EFSR: Evolutionary Fisheye State Routing Protocol in Mobile Ad Hoc Networks

A.Samuel Chellathurai *1, E.George Dharma Prakash Raj *2

1 James College of Engineering and Technology, Nagercoil, India
*2 Bharathidasan University, Tiruchirappalli, India

Abstract— Mobile ad hoc network is wireless network that utilizes multi-hop radio relayed and are capable of operating without the support of any fixed infrastructure. The absence of any central coordinator makes the routing more complex. In mobile ad hoc network, the routing and resource managements are done in which all nodes coordinate to enable communication among themselves. This requires each node to be more intelligent so that it can operate both as network host for transmitting and receiving data and as network router for routing packets from other nodes. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. This feature presents a great challenge to the design of a routing scheme, since bandwidth is very limited and the network topology changes as nodes move. This paper investigates the behavior of existing table driven protocol, Fisheye State Routing protocol (FSR) and proposes and implements a new routing approach which is the enhancement of FSR on the basis of genetic algorithm to find optimal path.

Keywords— Mobile ad hoc network, bandwidth, table driven protocol, genetic algorithm.

I. INTRODUCTION

Wireless network enables communication between computers using standard network protocols, without network cabling. The wireless network can be classified into two types: Infrastructured and Infrastructure less [1]. In Infrastructured wireless networks, the mobile node can communicate while moving, the base stations are fixed and as the node goes out of the range of a base station, it gets into the range of another base station [2]. In Infrastructureless, the mobile node can communicate while moving; there are no fixed base stations and all the nodes in the network act as routers. The mobile nodes in the ad hoc network dynamically establish routing among themselves to form their own network while moving.

A Mobile Ad hoc Network (MANET) is a collection of wireless mobile nodes forming a temporary and short-lived network without any fixed infrastructure where all nodes are free to move about arbitrarily and all the nodes configure themselves. In MANET, each node acts both as a router and as a host and even the topology of network may also change rapidly. The mobility of nodes is also a major factor within MANET due to limited wireless transmission range. This can cause the network topology to change unpredictably as nodes enter and leave the network [3]. Node mobility can cause broken routing links, which forces nodes to recalculate their routing information. This consumes processing time, memory, device power and generates traffic backlogs and additional overhead [4].

This paper is organized as follows: Section 2 provides a discussion on work related to the existing routing protocols and working of genetic algorithm and its features. Section 3 proposes proposed new protocol, Evolutionary Fisheye State Routing Protocol (SERP) and its working. Section 4 provides implementation and result analysis. Section 5 gives the conclusions.

II. RELATED WORK

A. Conventional Routing Protocols

In recent past, a lot of attention has been shown by the research community to various issues related to ad hoc networks [5]. Many protocols have been proposed for routing in such an environment. These protocols can broadly be classified into two types: proactive and reactive routing protocols [6]. Proactive or table-driven protocols try to maintain routes to all the nodes in the network at all times by broadcasting routing updates in the network. On the other hand, reactive or on demand protocols attempt to find a route to the destination, only when the source has a packet to send to the destination. So it is necessary that the obtained route must be optimal path in the available multiple paths. Proactive protocols maintain the routing information of one node to the other using routing tables. Whenever there is a need for the route to the destination, it is readily available incurring minimum delay. But, at the same time, they may lead to a lot of wastage of the network resources if a majority of these available routes are never used. Reactive protocols are usually associated with less control traffic in a dynamic network; nodes have to wait until replies to the route queries are received.

B. Fisheye State Routing Protocol

The fisheye state routing (FSR) protocol [7] is a generalization of the Global State Routing (GSR) protocol [8]. FSR uses the fisheye technique to reduce routing overhead. The basic principle behind this technique is the property of fish’s eye that can capture pixel information with greater accuracy near its eye’s focal point. This accuracy decreases with an increase in the distance from the center of the local point. This property is translated to routing in ad hoc wireless networks by a node keeping accurate information about nodes in its local topology, and no accurate information about far-away nodes. The accuracy of the network information decreases with increasing distance.
The scope of fisheye is defined in terms of the nodes that can be reached in a certain number of hops. The center node has most accurate information about all nodes in the first circle, and becomes less accurate with each outer circle. Even though a node does not have accurate information about distance nodes, the packets are routed correctly because the route information becomes more and more accurate as the packet moves closer to the destination.

FSR maintains the topology of the network at every node, but does not flood the entire network with information, as is done in link state routing protocols. Instead of flooding information, each node exchanges topology information only with its neighbors. A sequence numbering scheme is used to identify the recent topology changes. This constitutes a hybrid approach comprising of link-level information exchange of distance vector protocols and the complete topology information exchange of link state protocols. The complete topology information of network is maintained at every node and the desired shortest paths are computed as required. The topology information exchange takes place periodically rather than being driven by an event. This is because; instability of the wireless links may cause excessive control overhead when event-driven updates are employed. FSR defines routing scope, which is the set of nodes that are reachable in a specific number of hops.

The link state information for the node belonging to the smallest scope is exchanged at the highest frequency. The frequency of exchanges decreases with an increase in scope. This keeps the immediate neighborhood topology information maintained at a node more precise compared to the information about the nodes farther away from it. Thus message size for a typical topology information update packet is significantly reduced due to the removal of topology information regarding the far-away nodes. The path information for a distant node may be inaccurate as there can be staleness in the information. But this compensated by the fact that the route gets more and more accurate as the packet nears its destination. FSR scales well for large MANET, because of the reduction in routing overhead due to the use of above described mechanism, where varying frequencies of updates are used [5].

Fisheye Routing determines routing decisions using a table-driven routing mechanism similar to link state. The table-driven ad hoc routing approach uses a connectionless approach of forwarding packets, with no regard to when and how frequently such routes are desired. It relies on an underlying routing table update mechanism that involves the constant propagation of routing information. A table-driven mechanism was selected over an on-demand mechanism based on the following characteristics:

- On-Demand routing protocols on the average create longer routes than table driven routing protocols [7].
- On-Demand routing protocols are more sensitive to traffic load than Table-Driven in which routing overhead traffic and latency increase as data traffic source/destination pairs increase.
- On-Demand Routing incurs higher average packet delay than Table Driving routing which results from latency caused by route discovery from new destinations and less optimal routes.
- Table-Driven routing accuracy is less sensitive to topology changes. Since every node has a ‘view’ of the entire network, routes are less disrupted when there is link breakage.
- Table-Driven protocols are easier to debug and to account for routes, since the entire network topology and route tables are stored at each node, whereas On-Demand routing only contain routes that are source initiated and these routes are difficult to track over time.

For these reasons, a table driven scheme for the ad hoc routing protocol was chosen. Link state was chosen over distance vector because of faster speed of convergence and shorter-lived routing loops [9]. Link state topology information is disseminated in special link-state packets where each node receives a global view of the network rather than the view seen by each node’s neighbor. Fisheye routing takes advantage of this feature by implementing a novel updating mechanism to reduce control overhead traffic. The Fisheye routing algorithm has the following main tasks.

1) Neighbor Discovery: responsible for establishing and maintaining neighbor relationships.
2) Information Dissemination: responsible for disseminating Link State Packets (LSP), which contain neighbor link information, to other nodes in the network.
3) Route Computation: responsible for computing routes to each destination using the information of the LSPs.

Each node initially starts with an empty neighbor list and an empty topology table. After its local variables are initialized, it invokes the neighbor discovery mechanism to acquire neighbors and maintain current neighbor relationships. LSPs in the network are distributed using the information dissemination mechanism. Each node has a database consisting of the collection of LSPs originated by each node in the network. From this database, the node uses the Route Computation mechanism to yield a routing table for the protocol. This process is periodically repeated.

1) Neighbor discovery: This mechanism is responsible for establishing and maintaining neighbor relationships. Neighbors can meet each other simply by transmitting a special packet (HELLO packet) over the broadcast medium. In the wireless network, HELLO packets are periodically broadcasted and nodes within the transmission range of the sending node will hear these special packets and record them as neighbors. Each node associates a timeout value in the node’s database for each neighbor. When it does not hear a HELLO packet from a particular neighbor within the timeout period, it will remove that neighbor from the neighbor list.
Timeout values are reset when a HELLO message is heard. HELLO Packets also contain the list of routers whose HELLO Packets have been seen recently. Nodes can use this information to detect the presence of unidirectional or bidirectional links by checking if it sees itself listed in the neighbor’s HELLO Packets.

2) Information Dissemination: This mechanism is responsible for distributing LSPs to the nodes in the network. Two main functions of information disseminations are handling the LSP integrity and updating interval.

After the router generates a new LSP, the new LSP must be transmitted to all the other routers. A simple scheme is flooding, in which each packet received is transmitted to each neighbor except the one from which the packet was received. Because each router retains the most recently generated LSP from other nodes, the router can recognize when it is receiving a duplicate LSP and refrain from flooding the packet more than once. The problem with this flooding is that a router cannot assume that the LSP most recently received is the one most recently generated by that node. Two LSPs could travel along different paths and might not be received in the order in which they were generated. A solution to this is to use a scheme involving a combination of a sequence number and an estimated age for each LSP.

A sequence number is a counter. Each router keeps track of the sequence number it used the last time it generated an LSP and uses the next sequence number when it needs to generate a new LSP. When a router receives a LSP, it compares the sequence number of the received LSP with the one stored in memory (for that originating node) and only accepts the LSP if it has a higher sequence number. The most recently generated sequence number has higher value.

However, a sequence number alone is not sufficient. The sequence number approach has various problems:
- The sequence number field is of finite size. A problem arises when a node creates a LSP to case the field to reach the maximum value. Making the sequence number field wrap around is not a good idea because it causes ambiguity on the relation of the sequence numbers.
- Sequence number on an LSP becomes corrupt. If the sequence field is corrupted to a very large sequence number, it will prevent valid, newer LSPs (with smaller sequence numbers) to be accepted.
- Sequence number is reset. When a router goes down or forgets the sequence number it was using, newer LSPs cannot be distinguished from older LSPs.

To solve the preceding problems, an age field is added to each LSP. It starts at some value and is decremented by routers as it is held in memory. When an LSP’s age reaches 0, the LSP can be considered too old and an LSP with a nonzero age is accepted as new, regardless of its sequence number.

3) Route Computation: Once the router has a database of LSPs, it computes the routes based on the Dijkstra’s [10] algorithm which computes all shortest paths from a single vertex. The link metric used for path cost is the hop count.

C. Genetic Algorithmic Approach

Genetic algorithms [11] perform much better because of their population based approach. In this paper genetic algorithm has been preferred as the optimization algorithm [12] because of the confidence that it would work due to its robustness. Another big advantage of genetic algorithms is the ability to parallelize them on a large scale by spreading the evaluations across different machines. A large amount of work has been done on the application of genetic algorithms or evolutionary algorithms to communications networks.

The design of the genetic algorithm has components like genetic representation, population initialization, fitness function, selection scheme, crossover and mutation [13]. A routing path contains sequence of nodes in network. The genetic algorithm is applied to a path that has been obtained from the route discovery phase. A routing path is encoded by a string of positive integers that indicating the IDs of the nodes in the network. The length of the string should not be more than the number of nodes present in the network.

In genetic algorithm, each chromosome represents a potential solution and this can contain more than one solution initially. The paths obtained from route discovery phase are considered as initial chromosomes. For an obtained solution, quality can be evaluated accurately with the help of fitness function. The goal of using this genetic algorithm is to find the shortest path, lowest throughput between source and destination and the larger buffer size that the path has. It is important to obtain the shortest path and lowest delay time as the primary concern.

Selection plays a key role in improving the quality of population of chromosomes. The selection of chromosome is purely done on the basis of result of fitness function. Crossover is done by choosing crossover points to find the better solution from current one. Since chromosomes are used as path structure, every time two chromosomes are chosen, for crossover. The population will undergo mutation after the crossover is performed.

III. EVOLUTIONARY FISHEYE STATE ROUTING PROTOCOL

In this paper a new protocol has been proposed, called Evolutionary Fisheye State Routing Protocol (EFSR), which uses simple genetic algorithm for finding multiple shortest paths. Here route computation has been done by genetic algorithm instead of Dijkstra’s algorithm. This protocol is able to find the shortest optimal path from the source node to destination node.

The proposed algorithm Evolutionary Fisheye State Routing (EFSR) is given below,
BEGIN
    Initialize the source node and destination node
    Generate the initial population using nodes in each chromosome randomly
    Apply Remove-Sharp algorithm for initial population
    While NOT (Convergence condition) DO
        Evaluate the fitness value of each chromosome in the current population
        Rank the population based on fitness values
        Select the highest fitness chromosome
        Apply two point crossover operation between current parents
        Apply the mutation process with the given probability
        Generate the new population
    END
    Output the best individual path
END

The above proposed algorithm initializes the source and destination nodes for data transmission. Then a set of chromosomes containing all possible nodes in the network are randomly generated. Then Remove-Sharp algorithm is applied to remove the badly positioned nodes in the network. Then fitness of each chromosome is evaluated in the current population. Then the chromosomes can be ranked according to the higher fitness value. The following fitness function is used to evaluate the fitness value,

\[ F = \begin{cases} 
1 & \text{Feasible path} \\
\sum C_i (s_i, s_{i+1}) & \text{Infeasible path}
\end{cases} \]

According to the above fitness function, cost is less for feasible path and infeasible path will have maximum cost. Cost between these two extreme values is ranked for selection. Then two highest fitness chromosomes are selected for multipoint crossover operation. Here two crossover points are selected as shown in the “fig.1”.

After performing two point crossover operation two more offspring are generated. Then the new offspring can be mutated with the given probability. This evolutionary operation is repeated until to get the optimal path between the source and destination.

Remove-Sharp algorithm removes sharp increase in the path distance due to a node, which is badly positioned in the network. Remove-Sharp algorithm is given below.

Step 1: Create a list (NEARLIST) containing the nearest m nodes for a selected node in the network
Step 2: Remove the selected nodes from the network and form network with N-1 nodes.
Step 3: Reinsert the selected in the network either before or after any one of the nodes in NEARLIST and the cost of the new route is calculated for each case.
Step 4: Select the sequence, which produces the least cost path
Step 5: Repeat the above steps each node in the network.

In the above algorithm a list of node is generated which contains the neighbours of the selected node. The selected node is excluded from the network. Now the excluded node from the network is reinserted in the network randomly either before or after any one of the nodes in the network and cost of the new route is calculated. The sequence which produces the least cost route is selected. The above algorithm is applied for each node in the network to remove the sharp edges in the network. Fig.2 shows the network with sharp edges. Here node 5 is badly positioned in the network.

IV. IMPLEMENTATION AND RESULT ANALYSIS

To implement the proposed protocol using NS2 the initial simulation parameters has to be initialized as shown in the table 1 before entering the simulation phase. The parameters
declared here are to be used in the design of network where the protocol is going to be simulated.

The performance of the new proposed protocol can be analysed by considering the parameters packets delivery ratio, throughput, end to end delay, and packet overhead. Since the XGraph tool is used, the proposed protocol Evolutionary Fisheye State Routing Protocol (EFSR) is compared against the existing Fisheye State Routing Protocol (FSR). The performance of both protocols are analysed with reference to number of nodes available in the ad hoc networks.

<table>
<thead>
<tr>
<th>TABLE 1</th>
<th>SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>Simulator</td>
<td>NS2</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300s</td>
</tr>
<tr>
<td>Simulation area</td>
<td>1000m x 1000m</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>Transmission range</td>
<td>60m</td>
</tr>
<tr>
<td>Max speed</td>
<td>0,5,10 m/s</td>
</tr>
<tr>
<td>CBR flows</td>
<td>20</td>
</tr>
<tr>
<td>Data payload</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Sending rate</td>
<td>4 packets</td>
</tr>
<tr>
<td>Movement model</td>
<td>Random waypoint</td>
</tr>
</tbody>
</table>

Initially packet delivery ratio is taken as the metric and XGraph tool is used to compare two protocols. It is observed from the result that the packet delivery ratio of EFSR is better than FSR as shown in the “Fig. 4”.

Then end to end delay is taken as the metric and XGraph tool is used to compare two protocols. It is observed from the result that the end to end delay of EFSR is initially higher but decreases as nodes increases as shown in the “Fig. 6”.

Finally overhead is taken as the metric and XGraph tool is used to compare two protocols. It is observed from the result that the overhead of EFSR is less when compared to the FSR as shown in the “Fig. 7”.

V. CONCLUSIONS

This paper proposes a new routing protocol for mobile ad hoc networks called Evolutionary Fisheye State Routing Protocol which applies the genetic algorithmic approach for finding the optimal path to the existing Fisheye State Routing Protocol This gives the better delivery of packets to the destination and throughput and reduces overhead and delays on the network. The EFSR will be definitely a promising protocol of future. Extensive simulation results reveal that the
The proposed scheme features better transmission delay, route convergence time, system efficiency and system throughput. The future work of this research is to improve the performance in various aspects and to consider more number of parameters to improve the performance of the proposed genetic algorithm in order to find more accurate optimal path. The scheme can be extended to provide good QoS by considering other parameters such as buffers, bandwidth etc. Further simulation can be carried out by considering time.

REFERENCES


