Topology Discovery and Modified Tabu Search for Efficient Routing in Wireless Ad Hoc Networks

V. Jayaraj, J. Jegathesh Amalraj* and V. Mathimalar
Bharathidasan University, Tamilnadu, India; jaya_v2000@yahoo.com, amaljas@gmail.com

Abstract
For efficient transformation of information, Routing is needed. The requirement of quality constraints have been increased due to the variety of applications involved. This has escort to eminence conscious routing. An updated network structure is required to maintain efficient routing. This paper endow with different methods that act upon topology discovery and route formulation, along with route maintenance. The topology discovery is performed initially by discovering the neighbors and then clustering them and assigning leaders for each cluster. The route formulation phase finds multiple paths to a node, and finally optimization is performed using a modified form of Tabu Search. The route maintenance is performed by periodically checking the available routes for invalid nodes and updating them. The proposed research work gives a complete strategy for topology discovery along with dynamic routing and it performs well under all circumstances in an Ad Hoc network.

Keywords: Topology Discovery, Tabu Search, Routing, Clustering, SENCAST

1. Introduction
Wireless ad hoc networks have increased due to the ease of use of technologies that facilitate interconnectivity. But the problem arises due to the intrinsic properties of wireless networks, mobility. This intrinsic property of a wireless node formulate the topology discovery of the network a complicated task. Various topology discovery techniques have been designed for wireless networks. The topology discovery in a wireless network is basically a neighbor discovery mechanism that finds all nodes within the range of the current node and marks them as neighbors. A neighbor discovery algorithm called MultiPath OLSR (MP-OLSR) is discussed in which is based on OLSR. The Multipath Dijkstra Algorithm is projected to obtain multiple paths. Different link metrics and cost functions are used by this algorithm to attain flexibility and extensibility. It performs the process of route recovery and loop detection, along with improving the quality of service regarding OLSR. It is also backward compatible with OLSR. A neighborhood discovery algorithm is proposed in which is based on two algorithms. Region based discovery algorithm is followed in the initial step. It does not guarantee communication links. The next step integrates the reliability with the previous step.

Routing is the process which determines the available routes from a source node to a destination node. It is always optimal to use a routing algorithm that satisfies the quality constraints along with an optimal path provides an extensive survey on routing protocols and their categories. It divides the wireless routing protocols into proactive or reactive, and geographical, multipath, hierarchical, flow oriented, WMN, multicast, geo-cast, power aware and hybrid provides a comparative analysis on AODV, DSR, OLSR and DSDV protocols. All simulations were conducted on NS2, and the results provide a valuable insight on the working of these protocols presents a neighbor supported reliable multipath multicast routing in ad hoc networks. A mesh of multipath routes is established from the source to multicast destinations using neighbours that have a high reliability pair factor.

To group up the nodes and make them function as a single entity, clustering is required. A cluster leader is to be elected for each cluster to facilitate communication.
within a cluster and between clusters.\textsuperscript{7} describes a fast converging clustering approach on MANET.\textsuperscript{8} presents a graph clustering approach for MANETs. It describes a new graph clustering algorithm that automatically defines the number of clusters based on a clustering tendency connectivity-based validation measure. It describes three phases, the coarsening phase, partition phase and refinement phase for performing the clustering process.\textsuperscript{9} presents an efficient graph based clustering algorithm. In\textsuperscript{10}, a dynamic algorithm that performs unicasting and multicasting is used. The SENCAST\textsuperscript{10} helps in efficient data transfer. The same SENCAST is used in the work for the estimation of the bandwidth in the routes, that serves as a QoS parameter when determining the routes.\textsuperscript{17} provides an efficient algorithm that utilizes an existing cognitive network to exhibit better performance. This methodology also ensures that the existing channel frequencies are not wasted and the frequencies that are in use are utilized to the maximum extent without the need for bothering about the effects of interference.

\section{2. Proposed Work}

This paper is focused on providing an efficient routing mechanism that helps in optimal transmission of data.\textsuperscript{18} An awareness about the network topology is the base for any routing mechanism. The proposed work begins by an efficient topology discovery. Route formulation is followed by topology discovery and Modified Tabu search is used for route optimization. Further, an error handling routine is also incorporated, that troubleshoots the obtained routes for any disruptions in the network.

\subsection{2.1 Topology Discovery}

The topology discovery is the process of determining the nodes that are present in the network and their connections with other nodes.\textsuperscript{18} For efficient transfer mechanism, clustering and cluster head selection should be done along with topology discovery. Further, the topology refresh is also carried out in order to make sure that the topology that has been discovered remains latest and without any dead paths.

\subsubsection{2.1.1 Neighbor Discovery}

The neighbor discovery is performed by finding the one hop neighbors. In case of emergency data transfers, the next node in range can be used; this is due to the fact that the topology of the wireless ad hoc network does not always remain the same. The process of neighbor discovery is performed by using the ‘hello’ messages.\textsuperscript{16} Each node broadcasts the message, hence retrieves details about all other nodes in the network, that are within its range.

\subsection{2.1.2 Clustering}

Clustering is the process of grouping similar data, such that all nodes that are similar to each other occupy the same cluster, and those that have differences occupy different clusters. A graph based Clustering that determines the clusters along with the cluster leaders is proposed here.

Consider a graph $G = (V,E)$, with vertices $V$ and edges $E$. The number of nodes in the graph $G$ is $n$, and the edge set contains $(i,j)$ where $i$ and $j$ represent vertices from $V$. The weight matrix is represented by $W_{ij}$, that represent the cost of traversal from $i$ to $j$, here it represents the distance from node $i$ to node $j$.

$$W_{ij} = \begin{cases} 
\text{distance between } i \text{ and } j \text{ if a path exit} \\
0 \quad \text{otherwise} 
\end{cases}$$

(1)

The adjacency matrix $A_{ij}$ is created, that represents the connectivity in the network. Adjacency matrix is determined by the following

$$A_{ij} = \begin{cases} 
1 \quad \text{if there is a connection between } i \text{ and } j \\
1 \quad \text{if } i = j \\
0 \quad \text{otherwise} 
\end{cases}$$

(2)

The degree of a node $\text{Deg}_i$, determined by the adjacency matrix, is the number of connections that a node maintains in the network.

$$\text{Deg}_i = \sum_{j=1}^{n} a_{ij} - 1$$

(3)

In a wireless network, clustering is performed by dividing the network into partitions such that the distance constraint is satisfied. A node is added to the cluster, if it contains the threshold that is less than that of the defined threshold. The value of threshold\textsuperscript{9} is defined by,

$$\delta = D_{min} + (D_{max} - D_{min}) \times \text{constraint}$$

(4)

Where $D_{min}$ and $D_{max}$ defines the minimum and maximum distances exhibited between nodes in the network, and the
constraint defines the accuracy of the cluster. The value of a constraint ranges from 0 to 1. A very low constraint groups only closely grouped nodes and many clusters, while a high value provides clusters of large size and lower cluster count. It is very important to strike a balance in this value. There are no defined rules to identify this value, this can only be performed as a trial and error process, and it differs with networks.

**Topology Discovery Algorithm**

```
For each node in v {
    Select all nodes (x) with Deg_n > 0
    For each x,
        Find threshold \( \delta_x \) (using equation (4))
        If \( \delta_x < \text{thresh}_{\text{max}} \)
            Add it to cluster \( C_n \).
        End For
    End For
}
```

For every node \( n \) in the vertex set \( V \), its neighbours are determined. Every neighbour node is processed to find the threshold using equation (4). If the threshold is within the limits of the defined threshold \( \text{thresh}_{\text{max}} \), then the node is incorporated in the current cluster.

**Cluster Determination Algorithm**

```
For each cluster \( C_i \)
    For each Cluster \( C_j \) where \( j > i \)
        Check if all elements in \( C_i \subset C_j \)
        If yes delete \( C_i \)
    End For
End For
```

The algorithm for cluster finalization is described above. Each cluster is examined to see if the cluster elements are a subset of any other cluster. If so, the current cluster is removed from the cluster set. The clusters remaining after this process forms the final clusters and the cluster base values are assigned as the cluster leaders.

**2.1.3 Leader Selection**

In general, when considering the process of routing, one cluster leader alone is not sufficient to maintain a stable cluster\(^{12}\). The following properties describe the characteristics of a cluster node and a leader node for maintaining the stability.

**Property 1:** All nodes in a cluster should contain \( 2f + 1 \) neighbors and \( f + 1 \) disjoint routes to the leaders.

**Property 2:** If a leader connects to less than \( 2f + 1 \) nodes, then the cluster is taken apart.

Where \( f \) denotes the threshold for the occurrence of an anomaly.

**2.1.4 QoS Parameter Maintenance**

Due to the fact that the system also considers the QoS parameters during the transmission of data, the process of topology discovery performs an extra job of obtaining the QoS parameters of the neighbors during the neighbor discovery phase. Since the routes are selected in the network on the basis of both availability and the quality parameters, it becomes necessary for the network to maintain these parameters.

**2.1.5 Topology Refresh Phase**

It cannot be expected of a wireless network to remain stable\(^{15}\). Hence it becomes vital to be certain the routes determined earlier are live and working. This functionality is performed by the topology refresh phase. This module checks for the validity of the network after a particular determined time interval. If so, it intimates the rest of the cluster mates about the invalid route and tries to find a new route. This functionality is performed by all the leaders. The leaders take turns in performing this operation. Hence, it is sure that at a certain time, only one of the leaders is performing the refresh operation. Hence the network traffic created due to the process of refresh is maintained at a minimal level.

**2.2 Route Formulation**

A process requesting for data transfer requires a path for transmission of the messages. The process of finding the optimal path in which the data transfer can take place is referred to as routing. Our process of route formulation uses the available routes and the QoS parameters to determine a set of routes. The optimal route from these set of routes is determined by using the modified Tabu Search.
2.2.1 Multiple Route Selection

When a routing need arises, the source node sends a Route REQuest (RREQ) packet\(^1\) to all its leaders. The leaders determine if they have any routes already present for the source and destination in the Tabu Queue. If such a route exists, the route is added to the Routes list. The request is then transmitted to all the other leader nodes, along with the current node list embedded in the RREQ message. Every leader node transmits the information available within their cluster. This process returns all the available routes\(^1\) back to the base node. The base node performs reverse mapping of the received packet and determines all the available routes\(^1\) and populates them in the Routes list.

2.2.2 Route Shortlist

The process of bandwidth estimation is used as a QoS parameter to shortlist the available routes from the Routes list\(^4\). The bandwidth estimation is performed using SENCAS, a dynamic routing protocol. The QoS requirement of the required application is also taken into account and the shortlisting of the routes is performed. This process returns a set of shortlisted routes Route Shortlist. This list also contains a set of routes. The optimal service is determined by using the Modified Tabu Search algorithm.

2.2.3 Route Optimization using Modified Tabu Search

A modified form of Tabu search is used to determine the final optimal route. This method uses the intermediate term memory, which a random route \(RR_i\) is selected from the Route Shortlist. The cost of transmission and level of QoS support is determined and is provided a score \(s_{Best_i}\). Another random route \(RR_j\) is selected from the set, where \(RR_i \neq RR_j\). The \(s_{Best_i}\) is calculated and is compared with the \(s_{Best_j}\). The least cost \(s_{Best}\) is considered for transmission, and this route is added to the Tabu list.

Modified Tabu Search Algorithm

\[
\begin{align*}
\text{Select a random route } RR_i; \\
\text{Find cost incurred. Let it be } s_{Best_i}; \\
\text{Select another random route } RR_j, \text{ where } i \neq j; \\
\text{If cost } (RR_j) < \text{ cost } (RR_i); \\
\text{sBest } = RR_j; \\
\text{Use } s_{Best} \text{ to transfer data;} \\
\text{Add } s_{Best} \text{ to Tabu Queue.}
\end{align*}
\]

2.3 Route Maintenance

The process of route maintenance plays a vital role in maintaining the stability of the system. The process of route maintenance begins from the phase when the first route is selected. A wireless ad hoc network in general is dynamic, i.e. nodes join and leave the network, and moves from one location to another in a random manner. The random joining and leaving phase is handled by the topology refresh phase, while node movements are handled by this module.

When a node \(N\) wishes to move from its current position, it broadcasts a leave message to all its connected nodes (and leaders) intimating movement. All the nodes receiving the message perform the process of deleting \(N\) and all the routes concerning \(N\) and transfer the details to all the leaders connected to it. When a node enters a region through migration, it broadcasts a join message to intimate the surrounding nodes that it has entered into
their vicinity. This process will require some addition in
the neighbor data of the candidate nodes and the leaders.
All connected leaders are also intimiated about this newly
joined node.

Sometimes, the migration process might not have any
effect on some nodes. This occurs when a node moves,
but still under the vicinity of some of its previous peers or
leaders. Hence this process might trigger an unnecessary
deletion and addition of data. Further, there is also a high
probability of loss of some routing data. Hence all nodes
wait for a threshold time interval Refresh Wait before
adding or deleting any nodes. This provides an added
advantage of facilitating multiple additions or deletions,
since that transactions also be added occurred during the
wait time.

**Route Maintenance Algorithm**

```
If (movement triggered)
  Obtain neighbor set N
  Multicast leave message (leave_n) to N
If (Movement Stopped)
  Broadcast join_n msg
If(msg received)
  Add to node list Q_n
  Trigger RefreshTimer using refresh Wait value
If (RefreshTimer expired)
  For all msgs in list Q_n
    If (leave_n)
      Check for join_n
      If (not available)
        Remove n from neighbor list
      Else
        Remove join_n from Q_n
        Remove leave_n from Q_n
    If (join_n)
      Add n to neighbor list
      Remove join_n from Q_n
  End For
```

3. Results and Discussion

This paper can be divided into three broad functionalities,
namely, topology discovery, route formulation and route
maintenance. The neighbor discovery, QoS maintenance
and clustering is performed in topology discovery phase.
It is a time consuming process, since it has many sub
functionalities. A stable system that accommodate any type
of movement and nodes and leave request is established
after this phase. As there is close interconnections between
the leader and assigning multiple leaders in single cluster
makes the system stable.

The usage of Tabu search, which is a meta heuristic
algorithm helps in optimal routing. Since this process is
random, every route satisfying the QoS requirements has
equal chances of being selected. Hence a layer of rout
security is also incorporated in this process. Another
added advantage of using a Tabu list is that, it implicitly
maintains all the traversed routes. Hence there is no
necessity of storing the available routes again in a separate
storage. In this modified Tabu search algorithm, a
queue is used. The elements are taken out of the queue
only after the queue is full. Hence it can be seen that both
dynamicity of the Tabu process is maintained along with
the reduced storage requirement. Further, maintaining
the QoS parameters help in determining the most effi
cient route and that is best for transmitting the data in
accordance with the application being used. Further, the
problem of selfish leaders is eliminated by using multiple
leaders. Hence the elimination of an additional overhead
of surveillance for selfish leaders is eliminated.

Table 1 shows the simulation setup of NS2 for this
research work.

The graph in Figure 1 shows the drop rate % with
respect to the number of packets. It is clear from the graph
that using the proposed cluster algorithm, after a particu
lar point, the drop rate stabilizes. This is due to the fact
that the number of hops stabilizes after the particular time,
hence the drop rate can be brought to a constant rate.

The graph in Figure 2 shows the packet delivery ratio
with respect to the node velocity. It can be seen that the
delivery ratio drops constantly when the velocity of the
node increases, and after a threshold of 35m/s, it can
be observed to maintain the delivery ratio at 77%.

<table>
<thead>
<tr>
<th>Table 1. NS2 Simulation Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
</tr>
<tr>
<td><strong>Type of queue</strong></td>
</tr>
<tr>
<td><strong>Delay time</strong></td>
</tr>
<tr>
<td><strong>No of nodes</strong></td>
</tr>
<tr>
<td><strong>Queue Limit</strong></td>
</tr>
<tr>
<td><strong>MAC Type</strong></td>
</tr>
<tr>
<td><strong>Simulation Area</strong></td>
</tr>
</tbody>
</table>
The graph in Figure 3 shows the packet delivery ratio with respect to the number of nodes. Here the packet delivery rate increases to a certain extent and then starts dropping. This threshold can be used to maintain the node count for the network.

The graph in Figure 4 shows the total overhead with respect to the number of clusters. As the number of clusters increase, the overhead also starts increasing. After a threshold, the increase becomes rapid.

4. Conclusion
Routing in a wireless ad hoc network is a critical process, that involves many parameters and security issues. A complete strategy of topology discovery along with optimal route formulation and updation options is developed in this research work. From the results, it is observed that the process works efficiently considering the drop rate. The threshold for velocity and the maximum number
of nodes was determined using the packet delivery ratio. The overhead was reduced by handling the nodes join and leave requests in a time interval. The overhead calculation was performed with respect to the cluster size. By the topology refresh phase, the current topology of the network is obtained. Then the best route is found by the modified Tabu search for routing. Overall efficiency in the network is improved by these techniques. Hence the above method performs well under all random circumstances in an ad hoc network. Future work will include dynamic caching mechanisms to avoid excess network traffic and to improve the overall QoS of the ad hoc networks.
5. References