science*, vol. 2, no. 4, pp. 1–16, 2021.

[3] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile networks and applications*, vol. 24, no. 3, pp. 796–809, 2019.

[4] B. Pourghebleh and V. Hayyolalam, "A comprehensive and systematic review of the load balancing mechanisms in the internet of things," *Cluster Computing*, vol. 23, no. 2, pp. 641–661, 2020.

[5] A. Khanna and S. Kaur, "Internet of things (iot), applications and challenges: a comprehensive review," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1687–1762, 2020.

[6] Z. Ruan and H. Luo, "Scalable and efficient routing protocol for internet of things by clustering cache and diverse paths," *International Journal of Embedded Systems*, vol. 14, no. 2, pp. 160–170, 2021.

[7] R. Chaudhry and S. Tapaswi, "Novel routing mechanism for iot-based emergency rescue scenarios," *International Journal of Mobile Communications*, vol. 17, no. 4, pp. 465–482, 2019.

[8] M. Kumar, A. F. Minai, A. A. Khan, and S. Kumar, "Iot based energy management system for smart grid," in *2020 International Conference on Advances in Computing, Communication & Materials (ICACCM)*. IEEE, 2020, pp. 121–125.

[9] S. Kim, C. Kim, and K. Jung, "Cooperative multipath routing with path bridging in wireless sensor network toward iots service," *Ad Hoc Networks*, vol. 106, p. 102252, 2020.

[10] M. Adil, "Congestion free opportunistic multipath routing load balancing scheme for internet of things (iot)," *Computer Networks*, vol. 184, p. 107707, 2021.

[11] C. H. Tseng, "Multipath load balancing routing for internet of things," *Journal of Sensors*, vol. 2016, 2016.

[12] R. Yarinezhad and S. N. Hashemi, "Solving the load balanced clustering and routing problems in wsns with an fpt-approximation algorithm and a grid structure," *Pervasive and Mobile Computing*, vol. 58, p. 101033, 2019.

[13] H.-Y. Kim and J.-M. Kim, "A load balancing scheme based on deep-learning in iot," *Cluster Computing*, vol. 20, no. 1, pp. 873–878, 2017.

[14] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load balancing for ultradense networks: A deep reinforcement learning-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9399–9412, 2019.

[15] C. A. Gomez, A. Shami, and X. Wang, "Machine learning aided scheme for load balancing in dense iot networks," *Sensors*, vol. 18, no. 11, p. 3779, 2018.

[16] B.-S. Roh, M.-H. Han, J.-H. Ham, and K.-I. Kim, "Q-lbr: Q-learning based load balancing routing for uav-assisted vanet," *Sensors*, vol. 20, no. 19, p. 5685, 2020.

[17] B. Debowski, P. Spachos, and S. Areibi, "Q-learning enhanced gradient based routing for balancing energy consumption in wsns," in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2016, pp. 18–23.

[18] N. Javaid, O. A. Karim, A. Sher, M. Imran, A. U. H. Yasar, and M. Guizani, "Q-learning for energy balancing and avoiding the void hole routing protocol in underwater sensor networks," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 702–706.

[19] A. Lipare, D. R. Edla, and V. Kuppili, "Energy efficient load balancing approach for avoiding energy hole problem in wsn using grey wolf optimizer with novel fitness function," *Applied Soft Computing*, vol. 84, p. 105706, 2019.

[20] A. Nayyar and R. Singh, "Ieemarp-a novel energy efficient multipath routing protocol based on ant colony optimization (aco) for dynamic sensor networks," *Multimedia Tools and Applications*, vol. 79, no. 47, pp. 35 221–35 252, 2020.

[21] K. Jaiswal and V. Anand, "Eomr: An energy-efficient optimal multi-path routing protocol to improve qos in wireless sensor network for iot applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2493–2515, 2020.

[22] Jaiswal, Kavita, Anand, and Veena, "An optimal qos-aware multi-path routing protocol for iot based wireless sensor networks," in *2019 3rd international conference on electronics, communication and aerospace technology (ICECA)*. IEEE, 2019, pp. 857–860.

[23] M. Z. Hasan and F. Al-Turjman, "Optimizing multipath routing with guaranteed fault tolerance in internet of things," *IEEE Sensors Journal*, vol. 17, no. 19, pp. 6463–6473, 2017.

T. C. Jermin Jeaunita T. C. Jermin Jeaunita. She is currently a research scholar in the Department of Computer Science and Engineering in PESIT Bangalore South Campus, Bangalore, India. She completed her B. E in Computer Science and Engineering from St. Xavier's Catholic College of Engineering, Tamilnadu, India and M.E in Computer Science and Engineering from Noorul Islam College of Engineering, Tamilnadu India. She is pursuing Ph.D in Computer Science and Engineering in Visveswaraya Technological University, Bangalore, India. Her research interest lies in peer to peer networks, WSN, IoT and Reinforcement Learning.

Sarasvathi V Dr. Sarasvathi V. She is currently working as Associate Professor in Computer Science and Engineering in PESIT Bangalore South Campus, Bangalore, India. She has completed Ph.D in VIT University, Vellore, Tamil Nadu. Her research interest includes Wireless Ad-Hoc Networks, Sensor and Mesh Networks, Internet of Things, Cloud Computing, Network Optimization and Performance computing. She had nearly 12 research publications in reputed peer reviewed international journals and conferences. She served as Guest Editor for Special Issue on: "Emerging Trends, Applications and Services in Communication Networks", International Journal of Communication Networks and Distributed Systems-Inderscience Journal and as Editor for IGI Global "Handbook of Research on Applied Cybernetics and Systems Science".