Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 1 (2025)

Geographic Greedy Perimeter and Energy-Aware Routing (Ggpear) in Wireless Sensor Networks

[1]Mrs. J. Gnana Mano Sheebha, [2]Dr. D. Maheswari,

[1]Research Scholar, RVS College of Arts and Science(Autonomous),Sulur, Bharathiar University, Coimbatore,Tamil Nadu, India

sheebhabovas@gmail.com,

[2]Head & Research Coordinator, School of Computer Studies-PG, RVS College of Arts and Science (Autonomous), Sulur, Bharathiar University, Tamil Nadu, India

maheswari@rvsgroup.com

Article History:

Received: 12-11-2024

Revised: 15-12-2024

Accepted: 01-01-2025

Abstract: A novel routing method based on geographic routing and energy aware metrics named Geographic Greedy Perimeter and Energy Aware Routing (GGPEAR) is proposed in this thesis to improve the efficiency and the longevity of Wireless Sensor Networks (WSNs). Given the geographic information: node positions, greedy perimeter routing is used to construct an optimal route to the destination. However, when greedy forwarding is impossible, GGPEAR reverts to perimeter routing, guaranteeing ongoing packet delivery. GGPEAR is an energy aware protocol that aims to balance the network energy consumption by selecting high energy nodes first for participation and prevents the nodes from energy starvation before time. This method enhances WSN performance greatly since each node reduces wastage in energy and increases communication efficiency. The combination of greedy forwarding with perimeter routing allows GGPEAR to manage its energy consumption and increase network throughput in general. Unlike other schemes, it presents a significant scalability advantage with minimal increase in energy consumption, end to end delay, and overhead as the number of the nodes or packets grows. A higher throughput, lower energy consumption, a lower end to end delay and a better packet delivery ratio (PDR) can be achieved with the proposed protocol compared with the existing ones as EEM-CRP and EE-TLT. GGPEAR exhibits lower packet loss ratios and is more reliable sensor size, it is suitable for large sensor networks. Network lifetime and resource conservation are critical in time sensitive and energy constrained applications for which the efficiency of the protocol is particularly important. GGPEAR optimizes routing and energy consumption to lengthen the operational lifetime of WSNs and guarantee high communication quality over the network.

Keywords: WSNs, GGPEAR, EBRA, GEAR, Routing.

1. Introduction

Due to their efficiency in sensing and transmitting data, Wireless Sensor Networks (WSNs) are important in many domains. The area of routing in WSNs is an active area of research and many algorithms have been proposed to improve energy efficiency, network lifetime and performance. For energy efficient communication in challenging environments, EBRA: Energy balanced routing algorithm for underwater wireless sensor networks (WSNs) is proposed by Ahmed et al. (2018). Next, Amgoth and Jana (2015) suggested an energy-aware routing algorithm that optimizes energy consumption in traditional WSNs to increase network lifetime. Considering the geographical part, Aregawi and Haile (2018) added geographic energy aware routing protocols and presented solutions to extend network lifetime.

To improve routing performance in WSNs, Bairagi et al. (2024) develop a recursive geographic forwarding mechanism that uses energy efficient protocols. Bousnina et al. (2017) explored a greedy approach for resource allocation in virtual sensor networks that focuses on optimal usage of network resources. In Chaurasia et al. (2023), a meta heuristics cluster based routing protocol called EEM-CRP, was proposed for increasing the energy efficiency of WSNs. Guleria et al. (2019) propose energy-aware location based routing protocol that optimize routes based on energy of the nodes combined with nodes geographical positions.

Hadi (2017) analyzed energy aware geographic routing for WSNs by analyzing routing protocols which provided insight into balancing the amount of energy being used over the nodes. Hadir et al. (2017) evaluate the performance of DV-Hop localization algorithm on geographical routing to overcome localization challenges. For energy harvesting WSNs, Hao et al. (2022) put forward a greedy strategy based routing algorithm that uses harvested energy to prolong network operation.

In the case of continuous cover range sensors, perimeter coverage scheduling was studied by Hung and Lui (2010), where they formulated the problem to schedule sensor operations for effective coverage. An energy efficient model for perimeter surveillance was proposed by Imoize et al. (2021) to improve the performance of WSNs in security applications. In Jibreel et al. (2022) heterogeneous gateway based multi hop routing protocol was introduced to improve energy aware in WSNs. A comprehensive survey of energy efficient protocols using radio scheduling to conserve energy in WSNs was done by Jones (2020). Finally, Karkazis et al. (2012) investigated geographical routing methodologies giving useful insights into according routing in WSNs using node location as a basis.

Due to the complexity of WSNs geographic routing protocols that forward packets based on location information have been developed. Greedy Routing, Perimeter Routing and Energy Aware Routing are one among these protocols which largely contribute in reducing energy consumption and for effective communication in constraint environments. For instance, Nabavi et al. (2021) proposed a multi objective greedy method including some metrics like maximizing network longevity and maximizing energy efficiency in WSNs' routing decisions for energy efficiency in WSNs (Nabavi et al., 2021). In addition, exhaustive survey articles are presented regarding the routing strategies that consider the energy consumption under their perspectives, such as the work by Nakas et al. (2020), which comprises numerous methods devised to reduce the energy consumption over WSNs (Nakas et al., 2020).

However, other studies have concentrated on specific difficulties, like the existence of complicated holes in the network that have a sizable effect on routing execution. To solve the issue of routing in the presence of such holes with energy efficiency, while providing reliable data transmission, Nguyen et al. (2021) presented a solution (Nguyen et al., 2021). Besides, geographic routing protocols such as described by Pandith et al. (2022) have been looked into since they route the data using the geographic location of the node instead of manipulating complex routing table information and improving the scalability (Pandith et al., 2022). Furthermore, Shu et al. (2010) propose TPGF (Two-Phase Geographic Forwarding), a geographic routing protocol for wireless multimedia sensor networks in which the sensor network nodes have the ability to transmit multimedia data (Shu et al., 2010). However, these new forms of geographic routing are essential to jump start improvements of data forwarding in WSNs, particularly in multimedia applications where bandwidth and energy use are of the greatest concern.

In this overview, a range of techniques identified by researchers to address routing challenges in WSNs, striking a balance among energy efficiency, performance, and operational lifetime is presented.

2. Background Study

Chaurasia, S. et al. (2023) In this paper, EEM-CRP protocol is proposed which combines the meta-heuristic clustering techniques with saving the energy in WSNs. The work of this paper focuses on making use of optimal clusters, which reduce the energy consumption while in data aggregation and transmission, so as to prolong the network lifetime. Dynamic cluster head selection and balanced energy distribution across nodes are integrated in their methodology.

Khan, T., et al. (2016) A clustering depth based routing protocol is proposed by the authors for underwater WSNs. Using depth information, this protocol trades off energy savings by optimizing clustering and routing process. The object of the study is the communications underwater, specifically high latency and energy consumption and the ways to improve the efficiency of sensor networks in submerged environments.

Khedr, A. M. et al. (2009) these authors present a perimeter discovery algorithm for WSNs with the objective of finding and exploiting the perimeter of a sensor covered region. The protocol uses optimized routing around these perimeters to conserve energy and eliminate redundant data transmission ensuring efficient network operation even in complex scenarios.

Kheirabadi M. T. and Mohamad M. M., (2013) this paper do a survey of greedy routing techniques in underwater acoustic sensor networks (UASNs). Under various unique constraints of underwater environments, it identifies the advantages and limitations of different routing strategies. In addition, the authors offer comparative analysis of the design with a focus on energy efficiency and reliability that are critical in underwater network design.

Kumar, A., et al. (2017) In WSNs, this survey discusses the use of geographical information for optimizing routing decisions in location based routing protocols. The authors classify existing protocols according to their goals such as energy efficiency, scalability, and robustness, shedding light on how to construct powerful location aware systems.

Kumar, S., et al. (2020) based on a novel energy aware routing protocol using adaptive strategies in balancing energy consumption across nodes, the authors analyze it. The results show the improved network longevity and data transmission reliability obtained by incorporating energy thresholds and predictive modeling.

Lin, D. et al, (2020) these authors summarizes energy efficient approaches for static WSNs, in terms of deployments, power management and routing. These authors divide techniques into algorithmic and hardware-based approach and review some of the recent advances that have significantly reduced energy consumed in a static network.

Liu, J., et al (2017) these authors specifically characterizes data deliverability of greedy routing protocols for WSNs. The authors provide a mathematical framework that predicts where and when greedy routing will succeed or fail when the network topology, node density and mobility vary.

Mayasala, P. & Krishna, S. M. (2022) A Correlation Distance-Based Greedy Perimeter Stateless Routing Algorithm, which uses correlation distance metrics with perimeter routing in order to increase path reliability and energy efficiency, is proposed by the authors. This approach decreases packet drops and allows a more profitable usage of network resources.

Meng, X., et al. (2016) A grid based reliable routing protocol for WSN is proposed, which relies on randomly distributed clusters to improve the fault tolerance of the network and to reduce energy overhead. The protocol addresses this problem by dividing the network into grid cells, and dynamically adopting cluster roles to improve data delivery rates and energy efficiency.

Mittal, M., & Iwendi, C. (2019) these authors reviews energy aware routing protocols for WSNs with a view of reducing energy wastage while routing the data. The authors point out key challenges to tackle such as load balancing and scalability in networks and present a thorough account of the solutions to these problems.

Tan, N. D., & Nguyen, V. H. (2023) In order to increase the number of energy efficiency in WSNs, the EE-TLT protocol uses a two-level tree based clustering approach. The protocols exploit reduction in energy consumption and extended the network lifetime by organizing nodes into hierarchical clusters and optimizing intra and inter cluster communication.

Table 1: Comparison table on Routing Algorithm in WSN

Reference	**Routing Algorithm**	**Key Objective**	**Network Structure**	**Additional Features**
Wang et al. (2012)	Security Routing based on Geography and Energy Awareness	Secure and energy-efficient routing in WSNs	Geographic-based routing in WSNs	Security focus with energy and geographic awareness
Vamsi & Kant (2014)	Trusted Greedy Perimeter Stateless Routing (TGPSR)	Improving routing reliability and energy efficiency	Flat network with perimeter routing	Trust mechanism and perimeter routing for energy saving

Yang et al. (2015)	Hole Detection and Shape-Free Representation with Double Landmarks	Efficient routing in the presence of network holes	WSN with geographic and hole detection mechanisms	Addresses hole detection, shape-free representation for routing
Vahabi et al. (2019)	Geographic and Clustering Routing	Energy conservation through node clustering and geographic forwarding	Clustered WSN with node groups	Energy saving with pair node groups for routing
Tang & Lu (2020)	DS Evidence Theory-based Energy Balanced Routing	Enhancing network lifetime through energy balancing	WSN-assisted IoT	Focus on lifetime extension, evidence theory-based approach
Venkatasubramanian (2021)	Correlation Distance-Based Greedy Perimeter Stateless Routing	Reduce energy consumption and enhance routing reliability	Flat WSN with perimeter routing	Correlation-based energy efficiency and perimeter routing
Yan & Qi (2021)	CARA: Congestion-Aware Routing Algorithm	Minimize congestion and energy consumption	Flat WSN	Focus on congestion management along with energy awareness
Zagrouba & Kardi (2021)	Comparative Study of Energy-Efficient Routing Techniques	Comparison of different energy-efficient protocols	Various WSN structures (Flat, Clustered, etc.)	Comprehensive comparison across various WSN routing strategies
Yue (2024)	EEUC-based Inter-Aircraft Ultraviolet Communication Algorithm	Optimizes energy consumption in UAVs	Inter-aircraft communication in drone swarms	Energy optimization in drone networks

2.1 Problem identification

The problem that needs to be solved in energy efficient routing for Wireless Sensor Networks (WSNs) is to balance the energy consumption and the demands of reliable and efficient data transmission. Since sensor nodes are resource constrained, routing protocols need to be optimized to prolong the network lifetime while satisfying security, congestion, and problem of holes or irregular topologies remain an important issue. In addition, determining the most suitable routing strategy for dealing with different network conditions and application specific requirements complicates the design and deployment of these protocols.

3. Proposed Methodology

Create a Geographic Greedy Perimeter and Energy-Aware Routing (GGPEAR) an effective routing protocol. This protocol seeks to integrate geographic information and sensor node energy levels into the routing choices thereby extending the lifetime of wireless sensor networks. Combining geographic routing which uses the actual positions of sensor nodes for routing with greedy perimeter routing where packets are sent to the closest neighbor to the destination GGPEAR uses The protocol moves to perimeter routing to locate another route in situations when greedy forwarding is not feasible. Furthermore included are energy-aware metrics to guarantee that nodes with more energy reserves are given top priority in the routing process, therefore balancing the energy consumption throughout the network and preventing early depletion of any one node.

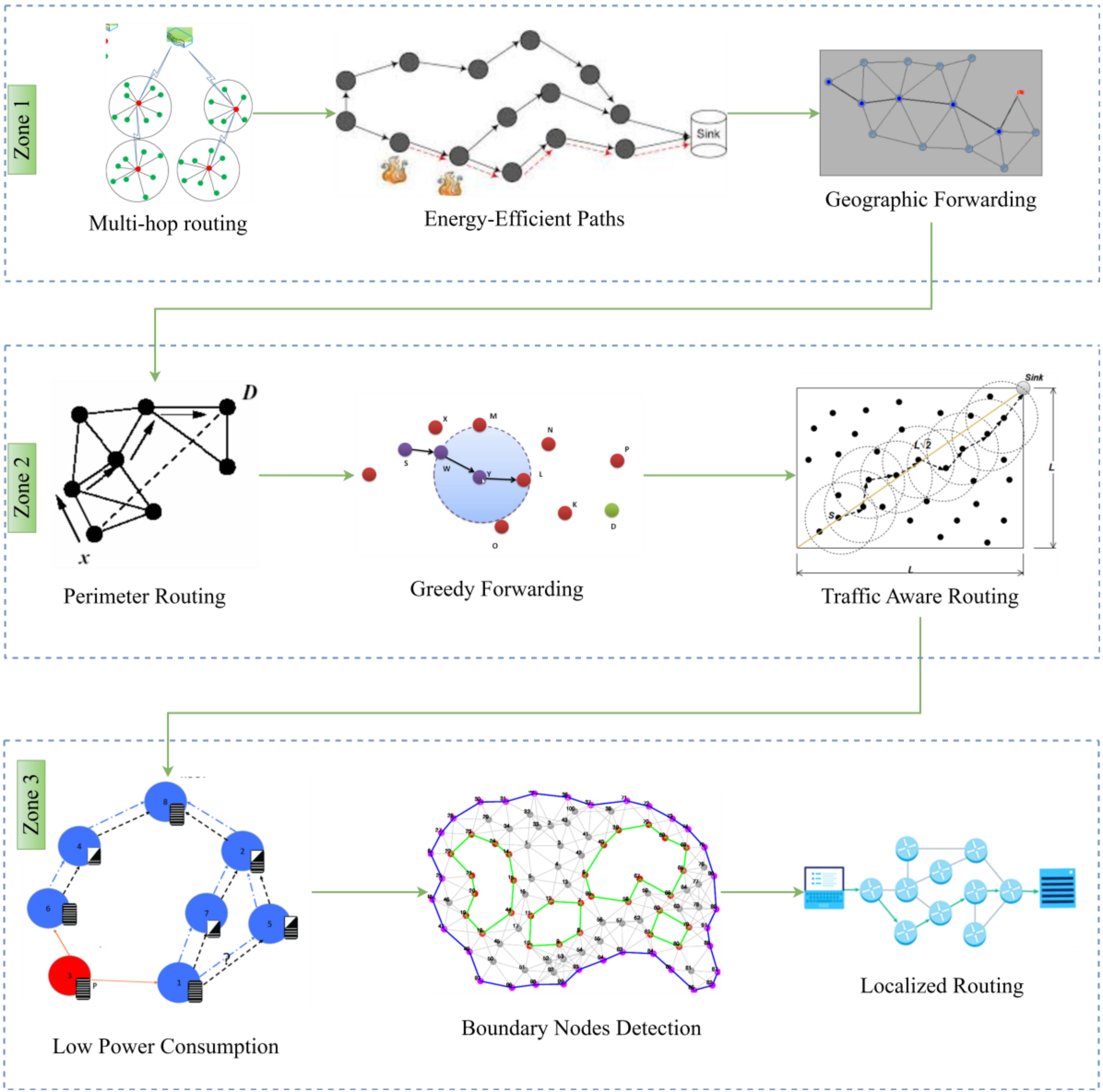

Figure 1: Proposed Method Architecture

This Figure 1 outlines a hierarchical routing approach in Wireless Sensor Networks (WSNs), divided into three zones to optimize energy efficiency and data delivery: Zone 1 utilizes multipath routing, optimal energy distribution and geographic forwarding techniques to ensure that data gets to the sink in the shortest time possible while at the same time taking the least amount of energy. Zone 2 uses more sophisticated techniques such as perimeter routing, greedy forwarding, and traffic-aware routing to move through intricate topologies while going around anything that interfere with its ability to deliver packets in an effective manner. Zone 3 emphasizes power consumption reduction with the help of boundary node detection for optimizing routing within the zones and localized routing for minimizing unessential communication and energy consumption enhancing thus network lifetime. The approach uses geographical and energy measurements to control the responsive and optimal routing.

3.1 Greedy Routing

In this paper, design a greedy routing protocol for Wireless Sensor Networks (WSNs) that works by selecting the neighbor of the transmitting node that is closest to the destination, in order to minimize the distance to the destination in each step. However, this method is a "greedy" method as it chooses the node that moves it the closest to the destination without any long–term planning. For large, dynamic networks, the algorithm operates without requiring global knowledge of the network topology. Greedy routing is simple and incurs low overhead, but it has the problem of local maxima: nodes not be able to make progress towards the destination as there are no forward neighbors. To solve this, some alternatives of greedy routing have been proposed that employ a backup routing strategy, e.g., perimeter routing, or a recovery mechanism. Overall, greedy routing is scalable and energy efficient, though it can result in unsuccessful routing due to sparsity or by virtue of blockage.

$D_{AB} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ -------- (1)

The equation (1) is Euclidean distance between two points. $A(x_1 - x_2)$ and $B(y_1 - y_2)$ in a 2D plane. It does that by computing straight-line distance (because distance is the straight line joining two points) by using Pythagorean Theorem between differences in the x and y coordinates of points.

$N_{next} = arg \min_{N_i \in N(N)} D_{N_i,D}$ -------------------- (2)

It is the selection equation (2) of the next node. N_{next}in a routing process. Here, $N(N)$ The set of nodes adjacent to current node is $N(N)$. N and $D_{N_i,D}$ Distance between each neighboring Node, D_{N_i} and the destination D. Only to choose the next node to go to is the minimum distance to the destination D.

$N_{next} = Backup\ neighbor\ with\ better\ progress\ toward\ destination$ ----------- (3)

The equation (3) proposes a strategy where, when the current routing path fails or becomes inefficient, another backup neighboring node is selected. This 'backup' neighbor is one that will make better progress towards the destination. The objective is to maintain the routing process by selecting any of the neighboring nodes which can deduce the data adjacent to the destination with minimal effort when primary nodes have stopped being intriguing or accessible.

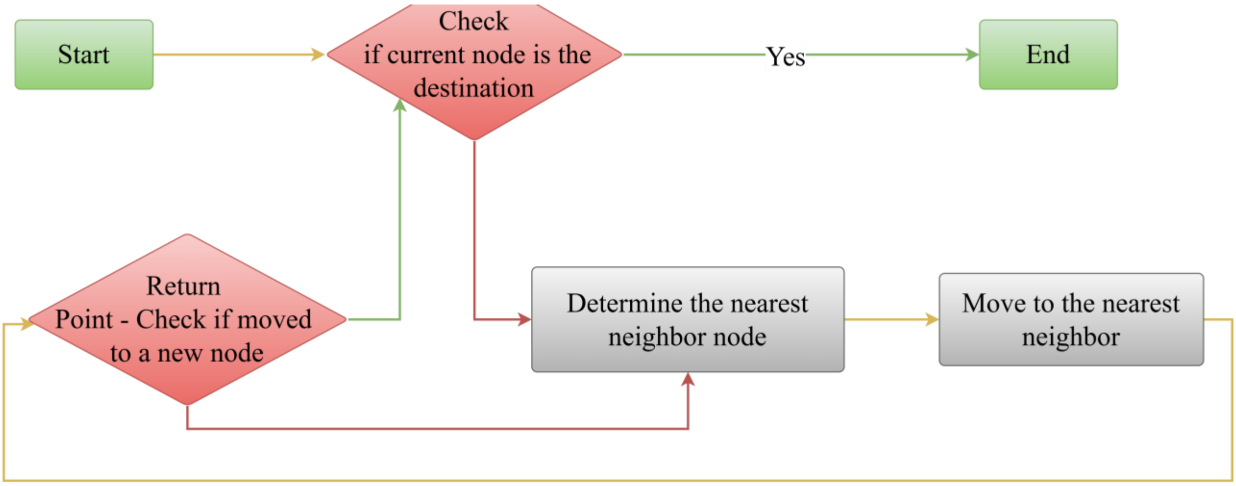

Figure 2: Greedy Routing Flow Chart

Figure 2 is a flowchart of a decision in nearest neighbor approach when finding a path or solving a problem. From here it starts with an initial node and then checks if the current node is the end node. If the destination is reached an organizations ends, Otherwise, it finds out the nearest node from its position and goes towards that nearest node. This process is followed by another check to determine if a new node is set to be invoked as part of navigation flow after the change of node. But if no new node is to be reached it returns to the refinement of the decision. This repetitive process goes on until the destination node is node is discovered.

Algorithm 1: Greedy Routing
Input: - Source node \(*S* \) - Destination node \ (*D* \) - Neighbor nodes of a node \ (N_i \) - Position coordinates of nodes - Distance function \ (*D (N, D)* \) between nodes 1. Initialize: - Current node \ (*N_ {\text{current}} = S* \) (source node) - If \ (*N_ {\text{current}} = D* \), return success (data already at destination) 2. Step 1: Greedy Routing Process - While \ (*N_ {\text{current}} \neq D* \): - If there are no neighbors left (network disconnected or all neighbors are farther from destination): - Return failure (local maxima or unreachable destination)

- Find the closest neighbor \ *(N_ {\text{next}} \)* to the destination \ *(D \):*
- \ *(N_ {\text{next}} = \arg\min_{N_i \in \mathcal{N}(N_{\text{current}})} D(N_i, D) \)*
- If *(D(N_{\text{next}}, D) < D(N_{\text{current}}, D) \)*, forward data to *(N_{\text{next}} \)*
- Set \ *(N_ {\text{current}} = N_ {\text{next}} \)*

3. Step 2: Termination
   - If \ *(N_ {\text{current}} = D \)*, return success (data successfully delivered to destination)
   - Else, if no further progress is possible, return failure (local maxima or no path found)

End Algorithm

Output:

- Data successfully delivered to destination or failure due to local maxima

Greedy Routing method is elucidated in algorithm 2. It prioritizes the neighbors in target's local area top during data delivery. This process stopping indicates data delivery or failure (local maximum or unavailable destination) in case of no closer neighbor.

3.2 Perimeter Routing

Perimeter routing is a solution for the limitations of greedy routing in Wireless Sensor Networks (WSNs) when routing out from local maximum, i.e., when the node has no neighbor that is closer to the target. In the case greedy routing fails the network changes to a different route in which data is routed around the perimeter of a boundary formed by the neighbors of the local maximum. Data is routed around the hull (a convex hull is typically used), until a node is found to which data can go to a destination. It continues to execute until either a different route strategy is employed, or the target is reached. While perimeter routing incur more cost of perimeter traversal, perimeter routing will guarantee delivery of data in sparse or blocked network areas. Hybrid systems that achieve a balance between economy and reliability are often built together with greedy routing. While perimeter routing increases the potential for data transmission to be successful, it can also yield more latency and energy consumption than greedy routing.

$N_{next} = arg \, {}^{min}_{N_i \in N(N_{current})} D(N_i, D)$ --------- (4)

In equation (4), N_{next} determines the next node in routing process using a geometric criterion N_i which minimizes distance D to the destination. Perimeter routing uses identification of the next node N_{next} to efficiently route data packets to their destinations. The next value is computed as argminN_i ∈ (current). *D* (N_i, *D)* selects the neighboring node N_i of the current node ($N_{current}$) which is geographically closest to the destination node D in terms of distance D (N , D). N ($N_{current}$) represents the set of all nearby nodes for Ncurrent.

$N_i \in N(N_{current}), \; D(N_i, D) \geq D(N_{current}, D)$ -------- (5)

This equation (5) assures any nearby node N_i examined for forwarding a data packet is not closer to the destination D than the current node $(N_{current})$ When dealing with routing loops or voids, this condition is critical for identifying situations in which forwarding does not shorten the distance to the destination.

$N_{next} = arg \min_{P_i \in P\,(H)} D\,(N_{current}, D)$----------- (6)

Equation (6) determines which node receives the data packet that comes after the previous one. This is accomplished by having the algorithm analyze every possible forwarding node P_i in the set of all prospective nodes *P (H)* located within a specific area or region *H*. *arg min* operator seeks to identify the node P_i which minimizes the distance $D\,(N_{current}, D)$ between the current node *N* and the destination *D*. It ensures that the packet goes efficiently to its destination by finding the node among the closest candidates.

$N_{current} = P_i$ ---------- (7)

and

$N_{next} = arg \min_{N_j \in N\,(N_{current})} D\,(N_j, D)$ ------- (8)

Denoted as $N_{current} =$, the current node is one of the candidates,oiPi, from the set of probable nodes in equation (7), $N_{current}$ = P_i. Seeking the node to forward the data packet after this, equation (8) searches by taking any adjacent nodes N_j which are included in the collection ($N_{current}$t). It comprises of nodes which are directly connected to $N_{current}$. *arg min* operator assist to identify the neighbor whose distance to the destination *D* is the shortest. By selecting the closest adjacent node, this guarantees that the packet moves towards its destination, therefore enhancing the routing process.

$N_{current} = D$ -------- (9)

As per equation (9); if *N* equals *D*, the node now being processed is the target node D. This indicates that the routing method has arrived at its target and no additional routing decisions are required. When this condition is met, routing operation is completed and the data packet reaches its destination. Routing algorithms typically use this equation as a termination condition to indicate that the packet has been successfully delivered.

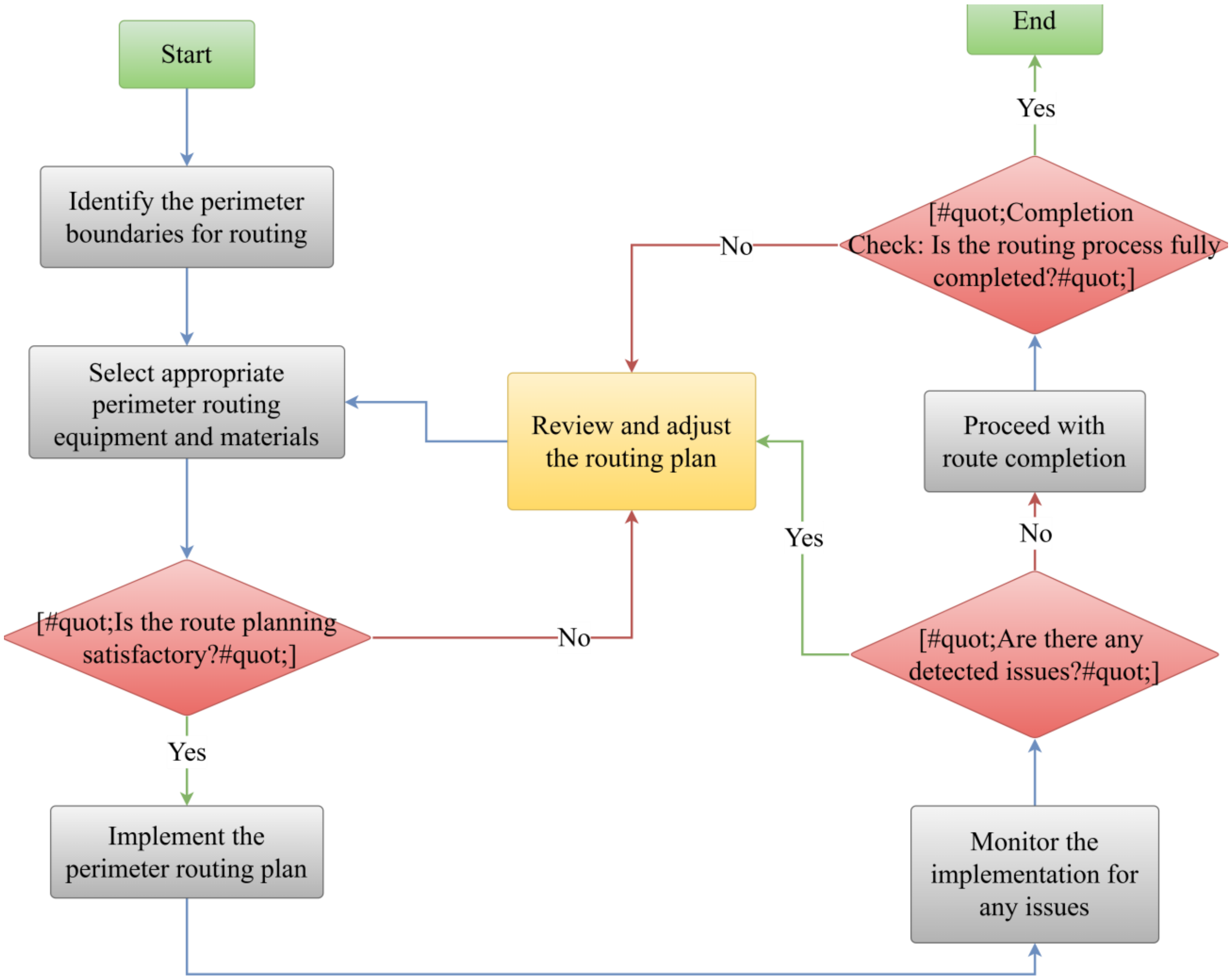

Figure 3: Perimeter Routing Flow Chart

The following Figure 3 illustrates the procedure of perimeter routing and its deployment. ESAB conclude that it begins with defining the sector borders and choosing the right equipment and materials for routing. If the route planning is fine, then a plan is set and followed. If not, then the routing plan is looked over and changed if necessary. After a while, the process looks for problems that might be around. When there is a problem it is solved before moving forward with the procedure. It goes round in circles until the routing exercise is accomplished, then ceases operation.

Algorithm 2: Perimeter Routing
Input: - Source node \ $(S$ \) - Destination node \ $(D$ \) - Neighbor nodes of a node \ $(N_i$ \) - Position coordinates of nodes

- Convex hull formation function
- Distance function $D(N, D)$ between nodes

1. Initialize:
   - Current node $N_{\text{current}} = S$ (source node)
   - If $N_{\text{current}} = D$, return success (data already at destination)
2. Step 1: Greedy Routing (Initial Attempt)
   - While $N_{\text{current}} \neq D$:
   - Find the closest neighbor N_{next} *to the destination* D:
   - $N_{\text{next}} = \arg\min_{N_i \in \mathcal{N}(N_{\text{current}})} D(N_i, D)$
   - If $D(N_{\text{next}}, D) < D(N_{\text{current}}, D)$, forward data to N_{next}
   - Set $N_{\text{current}} = N_{\text{next}}$
   - If N_{next} is not closer to D (local maxima detected), proceed to Step 2 (Perimeter Routing)
3. Step 2: Perimeter Routing (Handle Local Maxima)
   - If greedy routing fails (no closer neighbor found):
   - Identify the local maximum region and form the convex hull H of the nodes around the region
   - Start perimeter traversal:
   - Identify the first node P_1 in the convex hull H
   - Set $N_{\text{current}} = P_1$
4. Step 3: Traverse the Perimeter
   - While $N_{\text{current}} \neq D$:
   - Find the next node on the convex hull N_{next}:
   - $N_{\text{next}} = P_i$ (next node in the perimeter sequence)
   - Forward data to N_{next}
   - If $D(N_{\text{next}}, D) < D(N_{\text{current}}, D)$, return to greedy routing:
   - $N_{\text{current}} = N_{\text{next}}$
   - Proceed to Step 1 (Greedy Routing)
5. Step 4: Termination
   - If destination D is reached, return success (data successfully delivered to destination)
   - If no progress can be made (no path found), return failure (no route to destination)

End Algorithm

Output:

- Data successfully delivered to destination or failure due to no path

The method explained in algorithm 3 refers to Perimeter routing. It enters convex hull mode and navigates around the obstruction until it finds a node closer to the goal, overcoming local maxima. Otherwise, it returns to greedy routing in an attempt to send the data; if no path is found, the operation terminates.

3.3 Energy-Balanced Routing Algorithm (EBRA)

In WSNs, the Energy-Balanced Routing Algorithm (EBRA) seeks to maximise network uptime. This method decreases power consumption and distributes electricity fairly among all nodes. Uneven energy consumption causes some WSN sensor nodes to fail prematurely due to resource restrictions, fragmenting the network and resulting in lesser performance than it would otherwise have. To address this difficulty, EBRA finds the most effective routes for the network's autonomous routing. The approach will find energy-efficient pathways and employ load balancing, multi-hop routing, or clustering to reduce power consumption on each node.

EBRA improves network efficiency over time by reducing the power consumption of the most frequently utilised nodes. It also dynamically alters routing possibilities in response to topological changes, node density, and environmental conditions. This energy-efficient technique improves network longevity, data transmission reliability, and early node failure detection. Smart cities, military surveillance, and environmental monitoring are among the applications that will benefit the most from EBRA's continuous network capabilities.

$$E_{tx} = E_{elec} + E_{amp}.d^n \text{ --------- } (10)$$

Equation (10) estimates the total energy spent by the transmitter, where E represents total energy and n is the component count. The energy used by the transmitter's electrical components including signal processing and encoding, is originally indicated asE_{elec}. If the signal is to reach the receiver over a distance of d, it has to be boosted. This is seen by the second component, $E_{amp}.d^n$. The energy demand as a function of distance is expressed by the functiond^n, where n is the path loss exponent which considers the signal's attenuation with distance resulting from environmental conditions. It allows to roughly estimating the overall energy cost of data transmission via a wireless channel.

$$E_{rx} = E_{elec} \text{ --------- } (11)$$

Equation (11) states that the energy consumed by the receiver E_{rx} is equal to the energy used by its electronic componentsE_{elec}. This energy is used for signal reception, demodulation, and decoding.

$$E_{node}(t+1) = E_{node}(t) - \sum_i E_{tx}(i) - E_{rx}(i) \text{ -------- } (12)$$

The equation (12) shows that the node's energy usage decreases over time, taking into account both transmission E_{tx} and reception E_{rx}). The energy at the next time step E_{node} is calculated by subtracting the total energy expended by sending and receiving from other nodes from the current energy at each node (t+1). i and t refers to the node and time.

$$P_i = \frac{E_{node}(i)}{E_{max}} \text{ ---------- } (13)$$

This equation (13) calculates the energy ratio P_i for node i, which is the current energy level $E_{node}(i)$ divided by the node's maximum energy capacity E_{max}. This ratio allows one to calculate the remaining energy relative to the node's total capacity.

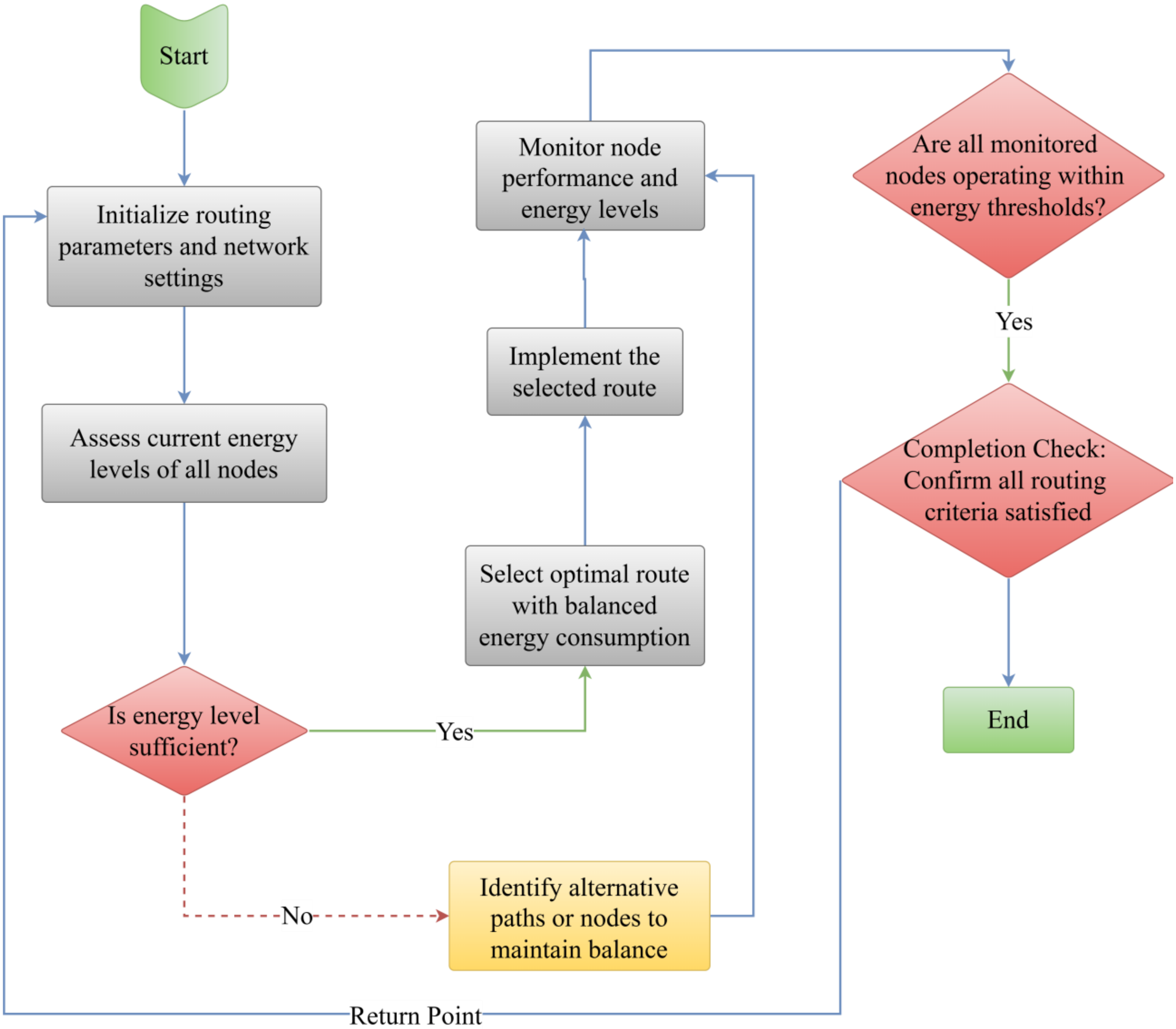

Figure 4: Energy-Balanced Routing Algorithm (EBRA) Flow Chart

The following is a Figure 4 of energy-aware routing process. S telecom starts with setting routing parameters and testing energy levels within the nodes. If the energy values are enough an optimal path with reasonable values of energy consumption for each geographical point is defined and then realized. When routing, information regarding node performance and energy level is collected. If nodes are running below or above the specified energy ranges, it comes to an end. When energy supply appear to be inadequate, other circuits or bridges are considered so as to restore equilibrium. They give this iterative or cyclic approach an assurance of the overall energy efficiency and stability of routing.

Algorithm 3: Energy-Balanced Routing Algorithm (EBRA)

Input:

- Source node \ (S \)
- Destination node \ (D \)
- Neighbor nodes of a node \ (N_i\)
- Position coordinates of nodes
- Energy levels of each node \ (E (N_i) \)
- Distance function \ *(D (N, D) \)* between nodes

1. Initialize:

- Current node \ *(N_ {\text{current}} = S \)* (source node)
- If \ *(N_ {\text{current}} = D \)*, return success (data already at destination)
- Set \ *(E_ {\text{threshold}} \)* (minimum energy threshold for routing decisions)
- Set energy levels \ *(E (N_ {\text{current}}) \)*

2. Step 1: Energy-Based Routing Decision

- While \ *(N_ {\text{current}} \neq D \):*
- Find the set of neighbors \ *(\mathcal{N}(N_{\text{current}}) \)*
- From the neighbors, select the node \ *(N_ {\text{next}} \)* that satisfies:
- \ *(E (N_ {\text{next}}) \geq E_ {\text{threshold}} \)* (Node has sufficient energy)
- \ *(D (N_ {\text{next}}, D) < D (N_ {\text{current}}, D) \)* (Node brings the data closer to the destination)
- Minimize energy consumption:
-*(\text{Energy consumption}(N_{\text{current}}, N_{\text{next}}) = f(D(N_{\text{current}}, N_{\text{next}}), E(N_{\text{current}})) \)*
- If such a neighbor \ *(N_ {\text{next}} \)* is found:
- Forward data to \ *(N_ {\text{next}} \)*
- Set \ *(N_ {\text{current}} = N_ {\text{next}} \)*
- Else, proceed to Step 2 (Energy Recovery)

3. Step 2: Energy Recovery (if no valid node found)

- If no valid neighbor is found (all neighbors have insufficient energy):

- Check for alternative paths or local energy recovery strategies (e.g., local energy harvest or node sleeping mechanisms)

- If an alternative path exists, switch to that path:

- \ *(N_ {\text{current}}* = \text{Alternative node with sufficient energy} \)

- If no path is found, return failure (network partitioned or energy depletion)

4. Step 3: Termination

- If \ *(N_ {\text{current}} = D \)*, return success (data successfully delivered to destination)

- If the energy of nodes in the path is insufficient and no alternative path exists, return failure (network failure or energy exhaustion)

End Algorithm

Output:

- Data successfully delivered to destination or failure due to energy depletion

Algorithm 4 describes the process carried out in EBRA method. To reduce power usage, this EBRA transmits data to nearby node which has sufficient power and is close to the eventual destination. If no such neighbor exists, it attempts recovery via alternative channels or fails due to energy exhaustion.

3.4 Geographic and Energy-Aware Routing (GEAR)

The Gear routing system combines geographic routing with energy-efficient approaches to maximize energy consumption while ensuring data transfer reliability. It is intended for use in wireless sensor networks (WSNs). Although each sensor node in GEAR is aware of its own geographic location, the location of the destination determines the routing alternatives. GEAR routes data effectively by taking into account both node physical locations and energy levels. It combines greedy forwarding with energy-aware approaches to prevent data from being transmitted to nodes that are about to run out of resources.

The strategy works well because it divides the network into portions and ensures that routing decisions in each area are based on the energy condition of the nodes. Nodes send data to the next node geographically closer to the goal while managing the network's energy consumption. GEAR takes into account geographical proximity and energy levels to provide a longer network lifetime and fewer energy hotspots, resulting in some nodes being overworked and dying prematurely. Given that the long-term operation of large-scale WSNs relies on energy efficiency, GEAR's method is ideal for such networks.

$$next\ hop = \arg\min\left(d\ (i, D) +\ \lambda.\frac{E_{node}\ (j)}{E_{max}}\right) \text{ -------- (14)}$$

Equation (11) selects the next hop node with the lowest sum of distances to the destination *(d(I, D))* and a factor λ times the candidate node's remaining energy ($E_{node}\ (j)$) to maximize E_{max}.

$$R_{energy}\,(i) = \frac{E_{node}\,(i)}{E_{max}}\,.R_{max} \text{ -------- } (15)$$

Equation (15) calculates the energy-related reward$R_{energy}\,(i)$ for each node *i* by taking the product of its maximum energyE_{max}, remaining energy after consumption, and maximum reward R_{max}. Nodes are graded depending on residual energy and rewarded accordingly. If

$$E_{node}\,(i) < \in \text{ -------- } (16)$$

node *i* is excluded from routing. This equation (16) allows assessing the energy level of node *i*, represented as $E_{node}\,(i) <$ to determine whether it is less than the threshold e. If the node's energy level falls below this level, it may not have enough power to carry out its functions properly.

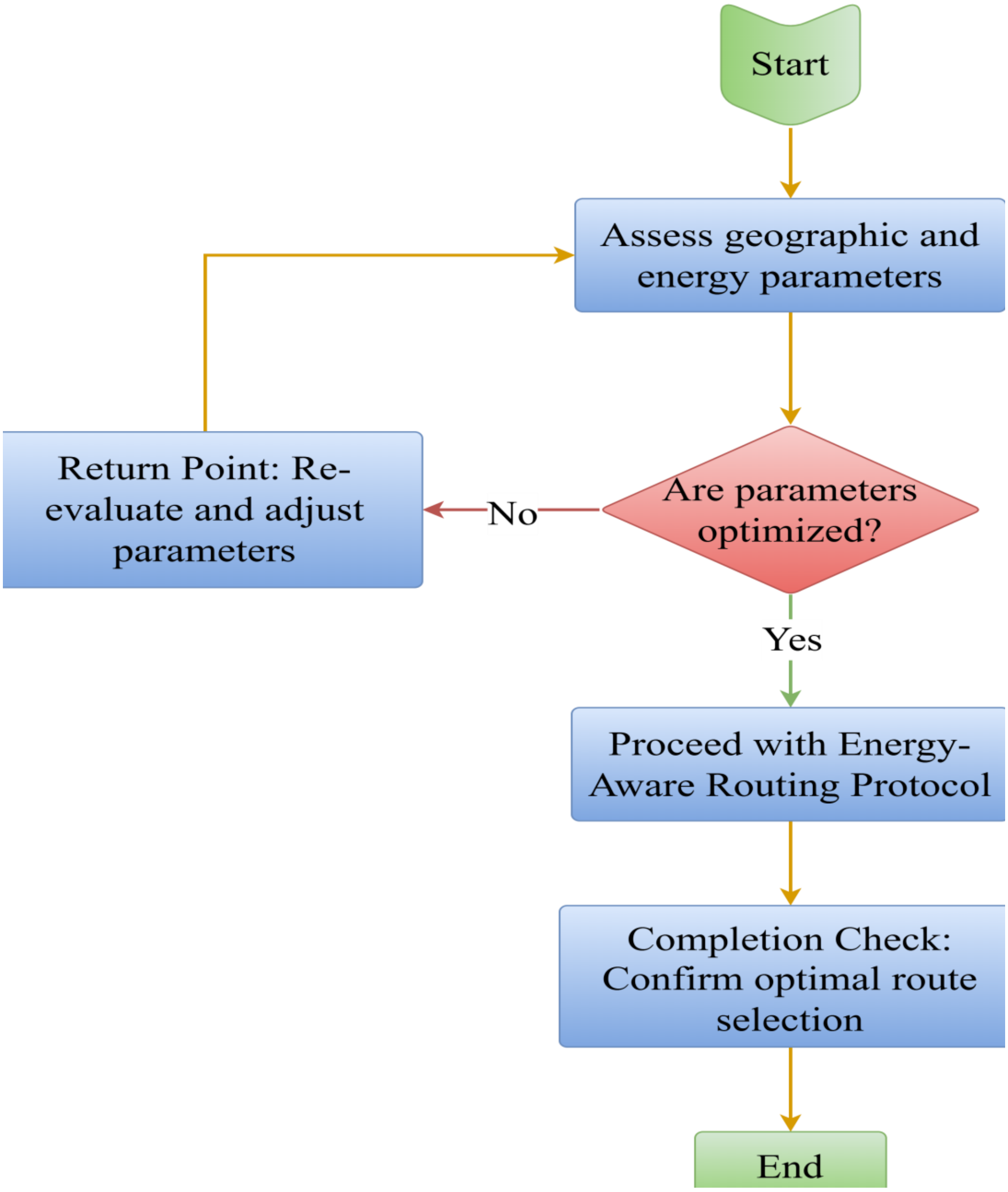

Figure 5: Geographic and Energy-Aware Routing (GEAR) Flow Chart

Algorithm 4: Geographic and Energy-Aware Routing (GEAR)
Input: - Source node \ *(S \)* - Destination node \ *(D \)* - Neighbor nodes of a node \ (N_i\) - Position coordinates of nodes - Energy levels of each node \ *(E (N_i) \)* - Distance function \ *(D (N, D) \)* between nodes 1. Initialize: - Current node \ *(N_ {\text{current}} = S \)* (source node) - If \ *(N_ {\text{current}} = D \)*, return success (data already at destination) - Set \ *(E_ {\text{threshold}} \)* (minimum energy threshold for routing decisions) - Set energy levels \ *(E (N_ {\text{current}}) \)* 2. Step 1: Geographic and Energy-Aware Routing Decision - While \ *(N_ {\text{current}} \neq D* \): - Find the set of neighbor \ *(\mathcal {N} (N_ {\text{current}}) \)* - From the neighbors, select the node \ *(N_ {\text{next}} \)* that satisfies the following criteria: 1. **Energy Requirement**: \ *(E (N_ {\text{next}}) \geq E_ \text{threshold}} \)* (Node must have sufficient energy) 2. **Geographic Proximity**: \ *(D (N_ {\text{next}}, D) < D (N_{\text{current}}, D) \)* (Node must be closer to the destination) 3. **Energy Efficiency**: Minimize energy consumption: - Energy consumption \ *(= f(D(N_{\text{current}}, N_{\text{next}}), E(N_{\text{current}})) \)* 4. **Avoid High Energy Depletion**: Prefer nodes with higher remaining energy to avoid early energy exhaustion. - If such a neighbor \ *(N_ {\text{next}} \)* is found: - Forward data to \ *(N_ {\text{next}} \)* - Set \ *(N_ {\text{current}} = N_ {\text{next}} \)* - Else, proceed to Step 2 (Energy Recovery and Alternative Path) 3. Step 2: Energy Recovery or Alternative Path (if no valid node found) - If no valid neighbor is found (all neighbors have insufficient energy or are not bringing the data closer to destination):

- Check for local energy recovery strategies:

- **Local energy harvesting** (if applicable), or

- **Node sleeping mechanisms** to save energy

- If an alternative path with sufficient energy exists, switch to that path:

- \ *(N_ {\text{current}} = \text{Alternative node with sufficient energy} \)*

- If no alternative path is available, return failure (energy depletion or no route found)

4. Step 3: Termination

- If \ *(N_ {\text{current}} = D \)*, return success (data successfully delivered to destination)

- If the energy of nodes in the path is insufficient and no alternative path exists, return failure (network failure or energy exhaustion)

End Algorithm

Output:

- Data successfully delivered to destination or failure due to energy depletion

Algorithm 5 represents the GEAR algorithm. This strategy chooses neighbors which are close to the goal and have enough energy to convey the data, hence improving energy efficiency. It seeks recovery through several channels or fails due to energy depletion in the absence of a suitable neighbor.

3.5 Geographic Greedy Perimeter and Energy-Aware Routing (GGPEAR) in Wireless Sensor Networks

GGPEAR combines geographic routing, perimeter-based routing, and energy-aware techniques to improve data transmission reliability and efficiency. This is a hybrid routing technology designed specifically for Wireless Sensor Networks (WSN). WSNs sometimes have limited resources, therefore a reliable routing system is required to control the network's power consumption and send data where it is needed. GGPEAR begins with a greedy routing strategy in which every node forwards data to the node closest to the eventual destination. This greedy strategy has limitations, including local minima, where the next node may not be much closer to the objective and even be at a dead end, despite its ability to shine in sparse networks and traverse short distances quickly.

In these cases, GGPEAR implements perimeter-based routing, which sends data beyond the network's outer boundaries until it arrives at a node that is geographically closer to its final destination. When greedy routing encounters local minima or blockages, data can still be routed in this perimeter mode before failing. While the energy-efficient component of GGPEAR aims to reduce total network energy consumption, greedy and perimeter-based routing addresses topological difficulties. The approach implements an energy-aware measure that prioritizes nodes with greater energy levels in order to avoid overusing low residual energy nodes and prematurely draining them while transmitting data. As a result, network utilization will be homogenous, with no energy hotspots developing.

GGPEAR can also accommodate various network topologies and node densities. The routing plan is constantly adjusted in real time based on energy levels and the distance to the target. The approach also makes use of multi-hop communication to effectively disperse energy around the network when direct transmission to the destination is not possible. GGPEAR, which combines global routing, perimeter routing, and energy awareness, strikes a balance between energy efficiency, resilience, and data delivery dependability, making it ideal for large-scale WSNs with limited resources and changing network conditions. This strategy extends the network's lifetime while reducing congestion and ensuring good communication even in demanding circumstances such as densely inhabited areas or sites with numerous barriers.

$E_{node}\,(t+1) = \, E_{node}\,(t) - \, E_{tx} + E_{rx}$ --------- (17)

This equation (17) calculates the node's new energy at time *(t+1)* using the transmission energy E_{tx} and the reception energyE_{rx}. This depicts the node's overall energy shift caused by data reception and transmission.

$perimeter\ next\ hop = \arg{}^{min}_{j} \left(d\ (i, D + \ \lambda.\frac{E_{node}\,(j)}{E_{max}}\right)$ ------- (18)

Equation (18) finds the *next hop* node in perimeter-based routing by minimizing the total of distances to the destination *d (i,D)* and a weighted proportion of remaining energy of the likely next hop. Weighted factor is represented as $\lambda\cdot$ Energy level of node *j* is $E_{node}\,(j)$. This allows to choose a node which maximizes efficiency while remaining geographically close to the end destination.

$E_{multi-hop} \ = \ \sum_{i=1}^{n} \sum_{i} E_{tx}\,(i) - \ E_{rx}\,(i)$ ------------ (19)

As shown in equation (19), the total energy consumption throughout *n* hops is calculated by adding the energy transmission E_{tx} and reception E_{rx} at each hop. The result displays the total amount of energy used to transfer data between network nodes.

$Next\ hop = \arg{}^{min}_{j} \left(d\ (i, D + \ \alpha.\frac{E_{node}\,(j)}{E_{max}}\right)$ -------- (20)

This equation (20) picks the next hop node by minimizing the total distances to destination $(i,)$ and weighted factor α of the energy consumption at the prospective next hop j, given by E node (b). E_{node} at position *j* is $E_{node}\,(j)$. *Next Hop* is calculated by taking into account both energy efficiency and distance to destination.

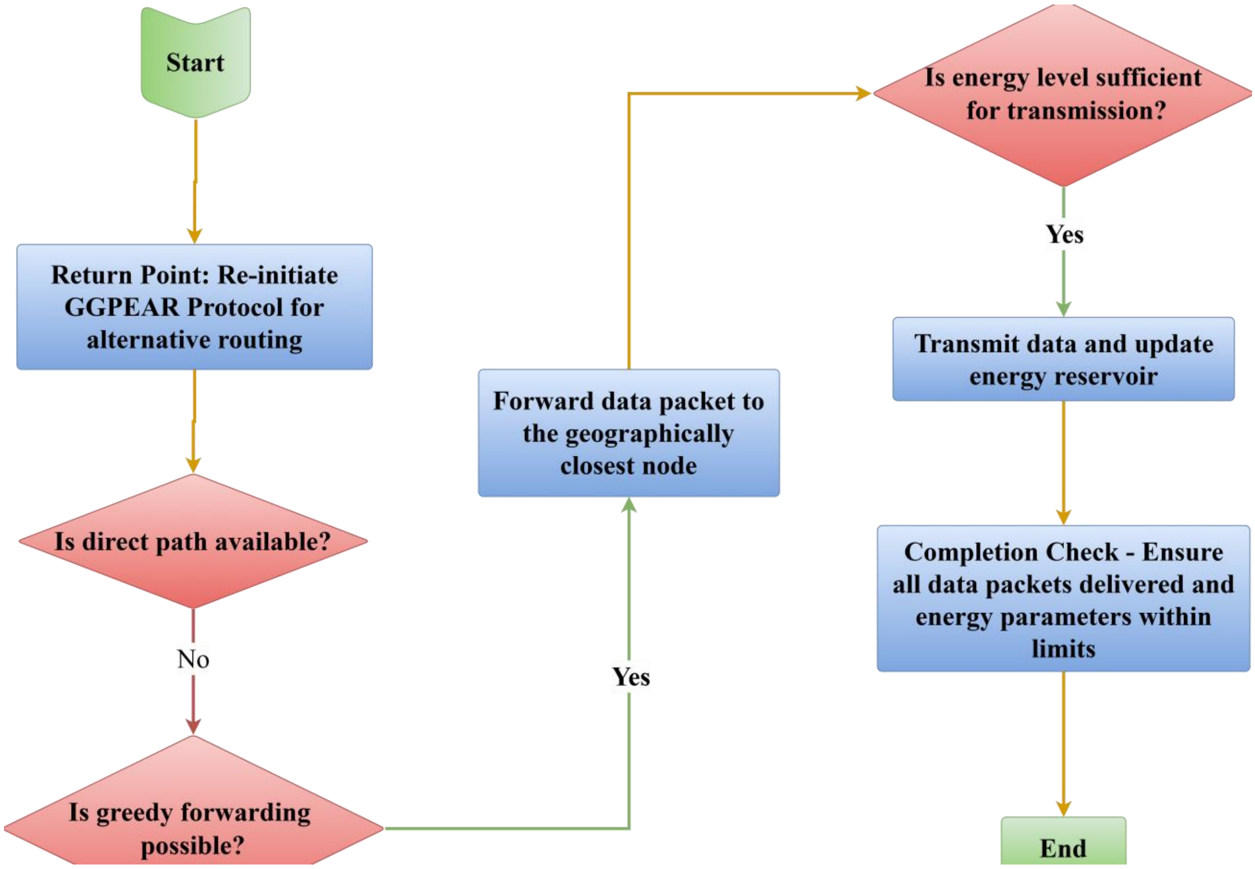

Figure 6: Geographic Greedy Perimeter and Energy-Aware Routing (GGPEAR) Flow Chart

Algorithm 5: Geographic Greedy Perimeter and Energy-Aware Routing (GGPEAR)
Input: - Source node \\(S \\) - Destination node \\ (D \\) - Neighbor nodes of a node \\ (N_i \\) - Energy levels of each node \\ ($E\ (N_i)$ \\) - Distance function \\ ($D\ (N, D)$ \\) between nodes 1. Initialize: - Current node \\(N_{\\$text${current}} $= S$ \\) (source node) - If \\(N_{\\$text${current}} $= D$ \\), return success - Set \\(E_{\\$text${threshold}} \\) (minimum energy threshold for routing decisions) - Set energy level \\($E(N$_{\\$text${current}}) \\) 2. Step 1: Energy-Aware Greedy Routing

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 1 (2025)

- While \ (N_{\text{current}} \neq D \):

- If \(E(N_{\text{current}}) < E_{\text{threshold}} \), switch to recovery or stop routing

- Find the closest neighbor \(N_{\text{next}} \) using greedy forwarding:

- \(N_{\text{next}} = \arg\min_{N_i \in \mathcal{N}(N_{\text{current}})} D(N_i, D) \)

- If \(E(N_{\text{next}}) \geq E_{\text{threshold}} \) and \(D(N_{\text{next}}, D) < D(N_{\text{current}}, D) \), forward data to \(N_{\text{next}} \)

- Else, proceed to Step 2 (Perimeter Routing)

3. Step 2: Energy-Aware Perimeter Routing (if greedy routing fails)

- If greedy routing fails (no closer neighbor found):

- Form the convex hull \ (H \) around the local maximum nodes

- Identify the first node \ (P_1 \) in the convex hull \ (H \)

- Set \ (N_ {\text{current}} = P_1 \)

- If \ (E (N_ {\text{current}}) < E_ {\text{threshold}} \), select the next available node with higher energy on the perimeter

4. Step 3: Traverse Perimeter with Energy Awareness

- While \ (N_ {\text{current}} \neq D \):

- Find the next node on the convex hull \ (N_ {\text{next}} \):

- \ (N_ {\text{next}} = P_i \) (next node in the perimeter sequence)

- Forward data to \ (N_{\text{next}} \)

- If \(E(N_{\text{next}}) \geq E_{\text{threshold}} \) and \(D(N_{\text{next}}, D) < D(N_{\text{current}}, D) \), switch to greedy routing:

- \ (N_ {\text{current}} = N_ {\text{next}} \)

- Return to Step 1 (Greedy Routing)

- Else, continue perimeter traversal

5. Step 4: Termination

- If destination \ (D \) is reached, return success

- If the data cannot reach the destination or energy is insufficient, return failure and trigger recovery or alternative routing

End Algorithm

Output:

- Data successfully delivered to destination or failure with recovery mechanism

Algorithm 1 describes the GGPEAR method**.** Regarding wireless sensor networks, GGPEAR method considers energy levels and uses a greedy approach to forward data to the closest node prior for determining their destination. If greedy forwarding fails to go over a barrier, it turns back to perimeter routing.

4. Results and Discussion

The Geographic Greedy Perimeter and Energy-Aware Routing (GGPEAR) protocol improves the outcomes of WSN efficiency by a two-dimensional approach that combines geographic greedy forwarding with energy-aware functions. Eliminate energy wastage while enhancing the efficiency of perimeter routing hence enhancing the network lifetime through attenuation of components hindering efficient communication.

4.1 Throughput

The throughput is the amount of data successfully transmitted at the source to the destination in a time interval. In other words, it is an ultimate metric of data transfer efficiency and is a metric that is typically expressed in bits per second (bps).

$$Throughput = \frac{Total\ Data\ Received\ (in\ bits)}{Total\ Time\ (in\ seconds)} \text{-------- (21)}$$

Table 2: Comparison table on Throughput

Number of Nodes	**GGPEAR (Proposed)**	**EEM-CRP (Existing)**	**EE-TLT (Existing)**
10	423	382	372
20	402	371	363
30	391	364	354
40	382	352	341
50	371	342	332
60	364	331	324

Table 2 examines the results of three methods namely GGPEAR (Proposed), EEM-CRP (Existing), and EE-TLT (Existing) for different nodes (10, 20, 30, 40, 50, 60). The values are some sort of metric or quantification of every method at each nodes value. In the case of gg-pear form for the values it is a different story since it starts at 423 at 10 nodes and goes down to 364 at 60 nodes. Also, the EEM-CRP and EE-TLT approaches follow the decrease in values, but with somewhat lower values as compared to the GGPEAR at every node count. For instance, EEM-CRPwhen 10 nodes = 382 and EE-TLT = 372, and the values reduce constantly as the number of nodes increasing. This implies that the GGPEAR method presented in this paper at some point surpasses the other methods improving at higher node counts with less difference in value. The table gives some idea of how each method behaves with respect to nodes and would be of use in comparing the efficiency or efficacy of the various strategies at greater scales.

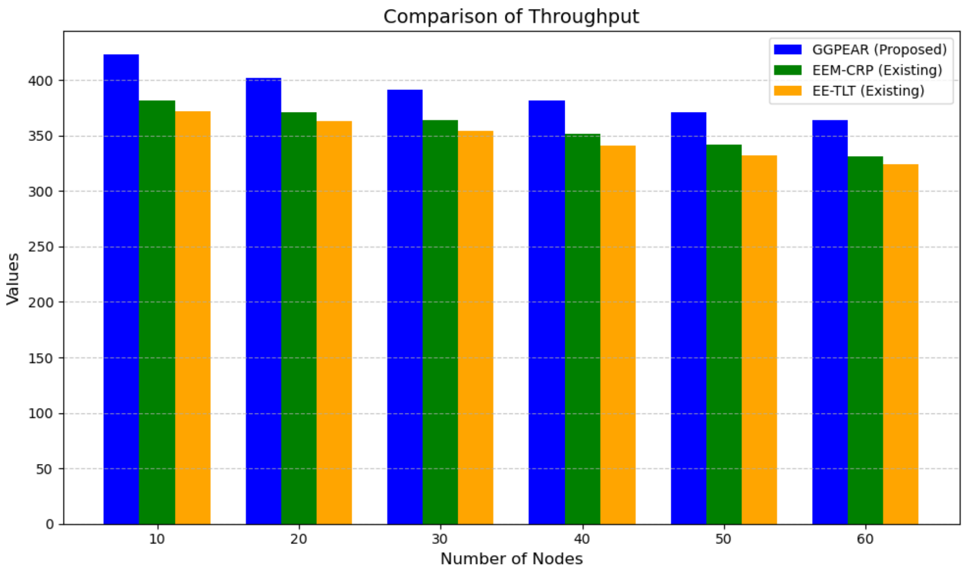

Figure 7: Comparison Chart on Throughput

The Figure 7 illustrates the comparison of throughput values for three protocols: plot the number of nodes (10 to 60) over linear piecewise fit (Proposed), EE-TLT (Existing) and EEM-CRP (Existing). The experiment results show a slightly higher throughput values for all the test cases than existing methods such as EEM-CRP and EE-TLT, and thus GGPEAR performs better. The performance of the EEM-CRP generally outperforms the EE-TLT; however, the difference between them is less pronounced under higher node counts. This is a trend that shows that GGPEAR can be scaled and be more efficient at, providing better throughput when are dealing with larger network sizes. It would be possible to see the proposed protocol advantages over the existing ones, hence why this protocol is more preferable in this case.

4.2 Energy Consumption

Energy Consumption in networks is the total power consumed by devices to transmit, receive and process data. In order to extend the lifespan of a battery powered device, like those used in wireless sensor networks and IoT systems, energy consumption must be minimized.

$Energy\ Consumption = Initial - Residual\ energy$ -------- (22)

Table 3: Comparison table on Energy Consumption

Number of Nodes	GGPEAR (Proposed)	EEM-CRP (Existing)	EE-TLT (Existing)
10	12.55	15.32	16.75
20	25.84	30.42	32.31

30	38.91	46.71	48.22
40	52.16	62.52	64.31
50	65.72	78.21	80.14
60	78.65	89.25	95.23

Table 3 compares responses of three methods namely GGPEAR (Proposed), EEM-CRP (Existing) and EE-TLT (Existing) in terms of number of nodes 10, 20, 30, 40, 50, and 60. The values are indicating some kind of metric or measurement for each method for each classes of nodes. For the GGPEAR approach the values reduce gradually with increase in the number of nodes from 423 on 10 nodes to 364 on 60 nodes. The EEM-CRP and EE-TLT trend looks similar to that of GGPEAR, although slightly lower for corresponding node counts at each of the grits. For example at 10 nodes EEM-CRP has a score of 382 while EE-TLT has a score of 372 and as the number of nodes rises the results begin to fall. This means that the proposed GGPEAR method can handle scenarios with higher node count better than the existing methods and as the node count increases the rate at which value reduces is lower. The table gives a sense of how each of these methods grows with the number of nodes and using the table, one is able to compare the efficacy or feasibility of these approaches at scale.

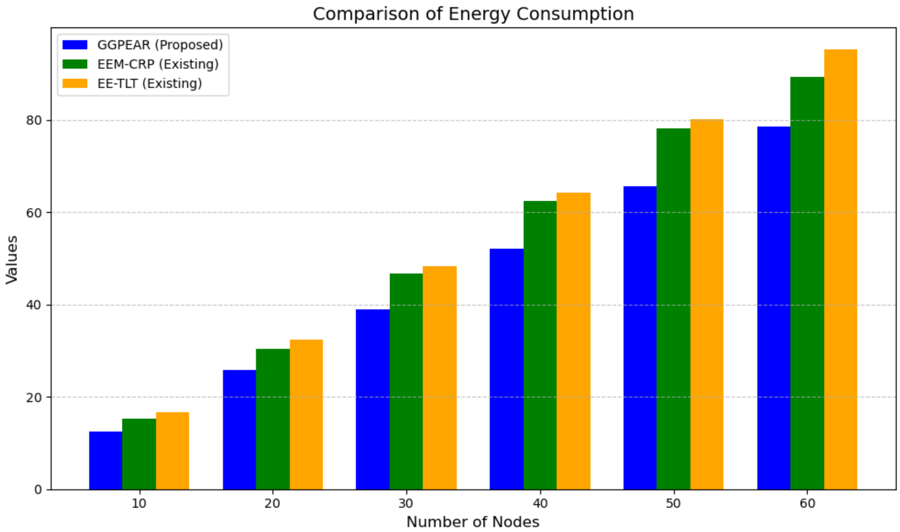

Figure 8: Comparison Chart on Energy Consumption

Figure 8 respect to varying number of sensor nodes compare the energy consumption of proposed GGPEAR method with two existing methods namely EEM-CRP and EE-TLT. In addition, GGPEAR shows lower energy consumption consistently compared to the other two methods, to make this case. With the increase in number of nodes, energy consumption of all methods increases, but the growth rate of GGPEAR is minimal, which leads to believe that it is scalable. The GGPEAR results validate that GGPEAR is able to optimize energy usage and therefore use it to prolong the lifeline of wireless sensor networks. The significance of such high efficiency is more important in large scale networks with larger node counts.

4.3 End-to-End Delay

The term End to End Delay in fact involves the time that a data packet takes passing through a Network from source to the destination. In its external context, it is a parameter that measures the performance and QoS of a network and encompasses transmission and propagation delays, processor delay, and delay due to queuing.

$$End - to - End\ Delay = \frac{\sum(Time\ Received - TimeSent)}{Total\ Number\ of\ Packets\ Received} \text{ -------- } (23)$$

Table 4: Comparison table on End-to-End Delay

Number of Packets	GGPEAR (Proposed)	EEM-CRP (Existing)	EE-TLT (Existing)
100	118	132	141
200	124	136	146
300	127	143	154
400	133	147	157
500	138	154	162

Table 4 analyses the performance of three methods GGPEAR (Proposed), EEM-CRP (Existing) and EE-TLT (Existing) by the no. of packets (100, 200, 300, 400, 500). The values in the table probably indicate the execution time or other utilization of the method by each implementation at various packet quantities. This indicates that as the number of packets increases the values for all three methods increase and that for the GGPEAR (Proposed) method the increase is minimal. For instance, at 100 packets, have GGPEAR = 118 while EEM-CRP = 132 and EE-TLT = 141. This phenomenon continues for the higher packet count, and this indicates that the GGPEAR method outperforms the other methods for the high packet count since the resource usage gain is slower than the other methods. This implies that GGPEAR be a better solution to handling large packet size payloads than is currently implemented across interfaces in Open Flow.

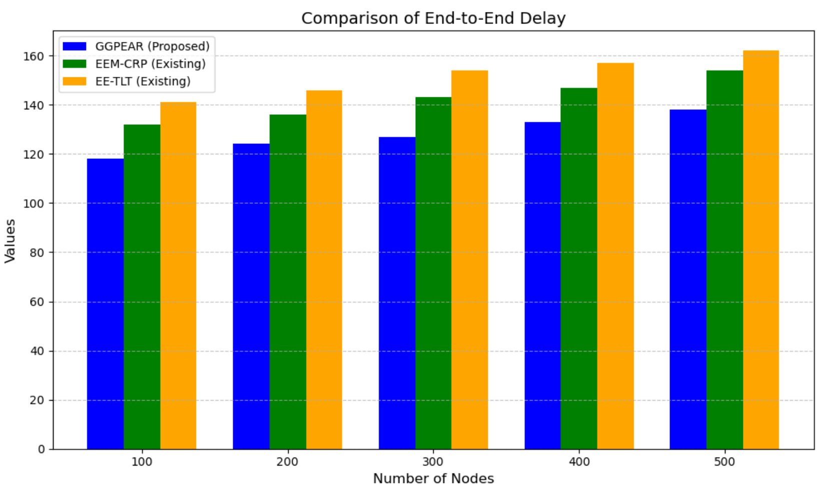

Figure 9: Comparison Chart on End-to-End Delay

A Figure 9 compares the end to end delay of the proposed GGPEAR method and two existing methods with different node counts. Furthermore, GGPEAR always gets the lowest delay and thus demonstrate better efficiency in transmitting data. EEM-CRP and EE-TLT have larger delays, and EE-TLT delays are the largest among all scenarios. With increasing number of nodes, delays of all methods slightly increase; however, GGPEAR still enjoys a large edge. This illustrates that GGPEAR is capable of minimizing communication delays and therefore suitable for time constrained applications in a wireless sensor network.

4.4 Packet Delivery Ratio (PDR)

PDR or Packet Delivery Ratio can be defined as the ability of the packets to be sent and received within a network correctly out of those packets that were sent. It is very important that it quantifies stability and efficiency of the network and any value higher than PDR indicates better link quality.

$$PDR = \frac{Number\ of\ Packets\ Received}{Number\ of\ Packets\ Sent} \times 100 \text{ -------- } (24)$$

Table 5: Comparison table on Packet Delivery Ratio (PDR)

Number of Packets	**GGPEAR (Proposed)**	**EEM-CRP (Existing)**	**EE-TLT (Existing)**
100	98.5	96	94.5
200	98.2	95.7	94
300	97.8	95.4	93.7
400	97.4	95	93.3
500	97	94.6	93

The following table 5 is the evaluation of three methods in terms of packets, 100, 200, 300, 400 and 500; GGPEAR is the Proposed method and EEM-CRP, EE-TLT are Existing methods. The values probably reflect another parameter like success rate, time, speed or some kind of quality standard for each methodology at various packets levels. In general, all three methods depict a very slight decline in performance as the packets rise but by a rather uniform trend across the table. As for the performance of different approaches, proposed GGPEAR sustains the best performance level ranging from 98.5 which was achieved after handling 100 packets down to 97 percent after handling 500 packets. Similarly and combined, but with slightly worse overall performance, is the EEM-CRP, which begins at 96 and declines to 94.6, while EE-TLT begins at 94.5 and declines to 93. This means that even though there is a general trend of a decrease in correlation as the packet count increases, GGPEAR is less inclined to this change, which means it better in sustaining its performance at large volume.

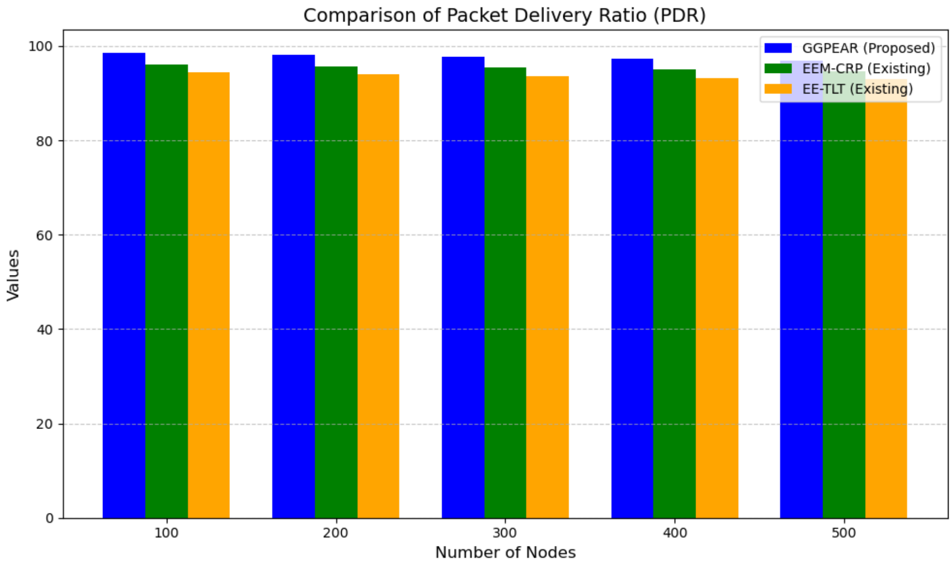

Figure 10: Comparison Chart on Packet Delivery Ratio (PDR)

The packet delivery Ratio (PDR) of the proposed GGPEAR method is compared with the existing EEM-CRP and EE-TLT methods for different numbers of nodes using Figure 10. Overall, GGPEAR consistently attains the highest PDR, exceeding 99%, and is qualitatively found to receive data reliably and efficiently. GGPEAR performs well, and EEM-CRP also performs well, but yields slightly lower PDR than GGPEAR, and EE-TLT yields the lowest PDR. All methods have relatively stable performance when the number of nodes increases, however, GGPEAR not only shows its superior PDR, but also demonstrates its robustness and reliability in wireless sensor networks.

4.5 Packet Loss Ratio

Packet Loss Ratio better known as PLR is defined as the proportion of the transmission data packets that did not reach their destination intact within the network layer throughout, this is because of congestion or errors or other hindrances. The value of PLR can give the possessor some useful information about the current state of a specific network: the lower the PLR, the better the network’s condition and reliability; in contrast, the higher the PLR, the more signs of inefficiency and troubles in the given communication network are to be expected.

$Packet\ Loss\ Ratio = 1 - PDR$ -------- (25)

Table 6: Comparison table on Packet Loss Ratio

Number of Packets	**GGPEAR (Proposed)**	**EEM-CRP (Existing)**	**EE-TLT (Existing)**
100	1.5	4	5.5

200	1.8	4.3	6
300	2.2	4.6	6.3
400	2.6	5	6.7
500	3	5.4	7

Table 6 shows packet by packet comparison of GGPEAR (Proposed), EEM-CRP (Existing), and Existing method EE-TLT with packets 100, 200, 300, 400, 500. The values are, apparently, a measure of efficiency, possibly time or resource consumption for each method depending on the number of packets. With the increase of packets, all of them have higher value of their corresponding metrics, but the rate of increase of GGPEAR (Proposed) is comparatively lower. For instance, at point 100 packets GGPEAR is equal to 1.5 whereas EEM-CRP and EE-TLT values are 4 and 5.5 respectively. This trend persists even in large packets, thereby confirming the findings that GGPEAR outperforms the current methods when the number of packets are raised. This is evidenced by a slightly lesser increase in the values of GGPEAR, which indicate that it is more resource efficient, or less time consuming as packet size increases.

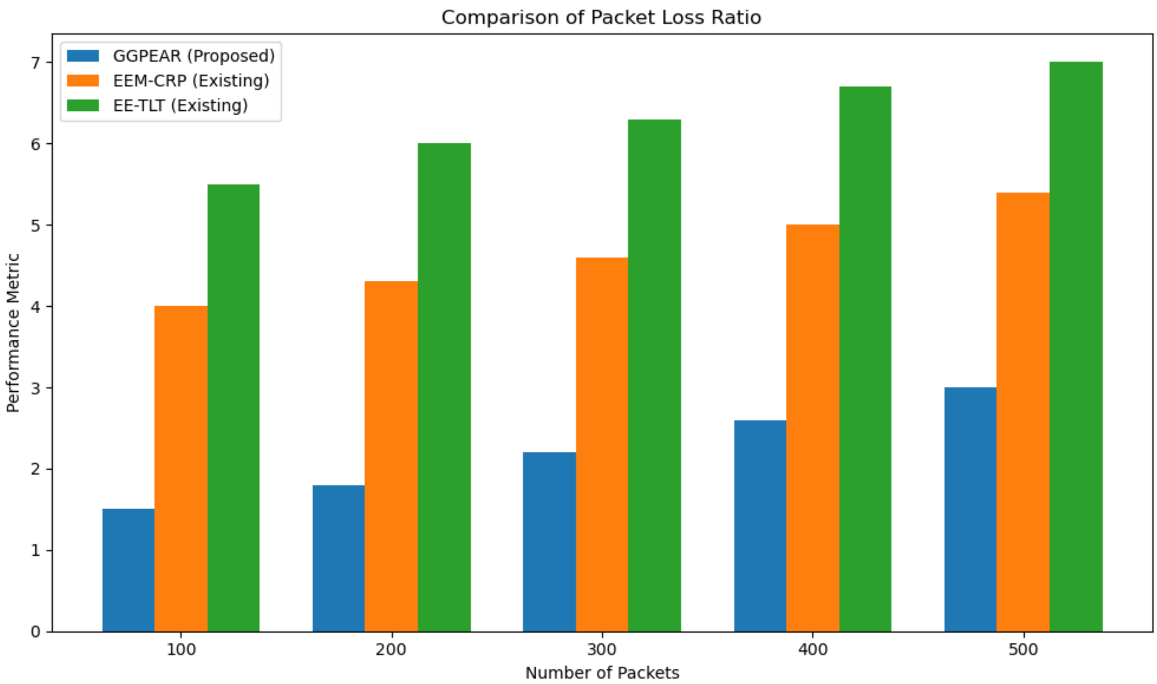

Figure 11: Comparison Chart on Packet Loss Ratio

The provided Figure 11 compares the packet loss ratio across three approaches: For varying numbers of packets (100, 200, 300, 400, and 500), GGPEAR (proposed) is compared with EEM-CRP (existing) and EE-TLT (existing). GGPEAR demonstrates the lowest packet loss ratio and hence proven to be more efficient in packet sending. On the other hand, the EE-TLT method is found to have the highest packet loss ratio for all values of packet counts, hence, comparatively worst performance. EEM-CRP operation is slightly good, attaining larger packet loss ratio value compared to GGPEAR, but less than EEM-TLT. Here compare packet loss reduction with GGPEAR over this set of environments and conclude that GGPEAR is substantially more reliable than MOBIKE.

4.6 Overhead

Overhead in networking, hence, includes all the extra efforts, time, bandwidth or energy needed to address the job of communication and to keep the system operational apart from the process of transferring data. When trying to reduce overheads, then companies are left with more time and space to work on becoming more efficient and making the best use of more limited resources that is a key feature of modern networks.

$$Overhead = \frac{Total\ Control\ Packets}{Total\ Data\ Packets} \text{ -------- (26)}$$

Table 7: Comparison table on Overhead

Number of Nodes	GGPEAR (Proposed)	EEM-CRP (Existing)	EE-TLT (Existing)
10	0.34	0.43	0.454
20	0.321	0.421	0.462
30	0.342	0.442	0.473
40	0.363	0.464	0.482
50	0.385	0.483	0.494
60	0.402	0.493	0.498

Table 7 performance of three methods is compared across various numbers of nodes (10, 20, 30, 40, 50, and 60) for the methods: GGPEAR (proposed), EEM-CRP (Existing), and EE-TLT (Existing). For each method, vary the number of nodes and plot out the values, which correspond to some measure of efficiency or resource usage. With an increase in the number of nodes, values for all three methods increase. The initial value of GGPEAR (Proposed) is set to 0.34 at 10 nodes which increases to 0.402 at 60 nodes. EEM-CRP and EE-TLT, by comparison, both display slightly higher values throughout all node counts, starting at 0.43 (0.43) at 10 nodes and climbing to 0.493 (0.498) at 60, and starting at 0.454 (0.45) at 10 and reaching 0. 498 (0.473) respectively. This indicates that GGPEAR increases less in the performance metric compared to existing methods with increasing number of nodes, meaning it is both more efficient and scalable.

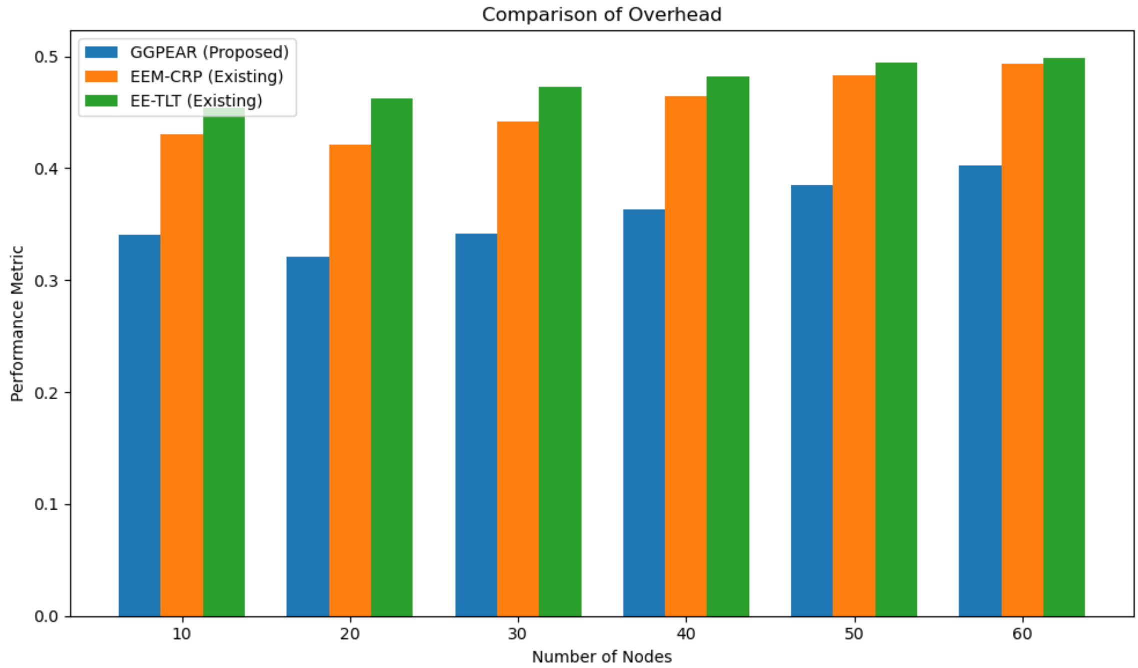

Figure 12: Comparison Chart on Packet Loss Ratio

The Figure 12 compares the overhead associated with three methods: For varying numbers of nodes, results of clustering's by GGPEAR (proposed), EEM-CRP (existing), and EE-TLT (existing) are given. What these figures illustrate is that GGPEAR shows relatively low overhead across all node counts, as expected. Both EEM-CRP and EE-TLT exhibited high overheads compared to other protocols, and EE-TLT consistently had highest overhead than the other approaches. GGPEAR has this advantage, in that it reduces the overhead compared with the existing methods and is more scalable and efficient.

5. Conclusion

In conclusion, Geographic Greedy Perimeter and Energy−Aware Routing (GGPEAR) protocol outperforms the existing protocols such as EEM−CRP and EE−TLT w.r.t. many performance criteria like throughput, energy expenditure, end−to end delay, packet delivery ratio (PDR), packet loss ratio (PLR) and overhead. Results demonstrate that GGPEAR has higher throughput regardless of node count, leading to better data transfer efficiency and lower power consumption than the existing methods, which is very important for prolonging network lifetime especially in large scale wireless sensor networks. Additionally, GGPEAR has lower end to end delay and thus suitable for time sensitive applications. Finally, demonstrate that in terms of packet delivery ratio, the protocol behaves much better than the case without any frequency declaration, successfully delivering packets with higher reliability and lower loss, with minimal loss, hence maintaining the network stable. GGPEAR's lower overhead also improves scalability and efficiency, which in effect leads to better resource utilization by GGPEAR and savings in unnecessary communication costs. GGPEAR is found to be better solution of wireless sensor networks over all making it a more scalable and efficient solution with a robust approach as it deals with large packet sizes and node counts while using less energy and improving the performance of the network. In particular, due to its ability to wick off wasted energy, be more intuitive (leading to lower delays) and improve data transmission efficiency, it has a great potential to be a candidate for modern wireless communication systems, especially in the cases where the network lifetime reliability are critical.

Reference:

[1] Ahmed, M., Malookani, R., ur Rehman, M., Naeem, N., & Parveen, S. (2018). EBRA: energy balanced routing algorithm for underwater wireless sensor network. *IJCSNS*, *18*(12), 49.

[2] Amgoth, T., & Jana, P. K. (2015). Energy-aware routing algorithm for wireless sensor networks. *Computers & Electrical Engineering*, *41*, 357-367.

[3] Aregawi, S., & Haile, C. (2018). IMPROVING GEOGRAPHICAL ENERGY AWARE ROUTING PROTOCOL TO ENHANCE LIFETIME OF WSNs.

[4] Bairagi, P. P., Dutta, M., & Babulal, K. S. (2024). An energy-efficient protocol based on recursive geographic forwarding mechanisms for improving routing performance in WSN. *IETE Journal of Research*, *70*(3), 2212-2224.

[5] Bousnina, S., Cesana, M., Ortín, J., Delgado, C., Gállego, J. R., & Canales, M. (2017, March). A greedy approach for resource allocation in virtual sensor networks. In *2017 Wireless Days* (pp. 15-20). IEEE.

[6] Chaurasia, S., Kumar, K., & Kumar, N. (2023). EEM-CRP: Energy Efficient Meta-heuristic Cluster based Routing Protocol for WSNs. *IEEE Sensors Journal*.

[7] Guleria, K., Kumar, S., & Verma, A. K. (2019). Energy aware location based routing protocols in wireless sensor networks. *World Scientific News*, *124*(2), 326-333.

[8] Hadi, K. (2017). Analyses of Energy-Aware Geographic Routing Protocols for Wireless Sensor Networks. *Lecture Notes on Information Theory Vol*, *5*(1).

[9] Hadir, A., Zine-Dine, K., Bakhouya, M., El Kafi, J., & El Ouadghiri, D. (2017). Performance evaluation of DV-Hop localization algorithm for geographical routing in wireless sensor networks. *Procedia computer science*, *113*, 261-266.

[10] Hao, S., Hong, Y., & He, Y. (2022). An energy-efficient routing algorithm based on greedy strategy for energy harvesting wireless sensor networks. *Sensors*, *22*(4), 1645.

[11] Hung, K. S., & Lui, K. S. (2010). Perimeter coverage scheduling in wireless sensor networks using sensors with a single continuous cover range. *EURASIP Journal on Wireless Communications and Networking*, *2010*, 1-17.

[12] Imoize, A. L., Ajibola, O. A., Oyedare, T. R., Ogbebor, J. O., & Ajose, S. O. (2021). Development of an energy-efficient wireless sensor network model for perimeter surveillance. *International Journal of Electrical Engineering and Applied Sciences (IJEEAS)*, *4*(1).

[13] Jibreel, F., Tuyishimire, E., & Daabo, M. I. (2022). An enhanced heterogeneous gateway-based energy-aware multi-hop routing protocol for wireless sensor networks. *Information*, *13*(4), 166.

[14] Jones, A. (2020). Survey: energy efficient protocols using radio scheduling in wireless sensor network. *International Journal of Electrical & Computer Engineering (2088-8708)*, *10*(2).

[15] Karkazis, P., Leligou, H. C., Orphanoudakis, T., & Zahariadis, T. (2012, July). Geographical routing in wireless sensor networks. In *2012 International Conference on Telecommunications and Multimedia (TEMU)* (pp. 19-24). IEEE.

[16] Khan, T., Ahmad, I., Aman, W., Azam, I., Khan, Z. A., Qasim, U., ... & Javaid, N. (2016, March). Clustering depth based routing for underwater wireless sensor networks. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)* (pp. 506-515). IEEE.

[17] Khedr, A. M., Osamy, W., & Agrawal, D. P. (2009). Perimeter discovery in wireless sensor networks. *Journal of Parallel and Distributed Computing*, *69*(11), 922-929.

[18] Kheirabadi, M. T., & Mohamad, M. M. (2013). Greedy routing in underwater acoustic sensor networks: a survey. *International Journal of Distributed Sensor Networks*, *9*(7), 701834.

[19] Kumar, A., Shwe, H. Y., Wong, K. J., & Chong, P. H. (2017). Location-based routing protocols for wireless sensor networks: A survey. *Wireless Sensor Network*, *9*(1), 25-72.

[20] Kumar, S., Soni, S. K., Randheer, & Priyadarshi, R. (2020). Performance analysis of novel energy aware routing in wireless sensor network. In *Nanoelectronics, Circuits and Communication Systems: Proceeding of NCCS 2018* (pp. 503-511). Springer Singapore.

[21] Lin, D., Wang, Q., Min, W., Xu, J., & Zhang, Z. (2020). A survey on energy-efficient strategies in static wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *17*(1), 1-48.

[22] Liu, J., Shen, H., Yu, L., Narman, H. S., Zhai, J., Hallstrom, J. O., & He, Y. (2017). Characterizing data deliverability of greedy routing in wireless sensor networks. *IEEE Transactions on Mobile Computing*, *17*(3), 543-559.

[23] Mayasala, P., & Krishna, S. M. (2022). Correlation Distance Based Greedy Perimeter Stateless Routing Algorithm for Wireless Sensor Networks. *International Journal of Computer Science & Network Security*, *22*(1), 139-148.

[24] Meng, X., Shi, X., Wang, Z., Wu, S., & Li, C. (2016). A grid-based reliable routing protocol for wireless sensor networks with randomly distributed clusters. *Ad Hoc Networks*, *51*, 47-61.

[25] Mittal, M., & Iwendi, C. (2019). A survey on energy-aware wireless sensor routing protocols. *EAI Endorsed Transactions on Energy Web*, *6*(24).

[26] Nabavi, S. R., Eraghi, N. O., & Torkestani, J. A. (2021). WSN routing protocol using a multiobjective greedy approach. *Wireless Communications and Mobile Computing*, *2021*(1), 6664669.

[27] Nakas, C., Kandris, D., & Visvardis, G. (2020). Energy efficient routing in wireless sensor networks: A comprehensive survey. *Algorithms*, *13*(3), 72.

[28] Nguyen, K. V., Nguyen, C. H., Le Nguyen, P., Van Do, T., & Chlamtac, I. (2021). Energy-efficient routing in the proximity of a complicated hole in wireless sensor networks. *Wireless Networks*, *27*(4), 3073-3089.

[29] Pandith, M. M., Ramaswamy, N. K., Srikantaswamy, M., & Ramaswamy, R. K. (2022). A comprehensive review of geographic routing protocols in wireless sensor network. *Information Dynamics and Applications*, *1*(1), 14-25.

[30] Shu, L., Zhang, Y., Yang, L. T., Wang, Y., Hauswirth, M., & Xiong, N. (2010). TPGF: geographic routing in wireless multimedia sensor networks. *Telecommunication Systems*, *44*, 79-95.

[31] Tan, N. D., & Nguyen, V. H. (2023). EE-TLT: Energy-efficient routing protocol using two-level tree-based clustering in wireless sensor network. *Journal of Communications and Networks*, *25*(6), 734-749.

[32] Tang, L., & Lu, Z. (2020). DS evidence theory-based energy balanced routing algorithm for network lifetime enhancement in WSN-assisted IOT. *Algorithms*, *13*(6), 152.
[33] Vahabi, S., Lahabi, M., Eslaminejad, M., & Dashti, S. E. (2019). Geographic and clustering routing for energy saving in wireless sensor network with pair of node groups. *Journal of Communication Engineering*, *8*(1), 146-157.
[34] Vamsi, P. R., & Kant, K. (2014). An improved trusted greedy perimeter stateless routing for wireless sensor networks. *International Journal of Computer Network and Information Security*, *6*(11), 13.
[35] Venkatasubramanian, S. (2021). Correlation distance based greedy perimeter stateless routing algorithm for wireless sensor networks. *International Journal of Advanced Networking and Applications*, *13*(3), 4962-4970.
[36] WANG, Y. J., YANG, J. Z., & MAN, D. P. (2012). Research on security routing protocol based on geography and energy aware. *The Journal of China Universities of Posts and Telecommunications*, *19*, 142-147.
[37] Yan, J., & Qi, B. (2021). CARA: a congestion-aware routing algorithm for wireless sensor networks. *Algorithms*, *14*(7), 199.
[38] Yang, J., Fei, Z., & Shen, J. (2015). Hole detection and shape-free representation and double landmarks based geographic routing in wireless sensor networks. *Digital Communications and Networks*, *1*(1), 75-83.
[39] Yue, K. (2024). Application of EEUC-based inter aircraft ultraviolet communication network algorithm in energy consumption optimization of drone swarm. *Energy Informatics*, *7*(1), 27.
[40] Zagrouba, R., & Kardi, A. (2021). Comparative study of energy efficient routing techniques in wireless sensor networks. *Information*, *12*(1), 42.