5-1-2007

Flooding control in route discovery for reactive routing in mobile ad hoc networks

Abedellatif Hussein

Flooding Control in Route Discovery for Reactive Routing in Mobile Ad Hoc Networks

by

Abedellatif Mohammed Hussein

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

Supervised by

Dr. Fei Hu
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
May, 2007

Approved By:

Dr. Fei Hu
Primary Advisor – Assistant Professor, Dept. of Computer Engineering

---

Dr. Nirmala Shenoy
Co- Advisor – Professor, Dept. of Information Technology

Dr. Muhammad Shaaban
Secondary Advisor – Associate Professor, Dept. of Computer Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title: Flooding Control in Route Discovery for Reactive Routing in Mobile Ad Hoc Networks

I, Abedellatif Mohammed Hussein, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

_____
Abedellatif Mohammed Hussein

_____
Date

Dedication

To my family, friends and teachers

Acknowledgments

I would like to thank my advisor Dr. Nirmala Shenoy for her dedication and support throughout the project. I would also like to thank my committee members, Dr. Fei Hu and Dr. Muhammad Shaaban for their valuable advice and feedback and finally I would like to thank all the staff and faculty of the department of computer engineering at Rochester Institute of Technology for their assistance.

Abstract

Routing is a very important function in the network layer of the OSI model for wired and wireless networks. Mobile Ad hoc Networks (MANETs) are a collection of wireless nodes forming a temporary network that is supposed to be constructed on the fly without infrastructure and prior setup. This fashion of setup demands that the nodes act as routers for other nodes. This necessitates the need of a robust dynamic routing scheme. Routing protocols are classified into three main categories: proactive, reactive, and hybrid. Reactive routing has been the focus of research in recent years due to its control traffic overhead reduction. Reactive routing operation involves three main steps: route discovery, packet delivery, and route maintenance. If a source node, initiating the message, knows the route to the destination, this route is used to transmit the message; otherwise, the source node will initiate a route discovery algorithm to build the route, which highlights the importance of this phase of the on-demand routing process. This thesis work will present a route discovery algorithm that will try to find the route between the sender and the intended receiver in relatively short periods of end-to-end delay, least amount of control traffic overhead, and a loop free path between the two communicating parties. Furthermore, performance comparison between the proposed algorithm and other standard algorithms, namely basic flooding and flooding with self-pruning, will be conducted. The proposed route discovery algorithm can be used in several approaches serving ad hoc network setup, where connectivity establishment and maintenance is important.

Table of Contents

List of Figures

List of Tables

Glossary

AF Alternating Flooding.

DTN Distributed Transient Network paradigm.

FSP Flooding with Self Pruning.

MANET Mobile Ad hoc Network.

MAC Media Access Control.

MPR Multipoint Relaying.

NB Neighbor Packet.

OPNET Optimum Performance Network.

RREQ Route Request Packet.

RRESP Route Response Packet.

SF Simple Flooding.

Chapter 1 Introduction

Recent advances in portable computing devices and wireless communication technology have made it possible to stay connected at any place and at anytime. In the near future, users will be able to move freely and still have seamless, reliable and high-speed network connectivity. Portable computers and hand-held devices will do for data communication what cellular phones are now doing for voice communication. Traditional network mobility focused on roaming, which is characterized by hosts connecting to the fixed infrastructure internet at locations other than their well known home network address. Hosts can connect directly to the fixed infrastructure on a visited subnet through a wireless link or a dial-up line, these so called traditional (or fixed-infrastructure mobile) networks raise issues such as address management, but do not require significant, changes to core network functions such as routing. Mobile Ad hoc Networks (MANETs) refer to mostly wireless - networks where all network components are mobile. In a MANETs there is no distinction between a host and a router since all network hosts can be endpoints as well as forwarders of traffic. In contrast to fixed infrastructure networks, MANETs require fundamental changes to network routing protocols, including multicast routing and packet forwarding. The next two sections will provide a general overview of the Mobile Wireless Ad hoc Networks and the motivation behind the study of flooding in these kinds of networks.

1.1. Overview of MANETs

The Latin term “ad hoc” literally means “for this purpose only.” Ad hoc Networks is the term for an autonomous collection of mobile nodes which are built on the fly for a specific purpose (e.g., emergency situations, rescue operations, battlefield situations). Ad hoc networks are described by the Distributed Transient Network paradigm (DTN), which defines networks with physically or logically scattered nodes that are free to move within the network. The nodes may join or leave the network at any time without notice. Early attempts to form ad hoc networks of mobile nodes go back to 1972 when the Department of Defense (DoD) initiated a program on Packet Radio Networks (PRNET). The intent was to create technology for the battlefield that does not need a previously deployed infrastructure and which would be robust – surviving the failure or destruction of some of the radios [1]-[2]-[3]. Over the years, interest in such network architecture increased due to its speed, easy deployment, robustness and low cost. The availability of license-free frequencies as well as advances in electronic chip design and fabrication facilitated the development of ad hoc networks. MANETs have several unique properties that differentiate them from fixed multi-hop networks [4]-[5].

- Dynamic Topology: Since the nodes are continuously moving in a random fashion, the network topology is also changing rapidly.
- Bandwidth-Constrained, variable capacity links: Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications (after accounting for the effects of multiple access, fading, noise, and interference conditions, etc.) is often much less than a radio's maximum transmission rate. One effect of the relatively

low to moderate link capacities is that congestion is typically very common and the mobile users will demand similar services like the ones served by its fixed counterpart. These demands will continue to increase as multimedia computing and collaborative networking applications rise.

- Energy-Constrained Operation: Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.
- Limited physical security: Mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered. Existing link security techniques are often applied within wireless networks to reduce security threats. As a benefit, the decentralized nature of network control in MANETs provides additional robustness against the single points of failure of more centralized approaches.

Devices in MANETs communicate with each other without relaying on a particular infrastructure. So if a device wants to communicate with another that is out of its range, it should use other devices as routers on its behalf so that they can effectively communicate with each other. Protocols used within these kinds of ad hoc networks need to be carefully designed because of the underlying assumptions and performance concerns that are created due to the above mentioned characteristics. Wireless communications are used very widely. Figure 1.1 shows an example of an ad hoc network.

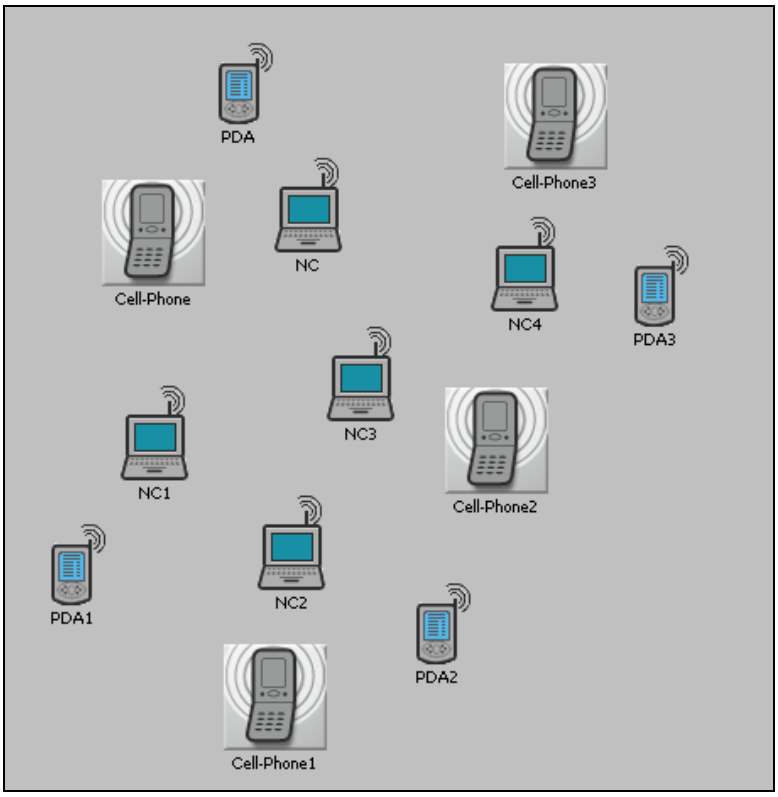


Figure 1.1- Ad hoc wireless networks example

The applications of MANETs are summarized below. MANETs find their applications in many fields.

- Personal area networking: Used for accessing, sharing, and processing data via the Internet. Examples are cell phones and laptop computers.
- Military environments: To equip military units and individual soldiers with battlefield communication devices so they can communicate with each other in tactical operations. Examples are soldiers equipped with Personal Digital Assistants (PDAs). Similar devices are used for tanks, planes and warships.
- Civilian environments: Used for accessing and sharing data with other potential users; e.g., distribution of presentations, exchange of information.

- Emergency operations: To setup communication in important or urgent situations such as search and rescue, policing, fire fighting and disaster recovery.

1.2. Motivation

The issue of routing packets between any two nodes in an ad hoc network is a challenging task because the nodes are randomly moving within the network. A path that was considered optimal at some point in time might not work a few seconds later. The wireless channel properties also add to the uncertainty of the path quality. Moreover the surrounding environment might cause problems for indoor scenarios. Nodes in ad hoc networks can communicate and exchange messages with nodes in their transmission range. To effectively communicate with nodes out of range a sending node will rely on its neighbors for message forwarding. The scenario of message exchange in MANETs depends on whether the source has the route to the destination, which in this case will use that route. If the source does not have any prior information about the destination, it will trigger its route discovery algorithm. Traditional routing protocols are proactive which means they maintain routes to all nodes. They require frequent control messages to maintain those routes. The other form of routing protocols is reactive which involves establishing the routes when they are explicitly needed. Reactive routing protocols are useful in large scale MANETs with moderate or low mobility [6]. Flooding forms the basis of nearly all communications in ad hoc networks and is fundamental to routing protocols [7]. Improving route discovery through flooding control is important in the reactive routing process, where, if not efficiently designed, flooding will result in the Broadcast Storm Problem [8]. On the other hand, quality of the routes and the network coverage should be taken into consideration. The existing route discovery algorithms

employ either simple or efficient flooding in their route request broadcast. If simple flooding is used, all nodes in the source's transmission range will rebroadcast the first copy of the message as shown in Figure 1.2 (a). If the same node is to send another route request, the same nodes will transmit the message as shown in Figure 1.2 (b). If efficient flooding is used in a fixed network topology and the retransmission decision is based on neighborhood information, the chosen nodes will rebroadcast the messages all of the time during the communication session as shown in Figure 1.3 (a) and Figure 1.3 (b), respectively.

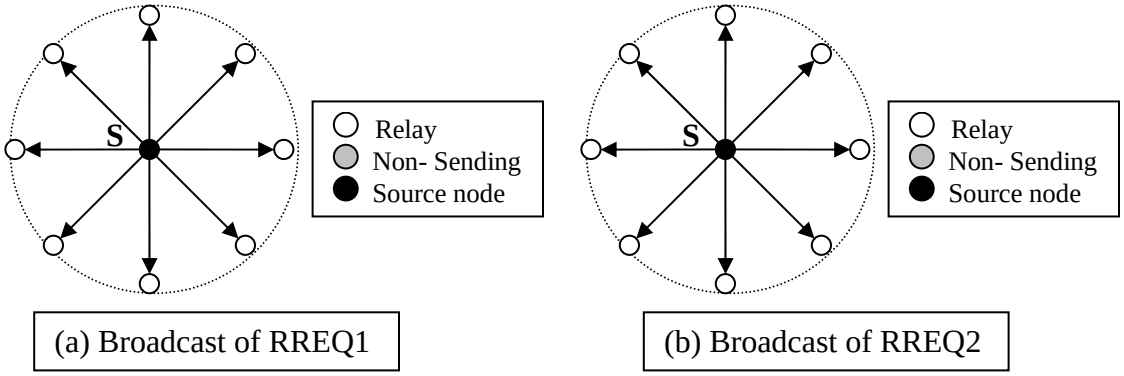


Figure 1.2- Simple flooding illustrations

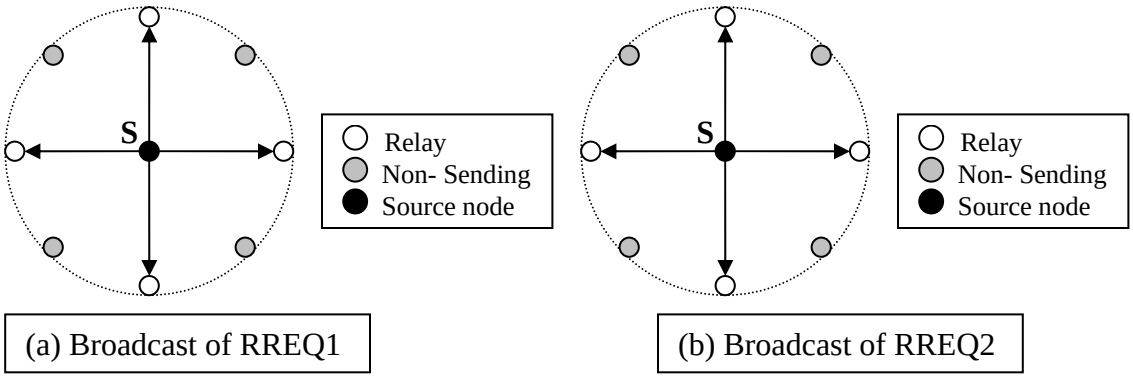


Figure 1.3- Efficient flooding illustrations

This thesis work will present a route discovery algorithm for reactive routing protocols, which will use a new rebroadcast strategy through the effective utilization of the second-hop neighborhood information. The sender node chooses a forwarding set of nodes for the first copy of a route request message as shown in Figure 1.4 (a) and uses a different forwarding set for another route request message as shown in Figure 1.4 (b). This approach reduces unnecessary broadcasts of control packets and decreases the chance of collision and contention. This remedies the Broadcast Storm Problem and eventually increases the amount of throughput as the network gets denser.

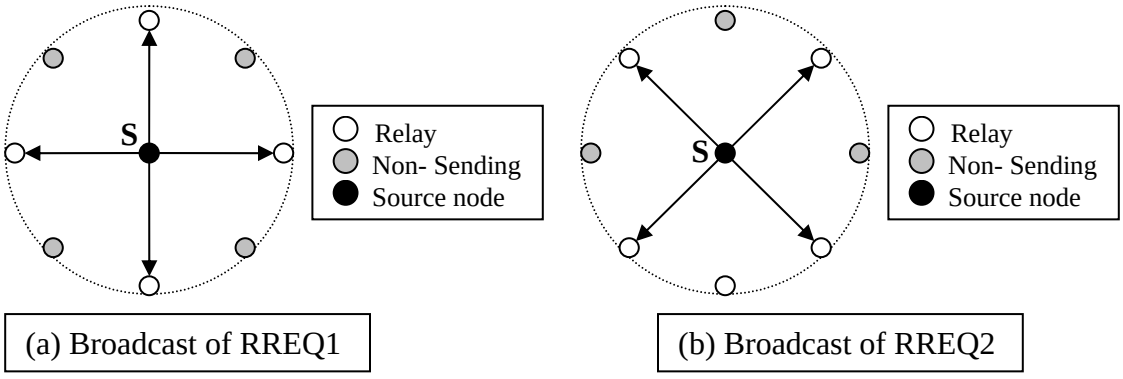


Figure 1.4- Proposed algorithm expected broadcast

Chapter 2 Related Work

There are two main characteristics of MANETs: they are wireless which means that broadcasting is inherent in their nature, and they are mobile which means that they are continuously moving and require frequent route updates. The following sections describe the above-mentioned properties.

2.1. Broadcasting in MANETs

Control traffic reduction is an important goal in route discovery operation in MANETs; several schemes and heuristics have been proposed that can be classified into four classes [9]:

- Blind flooding: This is the simplest form of flooding, where the receiving nodes retransmit the first copy of the message. If a node gets a duplicate message, it will refrain from transmission [8]. Even though blind flooding guaranties maximum coverage, it introduces high control overhead due to the unnecessary retransmissions. Blind flooding leads to reachability problems, collisions and causes contention, which degrades the overall performance.
- Probability based flooding: In this form of flooding, each node will retransmit, based on a predetermined probability, which will reduce the amount of flooding traffic but at the expense of network coverage, especially in sparse networks. In [10], gossiping is used (which is basically tossing a coin) to randomly decide whether or not to forward a message. This method exhibits a bimodal behavior, meaning that either the broadcast is successful in covering most of the network, or it dies early, covering only a small portion around the source. Another form of

broadcast that belongs to this category is the Counter Based Broadcast (CBB) [11], where a node only broadcasts if the counter is below certain threshold when a Random Assessment Delay (RAD) that the receiving node sets upon receiving the RREQ, is fired. The disadvantage of this kind of broadcast is that it can not ensure 100% reachability because some of the nodes that are supposed to transmit refrain from transmission because they exceed their threshold.

- Area-based methods: The retransmission decision is made based on the location; for example, if a node receives a packet from a source close to it in distance, it will not propagate the packet any further. Edge forwarding was proposed in [12], which uses location information as well as first hop neighbor information to decide on its retransmission strategy; the algorithm further divides the area around the source into six equal partitions, and if the receiving node lies in the partition edge and covers all of its first hop neighbors, then it forwards the packet. Border Aware broadcasting algorithm was proposed in [13], which uses the value of the signal strength in the received packet to estimate the node's distance from the sender. Nodes that are away from the sender and close to the border of their transmission range are only allowed to propagate messages, since the closer nodes to the sender do not contribute to the coverage. The disadvantage of these methods is that they rely on complex equipment such as GPS devices for location determination.
- Neighbor information methods: In this form of flooding, nodes have to keep track of first and second hop neighbors and use this information for retransmission decisions. Self-selection route discovery strategy [7], uses source-driven self-

selection (SDSS) and pure self-selection (PSS), where in the former, the source node decides which node should re-transmit based on a utility metric included in the transmitted RREQ packet and in the latter, each intermediate node will decide on rebroadcasting the RREQ packet by selecting its own utility metric that might be composed of (power, mobility, topology). A Look-ahead Unicast Routing algorithm (LAUR) [14], a source node, chooses one of its neighbors to rebroadcast; based on the number of packets in the queue, the node with fewer packets in the queue will be the most eligible node for rebroadcasting, thus avoiding nodes that are already congested. The Multipoint Relaying (MPR) flooding technique [15], proposes heuristics for choosing the forwarding nodes of the source so that the chosen relays would cover the second hop neighbors of the source. Flooding with self-pruning was proposed in [16], where each node exchanges the list of its first hop nodes with neighbors; upon receiving a broadcast, a node compares its own first hop neighbors against the ones listed in the message header, and if all of them are listed, it will refrain from retransmission.

2.2. Routing in MANETs

MANETs are gaining increasing attention, particularly in terms of routing protocols development, since the routing protocols that are present were designed for fixed wired networks where they cannot be applied to MANETs directly. Several routing protocols were designed, but all fall into three main categories: proactive, reactive, and hybrid routing protocols. The following section will briefly describe those categories.

2.2.1 Proactive Routing

Traditional routing protocols such as RIP and OSPF are proactive routing protocols. They employ periodic broadcasting to obtain network topology updates such as distance vector or link state information in order to compute the shortest path from the source to every destination. Periodic broadcasting consumes a lot of MANETs' limited bandwidth and increases the amount of control packet overhead since the nodes are moving. Optimized Link State Routing Protocol (OLSR) was proposed in [17] which is based on the link state algorithm and uses periodic exchange of messages to maintain topology information; it uses the MPR concept of flooding hello messages. Those hello messages have fields that hold the addresses of the sender's MPRs, if a node receives the hello message and it is a designated MPR, then it routes all of the data packets coming from the sender. Fisheye State Routing (FSR) was proposed in [18], which is aimed at large scale MANETs and MANETs with high mobility, route update information are compared to their distance; i.e. routes in the inner scope (within distance of 2 hopes) are maintained more regularly, whereas routes to a more distant nodes or outer scope are maintained less regularly. Cluster-heads manage routing inter- and intra-cluster which will reduce the total amount of route control information as noted in Cluster-head Gateway Switch Routing (CGSR) [19]. Topology Broadcast Based on Reverse-Path Forwarding Routing Protocol (TBRPF), proposed in [20], tries to reduce the control overhead by reducing the number of rebroadcast through optimized flooding. Because only the differences between the previous network state and the current network state are transmitted. Routing messages are smaller and may be sent more frequently. This means that nodes' routing tables are more up-to-date.

2.2.2 Reactive Routing

This category of routing protocols tries to reduce the total number of routing control information in the network by determining routes among nodes when needed only. The route discovery process goes through the following steps:

- Source S initiates a route request (RREQ) and broadcasts it to its neighbors.
- Upon receiving an (RREQ), each node rebroadcasts it.
- The destination sends a route reply (RRESP) to the source S when it receives an (RREQ) dedicated to it.

If the rebroadcast is done using blind flooding, routing control packets will be disseminated through the network and end up with a Broadcast Storm Problem; to overcome the problem in reactive routing, a number of different strategies have been proposed. Dynamic Source Routing (DSR) proposed in [21], utilizes the source routing option in data packets where the sender of the packet determines the complete sequence of nodes through which to forward the packet; it uses route caching and limits the number of hops in route discovery to reduce the effect of blind flooding. Ad-hoc On-Demand Distance Vector Routing (AODV) proposed in [22] uses the Expanding Ring Search technique for neighbor detection and maintains routing tables for the routes in the intermediate nodes. Routing On-demand Acyclic Multi-path (ROAM) [23], uses distance information between the sender and the receiver to construct directed acyclic sub graphs for the propagation of the flood. Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) [24], limits RRQ packets to a certain number of hops to reduce and localize the overhead associated with route discovery process, given that source and destination have prior communication; otherwise, RRQ will not be localized. Cluster-Based routing

Protocol (CBRP) [25], limits the amount of routing control packet information by making cluster heads responsible for exchanging and propagating RRQ.

2.2.3 Hybrid Routing

Hybrid Routing combines the advantages of proactive and reactive routing protocols to achieve a high level of scalability. As an example, Zone Routing Protocol (ZRP) [26] uses the zone concept where each node belongs to a specified zone. Routing within the zone is proactive, and when nodes need to communicate to other nodes in a different zone, reactive routing is used. This cooperation reduces the number of rebroadcasting nodes.

Chapter 3 Proposed Solution

The proposed algorithm uses the concept of source routing, where the sending node explicitly specifies in the transmitted packet header, the forwarding nodes set that are supposed to retransmit the broadcasted messages. The forwarding nodes sets are obtained by efficiently exploiting the second hop neighborhood information of each node in the network. The following sections provide a detailed explanation of the proposed algorithm.

3.1. Proposed Heuristics

3.1.1 Neighborhood Information

Finding neighborhood information is achieved by periodic transmission of *hello* messages where they are called *NB* messages in this thesis work. Each node in the network maintains a neighbor table that has entries for its immediate first-hop neighbors. The transmission range (*R*) covers the first-hop nodes. The second-hop neighbors are only reachable through the first-hop neighbors. In Figure 3.1, node (*X*) is a first-hop neighbor for the source (*S*) because it is within the transmission range (*R*). Node (*Y*) is a second-hop because if the source node wants to communicate with (*Y*), it must relay on the node (*X*). As the node is switched on, it sends frequent (around 4-5) *NB* packets in the first 10 ms, this provides quick neighbor table population. *NB* packet transmission frequency is set to 1 second plus a delay. This delay is calculated by generating a random value from a uniform distribution between (0, 200). The generated random value is multiplied by the slot time (0.000023 sec) and used as the delay. The neighbor checks the source id of the incoming packet, if it is in its local neighbor table, it reads the neighbors

of the source and updates the values in its local neighbor table, and otherwise it creates a new entry and adds the source node and his neighbors to the neighbor table. Figure 3.2 shows the algorithm for neighborhood table population.

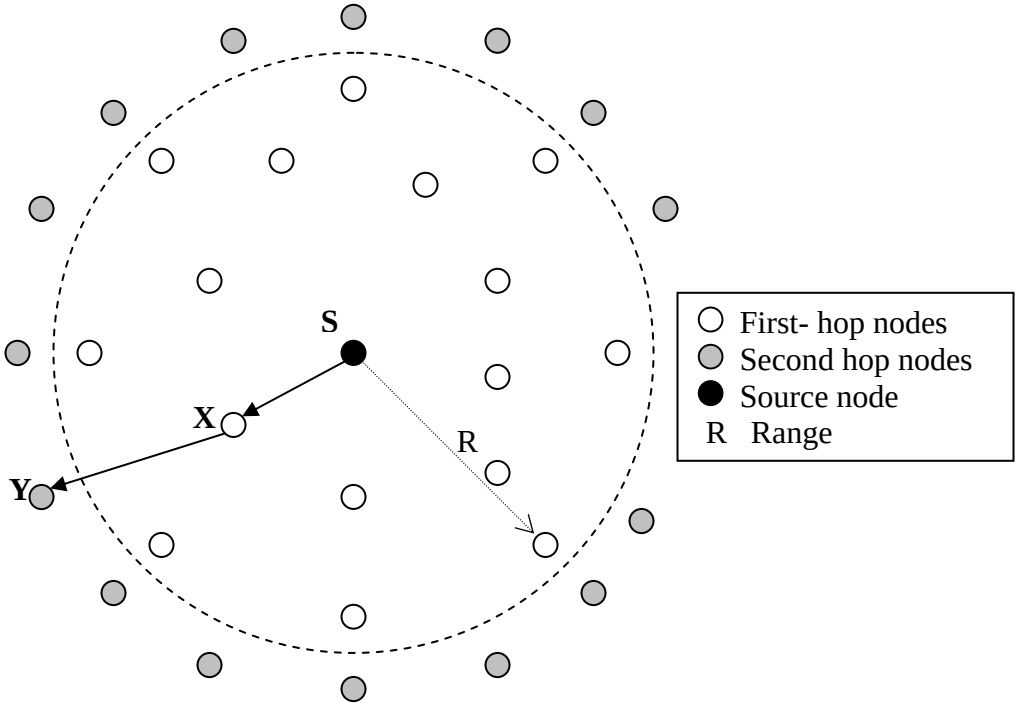


Figure 3.1- First and second-hop neighbors

```
If (the source id is in my neighbor table)
{
   Size   = number of the first hop neighbors of the source listed in the
            received packet
   Size2 = number of the source's neighbors listed in my neighbor table
      If (size > size2)
           Update_list of neighbors that corresponds to the source
}
Else
     { Create new entry for the source in my neighbor table
      Copy its first hop neighbors from the packet header and insert them
       in the list
      }
```

Figure 3.2- Algorithm to populate the neighborhood table

3.1.2 Forwarding Node Sets Calculation

With the neighborhood table ready, the process of calculating the forwarding nodes sets starts with the following assumptions:

- u & v are nodes in the network.
- $N(u)$ and $N(v)$ are neighbor sets of u and v, respectively.
- i & j are integer values from $\{0,1,2,3,.....n\}$, where n is the total number neighbor nodes in a neighborhood table.

Assume a node v has a neighbor set $N(v)$ and $u_i \in N(v)$ is the first neighbor node in the table. Take the node u_i and compare its neighbors $N(u_i)$ to the rest of the nodes in the table u_j where $j = \{i+1,2.....n\}$. If u_j is present, i.e. $u_j \in N(u_i)$, the node and the first neighbor in the list can hear each other. In other words, they overlap and are not disjoint. If $u_j \notin N(u_i)$, the node is a disjoint node and the algorithm states it should be added to the first node to form a disjoint set of nodes. The neighbors of u_i and u_j will form an extended set of neighbors $N_{ext} = N(u_i) \cup N(u_j)$ that replaces $N(u_i)$. This process continues to the next neighbor node using the extended set for the comparison. Figure 3.3 shows the algorithm applied to each node to calculate the forwarding node sets. The algorithm is employed for several rounds to examine the entire neighbors in the table.

```
Let u_i ∈ N(v) be the first neighbor node in the table
Let k =1

For ( i =0;i<n ; i++)

{
        Let rounds=1

        While (i<n and rounds<= n-i)
        {
          Let set(k) = u_i  //forwarding nodes set
          Let N_ext = N(u_i) //extended node set
          For (j=i+rounds ;j<n;j++)
            {
               If ( u_j ∉ N_ext )
                        {
                            set(k) = set(k) ∪ u_j
                            N_ext = N_ext ∪ N(u_j)
                        }
            }
        rounds ++
        K++

        }
}
```

Figure 3.3- Algorithm for forwarding nodes sets calculation

3.1.3 Forwarding Node Sets Filtration

The method used in calculating the forwarding node sets results in other sets. The subsets should be eliminated. Assuming there are a number of forwarding node sets (*i*), starting with the first set in the list R_j, if $R_j \subset R_i$ is a subset of R_i, discard this set, otherwise, R_j is not a subset of any previous set and is used as a qualified forwarding node set. This procedure is done until all sets in the list are covered and form the final list

of forwarding node sets that are to be used for RREQ transmission. Figure 3.4 shows the elimination algorithm taken by each node to determine its final forwarding node sets.

```
For (i = 1; i <max_number of forwarding nodes sets;i++)
{
For (j=i+1; j<max_number of forwarding nodes sets;j++)
   {
        If (set (j) is a subset of set (i))
             Discard set (j)
   }
}
```

Figure 3.4- Algorithm to eliminate forwarding node sets

3.1.4 Retransmission Strategy

Based on the neighborhood knowledge, a node determines its neighbors that can propagate a message without colliding with each others transmissions. In the RREQ messages, there are *n* fields for identifying non-overlapping nodes that are to forward the RREQ further. For example, in Figure 3.5, as node 12 forwards a RREQ it requests nodes 10, 13 and 14 forward the RREQ only. This scheme reduces the number of RREQ retransmissions as well as collision probabilities. Node 12 could have the following forwarding node sets that have the maximum number of nodes; (10, 13 and 14) and (19, 7 and 8). There could be other equal subsets of these two, which will be discarded while determining the final set.

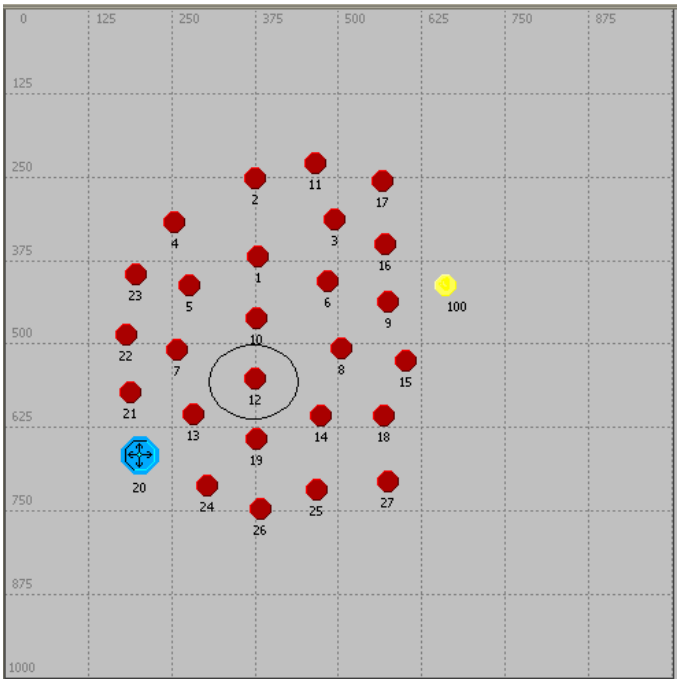


Figure 3.5- Network topology

3.1.5 Node Distance

Control traffic overhead is an important factor in flooding control. The nodes that do not contribute to new network coverage should not be allowed to rebroadcast RREQ messages any further. To achieve this goal, each node in the network has a specific value assigned to it upon the detection of a new RREQ message. This value is called (node distance), it represents the number of hops the node has from the original sender. If a node detects a RREQ directly from the original sender, it will have the value 1 as its node distance. Otherwise, upon detecting the first copy of a RREQ message, the receiving node examines the node distance in the packet header of the incoming RREQ message; the value is incremented by one to be set as its own node distance. Figure 3.6 shows the pseudo code for assigning the node distance.

```
If (source_id == original source)
        node_distance = 1
else {
      if (node_distance == 0)//first copy of RREQ to be detected
        node_distance = received node_distance + 1
    }
```

Figure 3.6- Algorithm to assign node distance

For example, referring to Figure 3.5, node (100) is the original sender, when the neighbor nodes (9, 15, 16) detect the broadcasted RREQ they set their node distance to 1. When node (9, 15 or 16) broadcasts the message further, its own node distance value is included in the packet header. As node (3) receives from node (16), it will set its node distance to 2 and rebroadcasts the message. If node (9) is in the forwarding node set of node (3), it will drop the packet because its node distance is less than the received node distance. This implies that node (9) detected the received RREQ before node (3) and its transmission will not contribute to any network coverage.

3.1.6 Example

To illustrate the operation of the algorithm, we provide an example that applies the above mentioned heuristics to node (12) in Figure 3.5. By examining the neighborhood table of node (12) shown in Table 3.1, node (10) is seen to be the first immediate neighbor. This node is picked first and its neighbors are compared with each neighbor of node (12). We see that the next node (13) is not a neighbor of node (10) which means that these nodes are disjoint (can not hear each other). This implies that nodes (10, 13) are added as the first forwarding set of nodes. Moreover, their neighbors are joined to form an extended neighbor list (6, 8, 1, 5, 7, 19, 21, 20, 22, 24). Next, this extended neighbor list is compared with the next node which is (14) in this case. Node

(14) is added to the first formerly formed set since it does not belong to the extended list. Thus, the forwarding node set becomes (10, 13, 14). The extended list is then updated to include the neighbors of (14) to be (6, 8, 1, 5, 7, 19, 21, 20, 22, 24, 15, 18, 25, 27). Note that comparing this extended list to the next nodes does not add any more new nodes to the first forwarding node set (all of the nodes are a subset of the extended neighbor list). The second neighbor in the list is picked next and the same algorithm is again applied to form a second forwarding set of nodes and so on until all the nodes are checked. Table 3.2 shows all the 16 resulting forwarding node sets. This approach results in sets that are subsets of each other. All subsets are eliminated to form the filtered forwarding node sets. Table 3.3 shows that the number of the forwarding node sets is reduced to 5 sets after the elimination process. To achieve optimal network coverage, the forwarding node sets with the maximum number of nodes (2 sets in this case) are chosen as the final forwarding node sets as shown in Table 3.4.

Table 3.1- Neighborhood table of node (12)

NB(12)	**NB(NB(12))**					
10	6	8	1	5	7	
13	19	21	7	20	22	24
14	15	18	19	25	27	8
19	14	26	24	25	13	
7	21	23	10	13	22	5
8	14	10	15	18	6	9

Table 3.2- Initial forwarding node sets

Forwarding set number	Nodes
0	10 13 14
1	10 14
2	10 19
3	10
4	10
5	13 14
6	13
7	13
8	13 8
9	14 7
10	14 7
11	14
12	19 7 8
13	19 8
14	7 8
15	8

Table 3.3- Filtered forwarding node sets

Forwarding set number	Nodes
0	10 13 14
1	10 19
2	13 8
3	14 7
4	19 7 8

Table 3.4- Final forwarding node sets

Forwarding set number	Nodes
Set 1	10 13 14
Set 2	19 7 8

3.2. Important Node Models

The network model consists of three basic mobile nodes: *Sending node*, *Intermediate* or *Relay node* and *Receiving node*. For the purpose of conducting this research each node will perform a designated function (sender, relay or receiver). Figure 3.7 shows a possible network topology with different nodes having different roles. Nodes marked with *S* represent a sender node, nodes marked with *D* represent a receiver node and nodes marked with *R* represent a relay node. The following section describes the function of the above-mentioned nodes in detail.

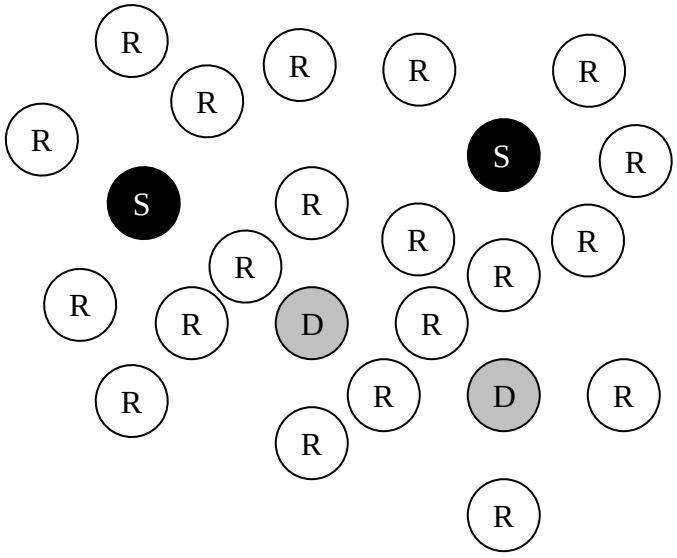


Figure 3.7- Example of a network model

3.2.1 Sending Node

The sending node initiates the route discovery algorithm in response to a communication request from a higher layer. The node creates a RREQ packet and populates the header with the destination address as well as the list of the forwarding

node set that are supposed to transmit the RREQ packet further. The node also sends *NB* packets for neighborhood updates as described in 3.1.1.

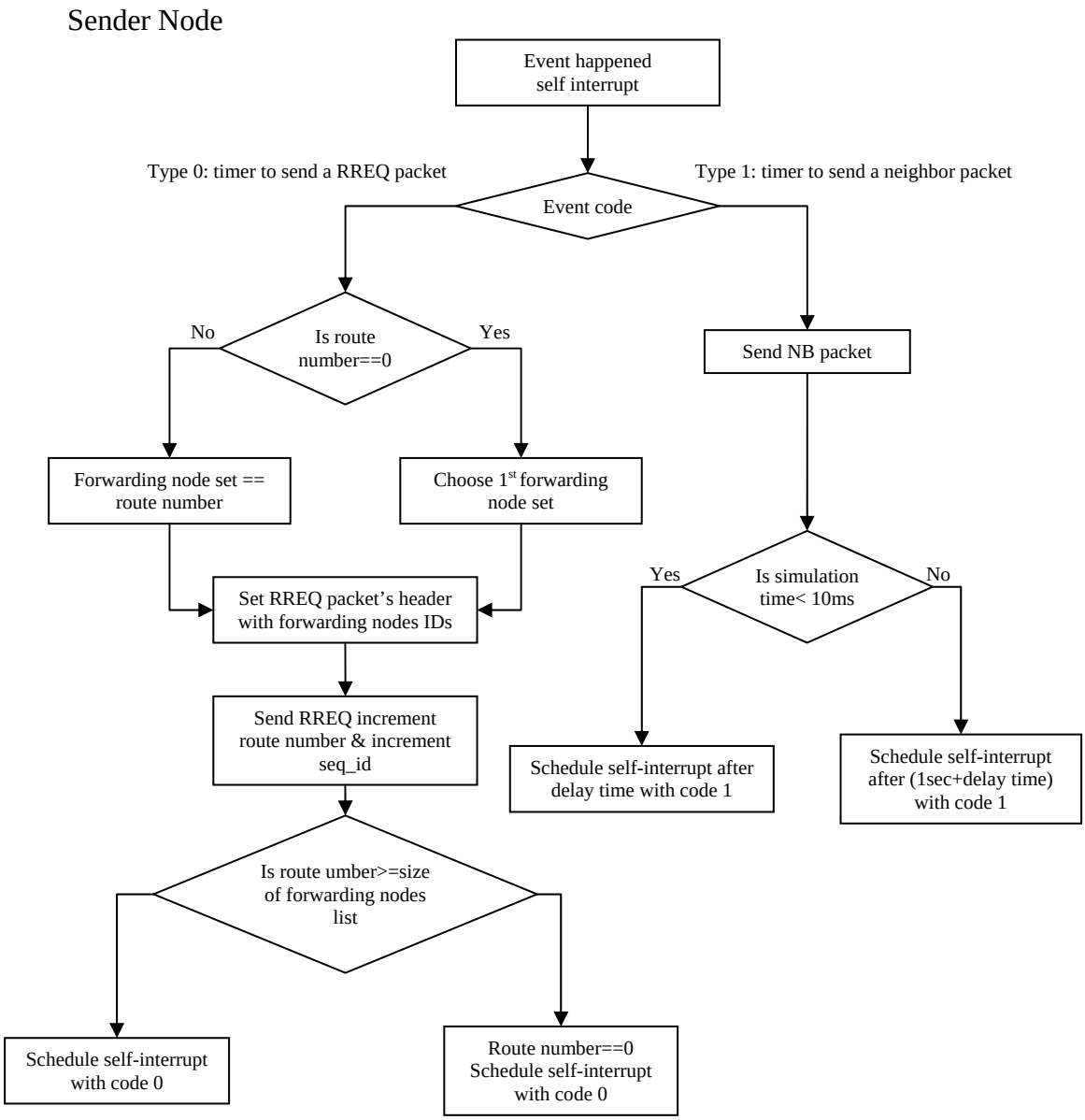


Figure 3.8- The algorithm followed by the sending node

3.2.2 Relay Node

The relay node or the intermediate node is a node that lies in between sender and receiver nodes. If the relay node id is listed in the forwarding node list, it will transmit the first copy of the RREQ. The relay node adds its own node id in the packets header to build the route. Furthermore, the relay node adds its list of the forwarding nodes set to the RREQ header. The node also sends *NB* packets for neighborhood updates as described in section 3.1.1. Figure 3.9 shows the algorithm followed by the relay node.

3.2.3 Receiving Node

The receiving node is the final destination that the original source node wants to communicate with. The receiving node does not retransmit route request packets. The node examines the destination id of the received packet header, if there is a match with its own id, the packet is accepted; otherwise the packet will be dropped. Upon accepting the RREQ packet, the receiving node creates a route reply packet with the original sender as the destination. The nodes listed in the RREQ packet header are used as a reverse route. The node also sends *NB* packets for neighborhood updates as described in 3.1.1. Figure 3.10 shows the algorithm followed by the receiving node.

Relay Node

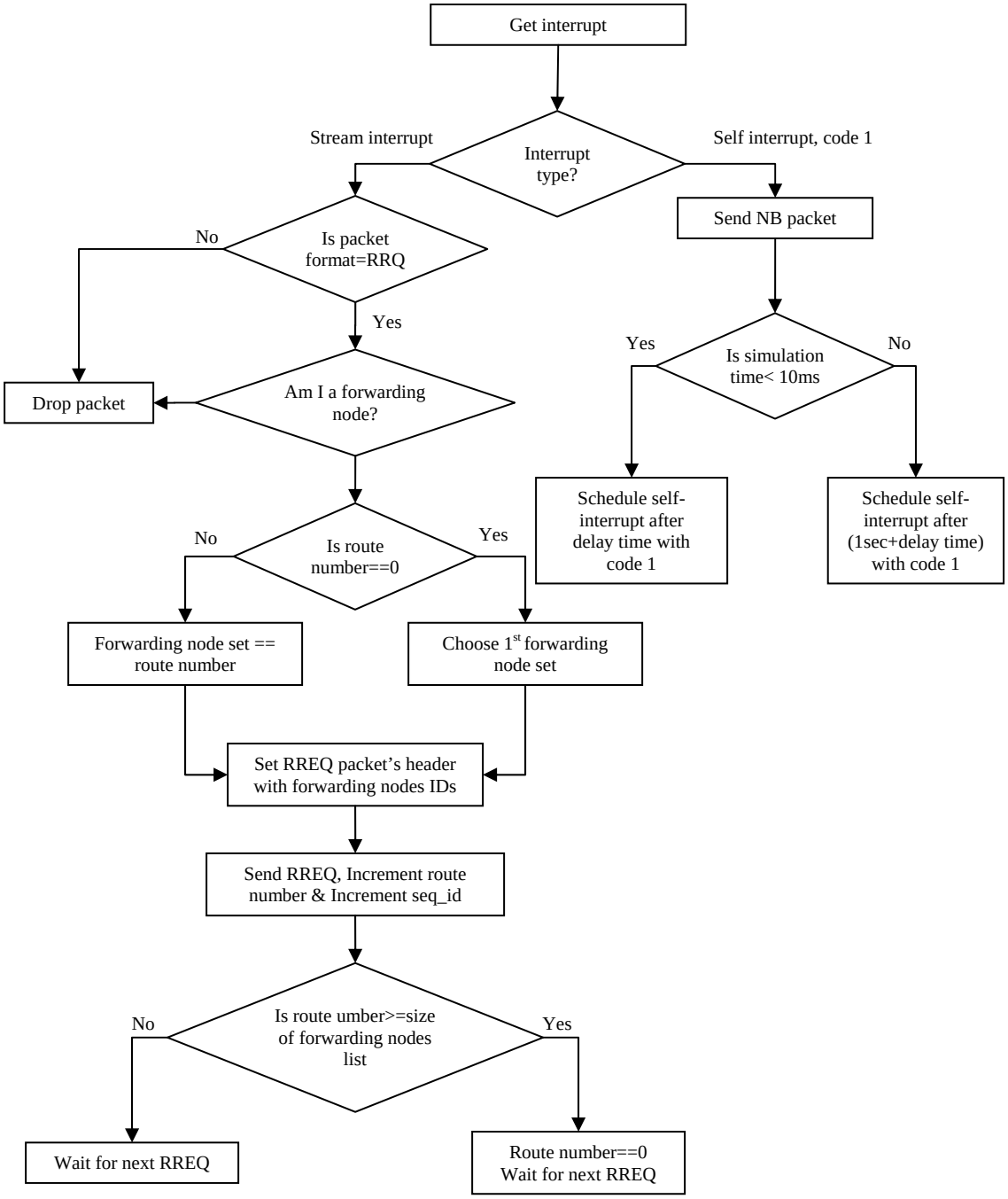


Figure 3.9- The algorithm followed by the relay node

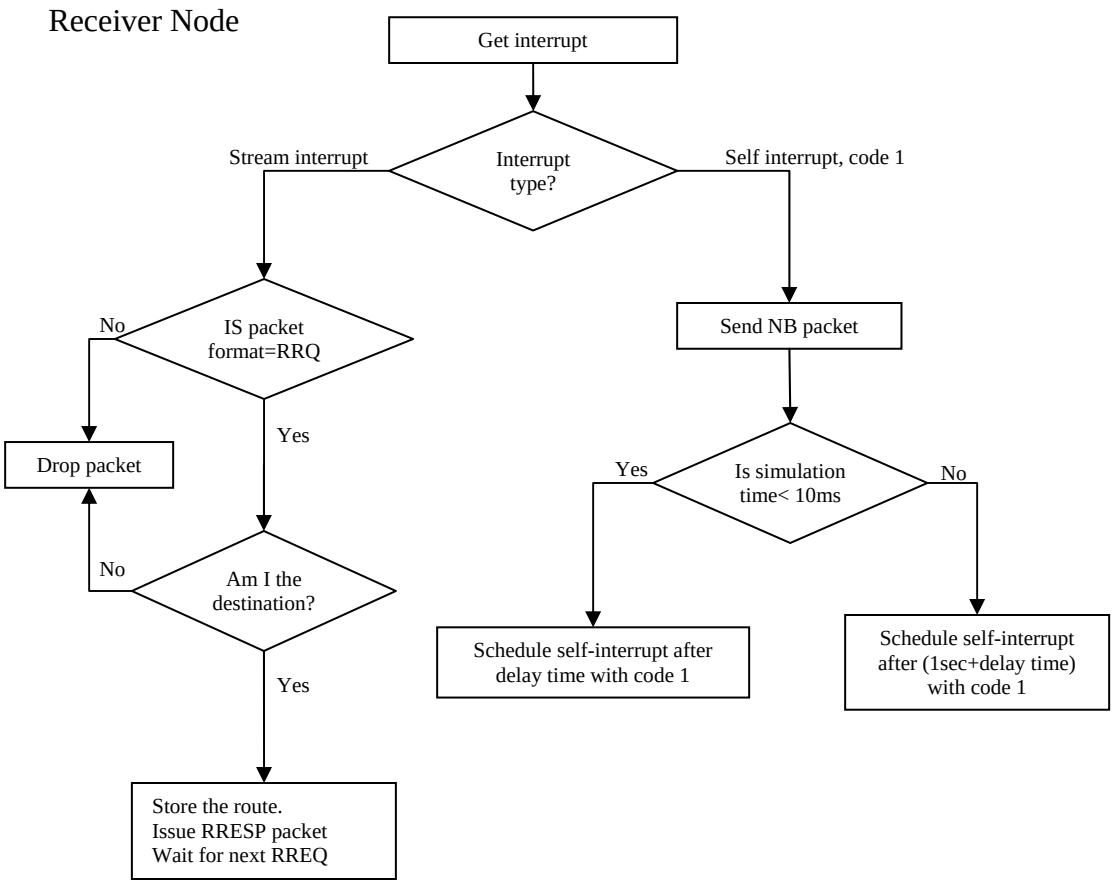


Figure 3.10- The algorithm followed by the receiving node

3.2.4 Medium Access Process

Wireless media is a shared medium which implies that contentions and collisions are common. To carry out the simulation for the proposed algorithm, carrier sense collision avoidance scheme is employed. When the MAC process receives a packet for transmission, it will be queued for a random time. When the timer is fired and it is the time to send the packet, the channel is sensed. If it is free, the node sends the packet. Otherwise, the packet is queued again for a random time before another attempt is made. The random time that the packet will stay in the queue is calculated by generating a random variable from a uniform distribution between (0, 0.5). The random value is then

multiplied by integer value equals to 60 (found through simulation). The result is then multiplied by the slot time which equals to 0.000023 seconds. Figure 3.11 shows the algorithm the MAC process follows.

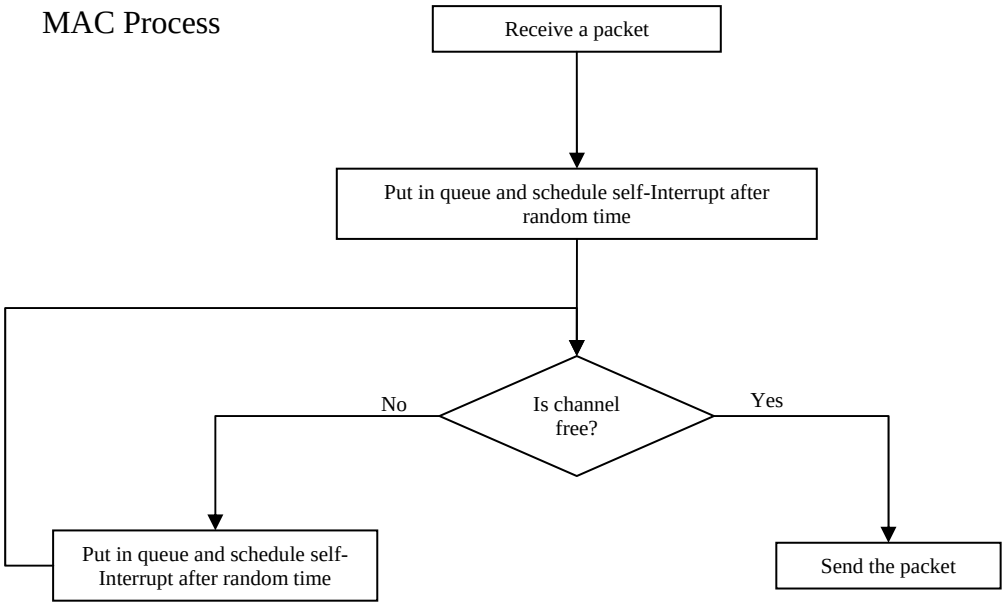


Figure 3.11- The algorithm followed by the MAC process

3.3. Important Packets Format

Several packet formats were used in this model as described below:

3.3.1 Route Request (RREQ)

This packet is created at the original source when the route discovery algorithm begins. The size of this packet is 256 bits to accommodate for the different fields. The structure of the packet and its explanation are shown in Figure 3.12 and Table 3.5, respectively.

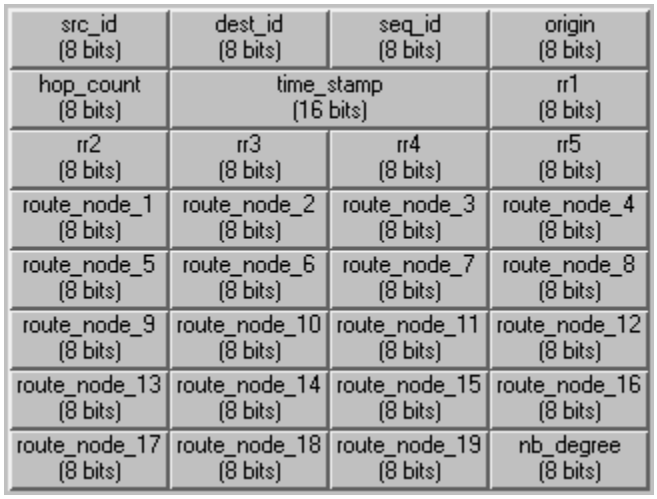


Figure 3.12- RREQ packet format structure

Table 3.5- RREQ packet fields description

Field	Explanation
source_id	Id of the sending node
dest_id	Final destination id
seq_id	Unique sequence number
Origin	Id of the original source, initiated the RRQ
hop_count	The number of hops this RRQ has traversed
time_stamp	The creation time of this RRQ at the origin
rr1-rr5	The forwarding node set ids
Route-node_1-19	The node ids of the forwarding nodes
nb_degree	Neighbor distance

3.3.2 Route Reply (RRESP)

This packet is created by the final destination node upon receiving a successful RREQ packet. The size of the packet is 160 bits to accommodate the various fields. The packet format structure and its description are shown in Figure 3.13 and Table 3.6, respectively.

Figure 3.13- RRESP packet format structure

Table 3.6- RRESP packet fields description

Field	Explanation
src_id	Id of the sending node
dest_id	Final destination node id
seq_id	The sequence id of the received RREQ packet
origin	The original source node id, issued the RRESP packet
1- 12	Node ids of the route
next-hop	The id of the next node that should forward the RRESP packet
time-stamp	The creation time of the RRESP packet

3.3.3 Neighbor Packets (NB)

The neighbor packet is created and sent by all nodes in the network. The purpose of this packet is to update neighborhood information among the neighboring nodes. The size of this packet is small (64 bits); it has one field for the source id and fourteen other fields to accommodate each node's neighbors. The frequency of sending this packet depends on the startup time; in the first 10 ms the node sends frequent (4-5) *NB* packets to quickly populate the neighborhood table, after the initial 10 ms the frequency decreases to 1 *NB* packet per second. Figure 3.14 and Table 3.7 show the structure of the packet format and its description, respectively.

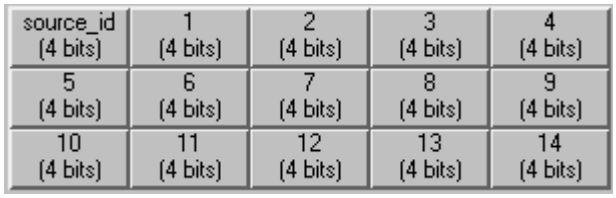


Figure 3.14- NB Packet format structure

Table 3.7- NB packet format description

Field	Explanation
Source_id	The id of the sending node
1- 14	The ids of the neighbor nodes of the sender

Chapter 4 Simulation Results and Comparisons

In order to verify the proposed algorithm, a reliable simulation tool is needed. The choice was on OPNET which is the de-facto standard in industry [27]. It is extensively used as a network technology development environment for designing and developing wired and wireless networks, devices, and communication protocols. Some of the features of the modeler are: highly scalable simulation engine, object-oriented modeling, finite state machine modeling, mobility modeling, integrated analysis tools, and many more. OPNET modeler provides a series of hierarchal models which mirror the structure of real-time networks, devices, and protocols. Each model is associated with specialized editors to allow development, modifications, and configurations at each specified hierarchical level. The wireless module offers comprehensive end-to-end performance analysis by modeling and simulating network topology, traffic, protocols, and end-user applications.

4.1. Simulation Model

Based on OPNET hierarchy, the simulation model consists of a network model that has a number of mobile wireless node models, which represents the entire network to be simulated. The number of the nodes ranges from (20-100) nodes depending on the simulation scenario. Each node has a designated function, a sender, a relay or a receiver node. Figure 4.1 shows a network topology where the yellow nodes represent sender nodes, red nodes represent rely nodes, and the blue nodes represent receiving nodes. The nodes are randomly thrown in a square area of 1000 m x 1000 m. Figure 4.2 shows the node model which represents the architecture of the network objects defined in the

network model. It provides the actual functionality and behavior of the network objects using functional elements called “node modules”. Each node model has two node modules or processes, routing process and medium access process (MAC) as well as transmitter module and receiver module. The different modules of the node model are connected by arrows that have different representations. Stream lines for packet movement between processes are represented by the blue arrows. Red arrows represent a statistical wire that connects the MAC process to the transceiver for channel detection.

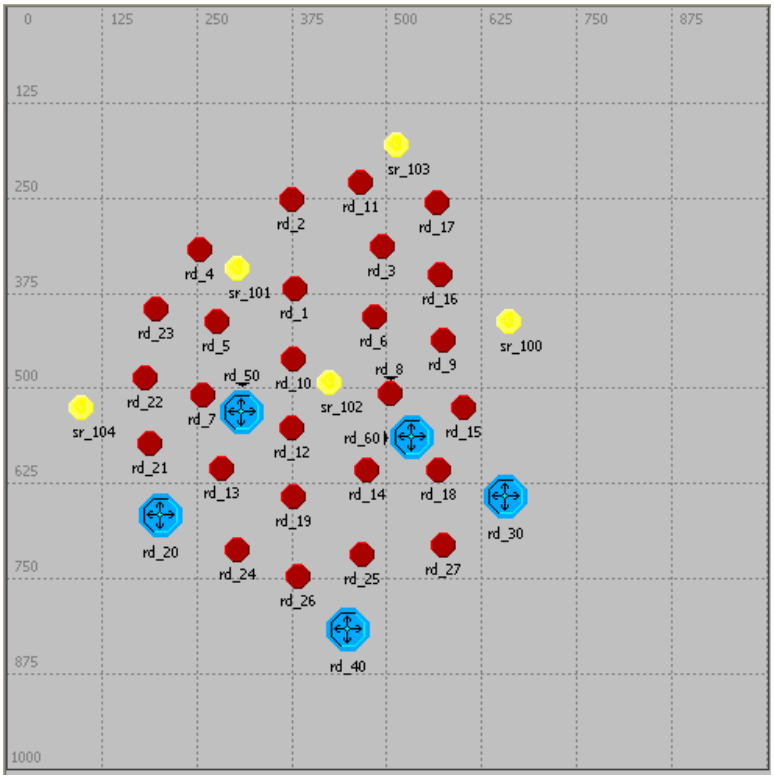


Figure 4.1- Network model example

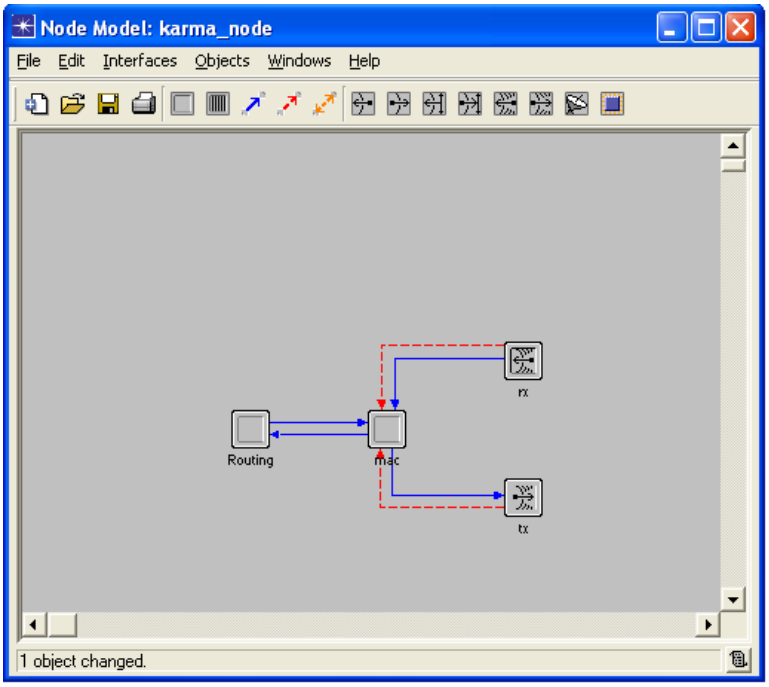


Figure 4.2- Node model

Process model shown in Figure 4.3 defines the actual functionality of the node model. It uses a finite state machine (FSM) approach to support the specifications and provide implementation of the protocols. The process model is interrupt-driven, the states and transitions define the progression of the process in response to events. The states are either forced or unforced. Forced states are non-blocked states and execute to completion and are represented by green circles. Unforced are usually blocked states between the executions of a state and are represented by red circles. Each state contains C code as shown in Figure 4.4, supported by extensive library of functions designed for protocol programming.

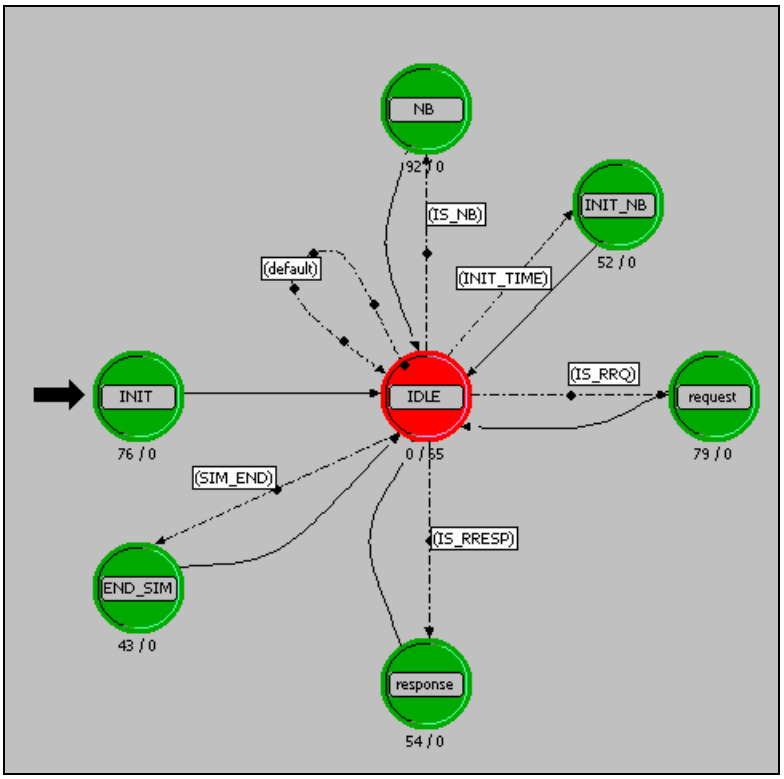


Figure 4.3- Process model

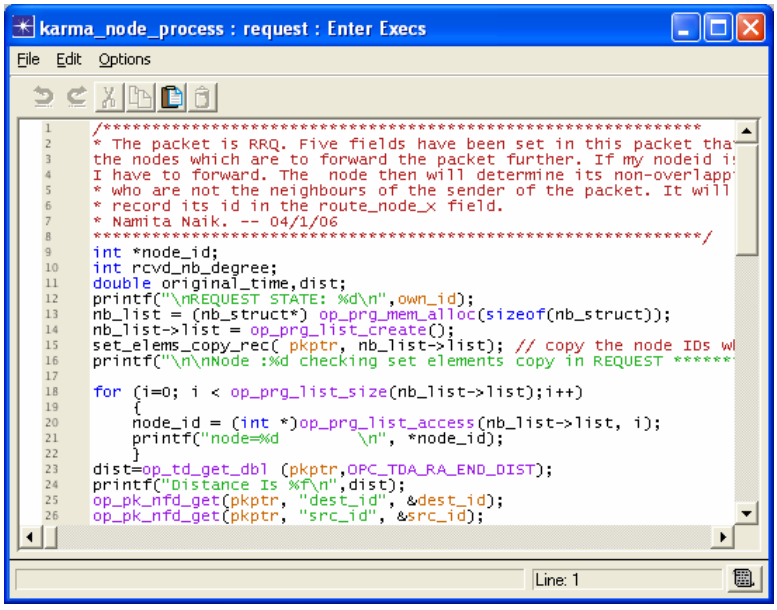


Figure 4.4- C code of the finite state machine

4.2. Simulation Parameters

The parameters are chosen, either empirically or by studying prior simulation models. Table 4.1 shows the various simulation parameters used to run the simulation model.

Table 4.1- Simulation parameters

Parameter	Default value
Simulation area	1000m X 1000m
Data rate	11Mbps
Simulation time	180 sec
Number of nodes	20- 100
Transmission range	160m
NB packet frequency	1packet/sec
Seed values	113, 128, 131, 409, 917

4.3. Simulation Results

The following subsection shows the effect of the simulation parameters on the results. Different simulation scenarios have been used to verify the proposed algorithm. The values obtained from five different seeds were averaged and plotted. The following metrics used for performance evaluation are borrowed from [11]-[13]:

- Efficiency: measures the number of RREQs transmitted to discover routes between a source and a destination.
- Network coverage: the percentage of the nodes who heard the route request. For example, if there are 28 nodes in a network and the source sends an RREQ. If 25 nodes receive this RREQ, then the coverage is 25/ 27 = 92.59%.
- Latency: the end to end delay in seconds to establish a route between a source and a destination.

- Route quality: the quality of the routes is evaluated for repeated route request scenario. Disjoint and non-overlapping routes are considered a better quality routes.
- Mobility: the performance of the algorithm is tested for mobile networks.

4.3.1 Efficiency

The graph in Figure 4.5 shows the total number of route requests obtained from simulating 28, 53 and 100 nodes. For 28 nodes, the total number of RREQs transmitted varies from 22 for one sender/ one receiver pairs to 42 for three sender/ receiver pairs. The number of RREQs for 53 nodes varies from 50 for one sender/ one receiver pairs to 127 for three senders/ receiver pairs. For the 100 nodes topology, route setup required 262 RREQs for three sender/ receiver pairs.

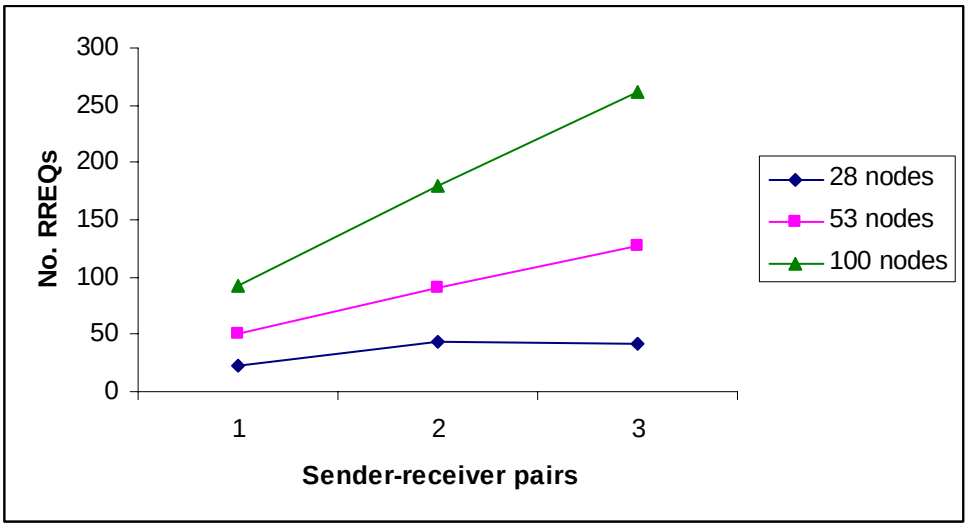


Figure 4.5- Total number of retransmission

The number of route requests increases with the number of nodes in the network. The number of sender/ receiver pairs also has major influence on the number of transmitted RREQs. Applying node distance concept decreased the total number of RREQs to 50% as

shown in Figure 4.6. The average number of RREQS decreased from 22.6 in 28 nodes to 16.4 transmissions. For 53 nodes, the average number of transmissions dropped to 28.6 and for 100 nodes the average number of RREQs dropped down from 92.8 to 51.8.

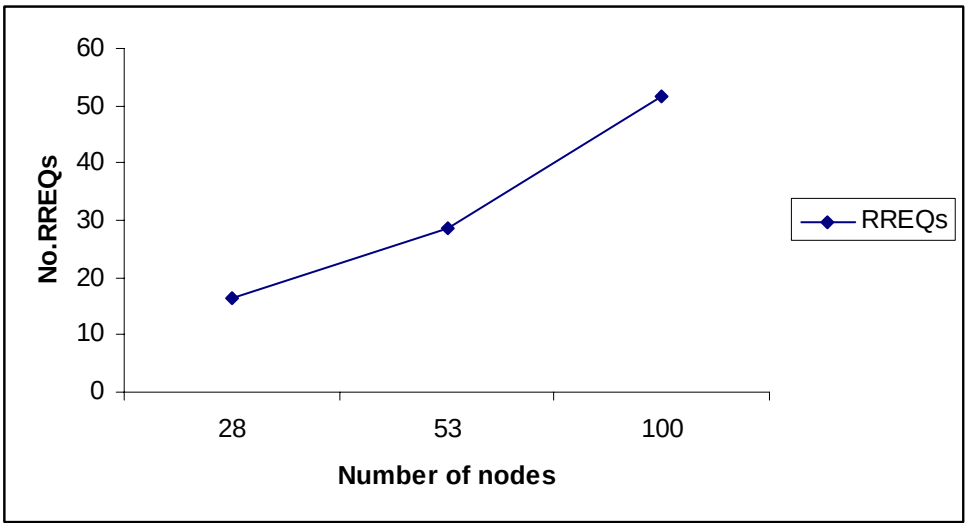


Figure 4.6- Number of RREQs using node distance

4.3.2 Network Coverage

Figure 4.7 shows plots for the network coverage for different number of nodes. The values are for one sender/ receiver pair in a static network topology. Coverage of around 100% is achievable. For 28 nodes, collisions cause a drop of the coverage to 99%. The affect of high network coverage is reflected on the number of routes discovered. The graph in Figure 4.8 shows the number of discovered routes for one sender/ one receiver pairs. An average of 1.6 routes is discovered for 28 and 2.2 routes for 53 nodes. The number of discovered routes increases with the number of nodes, due to the increase of the forwarding nodes sets. The average routes discovered for 100 nodes are 5.4 routes.

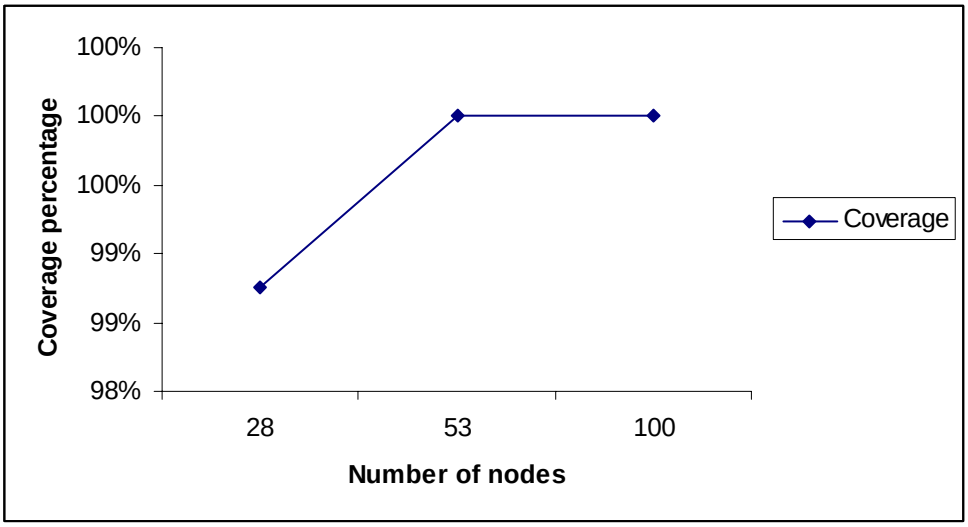


Figure 4.7- Coverage for different number of nodes

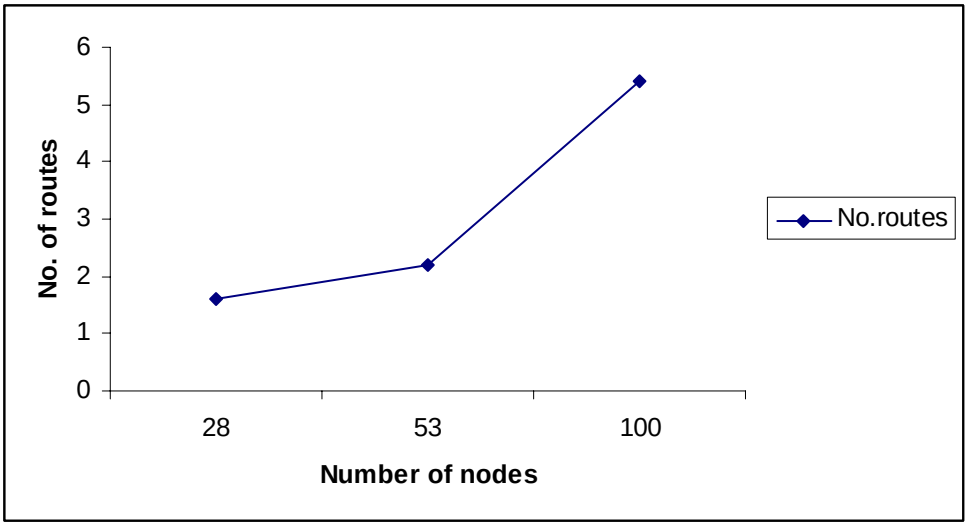


Figure 4.8- Number of discovered routes for one sender/ receiver pair

The graph in Figure 4.9 shows the routes discovered for each sender in three sender/ receiver pairs. The senders send their route request message at the same time. For 28 nodes the discovered routes for the first sender is 0.4 and 1.2 and 1.6 for the second and third senders respectively. The average discovered routes for the first sender in 53 nodes is 1.6 and for the second and third senders are 3.2 and 4.2 routes respectively. For 100 nodes, routes discovered for the third sender reaches to 6 routes. The number of

discovered routes increases for the second and third senders, and decreases for the first sender due to collisions.

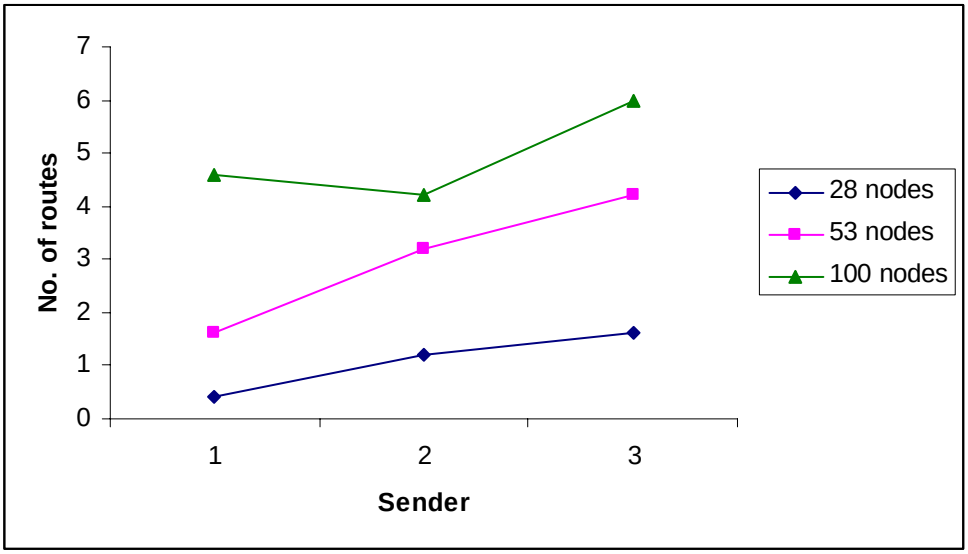


Figure 4.9- Number of discovered routes per sender

4.3.3 Latency

The graph shown in Figure 4.10 shows the route setup time for different node numbers. The route setup time represents the time taken from sending the route request till the reception of the route response message. The route setup time ranges from 0.003549 seconds for 28 nodes to 0.004612 for 100 nodes. The latency increases with the number of nodes in the network due to high traffic. As shown in Figure 4.8, the average number of the discovered routes is 5.4 in 100 nodes. This also contributes to the higher setup time.

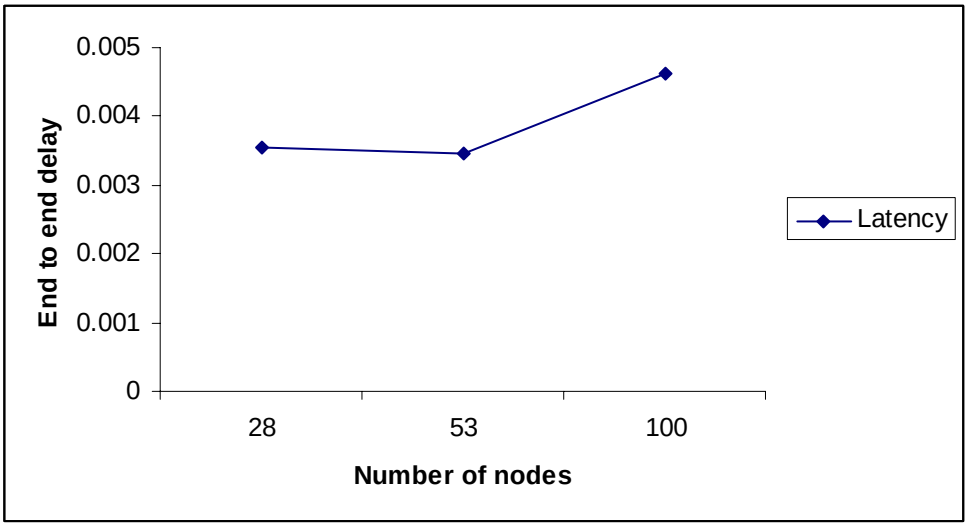


Figure 4.10- End to end delay for one sender/receiver pair

Figure 4.11 shows the latency in route setup for a three senders/ receivers scenario. For 28 nodes the average route setup time is 0.002023 seconds for the first sender. The second sender average route setup time increases to 0.004706 seconds due to more discovered routes than the first sender. For the third sender the route setup time is 0,002377 seconds because of fewer collisions. The route setup time in 53 nodes ranges from 0.003537 seconds for the first sender to 0.006183 seconds for the third sender because of the large number of discovered routes. The same setup time trend is also observed for the 100 nodes scenario. The route setup time varies with the number of discovered routes for each sender and with the number of nodes in the network as was shown in Figure 4.10.

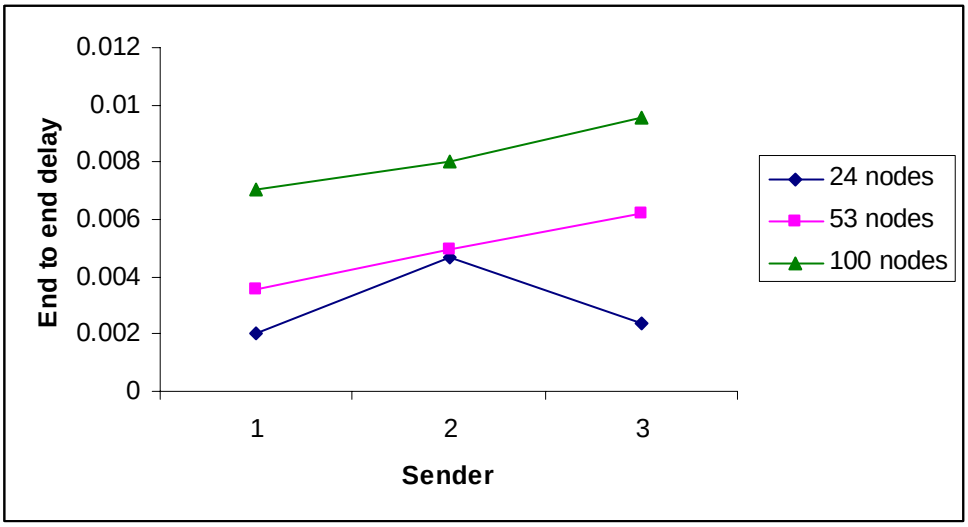


Figure 4.11- End to end delay per sender

4.3.4 Collisions

The graph in Figure 4.12 shows plots for lossy collision values for various node numbers. Lossy collisions happen at any node that belongs to the forwarding nodes set. Collision values increase with the number of nodes in the network due to high traffic. For 28 nodes with one sender/ receiver an average of 1.6 lossy collisions are present and an average of 5.6 lossy collisions for a network with 3 sources. The average of lossy collisions for 53 nodes is 8.2 for one source and 23.2 for a 3 sources network. The number of nodes and the number of sources greatly increase the average number of collisions for the 100 nodes with 3 sources.

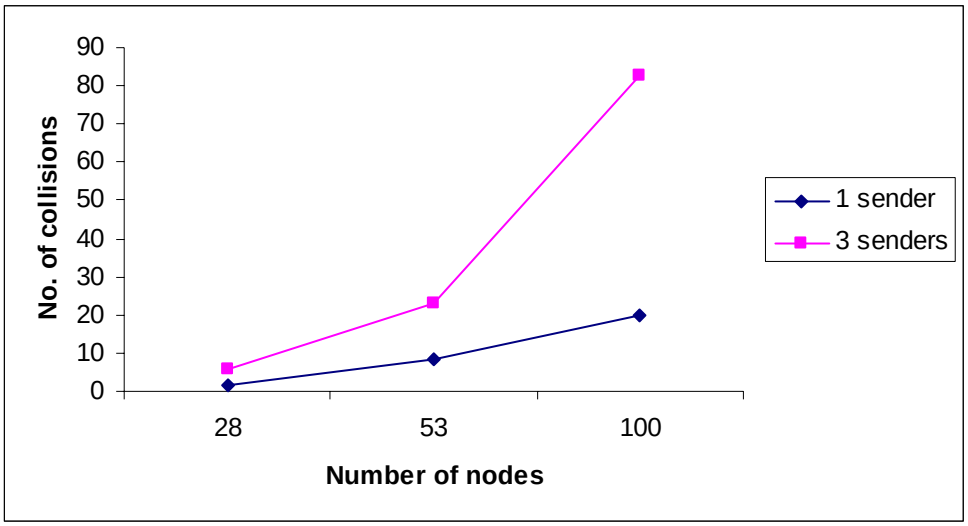


Figure 4.12- Lossy collisions

4.3.5 Quality of Discovered Routes

One of the main features of the proposed algorithm is its ability to provide different routes by broadcast alternation among the forwarding node sets. Figure 4.13 shows the routes formed from sending two successive route requests. The first route request is sent at time 10 seconds of the simulation time. The second route request is sent after 20 slot times (20*0.000023sec). The routes are disjoint and do not follow the same path. For the first route request message routes use the path shown in Figure 4.14.a whereas routes formed from the transmission of the second route request message are shown in Figure 4.14.b. The nodes that form the routes are shown in Table 4.2. There is an average of two routes per each route request; the average route setup time is around 0.005213 seconds for the first route request whereas it is a little bit higher (around 0.0353 seconds) for the second route request. The elevation of route setup time is due to higher network activity of sending route requests and receiving replies. The network coverage

and route request reachability is maintained to 100% as shown in Table 4.3. An average of 45 RREQs transmitted during the simulation time to discover the routes is also shown.

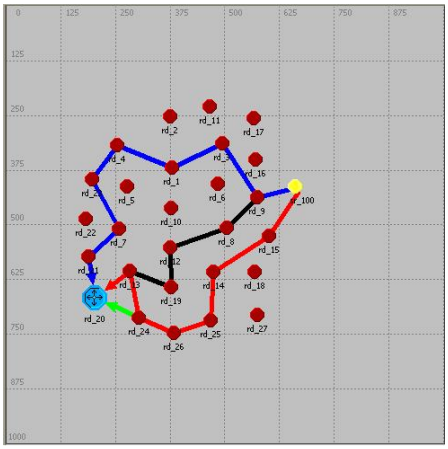
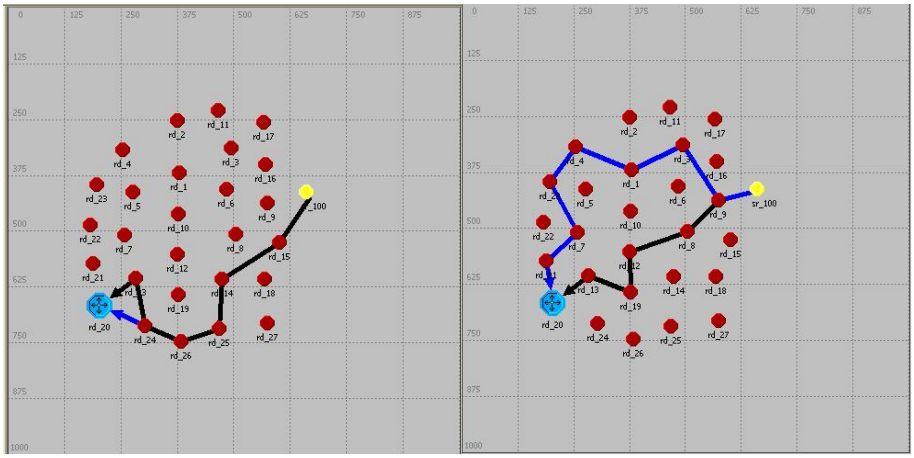


Figure 4.13- Route paths formed from sending repeated RREQs



a) First RREQ b) Second RREQ

Figure 4.14- Routes paths formed from each RREQ

Table 4.2- Routes resulting from sending repeated RREQ scenario

Seed	Number of routes	Nodes	Route setup time
113	4	16 6 8 10 7 13	0.004118
		16 6 8 10 7 13 24	0.006308
		9 15 14 19 13	0.042352
		9 16 17 3 1 4 23 7 21	0.050330
128	4	15 14 12 13	0.006600
		15 14 12 13 24	0.008311
		9 8 12 7 21	0.010979
		9 8 12 7 21 22 13	0.011676
131	4	9 8 14 12 13	0.005772
		9 8 14 12 13 24	0.007424
		16 6 8 12 7 21	0.009752
		16 6 8 12 19 13	0.011235
409	4	15 14 25 26 24	0.005090
		15 14 25 26 24 13	0.005719
		9 8 12 19 13	0.011702
		9 3 1 4 23 7 21	0.017786
917	3	15 14 25 26 24	0.005831
		15 14 12 13	0.006836
		9 3 1 5 22 13	0.008810

Table 4.3- Collisions and network coverage of repeated RREQ scenario

Seed	Total no. of RREQ	No. collisions in RREQ	Lossy collision	Coverage
113	46	0	0	100%
128	46	1	0	100%
131	47	1	1	100%
409	46	2	2	100%
917	43	7	5	100%

4.3.6 Mobility

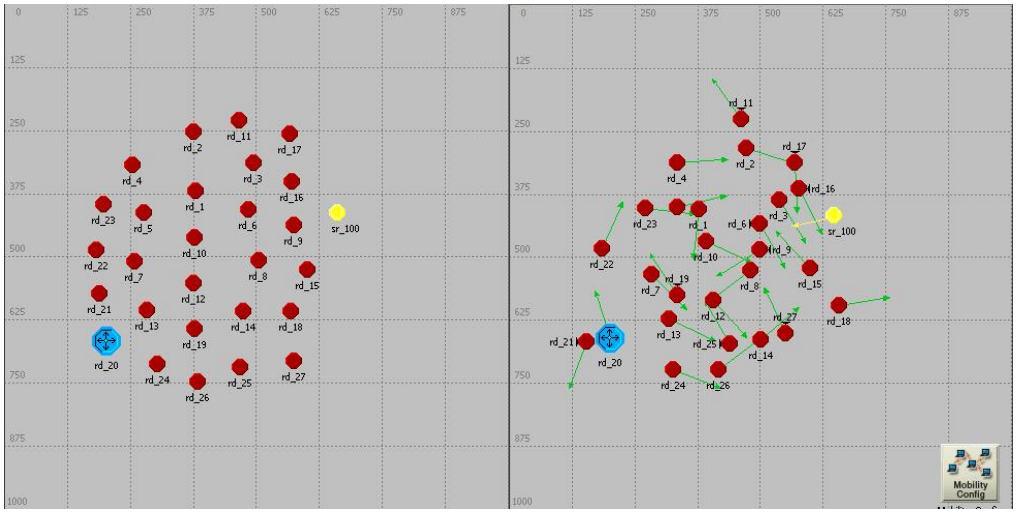
Mobility is an inherited feature of MANETs; nodes in the network area are allowed to move freely with a random speed. For nodes to communicate efficiently, routes have to be maintained and updated as necessary. This subsection discusses the results of simulating mobile nodes and the effects on the route discovery process. The random way mobility model is commonly used in MANETs [21], where random speed and destination are assigned for each node in the network. When the node reaches the predetermined destination, it pauses for preset time and then moves to another destination at either the same speed or a new speed. Low speed represents a slow moving node, whereas high speed signifies a highly mobile node. If the mobility area is small then nodes will move more frequently because they reach their destination faster. Pause time of zero represents continuously moving nodes whereas higher values of pause time represent slow network mobility. The proposed algorithm was simulated for a mobility scenario with a maximum speed of 20 meters per second which represents a vehicle moving at a speed of 45 miles per hour. Mobility region that span the entire simulation region and pause time of zero seconds are other parameters used in the simulation as summarized in Table 4.4.

Table 4.4- Mobility model parameters

Maximum speed	20m/s
Area	1000m X 1000m
Pause time	0 sec

Figure 4.15.a shows the topology before the start of movement and Figure 4.15.b shows the topology at the beginning of route request transmission. The simulation results shown in Table 4.5 illustrate that the route request reached the destination through three routes

with an average end to end delay of 0,002215 seconds. A total of 21 RREQ messages were transmitted during the simulation. It is clear from Figure 4.15.b that the neighborhood information has changed for every node in the network which required the frequent neighborhood table updates as mentioned in section 3.1.1. The changes are reflected on the average number of route requests that are sent. Some nodes move away from the transmission range and hence do not participate in the broadcast process.



a) Before mobility topology b) After mobility topology

Figure 4.15- Topology before and after mobility

Table 4.5- Results of simulating mobility scenario

No. routes	No. RREQ	Nodes	Delay
3	21	16 6 1 23 10 19	0.001896
		16 6 1 23 10 19 13	0.002231
		16 6 1 23 10 19 13 24	0.002517

The average number of route requests retransmissions decreases with the speed as shown in Figure 4.16. This is due to the random mobility scheme as well as collisions. For low speeds (0-5) meters per second, the trend resembles a static network (average of 20- 30 RREQs).

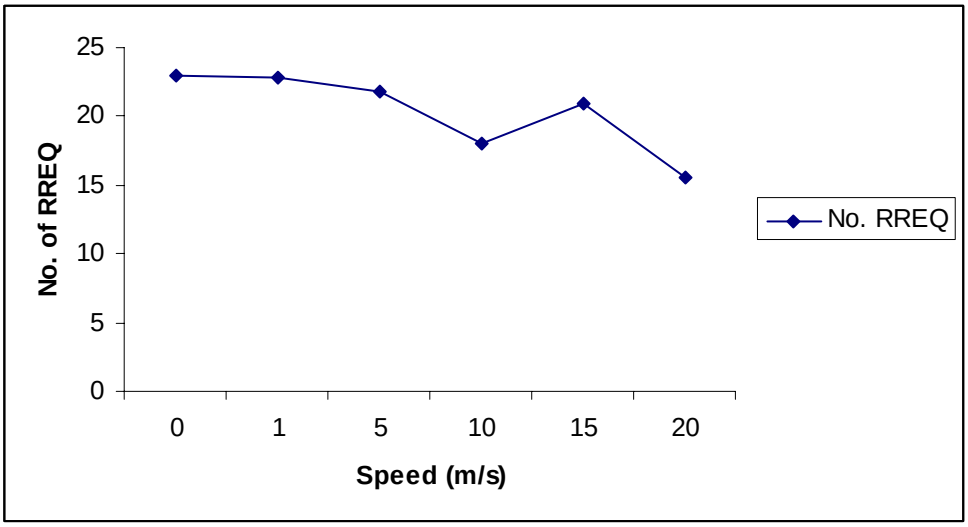


Figure 4.16- Average number of route requests versus speed

The network coverage is shown in Figure 4.17; higher speed decreased the reachability to 83% due to the decrease in the total number of retransmissions. For speeds (0-15) meters per second, reachability maintained to values more than 90%.

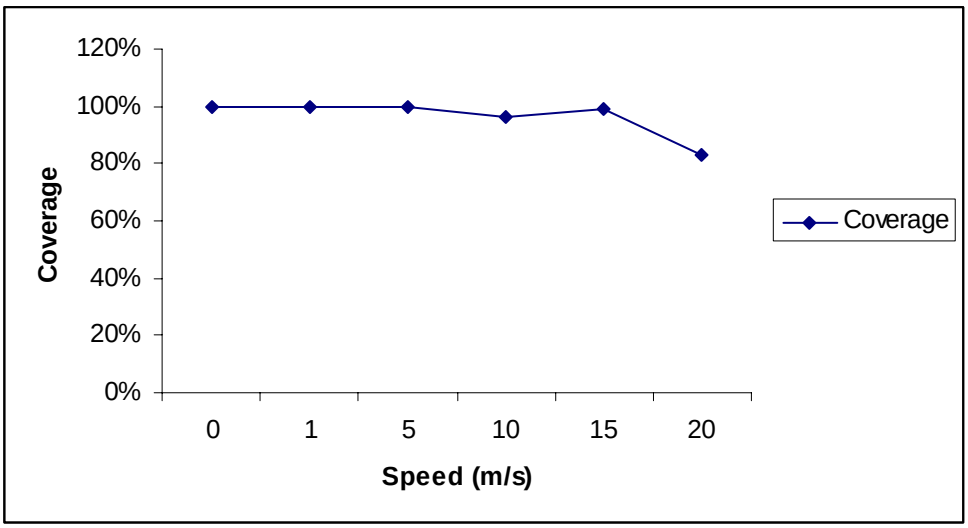


Figure 4.17- Network coverage for mobile scenario

4.4. Performance Comparison

Performance comparison of the proposed algorithm is done against two main flooding schemes; blind flooding [8], and flooding with self pruning [16]. They were discussed in details in section 2.1. The simulation parameters shown in Table 4.1 were used in the simulation. The proposed algorithm will be denoted by the letters (AF), simple flooding by the letters (SF), and flooding with self pruning by the letters (FSP) throughout this section. The performance metrics for comparing the algorithms are:

- **Efficiency**

The graph in Figure 4.18 shows the number of route requests transmitted to discover a route to a destination for a different number of nodes that ranges from 20 to 100 nodes; it is clear from the graph that the proposed algorithm has the least number of RREQs retransmissions. There is around 15% decrease in RREQs compared to simple flooding (SF), and an average of 10% decrease in RREQs compared to (FSP).

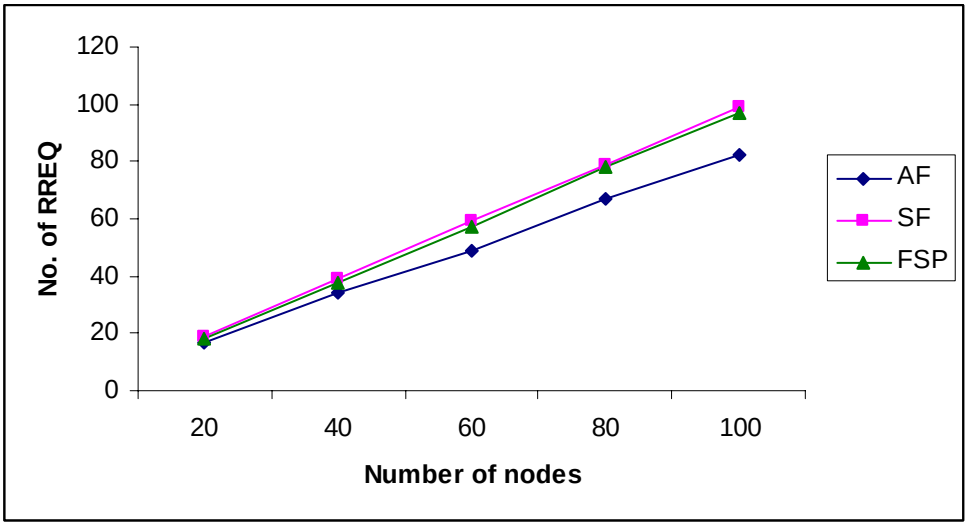


Figure 4.18- Number of route requests versus number of nodes

Applying the node distance concept discussed in section 3.1.5, substantial decrease of 45% is achieved in the transmitted RREQs of the proposed algorithm compared to (SF) and (FSP), respectively as shown in Figure 4.19 .

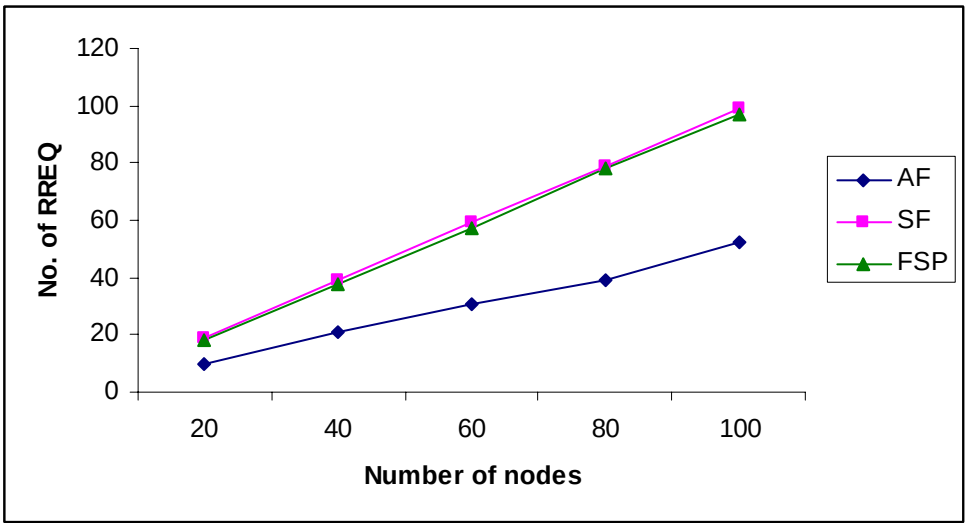


Figure 4.19- Number of RREQs transmitted using node distance

- **Network coverage**

The graph in Figure 4.20 shows the network coverage for the various algorithms. The proposed algorithm's reachability is 99% for 28 nodes and 100% for 53 and 100 nodes, respectively. The performance improved compared to simple flooding which achieves 100% reachability for all scenarios. The proposed algorithm outperforms (FSP) in a 28 nodes scenario.

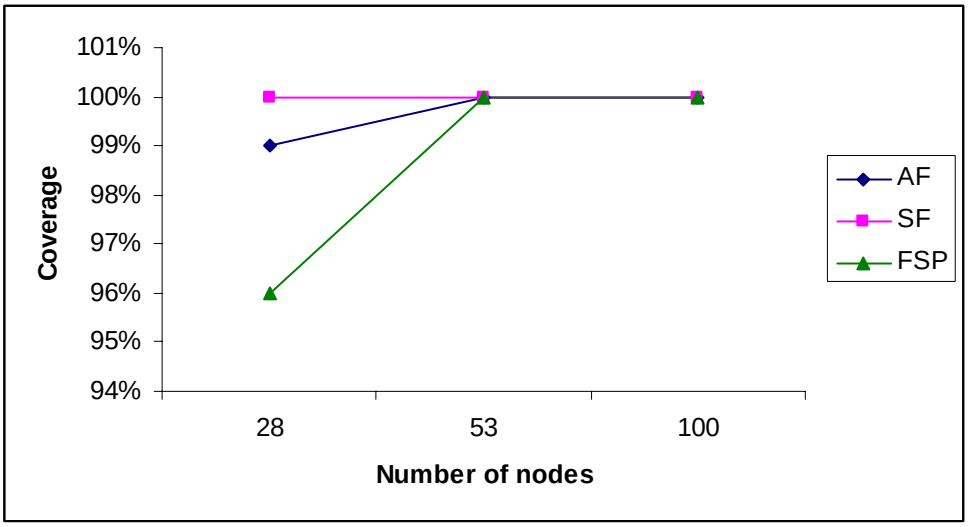


Figure 4.20- Coverage percentage among algorithms

- **Latency**

The graph in Figure 4.21 shows the route setup time for the algorithms. The proposed algorithm has a lower route setup time for 28 and 53nodes. For 100 nodes the route setup time reaches 0.004612 seconds. This value is still less than the simple flooding value. The reason for the higher setup time is due to increased traffic of the 100 nodes network.

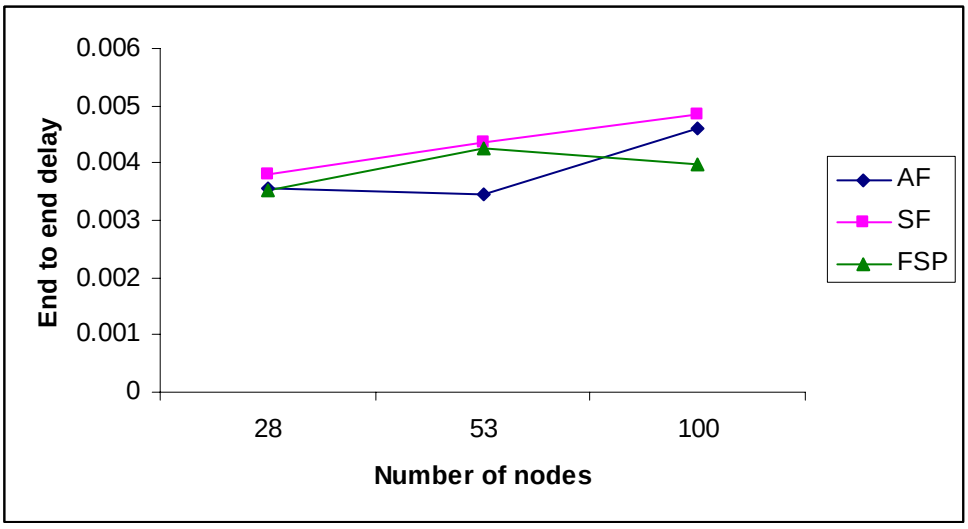


Figure 4.21- Latency for various algorithms

- **Collisions**

The proposed solution has a better performance in the number of collisions as shown in the graph in Figure 4.22. For 28 nodes the average number of collisions is 2.8 collisions. The average number of collisions for (SF) and (FSP) is 3.6 and 4.6, respectively. For 53 nodes, a higher performance is still achievable. The proposed algorithm achieves an average of 55% decrease in the average number of collisions for 100 nodes, compared to (SF) and (FSP), respectively.

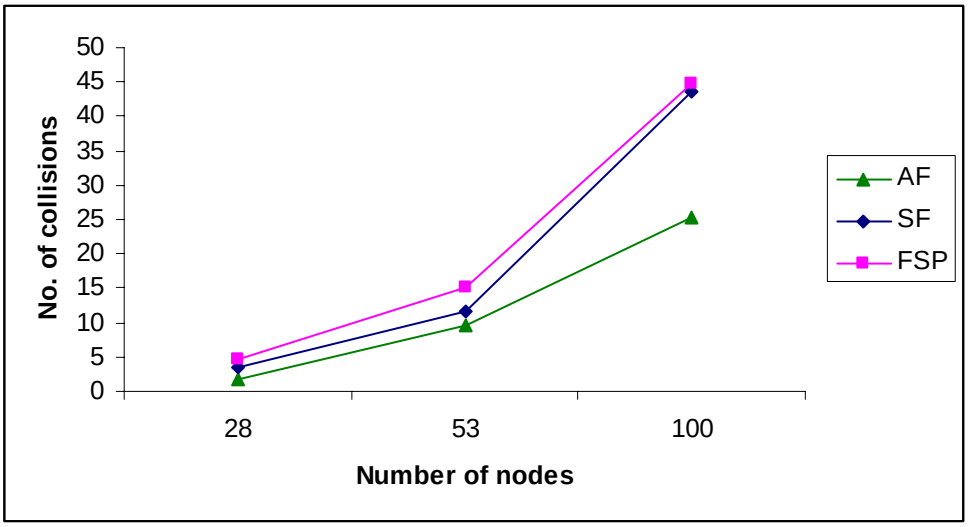


Figure 4.22- Collisions for various algorithms

- **Quality of discovered routes**

Repeated route request comparison which uses the topology in Figure 4.15.a is shown in Table 4.6, for example, node (100) has nodes (9, 15 and 16) as first-hop neighbors. In the proposed algorithm, for the first route request, nodes (15, 16) relay the message for node (100), whereas in (SF) and (FSP), nodes (9, 15 and 16) relay the message for node (100).

In the second route request only node (9) transmits the RREQ in the proposed algorithm, whereas in (SF) and (FSP) the same nodes (9, 15 and 16) transmit the message.

Table 4.6- Propagation of RREQ for node 100

Algorithm	First RREQ	Second RREQ
AF	15 16	9
SF	9 15 16	9 15 16
FSP	9 15 16	9 15 16

- **Mobility**

The graph in Figure 4.23 shows the average number of retransmissions for the different algorithms. The proposed algorithm has the fewest number of RREQs for the different speeds. The number of retransmissions decreases with the speed in the proposed algorithm; whereas the same average is maintained for (SF) and (FSP). This can be attributed to collisions and the random mobility scheme employed in the simulation.

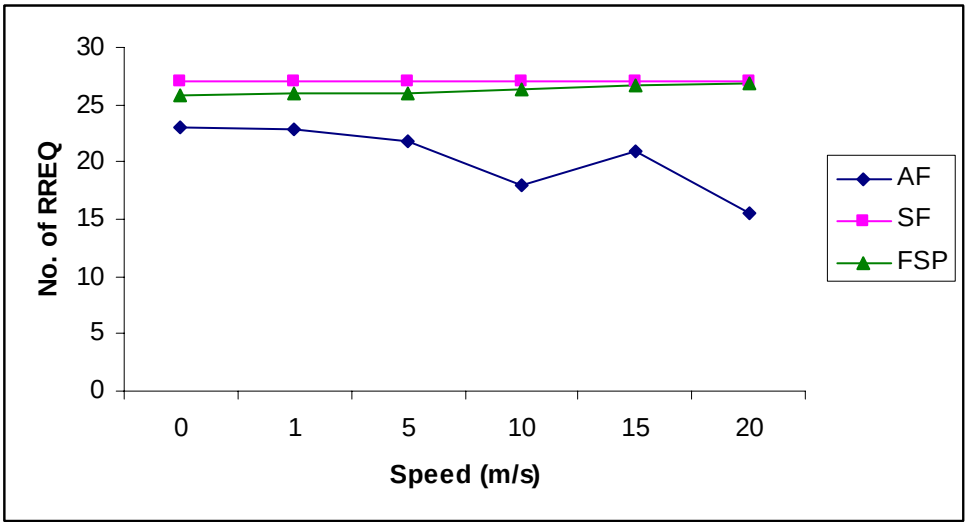


Figure 4.23- Number of retransmissions in mobile scenario

The network coverage is maintained to acceptable values compared to simple flooding as shown in Figure 4.24 . The coverage is maintained to values larger than 90% for speed ranges from 0 to 15 meters per second. Reachability drops down to 83% due to the decrease in the number of retransmission as was mentioned in Figure 4.23.

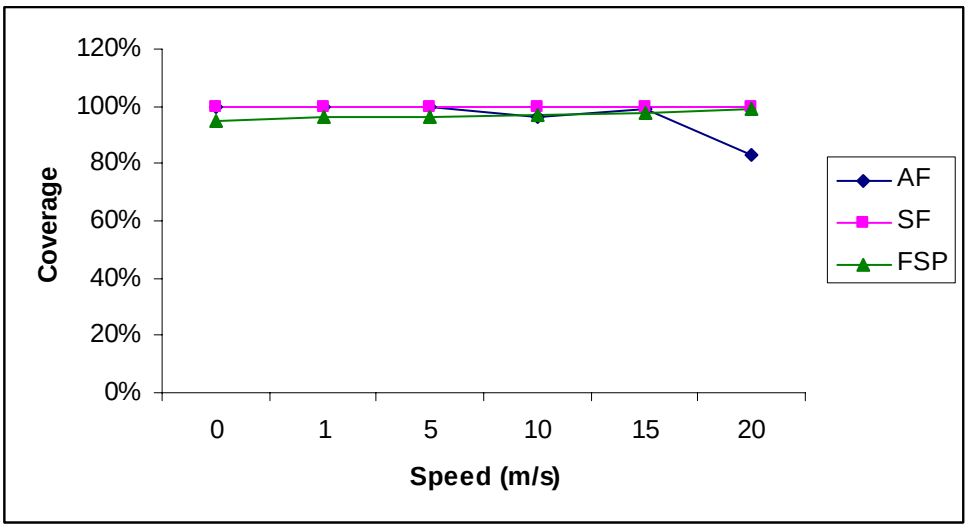


Figure 4.24- Network coverage of the different algorithms

The performance of the proposed algorithm is within the acceptable range of values compared to (SF) and (FSP). The proposed algorithm outperforms (SF) and (FSP) in handling repeated route requests as well as reducing the number of retransmissions.

Chapter 5 Conclusion and Future Work

Efficient broadcast is a flooding scheme that reduces the total number of control messages overhead while preserving optimal network coverage of route requests throughout the network. In this work, efficient broadcast is done by exploiting the neighborhood knowledge and forcing certain nodes to propagate the route request messages. Several route discovery algorithms use different broadcast schemes that exploit area, location, distance or neighborhood knowledge. Introducing the idea of reducing the total number of control traffic overhead as well as the quality of the discovered routes raise many challenges in algorithm design. Neighborhood knowledge is an accurate and reliable method for route discovery. In this work, neighborhood knowledge was used successfully to propose a solution for reducing the total number of route request broadcast in addition to discovering routes with better quality. The proposed solution was simulated using the network simulator OPNET which is an event driven simulator designed particularly to model and test network protocols and performance. Nodes in the network area are randomly placed. They keep track of their first hop neighbors in a local neighborhood table which is used for forwarding node sets calculation. A simple collision avoidance MAC was used to reduce packet collisions for correct performance evaluation. A high performance in decreasing the number of collisions is achievable compared to simple flooding and flooding with self pruning. The proposed algorithm reduces the volume of control traffic overhead. The proposed solution achieved the desired goal of alternating route request transmission among first hop-nodes. End to end delay in route request and the total time of the route discovery process was within acceptable range as proved by the comparison performed with simple

flooding and flooding with self-pruning. The routes discovered using the proposed solution were not the shortest path routes at all times; this can be improved by more in depth utilization of the neighborhood knowledge and applying more constraints on the retransmission criteria. The proposed algorithm uses source routing which may have high overhead when the number of nodes in the network gets large, the size of control packets will increase as the number of first-hop neighbors increases in dense networks. The high performance of the proposed algorithm is achieved as long as the number of repeated route requests does not exceed the number of the forwarding node sets, because there will be different forwarding set for each route request. When all forwarding sets are used the algorithm starts using the same sets again. Mobile scenarios simulation showed that the algorithm is efficient in mobile networks. Neighborhood table updates is an open issue for deciding on the best frequency for updates as well as the frequency of sending hello messages. A formal mathematical analysis has not been done, which can help in finding the optimal retransmission strategy. Future work may also involve, testing the algorithm in real ad hoc network as well as implementing the algorithm in routing protocols.

Bibliography

[1] S. Basagni, I. Stojmenovic, and S. Giordano, *Mobile Ad Hoc Networking*, Wiley-IEEE, 2004.

[2] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *in Proceedings of IEEE Special Issue on Packet Radio Networks*, vol. 75, pp. 21-32, Jan. 1987.

[3] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, "Issues in packet radio network design," *in Proceedings of IEEE Special Issue on Packet Radio Networks*, vol. 75, pp. 6-20, Jan. 1987.

[4] M. S. Corson, J. P. Macker, and G. H. Cirincione, "Internet-based mobile ad hoc networking," *IEEE Internet Computing*, vol. 3, pp. 63-70, Jul/ Aug. 1999.

[5] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC2501*, 1999.

[6] H. Zhou, "A Survey on Routing Protocols in MANETS," *MSU- CSE-03-08*, Mar. 2003.

[7] M. abolhasan and J. Lipman, "Self-selection route discovery strategies for reactive routing in Ad hoc networks," *in Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, p. 21, Nice, France, May 2006,

[8] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *in Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 151-162, Seattle, Washington, USA, 1999.

[9] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," *in MobiHoc'02: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 194-205, Lausanne, Switzerland, Jun 2002.

[10] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 479-491, June 2006.

[11] M. Jacobsson, C. Guo, and I. Niemegeers, "A flooding protocol for MANETs with self-pruning and prioritized retransmissions," *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 9, Nov. 2005.

[12] Y. Cai, K. A. Hua, and A. Phillips, "Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless Ad Hoc networks," *24th IEEE International Conference on Performance, Computing, and Communications*, pp. 347- 354, April 2005.

[13] C. Zhu, M. J. Lee, and T. Saadawi, "A border-aware broadcast scheme for wireless ad hoc network," First IEEE *Consumer Communications and Networking Conference*, pp. 134-139, Jan. 2004.

[14] Y. Qin, "A look-ahead unicast routing algorithm in MANETs," IEEE 61st *Vehicular Technology Conference*, vol. 4, pp. 2513- 2517, May/ June 2005.

[15] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: an efficient technique for flooding in mobile wireless networks," *Technical Report RR-3898, INRIA*, Mar. 2000.

[16] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," *MSWIM'00: Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 6168, Boston, MA, Aug. 2000.

[17] A. Qayyum, P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," *IEEE INMIC'01*, pp. 62- 68, Lahore, Pakistan, Dec. 2001.

[18] M. Gerla, X. Hong, and G. Pei, "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks," *In Proceedings of ICDCS Workshop on Wireless Networks and Mobile Computing*, pp.D71-D78, April 2000.

[19] C. Chiang, H. Wu, W. Liu, and M.Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," *In Proceedings of IEEE SICON*, pp. 197-211, April 1997.

[20] R.G. Ogier, M.G. Lewis, F.L. Templin, and B. Bellur, "Topology Broadcast Based on Reverse-Path forwarding (TBRPF)," Mar. 2002.

[21] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, 1996.

[22] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90- 100, New Orleans, LA, Feb. 1999.

[23] J. Raju and J. Garcia-Luna-Aceves, "A new approach to on-demand loopfree multipath routing," *in Proceedings of the 8th Annual IEEE International Conference on Computer Communications and Networks*, pp. 522-527, Oct. 1999.

[24] G. Aggelou and R. Tafazolli, “RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks,” *WoWMoM’99: Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia*, pp. 26-33, Seattle, Washington, USA, 1999.

[25] M. Jiang, J. Li, and Y. C. Tay, "Cluster Based Routing Protocol,” Internet Draft, 1999.

[26] Z. J. Haas, “A New Routing Protocol For The Reconfigurable Wireless Networks,” *Proceedings of 6th IEEE International Conference on Universal Personal Communications*, pp. 562- 566, San Diego, California, USA, October 12-16, 1997.

[27] http://www.OPNET.com.