MATTER: International Journal of Science and Technology
ISSN 2454-5880

Sebastian & Sivagurunathan, 2018

Volume 4 Issue 2, pp. 74-92

Date of Publication: 23rd August 2018

DOI-https://dx.doi.org/10.20319/mijst.2018.42.7492

This paper can be cited as: Sebastian, A., & Sivagurunathan, S. (2018). Load Balancing Optimization for RPL Based Emergency Response Using Q-Learning. MATTER: International Journal of Science and Technology, 4(2), 74-92.

LOAD BALANCING OPTIMIZATION FOR RPL BASED EMERGENCY RESPONSE USING Q-LEARNING

A. Sebastian
Dept. of Computer Science and Applications, Gandhigram Rural Institute, Dindigul, India
sebastinsj@gmail.com

Dr. S. Sivagurunathan
Dept. of Computer Science and Applications, Gandhigram Rural Institute, Dindigul, India
s.sivagurunathan@ruraluniv.ac.in

Abstract

Internet of Things technology has given rise to Smart Cities, Smart Health, Smart Transport Logistics, Smart Production and Supply chain management, Smart Home and many more. For IoT deployments, ROLL-WG has standardized Routing Protocol for Low Power and Lossy Networks (RPL) for urban environment (RFC 5548). RPL is designed to address the needs of constrained IoT environment. RPL uses Objective Functions (ETX & Hop Count) to optimize route selection. Many new Objective Functions for IoT applications are suggested by researchers to optimize path selection. Load Balancing Optimization for emergency response is least explored. In this article, we propose load balancing optimization for RPL based emergency response using Q-learning (LBO-QL). We have tested the proposed model in Contiki OS and Cooja simulator. Proposed model provides improved efficiency in Packet Delivery Ratio, Traffic Control Overhead and Power consumption. Hence, DODAG optimization using Q-Learning for disaster response is effective in optimized usage of constrained resources for disaster response operations with improved efficiency and reliability.

Keywords

Internet of Things, RPL, Load Balancing Optimization, Disaster Response, Multi Agent Q-Learning

1. Introduction

Internet of Things (IoT) network is a fast growing communication paradigm. Deployment of IoT networks range from smart agriculture, smart city, smart transport logistics, smart health and many more. IoT network also provides scope for communication in dangerous scenario like forest fire, detecting harmful chemical leakage, disaster situations like flood, cyclone, fire in the building, building collapse etc. Probing IoT network for disaster response is a felt need. When disaster strikes, power supply and communication networks are the most affected areas. In such scenario, fixed network or traditional IP based internet is inadequate. Hence, communication network which is dynamic, optimized, energy efficient and fault tolerant is essential. Further such network will be affected by inherent communication challenges such as lossiness, frequent loss of devices and links, sudden surge of nodes, congestion, collision due to interference, etc. IoT networks with low cost sensors and actuators are capable of delivering the desired results. RPL based IoT network is basically a collection based network. RPL based IoT network is shown in figure 1.

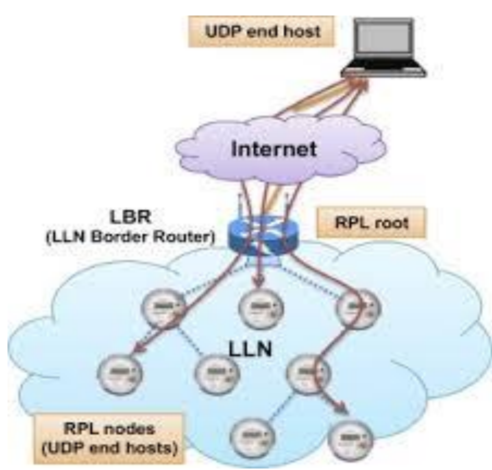

Figure 1: *RPL based LLN network*

Moreover, data collection through constrained resource objects and learning based algorithm are gaining popularity in real time applications. Reinforce Learning is very helpful in automation in dangerous scenarios due to its self learning capabilities (Abu-Elkheir et al., 2016).

Hence IoT has potential to serve emergency response by its smart connected environment. This article focuses on extending Q-learning algorithm to increase responsiveness to emergency situations and improve network efficiency of disaster response scenario. Hence load balancing Optimization in RPL using q-learning provides better performance than standard objective functions. Moreover, the network topology takes in to account the resource availability of all links and nodes with its rewards making the network reliable.

Emergency and Disaster Management has four pillars: Preparedness, Response, Recovery and Mitigation. Optimization of emergency response includes search and rescue, relief, evacuation, medical aid and coordination operations (Cisco, 2015). These operations are depicted in Figure 2.

In section Two, we discuss background of RPL, optimization and load balancing schemes. In section three, discuss related work. In section four, motivation and the need for load balancing optimization is specified. In section five, the design of load balancing optimization for RPL based emergency response using q-learning technique is explored. In section six, we analyze and discuss the results to justify our proposed method followed by conclusion and future works.

Figure 2: *Emergency Response Operations*

2. Background

IPv6 Routing Protocol for Low Power and Lossy Networks (LLNs) consist of low cost resource constrained nodes. The RPL based emergency response network will have sensors, actuators, wearable devices, monitoring cameras, etc. The network is also dynamic with large number of nodes joining and departing in quick time span. Since emergency response networks are very crucial to save life and infrastructure from further damage, network performance should have highest priority. The design and choice of routing protocols for emergency response must be scalable, fast converging and reliable, stable at the face of lossy links and unpredictable environment. In emergency response scenario, effective communication within LLN and with outside world is very important. Unlike other RPL based applications where the reporting interval is user dependent, in emergency response environment frequent reporting means efficient use of network resources.

2.1 RPL

Internet Engineering Task Force (IETF) has standardized RPL for applications such as home automation (RFC 5826), urban environment (RFC 5548), industrial control (RFC 5673) and building automation (RFC 5867) (A. Al-Fuqaha et al., 2015). RPL is a destination vector protocol for low power devices. In RPL, nodes organize themselves by forming a Destination Oriented Directed Acyclic Graph (DODAG) rooted towards the sink (DAG ROOT) identified by a unique identifier DODGID. A DODAG is uniquely identified by a combination of RPLInstanceID and DODAGID. A node Rank defines the nodes individual position relative to other nodes with respect to a DODAG root. Rank strictly increases in the DOWN direction and decreases in the UP direction. The exact way Rank is computed depends on the DAGs Objective Function (OF). The Rank is a simple topological distance, may be calculated as a function of link metrics and may consider other properties such as constraints. Expected Transmission Count (ETX) and Hop Count (of0) are the default OFs in RPL. The DODAG root may act as a border router for the DODAG and aggregate routes in the DODAG and may distribute DODAG routes to other routing protocols. RPL supports multipoint-to-point (MP2P), point-to- multipoint (P2MP) and point-to-point (P2P) traffic. Figure 3 explains the forming of DODAG (E. Borgia, 2014).

For the construction and maintenance of DODAG, RPL nodes transmit DODAG Information Object (DIO) messages. A DIO message contains information that allows a node to discover a RPL instance, learns its configuration parameters, learn the OF used and maintain upward routing topology. In order to join a DODAG, a node either can wait to receive DIO messages from nearby nodes or it can send a DOADG Information Solicitation (DIS) request to neighboring nodes. A node uses rank property in order to select another node as a parent. The rank property is a combination of one or more metrics and constraints into a value. Objective Function (OF) is used to calculate rank property. When the node has some data that needs to be sent to the root, it immediately sends this to the preferred parent. The parent node sends to its own parent and so on until it reaches the DODAG root (J. Gubbi et al., 2013).

2.2 Optimization in RPL

In RPL, parent and path selection are optimized by objective functions. Minimum Rank Hysteresis Objective Function (mrhof) and Objective Function Zero (of0) are provided in standard RPL. Mrhof uses Expected Transmission Count (ETX) metric to calculate node's rank. Of0 uses hop count metric to compute node rank. The parent selection optimization is done with objective function which provides low rank. Low parent rank indicates that the node is closer to the collection point or sink node. The node rank increases monotonically from the root to the leaf node. The node rank also suggests the depth or the density of the network. Many research scholars have suggested RPL optimization other objective functions (Han. D and Gnawali, 2012). These objective functions use more than one metric called composite metric for rank computation. The rank update is computed from equation (1) and (2).

$$\text{Rank (N)} = \text{Rank (PN)} + \text{RankIncrease} \quad (1)$$

$$\text{RankIncrease} = \text{Step} * \text{MinHopRankIncrease} \quad (2)$$

Where Step represents a scalar value and MinHopRankIncrease represents the minimum RPL parameter. Many of these objective functions are aimed at applications where optimized routing for energy, packet delivery ratio, stability, etc., is required. However, few works on load balancing optimization is available in the literature. For many applications and especially emergency response scenario needs load balancing optimization is essential for stability, efficiency and increased network life time.

2.3 Load Balancing in RPL

In the RPL-based mesh network, due to the lack of balance algorithm, large numbers of leaf nodes select the same parent node and leave others empty or act as a just forwarding node. At the next DODAG construction, the rank with many leaf nodes triggers all child nodes to re-select and switch their parent node due to increase in rank. This way frequent switching of parents greatly decrements the network efficiency, unstable topology and deplete constrained resources. Therefore, there was an urgent need to develop a mechanism to select optimized parent selection using load balancing.

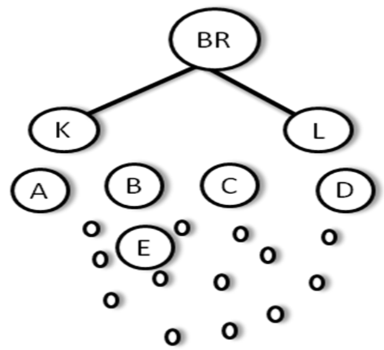

Figure 3: *Load Balance RPL (LB-OF)*

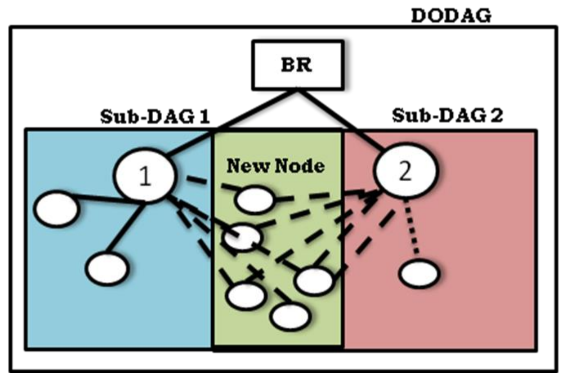

Figure 4: *Load Balance using child count*

2.3.1 Child Node Count (CNC)

CNC based optimized parent selection is shown in figure 3. As depicted the RANK of parents K and L are supposed to be equal, but parent K by chance dominates in the number of child nodes even though they share the common coverage. It is a waste of resource that parent L is left empty while many new nodes become child of node 1. When the network achieves a relative balance a new node will create imbalance. If the new node is situated in the vicinity of root node will create serious imbalance of network. The authors suggested CNC based parent selection (W. Almobaideen et al., 2016). CNC field gives the number of child node of preferred parent. Using a combined metrics of ETX and CNC, parent selection is carried out. Using CNC object as constraint load balancing is achieved.

2.3.2 Load Balanced Objective Function (LB-OF)

In Load Balanced Objective Function (LB-OF), data traffic is balanced by taking in to account the number of children for each preferred parent. When the leaf node receives DIO from parent node, it updates its own rank and broadcasts updated DIO to neighboring nodes including the preferred parent. In normal RPL, the preferred parent discards the DIO messages received from leaf node. But in LB-OF, the preferred parent records the DIO information details and when the parent address of leaf node matches the preferred parent address, it increments the child count. In DODAG construction process, new leaf nodes will avoid parents with more node count and thus avoid bottle neck and load imbalance in future (P. H. Gomes et al., 2016).

2.3.3 Child Count Based Load Balancing RPL (Ch-LBRPL)

Load balancing using CNC and LB-OF achieves some load balancing in the RPL network but in the long run it still creates imbalance. The authors introduced Child Count based Load balancing RPL (Ch-LBRPL), novel method to identify the common nodes between the preferred parents 1 and 2. This scenario is explained in figure 4. In the figure, a DODAG has 2 Sub-DAGs with parents 1 and 2. Parent 1 has 2 existing child nodes and parent 2 has 1 existing child node. 5 new nodes fall in the communication range between parent 1 and parent 2. In traditional case, the new nodes will accept DIO from the preferred parent and join the parent with low RANK property. In such case, the children selecting parent is not balanced. The low RANK parent will have more child nodes than the other parent. So they devised a method by which the new child nodes are distributed to parents in such a way that both parents have same number of nodes. The parent 1 and parent 2 report the nodes in their radio environment to the root node (BR) and the BR makes the load balancing decision about the number of child node each parent should acknowledge. This way any number of new nodes will be shared between preferred parents equally. In this case,

Load Imbalance = Total number of nodes – existing nodes in parent (1)- existing nodes in parent (2) (3)

Total node in DODAG=Existing nodes of parent (1) and parent (2) + Load Imbalance Nodes (4)

Equations 3 and 4 are used to balance the nodes among the parent 1 and parent 2.

3. Related Work

RPL load balancing is very important for LLN since its application scenarios have numerous nodes with high node density. Many authors have suggested solutions to load balance the RPL that would improve stability of the network, extended network life time and improved network performance. The authors (H. S. Kim et al., 2016) suggested queue utilization (QU-RPL). QU-RPL is designed for each node to select its parent node considering the queue utilization of its neighbor nodes as well as their hop distances to an LLN border router (LBR). QU-RPL is effective in lowering queue losses and increasing the packet delivery ratio compared to the standard RPL. The authors (Marwa Mamdough et al., 2016) proposed, Minimum Degree RPL (MD-RPL) which builds a minimum degree spanning tree to enable load balancing in RPL. MD-RPL modifies the original tree formed by RPL to decrease its degree. The authors (X. Liu et al., 2014) proposed a load balanced routing protocol based on the RPL protocol (LB-RPL) to achieve balanced workload distribution in the network. LB-RPL detects workload imbalance in a distributed and non-intrusive fashion. It also optimizes the data forwarding path by jointly considering both workload distribution and link-layer communication qualities. The authors (J. Guo et al., 2014) designed an energy-balancing routing protocol that maximizes the lifetime of the most constraint nodes. They proposed the Expected Lifetime metric, denoting the residual time of a node (time until the node will run out of energy). They also designed mechanism to detect energy-bottleneck nodes and to spread the traffic load uniformly among them. The authors (Minkeun Ha et al., 2015) propose three multipath schemes based on RPL: Energy Load Balancing (ELB), Fast Local Repair (FLR) and their combination (ELB-FLR). The authors (O. Iova et. al., 2015) address the imbalance of traffic load among gateways. The load balancing between gateways is suggested to reduce the traffic congestion thereby enlarging the network capacity. They proposed dynamic and distributed load balancing scheme to achieve a global load fairness motivated by water flow behavior named Multi-Gateway Load Balancing Scheme for Equilibrium (MLEq).

The authors (R. Jadhav et al., 2017) suggested optimization of parent node selection using Child Node Count. CNC of preferred parent is considered for Rank calculation and parent selection. All of the methods suggested above provide partial load balance to RPL. Emergency response environment involves load balancing as well as optimization.

4. Motivation: The Need for Load Balancing Optimization for RPL based Emergency Response Using Q-Learning

Standard objective functions such as mrhof and of0 fail to provide load balancing to RPL network. The RPL extensions provide partial load balance but fail to provide optimization. These inadequacies make RPL based emergency response vulnerable to bottle neck problems, hotspot problems, thundering herd problems and unstable network which affects severely life expectancy of the network. These problems can hamper the emergency response operations. Therefore there is a great need for load balancing optimization for emergency response in RPL based networks. In the proposed work, we use load balancing optimization the Sub DAG and DODAG level using machine learning algorithm such as Reinforcement algorithm which is also called Q-Learning.

5. Design of Load Balancing Optimization for RPL based Emergency Response Using Q-Learning (LBO-QL)

The proposed LBO-QL scenario for emergency response is shown in figure 5. Emergency response scenario has border routers BR_1, BR_2, BR_3, ……, BR_{n-1}, BR_n. The BRs has the capacity to manage the communications within LLN or DODAG. For any communication with any other IPv6 based networks, BRs use gateway. RPL based emergency response mesh network has 1,2,3,….., 34, n-1, n nodes and A,B,C, ….., O, N-1, N leaf nodes waiting to join the network.

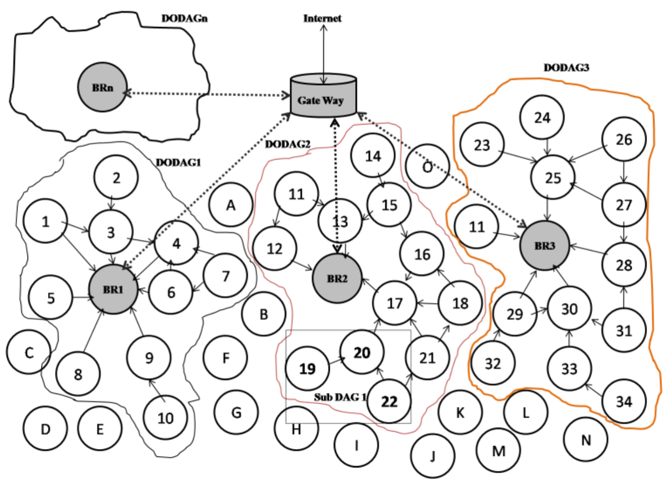

Figure 5: *Load Balancing Optimization for Emergency Response*

BR$_1$ has nodes 1 to 10, BR$_2$ has nodes 11 to 22 and BR$_3$ has nodes 23 to 34. The leaf nodes waiting to join the network are

At the edges of the communication range of BRs. The leaf nodes are at the edge or lie in between the communication range of BRs. BRs in RPL can be represented as DODAG1, DODAG2, …., DODAG$_{n-1}$, DODAG$_n$. DODAG also has many Sub DAGs. This way a node in RPL based emergency response can be topologically represented to be in a sub DAG and a DODAG. In a normal emergency response scenario, various operations are focused or space specific. A group of nodes or volunteers are instructed to do a specific task and there will be a group to coordinate all the tasks. In our case BR$_1$, BR$_2$, …, BRn-1 and BRn are collection points of specific emergency response teams. For example, BR$_1$ may do only sensing tasks. It will report the data related to the environment in emergency scenario. Humidity, temperature, pressure, smoke, toxic gas, etc are some of them. BR$_2$ may engage in search and rescue operations. BRn-1 may be coordinating other operations and reporting updates with the outer world. This way the communication requirements for each BR and related operation are different. Since RPL based IoT is a scalable network, emergency response network can have large number of nodes. We have omitted the mobility aspect in the emergency response scenario in our proposal. Hence in designing, we need to keep many of these requirements for successful network communication. The proposed work is carried out in three stages: RPL construction, Load Balancing, Load balancing optimization using Q-learning and finally load balancing optimization. Table 1 gives the notation summary used in the proposed design.

Table 1: *Notation Summary*

Symbol	Definition
DAG	Directed Acyclic Graph
DODAG	Destination Oriented Directed Acyclic Graph
ECc	Expected Child count
BR	Border Router
Exc	Existing Child Count
Pc Count	Parent Child Count
L	Learning Rate
Av	Assigned Value

5.1 Topology Construction in RPL

As shown in figure 6, standard RPL construction for DODAG3 or BR3 is explained here. The BR3 broadcasts DIO message to neighboring nodes 25, 28, 30, 29 and 11. Upon receiving the DIO, these nodes update their rank which is defined in the objective function. These nodes become one hop to the BR. They in return multicast DIO message to its neighbors 23, 24, 26, 27, 31, 33 and 32. These nodes update their rank with respect to their preferred parent and broadcast DIO message to its neighbors 34. This process continues till all nodes join the DODAG. DIO information includes RPLInstance ID, DODAGID, DODAG version number, Rank Number, Step, MinHopIncrease, etc. When a node receives DIO and the rank is higher than its own rank the node discards DIO message. Leaf Nodes O, K, L, M and N are waiting to join the DODAG3 in run time. Thus node 34 sends data to the BR3 through the preferred parent nodes 33, 30 using 3 hops. In the upward direction, the node has the address of preferred parent which routes the packets to the BR3. In the downward direction, BR3 uses DAO-ACK message to communicate to the leaf node in the DODAG.

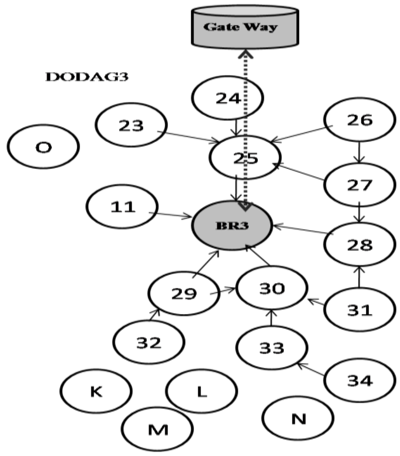

Figure 6: *Topology construction and Load Balancing for BR3*

5.2 Load Balanced RPL Construction

Once DODAG3 is constructed, load balancing of BR_3 using child count is initiated. BR_3 will execute Load Balancing Algorithm to assign new nodes to the preferred parents P(i) and P(j) for nodes O, K, L, M and N. Expected Child Count (ECc) is the total available nodes in the network. It calculated by eq. 5.

$$\text{Expected Child Count (ECc)} = Pc \text{ count (i)+Pc count (j) +LIM} \tag{5}$$

Applying the Load Balancing algorithm in figure 6, P(i), i.e., node 32 and P(j), i.e., node 33 have 2 existing children each. 4 common nodes (K, L, M and N) send DIO request to Preferred parent nodes 32 and 33. Using eq. 5, ECc (P(i)+P(j)+LIM) is calculated as 8. The load balancing algorithm suggests that if ECc is even then eq. (6) applies for P(i) and eq. (7) applies for P(j).

$$\text{Expected Child count (ECc) for P(i)} = \frac{ECc}{2} \tag{6}$$

$$\text{Expected Child count (ECc) for P(j)} = \frac{ECc}{2} \tag{7}$$

Applying the values in equation (6) & (7), we get ECc for P(i) is 4 and ECc for P(j) is 4. So for load balancing both the preferred parents P(i) and P(j) should have 4 nodes each.

If ECc is odd then eq. (8) applies for P(i) and eq. (9) applies for P(j).

$$\text{Expected Child count (ECc) for P(i)} = \frac{ECc+1}{2} \tag{8}$$

$$\text{Expected Child count (ECc) for P(j)} = \frac{ECc-1}{2} \tag{9}$$

In our proposed case, ECc is even. Hence eq. (6) & (7) are applied. Now, the BR_3 knows the value of ECc and Existing child count (EXc) of preferred parents P(i) and P(j). The BR_3 generates Assigned value (Av) of P(i) and P(j by equation (10) and (11).

The assigned value is the number of child nodes the preferred parents should accept. If there is DIO request more than the assigned value, the preferred parents should drop or neglect.

$$\text{Av of } P(i) = \text{ECc of } P(i) - \text{EXc of } P(i) \tag{10}$$

$$\text{Av of } P(j) = \text{ECc of } P(j) - \text{EXc of } P(j) \tag{11}$$

By assigning values in our proposed case, Assigned value (Av) of P(i) will be 4-2, i.e 2 nodes. And Assigned value (Av) of P(j) will be 4-2, i.e 2 nodes. The assigned value of P(i) is 4 and P(j) is also 4. Thus BR_3 iS successful in load balancing cing for any number of new nodes coming under common range. Figure 4 shows the Load Imbalance DAG (a) and Load Balanced DAG (b) using ch-LBRPL.

5.3 Load Balancing Optimization using Q-Learning

Once load balancing is done at the DODAG level, we apply q-learning technique for optimized path selection. Q-learning suits optimization of emergency response for many reasons. Emergency response environment is unknown and uncertain. Hence assigning node or link metrics for path optimization using q-learning is apt. In q-learning, the agent or node need to know only the immediate candidate parent nodes and hence traffic over head of the entire network is avoided. Once Q-tables are constructed then searching for nodes in the topology using n x n matrix becomes easy. In q-learning RPL, rewarding function can be both constrained metric and/or additive metric. Q-learning becomes more efficient when the number of steps to construct Q-table is more. This gives way for optimized path selection in scalable large networks. Q-learning algorithm is explained in table 2.

Table 2: *Q-Learning Algorithm*

Algorithm: Q-Learning Procedure	
1.	Set the Gamma parameter, and environment rewards in matrix R
2.	Initialize matrix Q to Zero
3.	For each episode
4.	Select a Random Initial State
5.	Do While the Goal State has not reached
6.	Select one among all possible actions for the current state
7.	Using the possible action consider going for next state
8.	Get maximum Q value for next based on all possible actions
9.	Compute Q(state, action) = R (state, action) + Gamma *
10.	Max[Q(next state, all actions)]
11.	Set the next state as the current state
12.	End Do
13.	End For

Load balanced RPL construction using Q-learning is shown in figure. As in our case, DODAG3 can be constructed by Reward table (R Matrix), i.e., figure 8. The Row represents the state and the columns represent the steps. BR is the goal state or sink node towards which all nodes send data. The nodes that are immediate neighbours and have link with BR are rewarded with the highest value, i.e., 100.

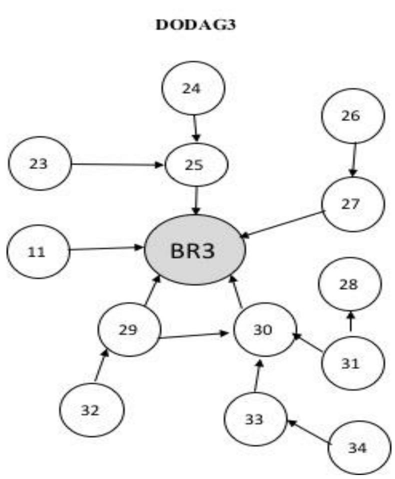

Figure 7: *The reward values in for DODAG3*

$$R = \begin{array}{c|cccccccccccccc}
 & BR & 11 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 \\
\hline
BR & 100 & 0 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\
11 & 100 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
23 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
24 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
25 & 100 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
26 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
27 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\
28 & 100 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & -1 & -1 \\
29 & 100 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & 0 & -1 & -1 \\
30 & 100 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & -1 \\
31 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & -1 & -1 & -1 \\
32 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & -1 & -1 \\
33 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & 0 \\
34 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0
\end{array}$$

Figure 8: *Reward table for DODAG 3*

The nodes with link is represented as "0" and nodes that carry no link carry value "-1". So the number of state and the number of steps every node takes to reach the goal is represented by n x n matrix. Where n represent the number of nodes taking part in the DODAG construction. For example, the node 25 has direct links with nodes BR, 23, 24, 26 and 27. But BR is the immediate neighbour to reach the goal state. Q-Learning is best optimization algorithm when the reward values are not assigned. The nodes in the network know the reward values of the nodes that are immediate neighbours and hence we can obtain optimality. The Q-learning update rule is given by eq. 8.

$$Q(a,i) = Q(a,i) + L(R(i) + Q(a_1,i) - Q(a,i)) \tag{8}$$

Where the following is true:

Q – table of Q-values

a - previous action

i - previous state

j - the new state resulting from the previous action

a_1 - the action that will produce the maximum Q-value

L – the learning rate (between 0 and 1)

R – the reward function

In Q-learning, during one step, the node analyzes data from the environment, chooses an action, and evaluates that action according to our equation. The agent will explore from state to state until it reaches the goal. Each episode consists of the agent moving from the initial state to the goal state. Each time the agent arrives at the goal state, the program goes to the next episode [13]. We will be setting the value of the learning parameter Gamma=0.8. The Gamma parameter has a range of 0 to 1 (0<=Gamma>1). If Gamma is closer to zero, the agent will tend to consider immediate rewards. If Gamma is closer to one, the agent will consider future rewards with greater weight, willing to delay reward. To use the matrix Q, the agent traces the sequence of states, from the initial state to goal state. Applying reward values for state 1 using the equation 2, we get

$$Q(1,2) = R(1,2) + 0.8 * Max[Q(2,3), Q(2,6)] = 0 + 0.8 * 0 = 0$$

$$Q(1,R) = R(1,R) + 0.8 * Max[Q(R,3), Q(R,4), Q(R,1)] = 100 + 0.8 * 0 = 100$$

If the agent learns through further episodes, it will finally reach convergence values in matrix Qn. This matrix can then be normalized (i.e., converted to percentage) by diving all non zero entries by the highest number. Once the matrix Q gets close enough to a state of convergence, we know our agent has learned the most optimal paths to the goal state. Tracing the best sequences of states is as simple as following the links with the highest values at each state.

6. Network Setup and Performance

The load balancing optimization using q-learning (LBO-QL) is carried out in Contiki OS Cooja Simulator in RPL protocol. Simulation parameters are given in Table 3. The simulated results are recorded for various performance metrics such as convergence time, packet delivery ratio, Control Traffic Overhead, energy consumption and path selection. The performance of the proposed load balancing optimization for emergency response (LBO-QL) is tested and the results are compared with standard objective functions in RPL (ofo & mrhof) and remaining energy in the nodes (RE). Convergence time is recorded from Cooja Mote Output log file. Energy consumption and path selection are recorded and analysed using collect view function in Cooja simulator. Control Traffic Overhead is analysed using 6LoWPAN analyser with Pcap in Wireshark.

Table 3: *Network simulation parameters*

Network Parameters	Values
Simulation Model	UDGM
No. Of Nodes	14
Area	120mx100m
Startup Delay	65s
Objective Functions	Of0, mrhof, RE, LBO-QL
Channel	Channel Check rate 8Hz and Radio Channel 26
TX and INT Range	Tx = 50m and INT = 55m
Simulation Time	600000ms

6.1 Convergence Time

RPL based IoT network consists of large number of nodes. The convergence time or network setup time is the time taken for the DODAG construction. Once DODAG is constructed the BR notifies that I am sink then nodes send data to the sink by multi hop preferred parents. Convergence time is an important factor for emergency response scenario. In this scenario, critical operations such as search and rescue operations are involved. Therefore, in a time bound emergency response system convergence time need to be low. The obtained results are shown in figure 9. In the results, objective function Of0 has the lowest convergence time. Since the number of nodes are only 14, many nodes are one or two hop neighbors to the BR. In the RE, the DODAG construction takes in to account the remaining energy of nodes and this more communication packets. So the energy consumption is more. The suggested method LBO-OF shows improved convergence time irrespective of the load balancing factor which contributes to added network life time.

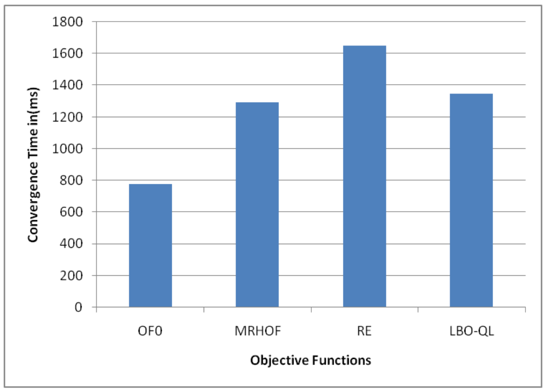

Figure 9: *Convergence Time*

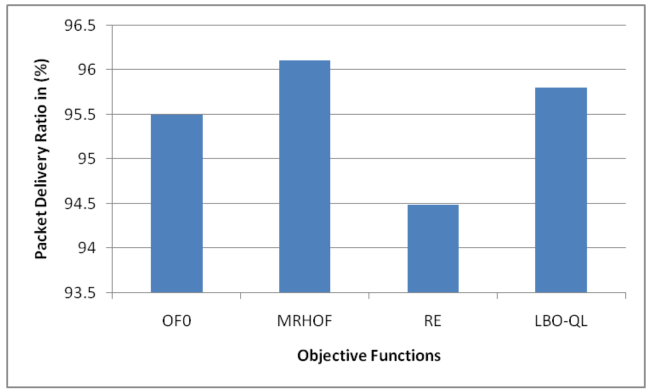

Figure 10: *Packet Delivery Ratio*

6.2 Packet Delivery Ratio

RPL based IoT networks are collection based networks. Packet delivery ratio is always an important parameter for network performance. The demand for high packet deliverer ratio is essential for emergency response system. Since the operations involve critical information, PDR need to be high comparing to other normal applications. The obtained results are shown in figure 10. The results suggest low packet delivery ratio for RE and high PDR for MRHOF objective function. Since MRHOF uses ETX metric for path selection, PDR is ensured. ETX metric calculates rank keeping the link quality in to account. On the other hand, LBO-OF has improved PDR than OF0 and little less than MRHOF. However, proposed method has other advantages such as improved network life time and balanced load. This will improve network efficiency in the long run.

6.3 Energy Consumption

IoT networks are basically resource constrained networks. The important resource constraint is energy. Often nodes in the emergency response scenario operate on battery. So network life time will depend on the optimized use of energy. RPL uses many techniques to save power of the nodes, i.e., duty cycle, sleep, etc. The obtained results are shown in figure 11.

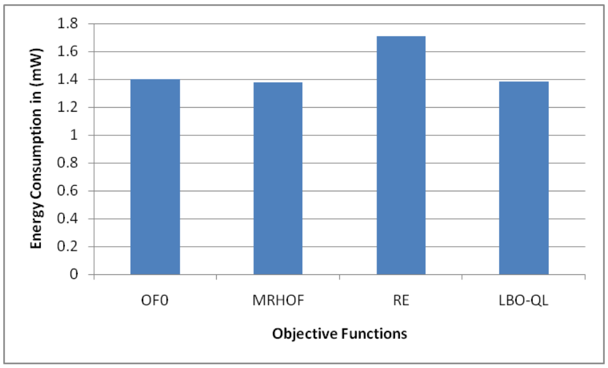

Figure 11: *Energy Consumption*

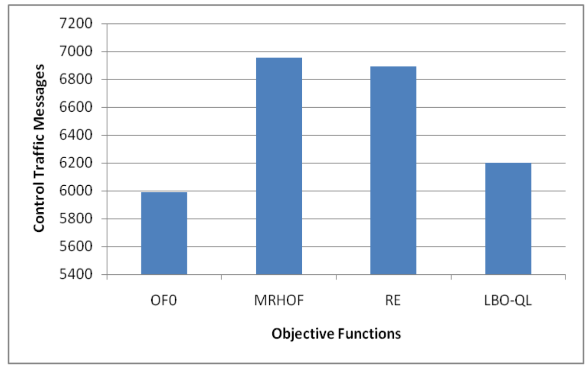

Figure 12: *Control Traffic Messages*

Figure 11 indicates that Energy consumption is more for Objective function RE and low for MRHOF and LBO-OF.

6.4 Control Traffic Overhead

RPL uses Internet Control Message Protocol (ICMPv6) for DODAG construction and maintenance. ICMP uses DIO, DIS, DAO messages for topology construction and repair mechanism. The data packets are UDP packets. At the initial set up DIO messages are sent by BR for nodes to calculate rank and join preferred parent. So DIO messages are more comparing to DIS, DAO and UDP. UDP packers are sent once the DODAG is constructed. The obtained results are shown in figure 12. From the results, it can be observed that Of0 has less control traffic as compared to MRHOF and RE. Hop count involves fewer control messages compare to the MRHOF and RE calculation. In comparison, LBO-OF has improved control traffic overhead. However, when the number of hops increases, the control overhead also increases. LBO-QL provides improved control traffic overhead, load balance and network stability. These network characteristics make the proposed method suit emergency response scenario.

7. Conclusion and Future Scope

The revolutionary IoT vision has given rise to many deployments. Rising demand for automation and information based control have popularized IoT networks. The low cost and low powered sensors and embedded devices possess the capability to provide improved and efficient emergency response scenarios. The proposed method load balancing optimization for RPL based emergency response using Q-learning (LBO-QL) is efficient in providing not only optimized network performance but also load balancing and stability to the entire network. The proposed method is successful in keeping the child count equal on all links. This gives better load distribution across nodes and links in the network. The proposed method also faces limitation of communication among multi DODAGs. As of now the BR can not communicate with another BR. A virtual framework needs to be created that establishes connection among the BRs. So there is a challenge to load balancing at Multi DODAG level. Mobility plays an important role in emergency response scenario. But sufficient research in mobility model is desired. When the nodes are mobile, Q-learning computation will increase the control traffic overhead and energy as the reconstruction of DODAG will be frequent. Hence, the proposed method is efficient for single BR. The emergency scenario will have many BRs and they also interact with other IPV_6 based networks. New optimization techniques for interoperability and load balancing in such environment in RPL need to be researched. The network requirement for emergency response is heterogeneous, so load balancing optimization for heterogeneous environment is a challenge. Multi agent q-learning and environment aware participatory q-learning are some solutions we plan to try in future.

References

Abu-Elkheir, M., Hassanein, H. S., & Oteafy, S. M. A., Enhancing emergency response systems through leveraging crowd sensing and heterogeneous data, 2016 International Wireless Communications and Mobile Computing Conferenc (IWCMC 2016), 2016, 188–193. https://doi.org/10.1109/IWCMC.2016.7577055

A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Commun. Surv. Tutorials, 2016, vol. 17, no. 4, pp. 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Cisco, Routing Protocol for LLN (RPL) Configuration Guide, Cisco IOS Release 15M&T, 2015

E. Borgia, The internet of things vision: Key features, applications and open issues, Comput. Commun., 2016, vol. 54, pp. 1–31. https://doi.org/10.1016/j.comcom.2016.04.024 https://doi.org/10.1016/j.comcom.2014.09.008

H.-S. Kim, H. Kim, J. Paek, and S. Bahk, Load Balancing under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks, IEEE Trans. Mob. Comput., 2016, vol. 1233, pp. 1-10 https://doi.org/10.1109/TCAD.2016.2597213

Han, D., & Gnawali, O., Performance of RPL under wireless interference. IEEE ommunications Magazine, 2012, Vol. 51(12), 137–143. https://doi.org/10.1109/MCOM.2013.6685769

J. Guo et al., G. Bhatti, P. Orlik and K. Parsons, Load Balanced Routing for Low Power and Lossy Networks, 2015.

J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Futur. Gener. Comput. Syst., 2015, vol. 29, no. 7, 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

Marwa Mamdough et al., RPL Load balancing via minimum degree spanning tree, IEEE transaction, 2016

Minkeun Ha, Kiwoong Kwon, Daeyoung Kim, Peng-Yong Kong , Dynamic and Distributed Load Balancing Scheme in Multi- gateway based 6LoWPAN, IEEE International Conference on Green Computing, 2015

O. Iova, F. Theoleyre, and T. Noel, Exploiting multiple parents in RPL to improve both the network lifetime and its stability, IEEE Int. Conf. Commun., 2015, vol. 2015–September, pp. 610–616. https://doi.org/10.1109/ICC.2015.7248389

P. H. Gomes et al., Reliability through Time-Slotted Channel Hopping and Flooding-based Routing, 2015

R. Jadhav et al, Optimization of Parent node selection RPL based Natworks", ROLL-WG INTERNET DRAFT, 2017, pp. 1–11.

W. Almobaideen, R. Krayshan, M. Allan, and M. Saadeh, Internet of Things: Geographical Routing based on healthcare centers vicinity for mobile smart tourism destination, Technol. Forecast. Soc. Change, 2016, April, pp. 0–1.

X.Liu, J. Guo, G. Bhatti, P. Orlik and K. Parsons, Load Balanced Routing for Low Power and Lossy Networks, 2016