

**EVALUATING TRANSMITTING QUALITY AND LOAD BALANCING USING DODAG GRAPH**

**<sup>1</sup>Kumudini. S, <sup>2</sup>Dr. M. P. Vani**

<sup>1</sup>Research Scholar, Score, Vellore Institute of Technology,

<sup>2</sup>Associate Professor senior, Score, Vellore Institute of Technology, Vellore, India.

kumudinisuresh2011@gmail.com, mpvani@vit.ac.in

**Abstract**

In the realm of Internet of Things (IoT) and wireless sensor networks (WSNs), efficient data transmission and balanced load distribution are critical for ensuring network reliability and longevity. This study focuses on the Directed Acyclic Graph (DAG) Optimized for Data Aggregation and Dissemination (DODAG) structure, employed within the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). The primary objectives of this research are to evaluate the transmitting quality and to optimize load balancing across the network using the DODAG graph.

**Keywords:** DODAG, RPL, IoT, Wireless Sensor Networks, Load Balancing, Transmitting Quality, Network Performance, Energy Consumption.

**I. Introduction**

Node status plays a significant role in routing the packets to the destination node. In the case of a network medium, each point-to-point link is technically assumed as a hop, and the count is considered only to those devices which are in between the two points of connection. Hopcount considers only those devices which perform routing, non-routing devices are not considered as a hop count. One of the crucial tasks in network is to define the importance of nodes relation with other elements in the network. The importance of node can be defined in various ways, one such way is to rank nodes according to the degree of connectivity.

In this project, we aim to enhance the packet transmission quality and load balancing in Wireless Sensor Networks (WSNs) using the Destination Oriented Directed Acyclic Graph (DODAG) approach. To achieve this, we use Mininet, a popular network emulator, to create and evaluate the DODAG topology. It provides a detailed description of the simulation setup, execution, and analysis of the results using various visualizations.

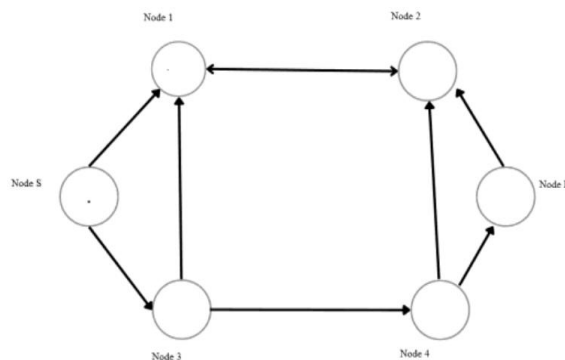


Fig 1:- Basic Network Topology

Network are naturally modelled as graphs. A broad application of importance of node (i) is its degree of connectivity by counting the number of links to (ii) we can understand the importance of nodes that point to (iii). In the figure a link by node 4 is more important than a link from node 1.

Metric called as ETX (Expected Transmission count) is proposed for routing reliability and how the energy is used to calculate energy utilized in packet transmission to derive the residual energy. Distribution of traffic across multiple servers is a core networking solution for load balancing. Application availability, responsiveness and server overload can be prevented and improved by Load Balancers. The loadbalancer resides in between the customer devices and servers, receiving and then distributing incoming requests to any available server capable of fulfilling them. To determine how requests are distributed across a server the load balancer follows an algorithm. Numerous options are available in this regard from a very basic option to a compound one.

## II. Solutes

Qualitative mode of transmitting the packets has 3 solutes: Status Determination, Total Aphasia, and Track Facilitator.

### A. Status Determination

The circumstance of wireless links will differ with things it must endure. The ratio of packet delivery will reduce if it involves unreliable links/nodes in the list. Therefore, it is necessary to choose a reliable link as the next forwarding node for transmitting the data in the network. A well-known metric to evaluate the quality of the nodes and to identify the reliable track to determine the status of the node is the expected transmission count ETX, which does not reflect the suitability of a node ,it is not the right strategy,therefore to increase the delivery ratio of the packet, link diversity is considered.In the initial state the destination node is set to

(0)Constant and source node is set to Constant

(1) Throughout the process.

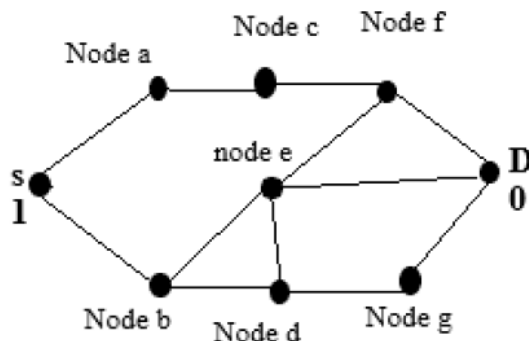


Fig 2. Status determination

Which makes it easy to trace the node which is not the destination, where the node status is less than its neighbour node for forwarding

The status of each node is identified by broadcasting Request to Calculate Status. The nodes are deployed with the current status of it. The non-source and non-destination node upon receiving the RTCS performs the algorithm to determine the status.

**Algorithm 1: Status Determination Algorithm**

Step 1: Read the neighbours of node (ng) and their respective IDs

Step 2: If neighbouring node is not source then initialize sum to be zero else jump to step 7

Step 3: for every  $i^{th}$  element to be equal to 1 repeat the step 3 for N number of neighbouring nodes.

Step 4: Calculate the ETX, ETX of the neighbouring node (ng) with the identifier of the  $i^{th}$  element.

To compute we need to divide 1 by 1 minus the probability of the forward node for each neighbours of node with the identifier of the  $i^{th}$  element multiplied by 1 minus the probability of reverse node with the neighbouring nodes identifying the  $i^{th}$  element.

Step 5: The probability of success of neighbouring nodes for  $i^{th}$  element can be computed by dividing 1 by the expected transmission count result from step 3.

Step 6: Determine the total nodes by summing up the total from step 4 multiplied by the probability of success for every identifier of  $i^{th}$  element

Step 7: end the loop

Step 8: we can calculate the probability of success of neighbouring node by the sum divided by the number of neighbouring nodes.

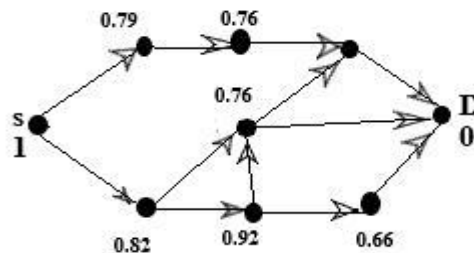
Step 9: Status of the node can be determined by subtracting constant 1 by the resultant derived in step7.

Step 10: if the difference between old status of the node and new status of the node is greater than predetermined threshold replace the current node with the new node.

Step 11: RTCS packet to be deployed if greater

Step 12: end of step 9

Step 13: end of step 2



DODAG table, calculate the cost of track for the DODAG information object sender and then decides whether to join DODAG table or not. Once a neighbour joins the DODAG table, the route is established and that root becomes a DODAG parent of the node. Once route is established the node calculates its status in the DODAG and replies with a DAO Message to its parent to inform its participation. Nodes which have not received any DIO messages and have not joined any DODAG table can request DODAG information by sending DIS messages to its neighbours periodically. This process is repeated by the neighbouring nodes until all have joined the DODAG table.

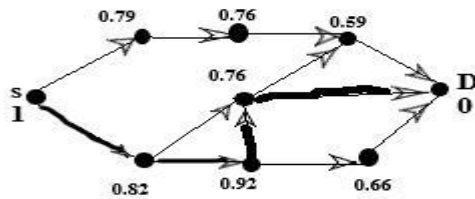


Fig 3. Result of the status determination

**4. Flow of The Dodag Formation Process**

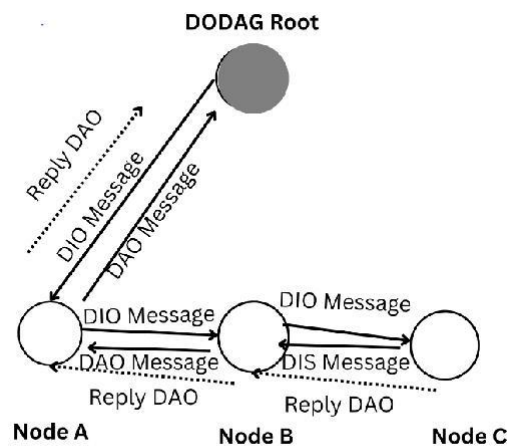


Figure 4. Flow diagram of the formation process for DODAG

Step 1: DIO messages with DODAG information are sent from the DODAG root; DIO messages are received by Node A and joins the DODAG root and Broadcasts a DAO message which has prefix information to the root.

Step 2: Next the DIO Message is sent by Node A which has the DODAG information and Node B which is in Node A’s transmission Scope receives the message, joins the DODAG, and broadcast with a DAO Message to Node A.

Step 3: DIS request message is received by node B from Node C but does not respond because it has not part of any DODAG table.

Step 4: DIO Message is sent from Node B to Node C inviting it to join the DODAG table once Node B joins the DODAG table

Step 5: Node C has joined the DODAG table now the Node C replies with a DAO Message to Node B.

Step 6: upon receiving the message Node B will collect all information and then send the DAO Message to its preferred parent Node A.

The DODAG root upon acquiring all nodes prefix information about their DAO messages to form a downward route.

Parent node is selected based on the Node which is nearer to the DODAG root, status of a node rises in the downward direction of the DODAG and reduces in the upward direction. The status of a parent node should be lesser than its nodes of its children, the algorithm prevents routing loops from occurring in the network. Parent candidate set is updated upon a node receiving a DIO message and a parent is chosen based on the status values of the nodes in this set later its status value is calculated. Upon finding if the status is more than its parents (i.e., the node is on a downward route from the parents), the DIO message is updated with the new status information. The DIO Message is passed to the neighboring nodes finally. Outcome by the usage of DIO messages every node in the network builds its upward path. With the help of DODAG, DAO messages tranenable a node to transmit its target information upwards, allowing the construction of downward path between the DODAG root and their connective nodes. DIO and DAO messages are used for the uncovering of upper and lower paths, respectively.

### Simulation Setup and Execution

Tools and Environment:

- Emulator: Mininet
- Programming Language: Python
- Operating System: Ubuntu 20.04 LTS
- Visualization Libraries: NetworkX, Matplotlib
- VM Environment: VirtualBox/VMware

Table 1: Simulation Parameters

Parameters	Value
Network Size	100 x 100 meters

Number of Nodes	50 nodes
Communication Range of Nodes	100 meters
Packet Size	512 bytes
Initial Energy of Nodes	0.5 Joules
Bit Error Rate	0.01 or (1%)
Energy Consumption of Radio Electronics	50nJ/bit
Energy Consumption of Power Amplifier	100 pJ/bit/m <sup>2</sup>

### Step-by-Step Implementation:

#### 1. Installing Mininet:

We install Mininet on the Ubuntu VM to create and simulate network topologies.

#### 2. Setting Up the Network Topology:

We create a network topology consisting of a controller, a switch, and two hosts. The topology is configured using a Python script that sets up the network, adds nodes and links, and defines the connections between them.

#### 3. Running the Mininet Script:

We execute the Python script to start the network simulation, which includes initiating the network, performing a ping test between the hosts, and starting the Mininet CLI for further interaction.

### Result Collection and Analysis

We collect data on packet transmission, residual energy, and load balancing efficiency during the simulation. The data is then processed and visualized using Python libraries.

### Visualization Setup

To analyze the results of the Mininet simulation, we create visualizations using Python libraries such as NetworkX and Matplotlib.

### Network Topology Visualization

This visualization illustrates the network topology consisting of a root node, intermediary nodes, and leaf nodes. It shows how nodes are connected in a DODAG structure.

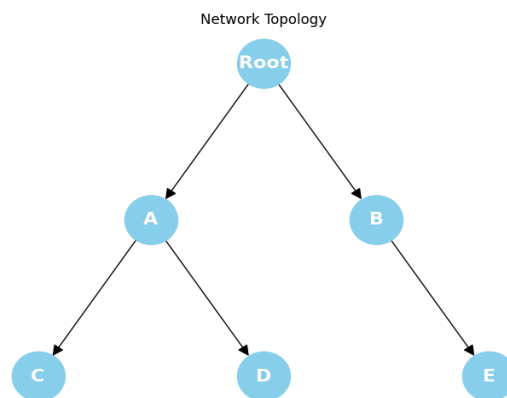


Figure 5: Network Topology Visualization

The network topology diagram represents the hierarchical structure of nodes in the DODAG. The root node is connected to intermediary nodes, which in turn are connected to leaf nodes. This structure helps in efficient data routing and load balancing.

### Ping Test Results Visualization

This bar chart shows the round-trip time (RTT) for ping tests between various nodes in the network. It helps in understanding the latency between nodes.

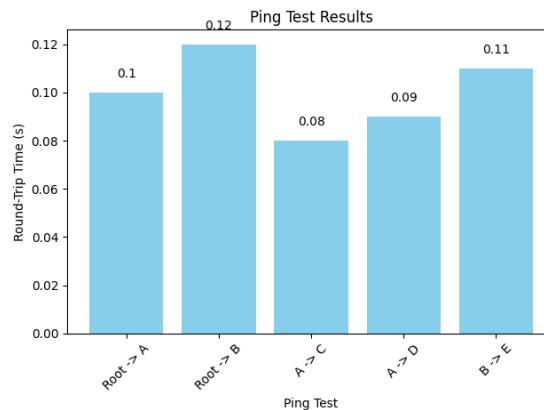


Figure 6: Ping Test Result

The ping test results indicate the time it takes for a packet to travel from the source node to the destination node and back. Lower RTT values suggest better network performance. The chart provides a clear comparison of the RTT for different node pairs.

### Residual Energy Analysis Visualization

This bar chart displays the initial energy, final energy, and energy consumed by each node in the network during the simulation. It helps in analyzing the energy efficiency of the nodes.

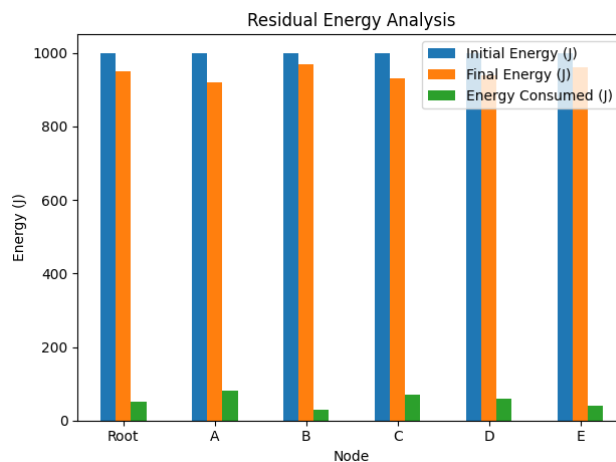


Figure 7: Residual Energy Analysis

The residual energy analysis shows how much energy each node consumes during the simulation. By comparing the initial and final energy levels, we can determine the efficiency of the network in terms of energy usage. Nodes with higher energy consumption may need optimization for better performance.

### Load Balancing Efficiency Visualization

This bar chart illustrates the load percentage distributed across different nodes in the network. It helps in evaluating the load balancing efficiency of the network.

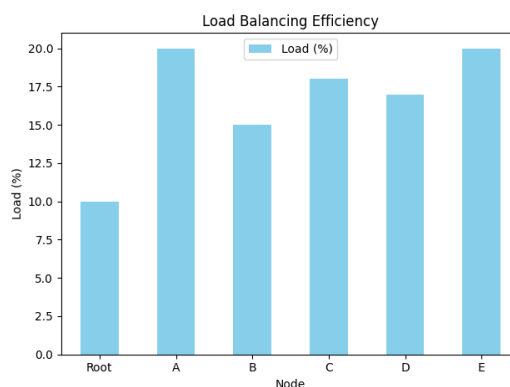


Figure 8: Load Balancing Efficiency

The load balancing efficiency chart shows how the network load is distributed among different nodes. A balanced load distribution ensures that no single node is overwhelmed, leading to improved network performance and reliability. The chart helps in identifying any imbalances that need to be addressed.

### III. Conclusion

This study successfully implements and evaluates the DODAG-based load balancing technique using Mininet. The results highlight the potential of DODAG in improving packet transmission quality and load balancing in WSNs. Future work will focus on optimizing the algorithm for larger networks and varying traffic conditions. The visualizations provide a clear insight into the network topology, latency, energy consumption, and load distribution across the network.

### IV. Future Scope of Work

This study enhance the algorithm to handle larger network sizes efficiently. Investigate scalability issues and propose solutions for maintaining performance in extensive networks. Adapt the load balancing technique to perform optimally with varying network loads and traffic patterns. Transition from simulation to real-world implementation and testing. Explore the applicability of the DODAG-based load balancing technique in various interdisciplinary fields such as smart agriculture, healthcare, and industrial automation. Incorporate encryption and secure communication protocols to protect against potential attacks. These future directions aim to enhance the performance, reliability, and applicability of the proposed DODAG-based load balancing technique in wireless sensor networks and IoT applications.

### References

- [1]. Sheng Shih Wang, Hsing-Tsai Liu, "EQ-RPL: An Energy-Efficient and Quality-Aware Routing Protocol for IOT-BASED Low -Power and Loosy Networks" Journal of internet technology Vol23 N0.3 May2022
- [2]. Daniel Kifetw, Prasanth Kumar, "A Residual Energy Monitoring in wireless sensor network" Sept2011.
- [3]. Kiran Kumar Y, Bendigeri, Jayashree D. Mallapur, "Multiple node placement strategy for Efficient routing in WSN", 101-112, Aug2015.
- [4]. Hongliang Tian, Zhihong Qian, Xue Wang and Xiao Liang "QOI Aware DODAG CONSTRUCTION in RPL-Based Event Detection Wireless sensor Networks" Journal of Sensors,2017.

- [5]. S. Rani R. Talwar, J. Malhotra, S. H. Ahmed, M. Sarkar and H. Song, "A Novel Scheme for an energy efficient internet of things based on wireless sensor networks", vol.15 no.11 PP-28603-2862,2015
- [6]. Rohit Kumar, Er. Sandeep Kad "Energy Efficient Protocols for Wireless Sensor Networks -A Review" Article ·International Journal of Engineering Trends and Applications (IJETA) – Volume 4 Issue 3, May-Jun 2017.
- [7]. M. Mamdouh, K. Elsayed, A. Khattab, RPL Load Balancing via Minimum Degree Spanning Tree, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, 2016, pp. 1-8.
- [8]. H. S. Kim, J. Paek, S. Bahk, QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications, IEEE International Conference on Sensing, Communication, and Networking (SECON), Seattle, WA, USA, 2015, pp. 265-273.
- [9]. M. Mahyoub, A. S. Hasan Mahmoud, M. Abu-Amara, T. R. Sheltami, An Efficient RPL-based Mechanism for Node-to-Node Communications in IoT, IEEE Internet of Things Journal, Vol. 8, No. 9, pp. 7152-7169, May, 2021.
- [10]. J. Tripathi, J. C. de Oliveira and J. Vasseur, "A performance evaluation study of RPL: Routing protocol for low power and lossy networks", *Conf. on Information Sciences and Systems CISS*, Mar. 2011.
- [11]. Qureshi, T. N., et al. "On performance evaluation of variants of Global energy balance in wsns."arXiv preprint arXiv:1208.2401 (2012).
- [12]. Qingyu Shi Fang Wang Dan Feng Weibin Xie "Adaptive load balancing based on accurate congestion feedback for asymmetric topologies", vol 157, 5 July 2019, pg133-145
- [13]. S. Jagir Hussain, M. Roopa "BE-RPL: Balanced-load and Energy-efficient RPL" vol-45no.1, pp. 785–801, 2023.
- [14]. A. Seyfollahi and A. Ghaffari, "A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks," *Computer Networks*, vol. 179, Article ID 107368, 2020.
- [15]. D. Pancaroglu and S. Sen, "Load balancing for RPL-based internet of things: A review," *Ad Hoc Networks*, vol. 116, Article ID 102491, 2021.

- [16]. T Winter, P Thubert, A Brandt, J Hui, R Kelsey, P Levis... - **2012** E. T. F. (IETF), Rfc 6550, rpl: Ipv6 routing protocol for low-power and lossy networks, 2012. [Online]. Available: <https://tools.ietf.org/html/rfc6550>
- [17]. W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An Application-specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670, Oct. 2002.