



SRG: Energy-Efficient Localized Routing to Bypass Void in Wireless Sensor Networks

Saurabh Singh, Sarvpal Singh and Jay Prakash

Dept of computer Science, MMMUT, Gorakhpur, India

saurabh9singh9@gmail.com, singh_sarvpal@yahoo.co.in, jpr_1998@yahoo.co.in

KEYWORDS ABSTRACT

minimum local situation; reverse; gradient; geographic routing; void bypassing

The Shift Reverse Gradient (SRG) approach presents a void-size-independent hole bypassing scheme for wireless sensor networks. It does not require establishing any chain or hierarchical tree structure to ensure reliable delivery. The proposed Shift Reverse Gradient (SRG) offers an energy-efficient solution with minimal overhead and consumes minimum power. It has a communication overhead equivalent to greedy forwarding. We have shown through the simulation that SRG energy consumption is minimal and is not much affected by an increase in the void size like other existing void bypassing methods.

1. Introduction

In places like a war zone, a volcano, or a forest, building a well-organised network is practically impossible. The network launched in these areas is prone to frequent failure due to extreme situations. Wireless sensor nodes are used to establish a network in such an area. These sensor nodes are distributed randomly. The sensor nodes have limited energy and stop working once they lose their power. Because of the extreme locations and randomized allocation, it is very energy-consuming to accurately collect and store the entire network configuration at each node. Due to this reason, the wireless sensor nodes with limited energy need more energy-efficient routing protocols than the popular network routing protocols (Akkaya and Younis, 2005). The wireless sensor protocols can be classified into two categories: (i) graph-based routing protocols and (ii) non-graph-based routing protocols. Graph-based routing protocols like Greedy Perimeter Stateless Routing (GPSR) guarantee the packet delivery but demand a large amount of energy and memory space (Karp and Kung, 2000). Thus they are not scalable because of the high energy and memory requirements. On the other hand, non-graph-based protocols do not require much energy and memory but generate longer routing paths. The non-graph-based routing is divided into two classes:

Saurabh Singh, Sarvpal Singh and Jay Prakash

SRG: Energy-Efficient Localized Routing to Bypass Void in Wireless Sensor Networks



(i) geographic routing protocols and (ii) non-geographic routing protocols. The non-geographic routing protocols have a significant localization-related overhead that makes them complicated. Geographic protocols assume that each sensor node has a GPS to extract its geographical coordinates (Savvides and Strivastava, 2003; Bahi et al., 2008). These approaches have the least routing overhead and they are easier to implement. Geographic routing techniques are based on greedy forwarding and use different boundary traversal techniques to bypass the void region (Jiang et al., 2008). The node where a situation of local minimum occurs and greedy forwarding fails (Boukerche et al., 2011). On this node, the void handling technique is applied. The Boundhole approach uses the right-hand rule for boundary traversal but fails to find an existing path in case of false boundary detection (Liu and Feng, 2008). The GAR algorithm resolves the false boundary detection problem using a rolling ball with a transmission range($R/2$) that is half of the actual transmission range (R) (Liu and Feng, 2008). This approach increases the number of nodes in the routing path. A large number of nodes makes it less energy-efficient and degrades its performance in terms of delay. The Curved Stick approach uses a curved stick to resolve the false boundary detection problem and reduces the number of routing nodes (Mostefaoui et al., 2013). All these approaches use different boundary traversal techniques that intensify the depletion of the energy of boundary nodes and increase the void size. We propose a new routing approach that efficiently bypasses the void region and finds an alternate path. In this approach, the packet shifts from the boundary node where the situation of local minimum occurs. The node where the local minimum occurs sends the packets to a distant node from the void region. After shifting from the node, the packet traverses by the side of the void. For instance, in Fig1, while forwarding from source node N_s to the destination node N_d the “local minimal situation” occurs at node N_2 is a stuck node. The curved stick protocols follow the path $N_s, N_1, N_2, N_3, N_8, N_9, N_{10}, N_{12}, N_{13}, N_d$. In the shift of reverse gradient (SRG) approach, we apply a SRG at the stuck node, in which we shift away from the stuck node by the two consecutive reverse gradients. In Fig. 1 On stuck node N_2 , there are two consecutive SRGs: one at node N_2 (that precedes node N_3) and another at node N_3 (that precedes N_4). From node N_4 , packets are sent using greedy forwarding. Thus, the routing path using the SRG The approach is as follows: $N_s, N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_d$. Here we can observe that the number of routing nodes in the SRG approach is less than the number of routing nodes visited by the curved stick approach. This approach can resolve the false boundary detection problem by selecting an alternative path. As shown in Fig. 1, the curved stick path length is longer than the SRG path. The main contribution of the proposed algorithm is as follows:

- The consumption of energy in the SRG approach is at least 10% lower in comparison to the existing algorithms and is least affected by void size.
- For moderate and large-sized void zones, the suggested SRG generates a minimum end-to-end delay in comparison to other existing algorithms.
- The SRG approach can simultaneously route through the dual-path around the void. It increases its throughput and reliability in comparison to the existing algorithms.

This paper has seven sections: Section 2 reviews the literature that has addressed the problem of the void region in the wireless sensor network. Section 3 defines the network model and preliminary knowledge required to understand the terminology of the SRG approach. Section 4 describes the SRG approach with detailed flow chart. The conceptual analysis of the SRG approach is presented in Section 5. Section 6 compares the SRG algorithm performance with the popular approaches for void handling. This paper concludes in Section 7.



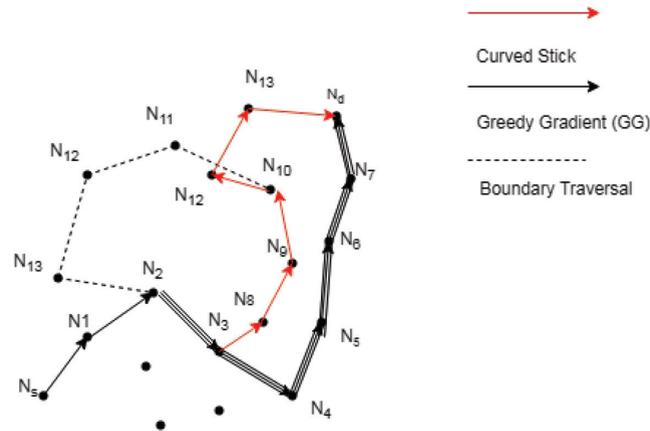


Figure 1. Different strategies routing path around void region

2. Related Work

Many wireless routing protocols use different strategies to avoid the void area in wireless networks. The geographical greedy forwarding approach performance is better because of its simplistic approach and the small overhead associated. These geographic routing protocols face the local minimum problem while passing the void regions. Void handling strategies are divided into two distinct classes (i) Graph-based and (ii) Non-graph based. Graph-based void handling strategies: The Graph-based approaches such as the Boundhole, Greedy Perimeter Stateless Routing (GPSR), Greedy Other Adaptive Face Routing (GOAFR), create a planar graph using Gabriel Graph or RNG techniques (Kuhn et al., 2008; Gabriel and Sokal, 1969; Toussaint, 1980). These techniques apply right-hand rule (or left hand rule) to traverse the boundary nodes around the void region, which is efficient in void handling. However, these techniques require a lot of computation and memory for graph creation. This graph creation consumes much energy as the number of nodes increases. These high memory and energy requirements make this strategy less scalable.

Non-graph-based void handling strategies: Non-graph-based strategies identify the stuck node around the void then different approaches opt for various techniques to get out of the stuck node. In Void Avoidance Algorithm (VAA) cost of the node is equal to its Euclidian distance from the destination (Joshi and Kim, 2009). A packet traverse from a node with a high-cost value to a node with a low-cost value. The cost of the nodes around the void increases to a value greater than their neighbour nodes. Hence the void region does not come in the routing path. The VAA algorithm is better than DUA as it does not require degrading the cost of a node (Chen et al., 2006). PAGER algorithm uses shadow node cost increment to avoid the void region (Arad and Shavitt, 2008). The shadow nodes are the node where the local minimum problem occurs inside the concave void.

Geographic and Energy Aware Routing (GEAR) protocol uses recursive geographic forwarding to the sub-regions and floods the final sub-region consequently (Hou and Zhang, 2018). The packets forward toward the target region. The target region divides into sub-regions. Among these sub-regions, packets forward toward the selected sub-region. The division process of the target region continues until a limit of hop-bound reaches. Then the packet floods the target sub-region. The flooding involved

makes it less energy-efficient. In Intermediate Node Forwarding (INF), negative acknowledgment is received by the source node if a packet fails to reach the destination (De Couto and Morris, 2001). On receiving the negative acknowledgment, the source node finds an intermediate node between the source and goal. The packets reach the destination through this intermediate node. This approach has a longer path and extra delay.

EDGR uses two anchor nodes around the void region (Huang et al., 2017). Random shift fixes these anchor nodes on two sides of the void. The packets are sent via these two anchor nodes simultaneously. It enhances the throughput of the network. SPEED protocol uses the backpressure technique to revert and find the alternate path (He et al., 2005; Kim et al., 2016). This alternate path may revert to the void and waste energy. In BVR-VCM, a virtual circle is drawn with its center inside the void (Zhang and Dong, 2015). The boundary nodes around the void projection on the circumference of this virtual circle are called virtual coordinates. This process is termed virtual coordinate mapping. These virtual coordinates are shared among the boundary nodes and used to bypass the void region. The high computation involved causes a waste of energy in the network. The GRACO is an ant colony optimization-based technique to avoid void areas (Rekik et al., 2018). In it, the ACO technique is used at a stuck node. It uses forward ant (Fants) to discover the network configuration and backward ant (Bants) to mark the recovery path from bypass. It has no overhead involved and is scalable for large size wireless networks. REACT proposed by Lima has a long-range sink (Lima et al., 2017). This sink has enough energy to send the signal to all the nodes. The nodes use received signal strength as a measure to apply greedy forwarding. A node that receives the stronger signal is closer to the sink node than the node which receives the weaker signal. RSSR works on the received signal strength indicator and has two version, Namely, RSSI election and RSSI selection (de Oliveira et al., 2015). In RSSI election, the next node receiving the strongest signal responds first and becomes the leader. Every node maintains a forward table that stores its distance from the sink. The leader node then broadcasts that message to its neighbours that it won the election and will be the next node. The other withdraws on receiving the message.

Table 1. Comparison chart of various void handling protocols

Technique	Overhead	Complexity	Scalability	Optimal Path	Guaranteed delivery
GPSR	High	Medium	No	No	Yes
GAR	High	Medium	No	No	Yes
Boundhole	Medium	High	Yes	No	Sometimes
Curved Stick	High	High	No	No	No
EDGR	High	High	Yes	Yes	Yes
VAA	High	High	No	Yes	Yes
Intermediate Node Forwarding	High	High	No	Yes	No
RSSI	High	Medium	No	Yes	Sometimes
BVR-VCM	High	High	No	Yes	Sometimes
GRACO	Medium	Medium	Yes	Yes	sometimes
Extended CPEQ	Medium	Medium	Yes	No	Sometimes



In RSSI selection, every node broadcasts a beacon message to its neighbours to get their distance. If it gets a query for itself, then forwards it to the neighbour closest to the sink after an interval. Q-Learning Inspired Hole Bypassing (QIH) approach uses Q-learning to bypass the void (Le Nguyen et al., 2021). The void information broadcast inside the convex hull is formed around the void region. It has a reward function with three parameters residual energy, distance from the void, and distance from the destination. It makes it loop-free, increases the lifetime, and makes it more energy-efficient. A fault-tolerant extension of the cluster-based CPEQ protocol has been proposed and it is a cluster-based, periodic, event-driven, and query-based protocol (Yilmaz et al., 2014). By reconstructing the corrupted links and bypassing the void inside the cluster by intra-cluster bypass. The holes spread over a group of clusters are bypassed by inter-cluster bypass. This protocol has used the intra-cluster and inter-cluster bypass techniques to make it more fault-tolerant and avoid the cost of localization. The fuzzy logic-based FLGR routing protocol uses the candidate node region to discard the void node (Hu et al., 2019). Table 1 represents the comparison of various void handling techniques and their different parameters such as complexity, scalability, overhead, and reliability.

3. Network Model and Preliminaries

A network is a group of randomly distributed wireless nodes aware of their geographic location. GPS installed inside the sensors gives the geographic coordinates. The network represents a $G(N, E)$ where the N represents the set of nodes and E stands for a link between the nodes. The node set N is a collection of all the n nodes distributed in the region as $N = N_1, N_2, \dots, N_n$. All the nodes are homogeneous and static with transmission range R . If the Euclidian distance between two nodes is less than the transmission range of the nodes, then only a link E exists between the nodes as $E = E_{ij} / d(N_i, N_j) \leq R$. The source node is aware of the location of the goal. The intermediate node gets the location of the goal via the packet it receives from the source.

In this assumption, the packet is forwarded from the source to the destination using the greedy forward approach. If the “local minimum situation” occurs around the void, it switches to the SRG approach. Once the void region gets over, it returns to the Greedy Forward approach. Thus, we focus on identifying the boundary node, where the “local minimum situation” occurs, to bypass the void region and find an efficient communication path.

Definition 1 (Void Region): The cyclic sequence of the boundary nodes $N_p, N_{i+1}, N_{i+2}, \dots, N_{i+j}, N_i$ so that this set of nodes form a minimum region around the void such that no node lies inside the void area.

Definition 2 (Stuck node): While traversing using greedy forward approach from source node N_s to the destination node N_d if there comes a node N_a in the path with coordinate (X_a, Y_a) with neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates (X_{b_k}, Y_{b_k}) for $k=1, 2, \dots, j$ such that its Euclidian distance from destination is less than its neighbour node is termed as stuck node.

Fig. 2 shows a void region between the source node N_s and the goal N_d . Node N_a on the boundary of the void where a situation of local minimum occurs. N_a is the stuck node on the void region.

$$d(N_a, N_d) \leq d(N_{b_k}, N_d) \quad \text{for } k = 1, 2, \dots, j \quad (1)$$



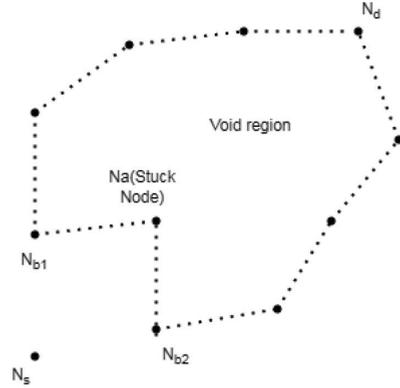


Figure 2: Void region and stuck node

Definition 3 (X-Gradient-Distance): Let N_a be a node with coordinate (x_a, y_a) . The node with its range R has neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates (x_{b_k}, y_{b_k}) for $k=1, 2, \dots, j$ then the X-Gradient-Distances between the node N_a and its neighbours N_{b_k} is

$$d(N_a, N_{b_k}) = |x_{b_k} - x_a|, \text{ for } k = 1, 2, \dots, j \quad (2)$$

Fig. 3 shows the X-gradient-distance between the node N_a and its neighbour N_{b_k} .

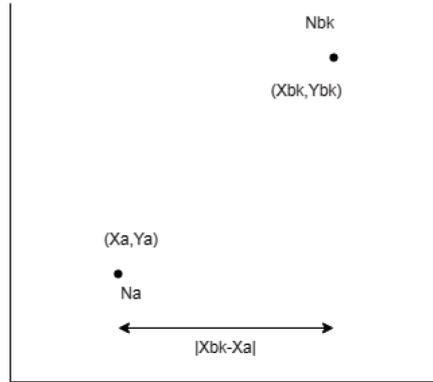


Figure 3. X-gradient-distance

Definition 4 (Y-Gradient-Distance): Let N_a be a node with coordinate (x_a, y_a) . The node with its range R has neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates (x_{b_k}, y_{b_k}) for $k=1, 2, \dots, j$ then the Y-Gradient-Distances between the node N_a and its neighbours is N_{b_k} is

$$d(N_a, N_{b_k}) = |y_{b_k} - y_a| \text{ for } k = 1, 2, \dots, j \quad (3)$$

Fig. 4 shows the shows the Y-gradient-distance between the node N_a and its neighbour N_{b_k} .

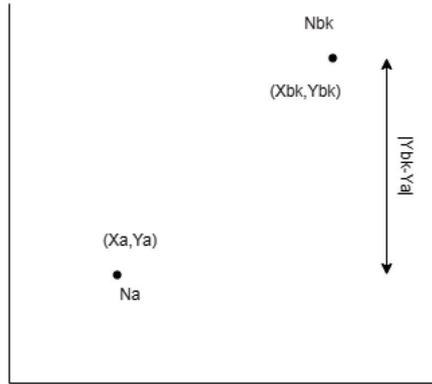


Figure 4. Y-gradient-distance

Definition 5 (Reverse-X-Gradient-Neighbour): Let N_a be a node with coordinate (x_a, y_a) . The node with its range R has neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates values (x_{b_k}, y_{b_k}) for $k=1, 2, \dots, j$ then the X-Gradient Distances between the node N_a and its neighbours is $d = |x_{b_k} - x_a|$ for $k=1, 2, \dots, j$. The node with maximum X-Gradient-Distance is termed as the N_a , Reverse-X-Gradient-Neighbour with

$$d = \max |x_{b_k} - x_a| \quad \text{for } k = 1, 2, \dots, j \quad (4)$$

In Fig. 5, the node N_a has three neighbours $N_{b_1}, N_{b_2}, N_{b_3}$. Among these neighbours the node N_{b_3} has the maximum X-gradient-distance. So, the Node N_{b_3} is Reverse-X-Gradient-Neighbour of the node N_a .

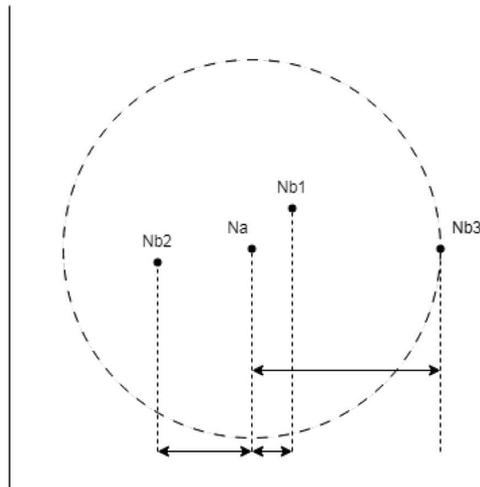


Figure 5. Reverse-X-gradient-neighbours

Definition 6 (Reverse-Y-Gradient-Neighbour): Let N_a be a node with coordinate (x_a, y_a) . The node with its range R has neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates (x_{b_k}, y_{b_k}) for $k=1, 2, \dots, j$ then the Y-Gradient-distances between the node N_a and its neighbours is $d = |y_{b_k} - y_a|$ for $k=1, 2, \dots, j$. The node with maximum Y-Gradient-Distance is termed as the N_a Reverse-Y-Gradient-Neighbour with

$$d = \max |y_{b_k} - y_a| \quad \text{for } k = 1, 2, \dots, j \quad (5)$$

In Fig. 6, the node N_a has three neighbour $N_{b_1}, N_{b_2}, N_{b_3}$. Among these neighbours the node N_{b_1} has the maximum Y-gradient-distance. So, the Node N_{b_1} is Reverse-Y-Gradient-Neighbour of the node N_a .

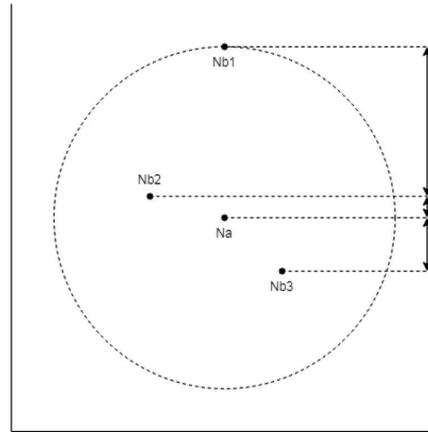


Figure 6. Reverse-Y-gradient-neighbours

3.1 Reverse Gradient Shift

In this approach, we do not use any extra overhead other than the information required by the greedy forward. Once the SRG last, it returns to the Greedy Forward approach. The Reverse-Gradient-Shift is not a backpropagation technique. It helps the packet come out of the “local minimum situation” and explore a new path. In the SRG approach, the node where the “local minimum situation” occurs performs a reverse gradient shift by traversing to a node at a two-hop distance away from the stuck node. There are two types of Reverse-Gradients-shifts: (i) Reverse-XX-Gradient-Shift (ii) Reverse-YY-Gradient-Shift. It maintains the simplistic requirements of the greedy forward approach. It tries to overcome this dead-end-like situation through Reverse-Gradient-Shift. To understand the working, we start with the following definition.

Definition 7 (Reverse-XX-Gradient-Shift): The process of shifting from the node N_{stuck} by two consecutive Reverse- X-Gradient-Neighbours is called Reverse-XX-Gradient-Shift. Let N_2 be the node where a “local minimum situation” has occurred and it becomes the stuck node. Starting from node N_{stuck} . In Fig. 7, first, at the stuck node N_2 , it finds out the first Reverse-X- Gradient-Neighbour that is N_1 , then again at node N_1 , it finds the second Reverse X -Gradient Neighbour that is N_3 . While selecting the second Reverse-X-Gradient-Neighbour, it considers all the neighbour nodes except

the stuck node. It is called Reverse-XX-Gradient-Shift. Algorithm-1 summarizes the Reverse-XX Gradient Shift.

Definition 8 (Reverse-YY-Gradient-Shift): The process of shifting from the node N_{stuck} by two consecutive Reverse-Y-Gradient-Neighbours is called Reverse-YY-Gradient-Shift. Let N_2 be node where local minimum situation occurs then N_2 is termed as the stuck node N_{stuck} . In Fig. 8, at the stuck node N_2 first we find out the first Reverse-Y- Gradient-Neighbour i.e. N_6 then again at node N_6 we find the second Reverse-Y-Gradient- Neighbour i.e. N_7 . While selecting the second Reverse Y-Gradient-Neighbour all the neighbour are considered except the N_{stuck} . It is called Reverse-YY-Gradient-Shift. Algorithm 2 summarizes the Reverse-YY -Gradient-Shift.

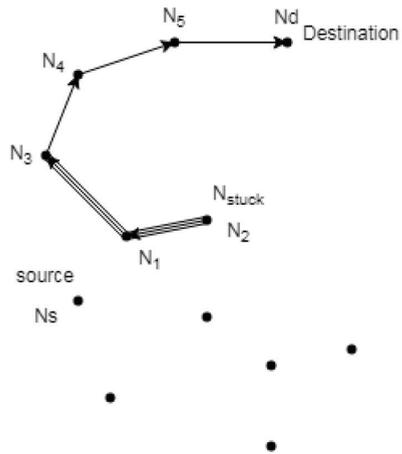


Figure 7. Reverse-XX-gradient-shift

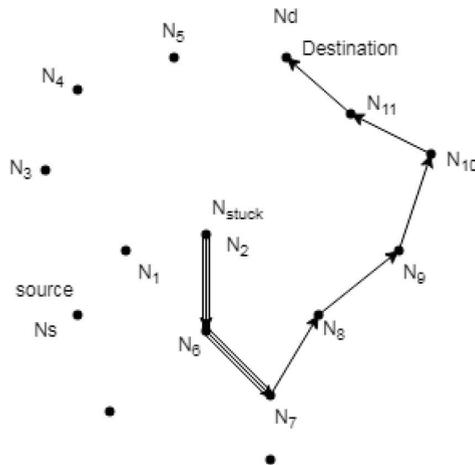


Figure 8. Reverse-YY-gradient-shift

4. Proposed Shift Reverse Gradient Approach

In this approach, we find out the node in the path from source node N_s to the destination node N_d where the local minimum problem occurs. In this approach, we identify this node as a stuck node. At the stuck node, we initiate the Shift Reverse Gradient approach. The Shift Reverse Gradient approach only needs the location of the neighbours of a node. It does not require any other information overhead. It does not repeatedly use boundary nodes to get out of the stuck node. In this way, boundary nodes around the void region do not lose their energy rapidly. Hence the void size does not increase at a fast pace.

Algorithm 1. Reverse-XX-Gradient Shift Algorithm

Set the node with the local minimum situation as N_{stuck} .

Find the Reverse-X-Gradient-Neighbour of node N_{stuck} as the next node N_i .

Find the Reverse-X-Gradient-Neighbour of node N_i excluding the node N_{stuck} . Set Greedy-X-Gradient-Neighbour as the next node N_k .

Switch back to the simple Greedy Forward at node N_k .

Algorithm 2. Reverse-YY-Gradient-Shift Algorithm

Set the node with the local minimum situation as N_{stuck} .

Find the Reverse-Y-Gradient Neighbour of node as N_{stuck} the next node N_i .

Find the Reverse-Y Gradient Neighbour of node N_i excluding the node N_{stuck} . Set Reverse-Y – Gradient Neighbour as the next node N_k .

Switch back to the simple Greedy Forward at node N_k .

4.1 Shift Reverse Gradient Routing Algorithm

Algorithm 3. Shift Reverse Gradient Algorithm

Let N_s is source node and N_d is the destination node. Initially $N_{stuck} = \emptyset$

If at some intermediate node N_i local minimum situation occur then $N_{stuck} = N_i$

if $N_{stuck} = \emptyset$ **then**

 Select the next hop using the Greedy Forward

else

 Find out the Reverse X-Gradient-Neighbour of node N_{stuck} and set it as the next node N_j . Find the Reverse X-Gradient-Neighbour of the node N_j excluding the node N_{stuck} . Set this node as the next node N_k . On node N_k switch back to the Greedy Forward.

end if

```

if Next Node  $\neq N_d$  then
  if Next node =  $N_{stuck}$  then
    Select the next node using Reverse-Y-Gradient- Neighbour of the node  $N_{stuck}$  and set it the
    next node  $N_l$ . Find the Reverse-Y-Gradient-Neighbour of the node  $N_l$  excluding the node  $N_{stuck}$ . Set
    this node as the next node  $N_m$ . On node  $N_m$  switch back to the Greedy Forward.

    if Next node =  $N_{stuck}$  then
      Acknowledge the source node no route is available

    else
      Select the next node using the greedy forward

    end if
  end if
end if
else
  Show routing nodes from source to destination
end if

```

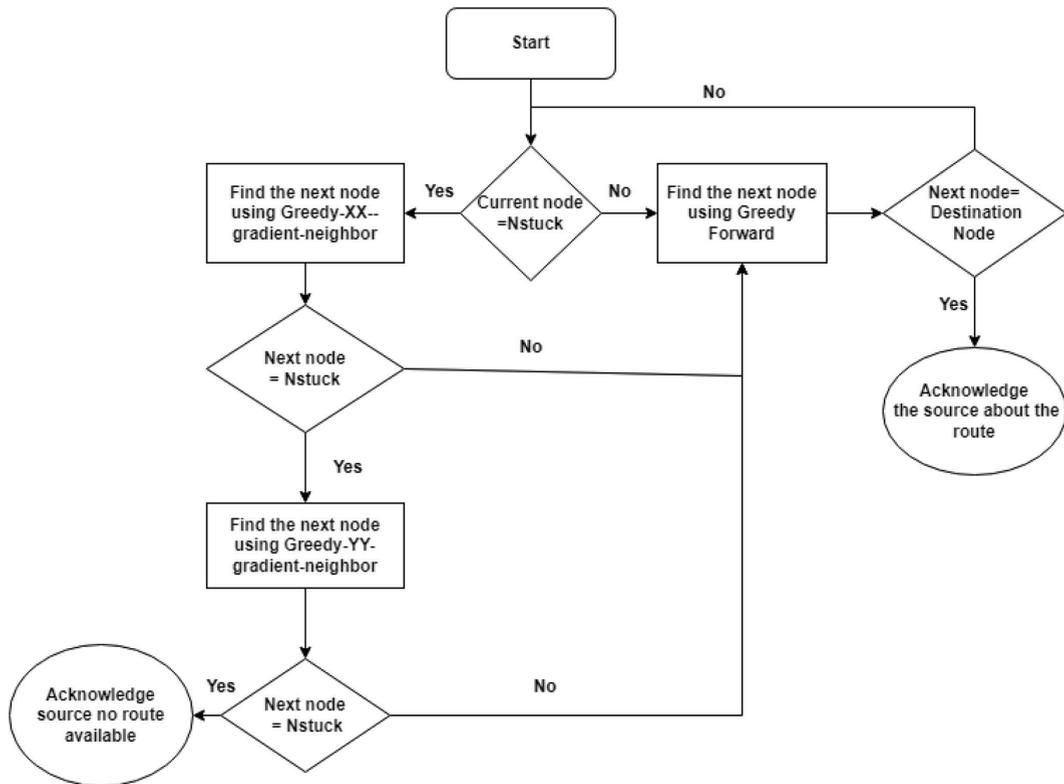


Figure 9. Flow chart of shift reverse gradient approach

This algorithm is an advancement of the greedy forward approach that is considered the most efficient technique in wireless routing [5]. This approach can deal with the local minimum problem efficiently. By Reverse Gradient Shift, once it bypasses the void, it returns to greedy forwarding. The SRG Algorithm has the following three phases: (i) Greedy Forward Phase (ii) Reverse-Gradient-Forward Phase (iii) Termination Phase

4.2 Greedy Forward Phase

In the Greedy Forward Phase, the packet proceeds towards the destination using the Greedy Forward method. A node forwards the packet to a neighbour node closest to the destination. In a situation, a packet gets stuck at some node having no neighbour node closer to the destination node, termed a stuck node. At this node Reverse-Gradient N_{stuck} -Forward is applied.

4.3 Reverse-Gradient-Forward Phase

In this phase, a stuck node N_i with coordinate (x_i, y_i) with neighbour nodes N_j and N_k which have coordinates (x_j, y_j) and (x_k, y_k) , respectively. The X- Gradient- Distance of node N_i with node N_j is $d(x_j - x_i)$. Similarly, the X-Gradient-Distance between the node N_i and node N_k is $d(x_k - x_i)$. At the stuck node N_{stuck} , we initiate Reverse XX-Gradient-Shift. In Reverse XX-Gradient-Shift, the node N_{stuck} finds its neighbour's X- Gradient Distance. Among the neighbour node of N_{stuck} , it selects the node with the maximum X-Gradient- Distance (Reverse X-gradient-Neighbour) as the next node. At the next node, it again finds the Reverse-X-Gradient- Neighbour from its neighbours except the stuck node. Thus, at N_{stuck} the Shift Reverse Gradient is applied over the two consecutive nodes once the local minimum problem occurs. This is called the Reverse-XX-Gradient-Shift. After Reverse-XX-Gradient-Shift, it returns to the Greedy Forward approach. Fig. 12 shows the routing path by applying the Reverse-XX Gradient-Shift at the stuck node. If after Reverse-XX-Gradient- Shift, the packet returns to the same stuck node, it attempts the Reverse-YY-Gradient-Shift at the stuck node. Fig. 13, shows the routing path on applying the Reverse-YY-Gradient-shift at stuck node. It again switches back to the Greedy Forward approach after the shift. Fig. 9 represents the flow chart of the Shift Reverse Gradient algorithm.

4.4 Termination Phase

In this phase, if the packet reaches the destination, a path exists between the source and destination. In case both the Reverse-XX-Gradient-Shift and Reverse-YY-Gradient-Shift packets return to the N_{stuck} , the process terminates as no path exists.

5. Proof of Correctness

Theorem 1: The Shift Reverse Gradient approach always finds a path if it exists and guarantees packet delivery.

Proof: The Shift Reverse Gradient approach uses the greedy forwarding approach till the local minimum problem occurs. On the occurrence of a local minimum situation, the Reverse-XX-Gradient-Shift applies on the stuck node. At the end of the Reverse-XX-Gradient-Shift, it switches

back to the greedy forward approach. Now depending on the next node position and stuck node position following two conditions may occur.

Case 1: If (Next Node $\neq N_{stuck}$) greedy approach can be applied further to find the next node till the destination node comes.

Case 2: If (Next Node = N_{stuck}) Reverse-YY-Gradient-Shift is applied again over the current node. If again the next node reaches the stuck node N_{stuck} the process is terminated. The sequence of Reverse-YY-Gradient Shift and Reverse-XX Gradient-Shift can be applied in any order but only once on N_{stuck} .

Lemma 1: A reverse gradient neighbour of a stuck node does not always lie on the boundary of the void.

Proof: Let N_{stuck} is a stuck node with coordinates (x_a, y_a) that lies on the boundary of the void region. The node N_{stuck} has neighbour nodes $(N_{b_1}, N_{b_2}, N_{b_3}, \dots, N_{b_j})$ with coordinates (x_{b_k}, y_{b_k}) for $(k=1, 2, \dots, j)$ then for the node N_{stuck} the Reverse Y-Gradient Neighbour has

$$d = \max |y_{b_k} - y_a| \quad \text{for } k = 1, 2, \dots, j \quad (6)$$

Similarly, the Reverse X-gradient Neighbour has

$$d = \max (|x_{b_k} - x_a| \quad \text{for } k = 1, 2, \dots, j) \quad (7)$$

In Fig. 10, this condition is satisfied by the nodes that exist away from the boundary node as N_{GY} (Reverse-Y Gradient Neighbour). It is more distant than N_{b_1} as per Y-gradient that exists on the boundary. Similarly, the node N_{GX} (Reverse-X Gradient Neighbour) is more distant from N_{stuck} than N_{b_2} as per X-gradient.

Theorem 2: The Shift Reverse Gradient Algorithm terminates within certain time and generate finite sequence.

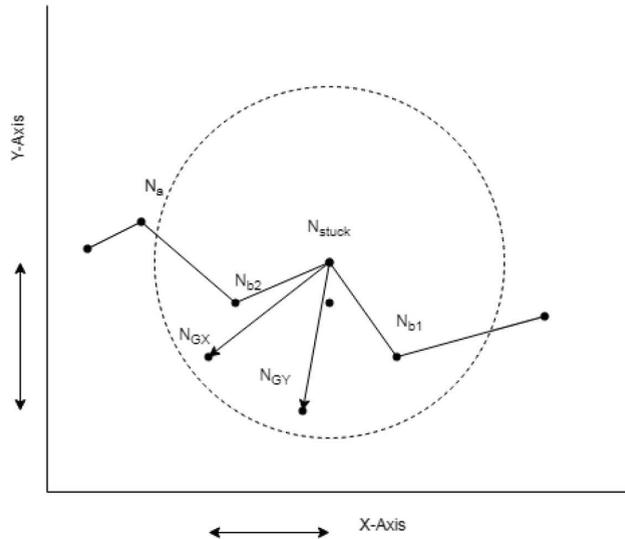


Figure 10. Reverse gradient neighbour location-free

Proof: In the SRG Algorithm, the stuck node N_{stuck} where the local minimum problem arises does not come again. The first node where the local minimum problem arises marked as the N_{stuck} node. In the SRG Algorithm, a packet does not revisit the stuck node N_{stuck} where the local minimum problem arises. The node at the end of the Reverse-XX-Gradient-Shift (or Reverse-YY-Gradient-Shift) returns to the Greedy Forward approach. If after performing Reverse-XX-Gradient-Shift (or Reverse-YY-Gradient-Shift) on N_{stuck} , it gets N_{stuck} as the next node, then it applies the Reverse-YY-Gradient-Shift (if a Reverse-YY-Gradient Shift on N_{stuck} gets N_{stuck} as the next node, it does Reverse-XX-Gradient-Shift). If initially at N_{stuck} it has applied Reverse-XX-Gradient-Shift, then on getting the N_{stuck} as the next node, it does Reverse-YY-Gradient-Shift. If initially at N_{stuck} it has done Reverse-YY-Gradient-Shift, on reaching the N_{stuck} as the next node, it does Reverse-XX-Gradient Shift. If even after doing both the Reverse-XX-Gradient-Shift and Reverse-YY-Gradient-Shift, it fails to get out of stuck node N_{stuck} . It terminates the process and declares that no route is available.

Lemma-2: Any node can have more the one node as Reverse- X-Gradient-Neighbours or Reverse-Y-Gradient-Neighbours

Proof In Fig. 11, node N_0 has N_5 and N_{11} , the two Reverse-X-Gradient- Neighbours as both are on equal distance on the circumference at distance R from the node along the x-axis. Similarly, the two Reverse-Y- Gradient-Neighbour of node N_0 is node N_2 and N_7 at an equal distance R from node N_0 along the y-axis on the circumference.

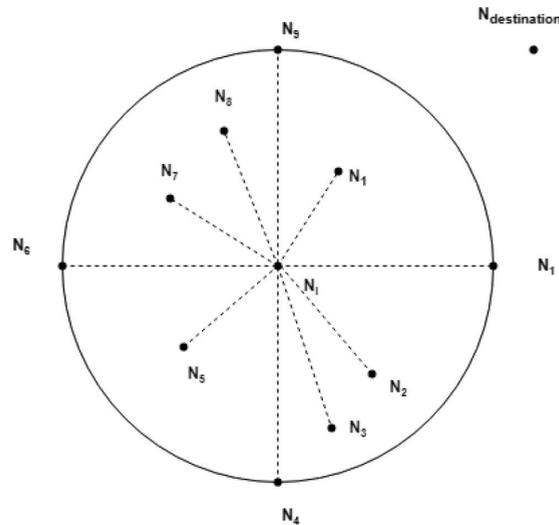


Figure 11. (i) Multiple Reverse-XX-Gradient-Neighbours (ii) Multiple Reverse-YY-Gradient-Neighbours

Theorem 3: The SRG Algorithm could explore the dual-path around the void region. These distinct routes increase the throughput.

Proof: The SRG approach function in two ways. In the first method, if one gets stuck at N_{stuck} , the initial shift is Reverse-XX Gradient Shift and returns to the greedy forward approach. If it gets the N_{stuck} as the next node, it switches to Reverse-YY Gradient Shift. In the second method at N_{stuck} , the initial shift is Reverse-YY Gradient Shift, and returns to greedy forward approach. If it gets the N_{stuck} as the next node, it switches to Reverse-XX Gradient Shift. in this way, the SRG algorithm could explore

the dual paths around the void region. The different gradient shifts can be applied alternatively to send the packet on the two routes. The two routes increase the throughput and the lifetime of the network. Here in Fig. 12 shows the route using the Reverse-XX Gradient shift at N_{stuck} is $N_s, N_1, N_2, N_{stuck}, N_4, N_5, N_6, N_7, N_8, N_d$. Fig. 13 shows the second route by using Reverse-YY Gradient shift at N_{stuck} is $N_s, N_1, N_2, N_{stuck}, N_7, N_8, N_9, N_{10}, N_{11}, N_{12}, N_d$. These two routing paths use distinct nodes that forward packet simultaneously.

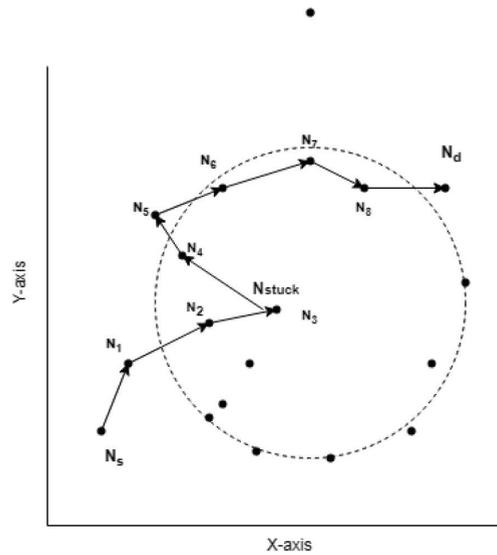


Figure 12. Routing path using reverse-XX-gradient shift

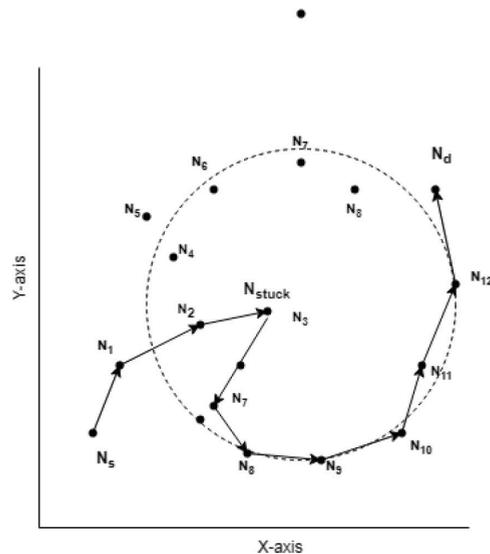


Figure 13. Routing path using reverse-YY-gradient-shift

5.1 Time Complexity Analysis

The SRG Algorithm uses the greedy forward approach to select the next node among its n number of neighbours with the least Euclidean distance from the destination. The required time for this step is $O(n)$. At the stuck node, where the situation of local minimum occurs, it applies SRG. In the SRG, first, it finds out the next node with maximum Reverse-Gradient-Distance among the stuck node N_{stuck} neighbours. Then from the node that we get after the first step, it finds the next node with maximum Reverse- Gradient-Distance among its neighbours. The time required to find the neighbour with maximum Reverse-Gradient-Distance is $O(n)+O(n)=2(n)$. So the SRG requires $O(n)$ amount of time. Thus the overall time complexity of the SRG approach is $O(n)$.

6. Performance Evaluation

In our experiment, we have compared the performance of the SRG approach with the three other existing void handling protocols. The three other protocols are the Boundhole approach, Greedy Anti Void Routing (GAR) approach, and Curved Stick (CS) approach through the simulation using MATLAB. The following parameters have been used to compare the performance of these protocols: (i) The length of the routing path (hop count); (ii) The average time for delivering a packet from source to goal; (iii) The average energy consumption for delivering a packet from source to goal. For this, we have simulated several simulation series in different settings.

6.1 Simulation Setting

In our simulation environment, we set a void region inside the network. The nodes are randomly allocated in the network except inside the void area. Every node has the same communication range with the increase in the size of a void, the phenomena of local minimum increases. Besides the artificially created void area, there are other small void areas in the network when the node density becomes low. The number of such voids and their sizes increases with the increasing number of dead nodes. The network has been configured by randomly distributing the nodes except inside the void region. The location of the source is on the very left end of the network area. The x coordinate of the source node has a value below a “Source Maximum”. Similarly, the destination node location value is above a “Destination Maximum” on the very right end of the network area. We have repeated the simulation scenario over the same parameters for 20 networks for every set of source and destination pairs. The parameter

Table 2. Simulation parameters

Parameters	Value(s)
Network Area	1000X900
Number of Nodes	500
Transmission Range	100
Hole Width	300
Hole Height	100,200,300,400,500
Source Maximum	100
Destination Maximum	900

mentioned in Table 2 is the same as those used in the [8]. MATLAB was used as a platform to perform the simulation. The simulation was done for an interval of 180 seconds. The sensor nodes start with an energy level 100mw. The node becomes dead if the energy level reaches below 80mw. The energy consumption per cycle is 0.35 mw. The number of packets sent per unit of time derives the average delay. The difference between the total initial energy of the nodes before the transmission begins and the total residual energy till all the possible route exhaust, is called the total energy consumption. Average energy consumption is calculated by dividing the total energy consumption by the number of nodes. We did the simulation with two different communication ranges. The first communication range (for the rare network) is 100m. The second communication range (for the dense network) is 150m.

Because of the small range of the rare network the node will have lower number of neighbours. While in the dense networks where the communication range is long, every node will have more neighbours. The higher number of neighbour nodes reduces the possibility of a situation of local minimum. The condition of the local minimum causes the difference in the routing path lengths of these algorithms, otherwise all of these follow the greedy forwarding approach. To measure the effect of numbers of the local minimum situations, we have conducted the simulation on two different communication ranges for these four algorithms.

6.2 Simulation Result

We have repeated the simulation over the two network configurations (i) Rare network (communication range 100m) (ii) Dense network (communication range 150m).

6.2.1 Routing Path Length

6.2.1.1 Rare Network (communication range 100m)

In Fig. 14, The plotted graph represents the routing path length for the four routing algorithms with a communication range equal to 100m (Rare Network). Results shows that for all the four algorithms, the routing path length increases with an increase in the size of the void area. The boundhole algorithm

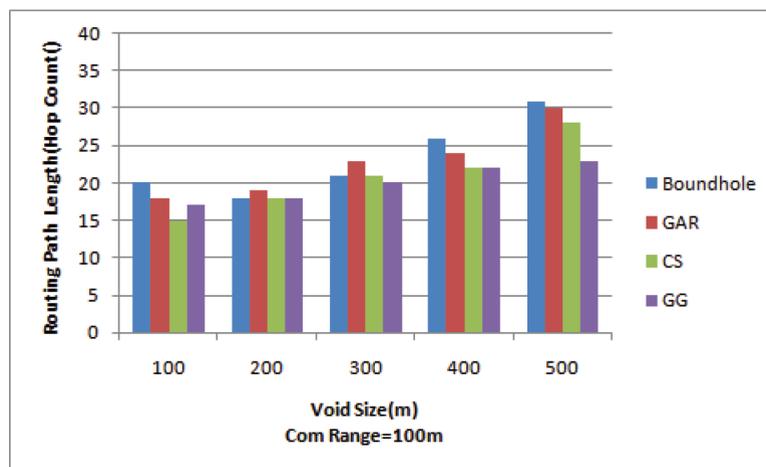


Figure 14. Routing path length for rare network

shows an exception when an intersection situation occurs because it cannot resolve the false boundary detection problem and visit longer paths (Liu and Feng, 2008, Mostefaoui et al., 2013). To overcome the false boundary detection, the GAR reduces its communication range to its half and performs better than the boundhole. But this approach visits unnecessary nodes. The CS approach uses a projection point which reduces the number of visited nodes by selecting only the potential ones. The performance of the CS approach is much better than GAR, and path length is at least three hops less in comparison. All these approaches perform the boundary traversal, and routing path length increase with the size of the void region. The SRG approach shows a slightly longer path for a small size void region (the void region with a length equal to 100), but it generates smaller routing path for the void of medium and large size.

6.2.1.2 Dense Network (communication range 150m)

In Fig. 15, the plotted graph represents the routing path length for the four routing algorithms with a communication range equal to 150m (Dense Network). All the algorithms show a relation between the routing path length and the size of the void area. The Boundhole algorithm with an increased range of communication can effectively eliminate the intersection situation. It improves the performance of the Boundhole algorithm. For the void region of moderate size, the Boundhole performance matches the GAR approach. We observe a longer path and poor performance by the Boundhole algorithm for large voids. CS approach consistently generates a routing path with a length of fewer than three hops in comparison to the GAR approach. The SRG Algorithm generates a longer routing path than the CS approach for a moderate-size void region. The cause for the longer route is that SRG visits two extra nodes to bypass the stuck node. It shows that the SRG approach generates a shorter path for the larger size void region (the void region with a size greater than 200m) than the CS approach. Here one can observe that among all these approaches for SRG, the routing path length is least affected by the increase in the size of the void region. The route length for large size void is 2-4 hop less than the CS approach. The SRG approach applies a void size-independent method to bypass the void area. It outperforms other boundary traversal-based algorithms in dealing with the large-sized void area.

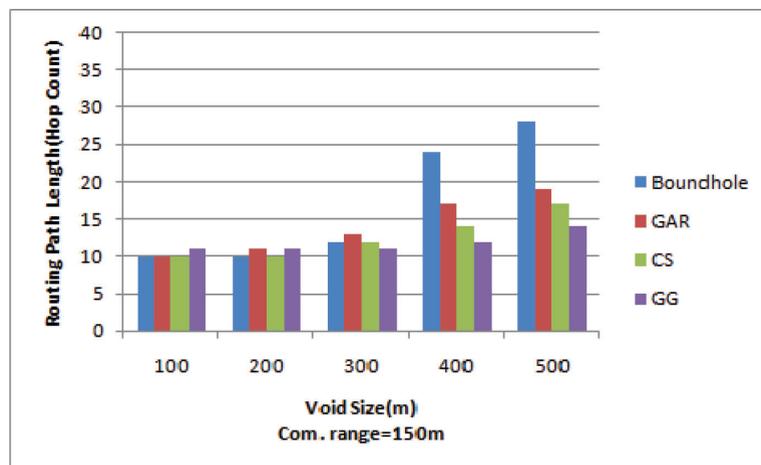


Figure 15. Routing path length for dense network

6.2.2 Average Transmission Delay

6.2.2.1 Rare Network (communication range 150m)

In Fig. 16, the graph plots the transmission delay for the four routing algorithms with a communication range equal to 100m (Rare Network). The transmission delay of the approach depends on the routing path length and complexity. The Boundhole algorithm has a longer path that increases transmission delay. The GAR has the highest processing time due to visiting an increased number of nodes to resolve the intersection situation. The CS approach reduces the number of nodes by selecting the nodes by their projection point. It reduces the processing time. The SRG approach visits only two extra nodes to get out of a stuck node. SRG is independent of the size of the void. This void size-independent algorithm reduces the processing time and transmission delay time. SRG transmission delay time for the medium and large-sized void is 7-17 percent less than the CS approach.

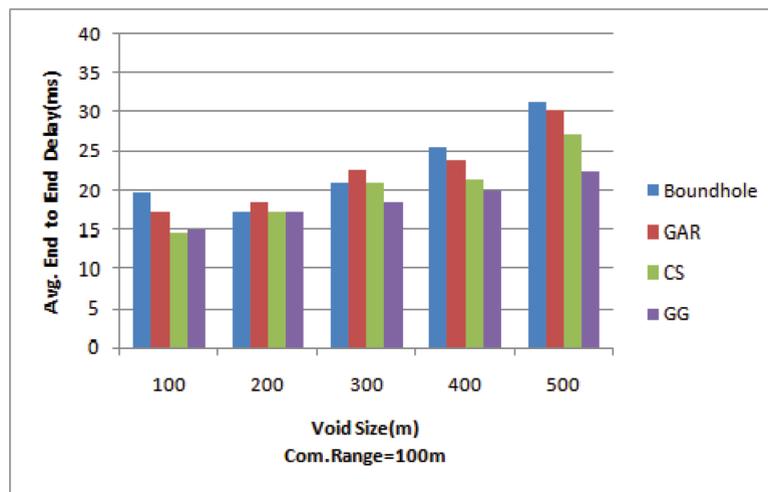


Figure 16. End to end delay for rare network

6.2.2.2 Dense Network (communication range 150m)

In Fig. 17, the graph plots the delay for transmission for the four routing algorithms with a communication range equal to 150m (Dense Network). When the communication range increases, the intersection situation eliminates for a small size void region. Here, for small size voids (100m and 200m), Boundhole, GAR, and CS approaches are nearly the same as they all follow the same greedy forwarding until the intersection situation arises. The SRG transmission delay for small size void is higher because, in SRG, it shifts by two extra nodes, causing increased route length and delay. The SRG approach outperforms the CS approach as the size of the void increases. The SRG void size-independent shifting proves better for the large-sized void regions. The transmission delay for medium and large-sized void is 12-16 percent less than the CS approach.

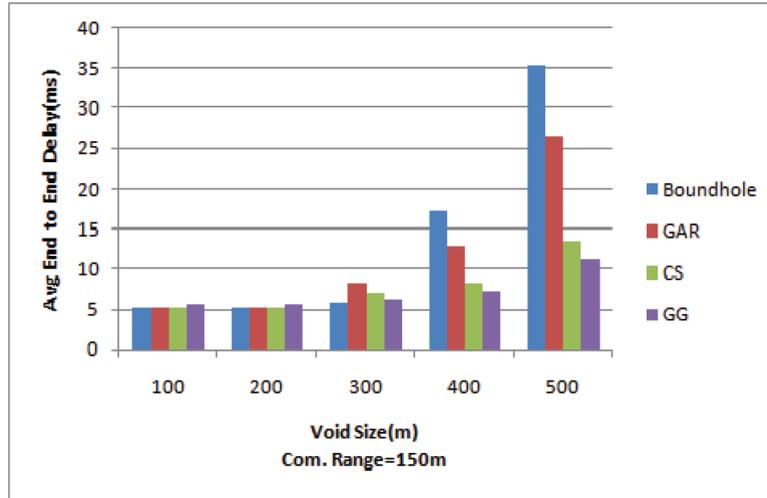


Figure 17. End to end delay for dense network

6.2.3 Average Energy Consumption

6.2.3.1 Rare Network (communication range 100m)

In Fig. 18, the plotted graph represents the average energy consumption for four routing algorithms for the communication range of 100m. The Boundhole algorithm cannot resolve the intersection problem and create a very long path in such a situation. The long route involves a higher number of nodes and more energy consumption. If the intersection situation does not occur, the Boundhole energy consumption reduces significantly. GAR approach efficiently deals with the intersection situation and perform better than the Boundhole. The CS approach gives better results in energy consumption by reducing the visited node using the projection point of nodes instead of every neighbour node. The SRG approach is very energy efficient in handling the void region, and average energy consumption remains nearly stable. The average energy consumption does not drastically increase with the size of the void. The energy consumption does not increase more than 12 percent with an increase in the hole size.

6.2.3.2 Dense Network (communication range 150m)

In Fig. 19, the graph represents the average energy consumption for four routing algorithms for the communication range of 150 m. The increased communication range eliminates the intersection situation in small and moderate- sized void areas. The Boundhole, GAR, and CS approach work similarly, using greedy forward until void area (up to 300m). The Boundhole, GAR, and CS generate the same route, and average energy consumption remains the same. For large-sized void areas, the intersection situation occurs. So a moderate size void creates a problem. GAR approach reduces its transmission range to half to resolve the intersection situation. It increases the number of nodes and energy consumption. The CS approach uses a full transmission range and projection points to reduce the number of visited nodes. The CS approach performs much better than the GAR approach for the large-sized void region. CS approach establishes a hit order and computes projection point for neighbours node that consumes energy. The SRG approach uses the same parameters used by greedy forwarding. It makes it more scalable and energy-efficient. The SRG approach is 6-16 percent more energy-efficient than the CS approach.

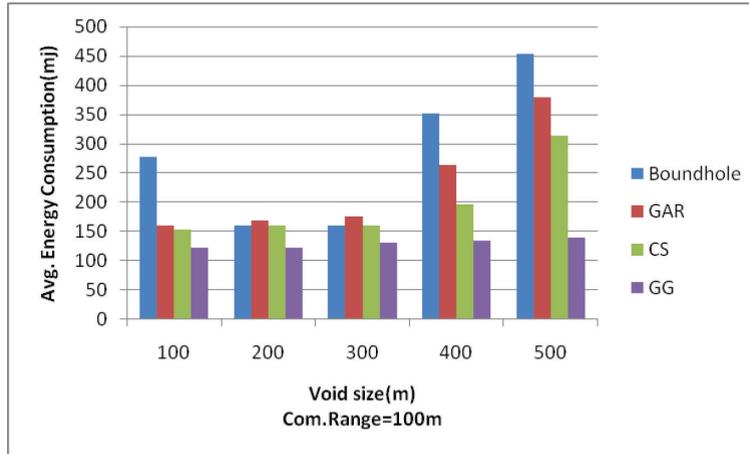


Figure 18. Avg Energy consumption for rare network

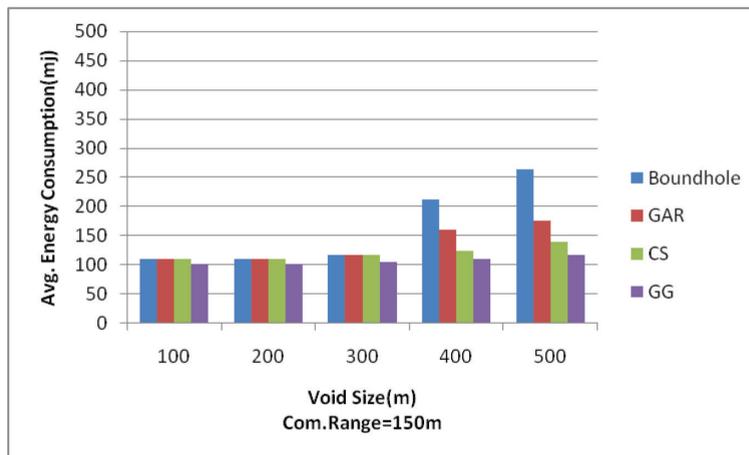


Figure 19. Average energy consumption for dense network

7. Conclusion

In this paper, we have proposed a new SRG routing algorithm for wireless sensor networks that is very efficient in bypassing medium- and large-sized void regions. In the simulation, we have shown that for both rare and dense networks with moderate and large-sized void areas, the SRG approach is at least 10 percent more efficient in average energy consumption. Its average energy consumption does not increase more than 12 percent with an increase in void size. Its transmission delay is 12–22 percent less than the state-of-the-art approaches. It makes this approach much more scalable and energy-efficient than other state-of-the-art approaches for medium- and large-sized holes.

7.1. Future Work

The experiments and analysis done show that the SRG approach generates a longer route path and delay for a small void area. The proposed SRG algorithm's performance is significantly better for moderate- and large-sized void areas than the popular approaches. But its results need improvement in the case of small-sized void areas. There is scope to improve its performance for small void areas. There is a lot of potential in this approach to extend it further for the 3-D network, vehicular transmission, and underwater transmission.

8. References

- Akkaya, K. and Younis, M., 2005. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3), 325-349. <https://doi.org/10.1016/j.adhoc.2003.09.010>
- Arad, N. and Shavitt, Y., 2008. Minimizing recovery state in geographic ad hoc routing. *IEEE Transactions on Mobile Computing*, 8(2), 203-217. <https://doi.org/10.1109/TMC.2008.86>
- Bahi, J. M., Makhoul, A., and Mostefaoui, A., 2008. Localization and coverage for high density sensor networks. *Computer Communications*, 31(4), 770-781. <https://doi.org/10.1109/PERCOMW.2007.61>
- Boukerche, A., Turgut, B., Aydin, N., Ahmad, M. Z., Bölöni, L., and Turgut, D., 2011. Routing protocols in ad hoc networks: A survey. *Computer networks*, 55(13), 3032-3080. <https://doi.org/10.1016/j.comnet.2011.05.010>
- Chen, S., Fan, G., and Cui, J.-H., 2006. Avoid 'void' in geographic routing for data aggregation in sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 1(4), 169-178. <https://doi.org/10.1504/IJAHUC.2006.010498>
- De Couto, D. S. and Morris, R., 2001. Location proxies and intermediate node forwarding for practical geographic forwarding.
- Gabriel, K. R. and Sokal, R. R., 1969. A new statistical approach to geographic variation analysis. *Systematic zoology*, 18(3), 259-278. <https://doi.org/10.2307/2412323>
- He, T., Stankovic, J. A., Abdelzaher, T. F., and Lu, C., 2005. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(10), 995-1006. <https://doi.org/10.1109/TPDS.2005.116>
- Hou, J. and Zhang, Y., 2018. Mobile-service based approach for topology control of wireless sensor networks. *Wireless Personal Communications*, 102(2), 1839-1851. <https://doi.org/10.1109/TPDS.2005.116>
- Hu, X., Ma, L., Ding, Y., Xu, J., Li, Y., and Ma, S., 2019. Fuzzy logic-based geographic routing protocol for dynamic wireless sensor networks. *Sensors*, 19(1), 196. <https://doi.org/10.3390/s19010196>
- Huang, H., Yin, H., Min, G., Zhang, J., Wu, Y., and Zhang, X., 2017. Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 17(6), 1339-1352. <https://doi.org/10.1109/TMC.2017.2771424>
- Jiang, Z., Ma, J., Lou, W., and Wu, J., 2008. An information model for geographic greedy forwarding in wireless ad-hoc sensor networks. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 825-833. IEEE. <https://doi.org/10.1109/INFOCOM.2008.134>

- Joshi, G. P. and Kim, S. W., 2009. A distributed geo-routing algorithm for wireless sensor networks. *Sensors*, 9(6), 4083-4103. <https://doi.org/10.3390/s90604083>
- Karp, B. and Kung, H.-T., 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243-254. <https://doi.org/10.1145/345910.345953>
- Kim, S., Kim, C., Cho, H., Yim, Y., and Kim, S.-H., 2016. Void avoidance scheme for real-time data dissemination in irregular wireless sensor networks. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pages 438-443. IEEE. <https://doi.org/10.1109/AINA.2016.59>
- Kuhn, F., Wattenhofer, R., and Zollinger, A., 2008. An algorithmic approach to geographic routing in ad hoc and sensor networks. *IEEE/ACM Transactions On Networking*, 16(1), 51-62. <https://doi.org/10.1109/TNET.2007.900372>
- Le Nguyen, P., Nguyen, N. H., Dinh, T. A. N., Le, K., Nguyen, T. H., and Nguyen, K., 2021. QIH: An Efficient Q-Learning Inspired Hole-Bypassing Routing Protocol for WSNs. *IEEE Access*, 9:123414-123429. <https://doi.org/10.1109/ACCESS.2021.3108156>
- Lima, M. M., Oliveira, H. A., Guidoni, D. L., and Loureiro, A. A., 2017. Geographic routing and hole bypass using long range sinks for wireless sensor networks. *Ad Hoc Networks*, 67, 1-10. <https://doi.org/10.1016/j.adhoc.2017.08.010>
- Liu, W.-J. and Feng, K.-T., 2008. Greedy routing with anti-void traversal for wireless sensor networks. *IEEE transactions on mobile computing*, 8(7), 910-922. <https://doi.org/10.1109/TMC.2008.162>
- Mostefaoui, A., Melkemi, M., and Boukerche, A., 2013. Localized routing approach to bypass holes in wireless sensor networks. *IEEE transactions on computers*, 63(12), 3053-3065. <https://doi.org/10.1109/TC.2013.180>
- de Oliveira, H. A., Boukerche, A., Guidoni, D. L., Nakamura, E. F., Mini, R. A., and Loureiro, A. A., 2015. An enhanced location-free Greedy Forward algorithm with hole bypass capability in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 77, 1-10. <https://doi.org/10.1016/j.jpdc.2014.10.007>
- Rekik, M., Mitton, N., and Chtourou, Z., 2018. GRACO: a geographic greedy routing with an ACO based void handling technique. *International Journal of Sensor Networks*, 26(3), 145-161. <https://doi.org/10.1504/IJSNET.2018.090148>
- Savvides, A. and Strivastava, M. B., 2003. Distributed fine-grained localization in ad-hoc networks. *IEEE Transactions of Mobile Computing*.
- Toussaint, G. T., 1980. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, 12(4), 261- 268. [https://doi.org/10.1016/0031-3203\(80\)90066-7](https://doi.org/10.1016/0031-3203(80)90066-7)
- Yilmaz, O., Dagdeviren, O., and Erciyas, K., 2014. Localization-free and energy-efficient hole bypassing techniques for fault-tolerant sensor networks. *Journal of network and computer applications*, 40, 164-178. <https://doi.org/10.1016/j.jnca.2013.09.002>
- Zhang, D. and Dong, E., 2015. A virtual coordinate-based bypassing void routing for wireless sensor networks. *IEEE sensors journal*, 15(7), 3853-3862. <https://doi.org/10.1109/JSEN.2015.2398852>



9. Vitae

	<p>Saurabh Singh is a research scholar (pursuing PhD) in the Department of Computer Science at the Madan Mohan Malviya University of Technology, Gorakhpur. He has done his M.Tech (Computer Science) from Jamia Hamdard, New Delhi. He has done his B.Tech (computer science) from Dr A.P.J. Abdul Kalam Technical University, Lucknow. His research interests include the wireless sensor network, underwater routing protocols and cryptography.</p>
	<p>Sarvpal Singh is a Professor in the Department of Information Technology and Computer Application at the Madan Mohan Malviya University of Technology, Gorakhpur. He did his BE (Computer Science) at Marathwada University, Aurangabad. He did his ME (Computer Science) from Thapar Institute of Engineering and Technology, Patiala and PhD (Computer Science) from Deen Dayal Upadhyay, Gorakhpur University. His field of research includes Wired and Wireless networking, Mobile and Cloud Computing and Linux OS</p>
	<p>Jay Prakash is an assistant professor in the Department of Information Technology and Computer Application at the Madan Mohan Malviya University of Technology, Gorakhpur. He has done his B.Tech (Computer Science) from BIET, Jhansi. He did his M.tech (Computer Science) from MNNIT, Allahabad. Prof. Jay Prakash did his Ph.D (Computer Science) at Uttarakhand Technical University. His research interest includes various protocols of the network, antennas in communication and gateway discovery in Adhoc networks for internet access. He has also researched in field of machine learning and image processing.</p>