IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. VII (May-Jun. 2014), PP 47-53
www.iosrjournals.org

Design and Implementation of a Simulator for Ad Hoc Network Routing Protocol

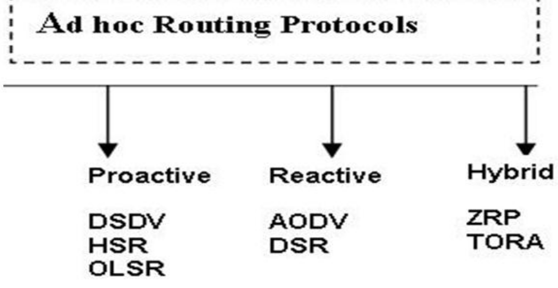
Priya Thakral , Manu Phogat
Department of Computer Science,GITM,Gurgaon

Abstract: *Ad hoc wireless network is defined as a collection of mobile platforms or nodes where each node is free to move. Each node is logically consisting of a router that may have multiple hosts and that also may have multiple wireless communications devices.An ad hoc wireless network has the features like autonomous terminals, distributed operation, multihop routing and dynamic network topology.As nodes are mobile and their battery power is less there are lots of possibilities to design a protocol. To test these routing protocols, there are some simulator like NS2 and Qualnet available. These routing protocols are augmented on NS2 and Qualnet for standard results. a power aware routing protocol methodology is proposed in this paper. The major aim of proposed work is to design a simulator for the routing protocols so that we will find out the some parameters like average number of nodes needed to get path from source to destination, average number of retransmission during finding the path from source to destination, average number of throughput and average power remaining of each node. In this work a routing protocol is designed in C++ and run on standard condition.*

I. Introduction

An ad hoc wireless network [16] consists of a set of mobile nodes (hosts) that are connected by wireless links. Since nodes in ad hco wireless are mobile, network topology in such a network may changes frequently. The routing protocols used in traditional wired networks can not be used for ad hoc wireless networks due to their highly dynamic topology, absence of established infrastructure for centralized administration, bandwidth constrained wireless links, and resource (energy) constrained nodes.MANETs are the collection of wireless nodes that can dynamically form a network anytime and anywhere to exchange information without using any pre-existing fixed network infrastructure. It is an autonomous system in which mobile hosts connected by wireless links are free to move randomly and often act as routers at the same time. This is a very important part of communication technology that supports truly pervasive computing, because in many contexts information exchange between mobile units cannot rely on any fixed network infrastructure, but on rapid configuration of a wireless connections on-the-fly. MANETs themselves are an independent, wide area of research and applications, instead of being only just a complement of the cellular system. Mobile Ad-Hoc Networks (MANETs) are wireless networks that continually re-organize themselves in response to their environment without the benefit of a pre-existing infrastructure. A MANET consists of a set of mobile participants who must communicate, collaborate, and interact to complete an assigned mission. The challenges of MANETs are to provide wireless, high-capacity, secure, and networked connectivity. Participants must communicate using bandwidth limited wireless links with potential intermittent connectivity, as compared to stable wired links and infrastructure. MANETs are a key enabler for achieving the goals of net-centric operations and warfare; they provide the right information at the right place at the right time.A MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communications devices)--herein simply referred to as "nodes"--which are free to move about arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices, and there may be multiple hosts per router. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network.

II. Ad-Hoc routing Protocols



2.1 Classifications Of Routing Protocols

The routing protocol for ad hoc wireless networks [30, 42, 46, 49 , 50] are classified as-

- **Table Driven Routing Protocols**

These protocols are extensions of the wired network routing protocols. They maintain the global topology information in the form of tables at every node. These tables are updated frequently in order to maintain consistent and accurate network state information.

- **Distance Sequenced Distance Vector (DSDV) Routing Protocol**

DSDV [5, 35] is one of the first protocols proposed for ad hoc wireless networks. It is enhanced version of the distributed Bellman-Ford algorithm where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network. It incorporates table updates with increasing sequence number tags to prevent loops, to counter the count-to-infinity problem, and for faster convergence.

As it is a table driven routing protocol, routes to all destinations are readily available at every node at all times. The tables are exchanged between neighbors at regular intervals to keep an up-to-date view of the network topology. The tables are also forwarded if a node observes a significant change in local topology. The table updates are of two types: - incremental updates and full dumps. An incremental update takes a single network data packet unit (NDPU), while a full dump may take multiple NDPUs . Incremental updates are used when a node does not observe significant changes in local topology. A full dump is done either when the local topology changes significantly or when an incremental update requires more than a single NDPU. Table updates are initiated by a destination with a new sequence number which is always greater than the previous one. Upon receiving an updated table, a node either updates its tables based on the received information or holds it for some time to select the best metric (which may be the lowest number of hops) received from multiple version of the same update table from different neighboring nodes. Based on the sequence number of the table update, it may forward or reject the table.

- **Reactive Rouitng Protocols**
- Reactive protocols, on the other hand, invoke a route determination procedure on demand only. Thus, when a route is needed, some sort of global search procedure is employed. The family of classical flooding algorithms belongs to the reactive group. e.g. **AODV** (Ad-hoc On Demand Distance Vector routing protocol), DSR (Dynamic Source Routing protocol)

- **AODV**

AODV [7, 24, 36] routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence number to identify the most recent path. The major difference between AODV and DSR stems out from the facts that DSR uses source routing in which a data node and the intermediate nodes store the next hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the RouteRequest packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single RouteRequest. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater than the last DestseqNum stored at the node.

A RouteRequest carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcseqNum), the destination sequence number (DestSeqNum), the broadcast identifier(BcastID), and the time to live(TTL) field. destSeqNum indicates the freshness of the route that is accepted by the source. When an intermediate node receives a RouteRequest, it either forwards it or prepares a RouteReply if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the RouteRequest packet. If a RouteRequest is received multiple times, which is indicated by the BcastId-SrcID pair, the duplicate copies are discarded. All the intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RouteReply packets to the source. Every intermediate node, while forwarding a RouteRequest, enters the previous node address and it's BcastID. A timer is used to delete this entry in case a RouteReply is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a RouteReply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop towards the destination.

- **DSR**

Dynamic source routing(DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The major difference between this and the other on-demand routing protocols is that it is beacon-less and hence does not require periodic hello packet (beacon) transmissions, which are used by a node to inform its neighbors of its presence. The basic approach of this protocol (and all other on-demand routing protocols) during the route construction phase is to establish a route by flooding RouteRequest packet, responds by sending a RouteReply packet back to the source, which carries the route traversed by the RouteRequest packet received.

III. Designing The Routing Protocols

The aim of the routing protocol is to find a path that consumes minimum battery power from source node to destination node. The strategy used in routing protocol design is as follows:

The simulator takes following inputs from the user-

- Number of nodes that a network contains.
- Number of iterations.
- Transmission radius or range that is used to find out the neighbor list of each node.
- Source node and destination node.

After having the following information a neighbour list is generated in increasing order of their distances from the neighbouring nodes. Using this sorted list the sorce node checks whether the destination node is in its vicinity or not if the destination node is in its vicinity then it directly unicast the request packet to the destination otherwise the first member of the sorted list is given the request packet if it has not seen this packet before otherwise the packet is given to the next member of the sorted list. The next node in turn repeats the source node procedure to find the destination node. The process goes on until the hop count maximum limit exceeds or the packet reaches the destination. The following parameters were recorded in text file as follows:

- Average number of hopes
- Average number of retransmission
- Average number of throughput
- Average power decapitated per node

IV. Implementation Of Routing Protocols

Here only one class named Network is created. The description of data members and member functions of this classe are as follows-

4.1 Data Members

- **n-** number of nodes
- **no_of_trans-** number of transmission do you want to run the program
- **sng-** source node
- **dng-** destination node
- **trans_range-** transmission range
- **nodes[ ][ ]-** double dimension array that store the x and y coordinates of the nodes.
- **d[ ][ ]-** double dimevnsion array that store the distance between all the nodes.
- **neighbors_list[ ][ ]-** double dimmension array that store the neighbor list of each node.
- **count[ ][ ]-** double dimension array that store the number of neighbors of each node.
- **fail_count-** store the number of failures.
- **succ_count-** store the number of successful transmission.
-

4.2 Member Functions

- **getdata()-** This function is used to take the coordinates of the nodes.
- **get_range()-** This function is used to take input number of nodes and transmission range.
- **get_power()-** This function is used to initialize the power to each node.
- **create_network-** This function is used to place the nodes in the network.
- **distance()-** This function is used to calculate the distance between all the nodes.
- **neighbors(int)-** This function is used to generate the neighbor list of each node.
- **path_found()-** This function is used to calculate the path from source to destination.
- **avg_power(int)-** This function is used to calculate the average power left of each node.

- **avg_hopes(int)-** This function is used to calcualte the average number of intermediate hopes.
- **avg_thrput(int)-** This function is used to calculate the average successful transmission of the packet.
- **avg_retrans(int)-** This function is used to calculate the avrage retransmission of the packet.

display(int,int,int)- This function is used to display the various outputs

V. Simulation And Results

5.1 Example

This section presents an example simulation of the proposed protocol
No. of times do you want to run program= 1
No. of nodes= 10
Transmission range=10
Transmission range=10

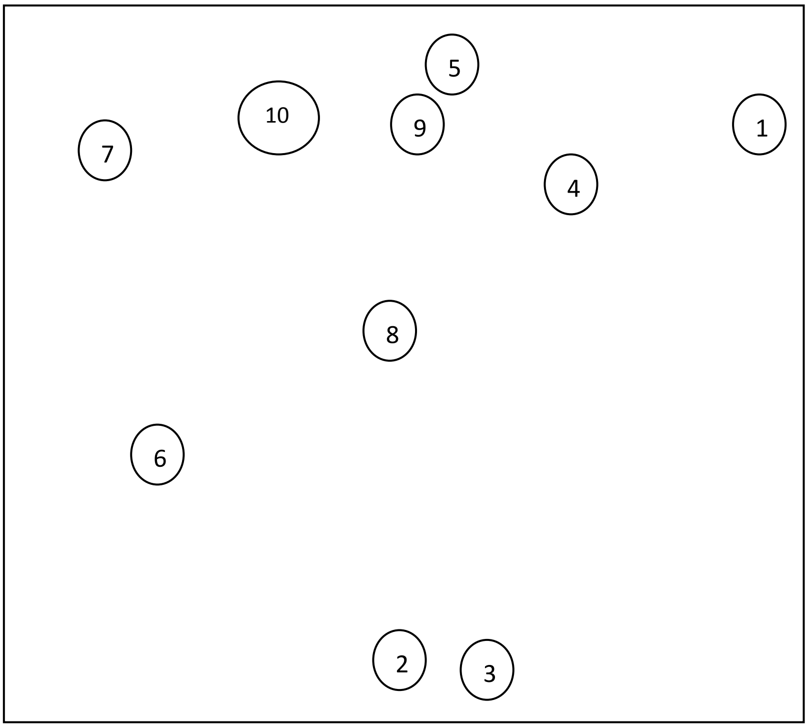


Fig. 5.1 A network with 10 nodes

Table 5.1 Neighbor list table of each node

Nodes	Neighbor list				
1	4				
2	3				
3	2				
4	9	5	1	8	10
5	9	4	10	8	7
6	8				
7	5	9	10		
8	4	9	10	6	5
9	5	4	10	8	7
10	9	5	4	7	8

Source Node is 4 and Destination Node is 9.
Destination Node 9 is found in the neighbor list of source node 4. So Path found from source to destination is 4, 9.
Average power left per node is 996.5.
Average throughput is 1
Average number of hopes are 1.
Average number of retransmission is 0.

5.2 Results

The simulator was designed in C++. It takes the following parameters as input:

1. Number of nodes
2. Transmission Radius of each node
3. Number of iterations

The following outputs were recorded.

1. **Average power left per node:** For measuring average power the following assumption was made

- Each node is assigned 100 units of power
- The node consumes 2 units of battery for transmiting a packet
- The node consumes 1.5 units power in receiving a packet

The average power decreases for lower transmission radius but as the transmission radius is increased the average power left per node also get increased as shown in Fig. 5.2. The reason for such behaviour is at lower transmission radius the number of retransmission is quite large.

	Average Power left per node		
Transmission Range	**Number of Nodes=20**	**Number of Nodes=25**	**Number of Nodes=30**
5	93	93.98	89.26
6	86.69	89.08	80.75
7	80.24	78.86	75.25
8	69.02	80.68	79.8
9	73.05	72.83	53.25
10	74.44	72.59	52.26
11	82.67	55.48	64.63
12	29.82	75.32	82.84
13	55.7	91.45	49.45
14	81.09	86.83	93.46

Table 5.2 Average power left per node

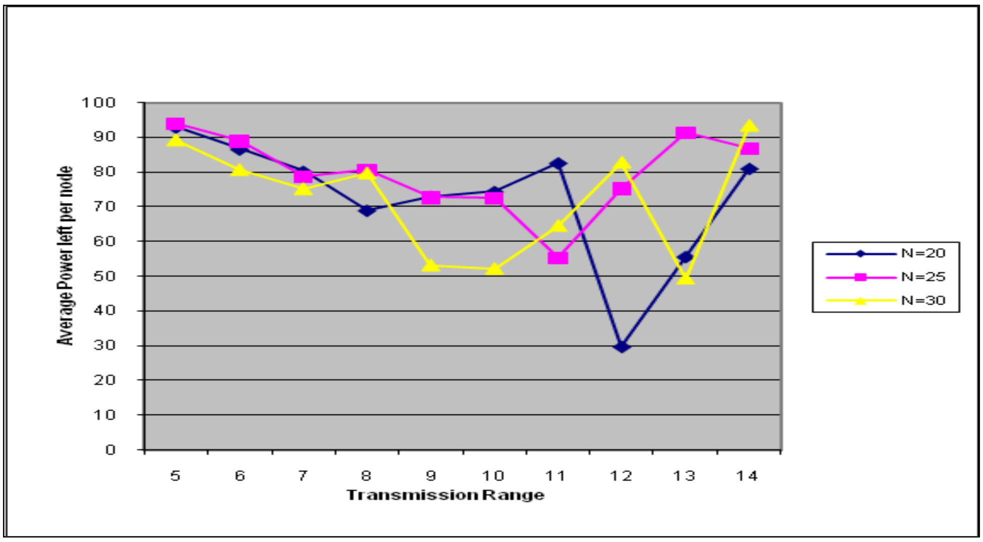


Fig. 5.2 Average power left Vs transmission range graph

1. **Throughput:** It may be defined as the number of successful transmission to the total number of transmissions. The average throughput increases as the transmission range increases due to the fact that the information regarding neighbouring nodes gets increased as shown in Fig. 5.3.

Table 5.3 Average throughput

	Average Throughput		
Transmission Range	**Number of Nodes=20**	**Number of Nodes=25**	**Number of Nodes=30**
5	0.2	0.15	0.05
6	0.15	0.15	0.15
7	0.25	0.25	0.2
8	0.35	0.6	0.4
9	0.55	0.65	0.55
10	0.75	0.75	0.65
11	0.9	0.85	0.9
12	0.65	0.85	1
13	0.85	1	0.9
14	1	1	1

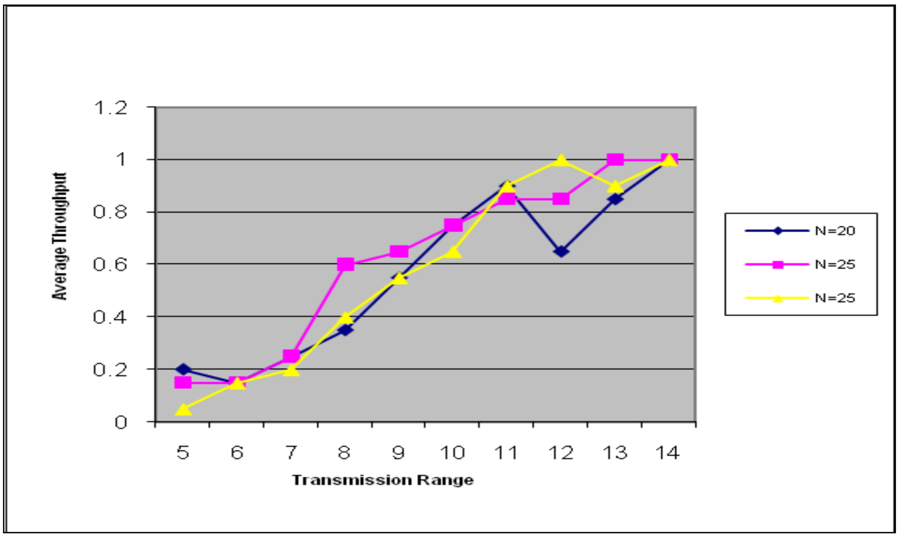


Fig. 5.3 Average Throughput Vs transmission range graph

3. Hop count: Defined as the number of intermediate nodes between a source and destination. As shown in the Fig. 5.4 with the increase in the transmission radius, the hop count gets increased.

Table 5.4 Average no. of hopes for successful transmission

	Average no. of hop count for successful transmission		
Transmission Range	**Number of Nodes=20**	**Number of Nodes=25**	**Number of Nodes=30**
5	2	1.66	1
6	2.33	1.66	4.66
7	2.4	2.2	3.75
8	2.42	2.5	3.125
9	1.54	4	2.81
10	3.26	4.2	3.53
11	2.5	6.29	5.72
12	4.3	3.52	6.3
13	4.47	3.05	4.22
14	3.55	4.7	2.8

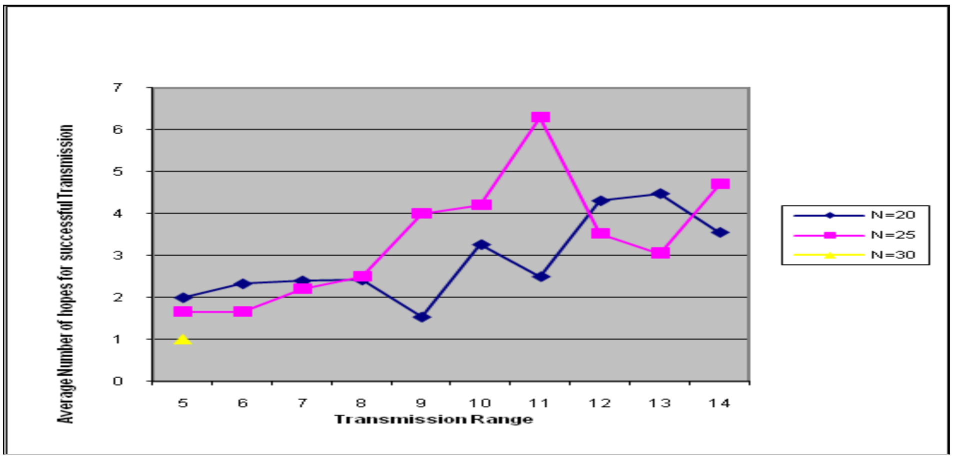


Fig. 5.4 Average no. of hopes for successful transmission Vs transmission range graph

4. Average number of retransmission: The average number of retransmission is more when the transmission radius is low since there are limited numbers of neighbour but as the transmission radius is increased the probability to reach destination gets increased and hence retransmission reduces as shown in Fig. 5.5.

Table 5.5 Average no. of retransmission

	Average no. of retransmission		
Transmission Range	**Number of Nodes=20**	**Number of Nodes=25**	**Number of Nodes=30**
5	1.5	1.75	2.55
6	1.85	2.2	2.35
7	2.3	2.55	2.7
8	2.2	1.55	1.8
9	1.85	1.75	2.7
10	1.3	1.25	2
11	0.65	1.3	0.8
12	2.6	0.7	0.15
13	1	0	1.2
14	0.25	0	0

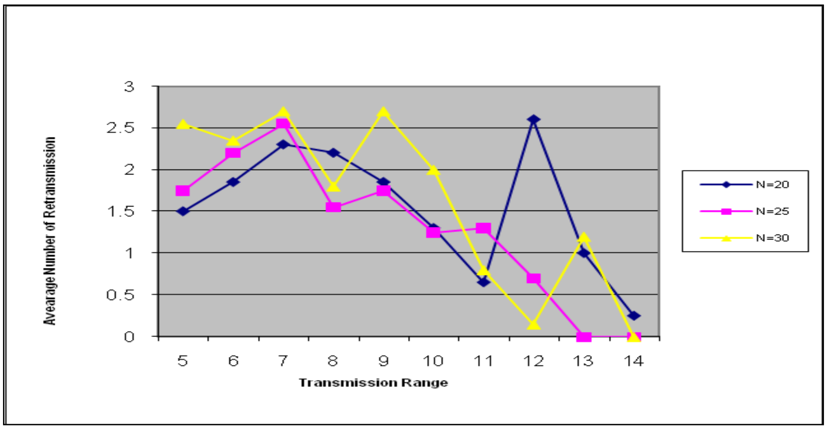


Fig. 5.5 Average no. of retransmission Vs transmission range graph

5.3 Discussion About Result

The power consumption in transmitting a packet is directly proportional to the square of the distance between the source and destination, more is the distance more is the power consumed and lesser is the effective network life time. The nodes thus tries to select their intermediate nodes to relay the packets inorder to increase its effective life time, reduce average power consumption of the overall network but at the same time introduces congestion since the number of nodes involved in routing process gets increased by adopting the staraegy proposed.