

# SECTOR BASED MULTICAST ROUTING ALGORITHM FOR MOBILE AD-HOC NETWORKS

Murali Parameswaran<sup>1</sup> and Chittaranjan Hota<sup>2</sup>

<sup>1</sup>Department of Computer Science & Information Systems, BITS-Pilani, Pilani, India

<sup>2</sup>Department of Computer Science & Information Systems, BITS-Pilani, Hyderabad,  
India

## **ABSTRACT**

*Multicast routing algorithms for mobile ad-hoc networks have been extensively researched in the recent past. In this paper, we present two algorithms for dealing with multicast routing problem using the notion of virtual forces. We look at the effective force exerted on a packet and determine whether a node could be considered as a Steiner node. The nodes' location information is used to generate virtual circuits corresponding to the multicast route. QoS parameters are taken into consideration in the form of virtual dampening force. The first algorithm produces relatively minimal multicast trees under the set of constraints. We improve upon the first algorithm and present a second algorithm that provides improvement in average residual energy in the network as well as effective cost per data packet transmitted. In this paper, the virtual-force technique has been applied for multicast routing for the first time in mobile ad-hoc networks.*

## **KEYWORDS**

*Network Protocols, Wireless Network, Mobile Network, Routing Protocols, Multicasting,*

## **1. INTRODUCTION**

A mobile ad-hoc network (MANET) involves independent nodes that are rapidly deployed in an environment. These mobile nodes must make the correct routing decisions in the presence of node mobility, transient link failures and energy and other constraints. In the event of sending multiple copies of the same information to multiple nodes, use of multicast routing is the preferred option. Depending on how the distribution paths among multicast group members are created, multicast routing protocols can be classified into tree-based, mesh-based and hybrid multicast routing protocols [1]. Tree-based protocols can further be sub-divided into source-tree and core-tree protocols. In our algorithm, we take the middle path by letting the multicast tree be created by the initial packet in transit.

The creation of multicast tree can be equated with the Steiner tree problem. Steiner tree problem for a graph  $G(V,E)$  is the problem of finding a tree spanning all nodes in  $Q$ , where  $Q \subseteq V$ , such that the length of the resultant tree is minimized [2],[3]. The set of nodes  $S$ , where  $S \in V, S \cap Q = \emptyset$ , are known as Steiner nodes. A Steiner minimal tree constructed for a local sub-graph of a  $G$  need not be part of the minimal tree constructed for the entire graph [2]. Steiner tree problem is known to be NP-complete for planar graphs. Approximation algorithms like the one given in [4] have been suggested for the Steiner tree problem. A related notion is to classify a tree as relatively minimal Steiner tree. In a relatively minimal Steiner tree, the length of

the tree is minimal for the specific set of nodes chosen to be part of the tree. The problem of multicasting reduces to finding a minimal Steiner tree in a graph. The task of a multicast routing problem is to determine the correct set of Steiner nodes  $S$  so that the tree can reach all terminal nodes in  $Q$ . For efficient communication, any multicast tree created must at least be relatively minimal. In our approach, we attempt to determine a relatively minimal Steiner tree with the help of virtual force approach.

The notion of virtual force had been explored in deployment problem in wireless sensor networks [5],[6], and in routing in MANETs [7]. In the virtual force approach (VFA), the participating nodes and/or the packet in transit are applied with some electric charge. The electrostatic forces are computed and the routing decision is made based on the magnitude and direction of the resultant force. Poduri, *et al.* [5] demonstrated the use of a combination of attractive and repulsive forces for solving sensor deployment problem in wireless sensor networks. Liu, *et al.* [7],[8] used the concept of virtual force in unicast routing for MANETs. In the routing algorithm for MANETs, the resultant force is used only to make a decision to move the packet forward.

Our main contribution in this paper is the use of virtual force in multicast routing for MANETs. The notion of virtual force is used to guide the packet towards the destinations. The effective force on the packet is a sum of contributing force values from the destinations, the source node and the packet itself. In addition to the forward guiding force, we use dampeners to limit choice of the next node in the path to accommodate QoS parameters. The combined result of these forces will let the packet know whether it is currently in a Steiner node. The resultant force will be towards the node that will be in the path to the majority of the destination nodes in one direction of a branch in the multicast tree.

In this paper, we have used the virtual force technique for multicasting. While attempting to use the technique for multicasting, we identified several new issues that were not present in unicast routing algorithms. The first algorithm proposed in this paper deals with directly imposing the virtual force technique for MANETs. We provide a refined algorithm that uses virtual force in a slightly different manner. We divide the region around the current node into sectors and perform a virtual-force based multicast routing on each of the sectors. Multicast routing using virtual force technique is attempted for the first time in this paper. As far as we know, no other paper has suggested the use of sectors of variable arc-lengths for performing multicast routing, in conjunction with the virtual force technique.

The remainder of the paper is organized as follows: section 2 describes the related work for our algorithm. Section 3 provides overall description of the technique, along with basic idea, assumptions used and the modelling details. Section 4 describes the first algorithm proposed, viz. multicast routing algorithm using virtual-force. Section 5 describes the details of the sector-based virtual force-based multicast routing algorithm. Section 6 contains the simulation results, while section 7 concludes the paper.

## 2. RELATED WORKS

Gilbert, *et al.* in [2] had put forth the criteria that the angle between out-going edges at a branch node for a Steiner node is  $120^\circ$ . A relatively minimal Steiner tree is a Steiner tree when this angle is greater than or equal to  $120^\circ$  for all internal nodes. In their paper, they have suggested an inductive property for constructing relatively minimal Steiner trees with the help of unit tension force from the group nodes. However, in a real-world network, we do not have control over the location of various nodes in the network. Thus, we should ideally be choosing branching nodes such that the angle between the out-going edges are exactly  $120^\circ$ , but still be able to handle cases where appropriate nodes are not available in those positions.

Mauve, *et al.* suggested a multicast routing algorithm using geometric information in [9]. Their greedy algorithm uses a heuristic dependent on the normalized number of next hop neighbours to determine whether a branching node in multicast tree has been reached. The parameter  $\lambda$  -used in their algorithm determines how late the split of the multicast forwarding will take place, with  $\lambda = 0$  indicating an early split and  $\lambda \approx 1$  for a very late split.

Fotopoulou, *et al.* [10] suggested the use of geo-circuits for unicast routing. They computed the unicast path from source to destination based on greedy position-based strategy and assigned a virtual circuit number to such paths. Once a geo-circuit was established, they re-used this geo-circuit to send future packets to the destination till the end of traffic or till the path is updated due to node mobility. The idea of using geographic virtual circuits can be used for multicasting as well, with a few minor revisions in the route caching system.

Liu, *et al.* suggested two routing algorithms, SWING [8] and SWING+ [7], for unicast routing in MANETs based on the notion of small world networks. These algorithms expanded the notion of virtual circuits as described by Fotopoulou, *et al.* [10] to include long virtual logical links to the surrounding neighbourhood of the current node with the help of virtual force. These two protocols were the first unicast routing protocols to use virtual force to compute the path towards destination. By computing the repulsive force exerted on the neighbours of the current node, the next hop neighbour was chosen as the node with the maximum repulsive force from the source node towards the destination node. They demonstrated that their approach works efficiently in a 3-D environment, as well. In this paper, we extend the notion of virtual-force suggested by Liu, *et al.* for multicast routing.

Rahman *et al.* [11] divided the region around the current node into quadrants, and considered four closest nodes in each quadrant for determining the next hop in the multicast tree. Their contention was that a split in multicast packet will occur when the destination nodes belong to multiple quadrants around the current node. They later expanded their algorithm in [12] used an intelligent energy matrix to increase the average life of the nodes in the network. Their algorithm provides a very reasonable length of the multicast tree. However, we found that by statically fixing the quadrants, their algorithm was generating slightly longer multicast trees.

The notion of virtual force as described in [7] cannot be used directly for the multicast routing problem. If we apply electrical charges in only the source node and the packet, as was described in Liu, *et al.* [7], we found that we could not really accommodate for multiple destinations typical in a multicasting environment. Initially, when we attempted to apply the electrical charges in destination nodes and the packet in transit, we found that the packet was moving towards a force-equilibrium position over and over again, increasing the likelihood of loops in the multicast tree. Also, supporting QoS parameters in a multicast route is slightly different than adding it in a simple virtual circuit because of the nature of path establishment. Instead of limiting the decision into fixed quadrants around the current node, as described in Rahman, *et al.* [12], we dynamically divide the region around the current node into  $\alpha$  sectors based on the direction of the effective force due to the nodes in the multicast group. In this paper, we have devised simple mechanisms to adapt the concept of virtual circuits into the multicast environment.

### 3. MULTICAST ROUTING USING THE VIRTUAL FORCE TECHNIQUE

In this section, we start with a discussion on the assumptions and basic working of our multicast routing algorithms with the help of the virtual force technique. We then introduce the system model used.

### 3.1. Assumptions

We assume that location information is available in all nodes in the network and that all nodes in the network will become eventually aware of location and mobility information. Most off-the-shelf products currently come with GPS facility. Our algorithm assumes that there is an appropriate data structure that can store this location information. Such a data structure will allow look-up of node location from the node's address. As far as proper working of the algorithm is concerned, the location information in the immediate vicinity of the current node needs to be accurate. There can be slight imprecision in the location values of farther nodes, as the algorithm only computes the best likely members of the multicast route for the farther nodes. As the query for computing the multicast route reaches closer to the actual destination(s), the intermediate node can divert the query appropriately as it is assumed that the closer nodes will have more accurate information.

All mobile nodes are assumed to be having omni-directional antennas, as is the case with most of the off-the-shelf devices. As our algorithm relies on the neighbourhood information derived from the Hello protocol, all mobile nodes need not be in the same plane.

### 3.2. Basic Idea

In this algorithm, we assign positive charges to sender, the multicast destinations, and the packet. The packet is placed at a virtual point near the sender node, and will be guided by the effective repulsive force to move towards the nodes in the multicast group. All the other nodes are given a tentative charge based on QoS parameters.

When a packet reaches node  $u$ , we need to decide whether  $u$  is a branching node (a node at which the multicast tree branches) or not. If  $u$  is a branching node, we need to determine the number of branches to be taken, and the specific neighbours that will become part of the branch. If  $u$  is not a branching node, then we need to determine which of the neighbours in the forwarding direction has to be chosen for the next hop.

Let  $M$  be the set of nodes that are part of the multicast group. At the node  $u$ , we compute the effective force exerted on the packet due to nodes belonging to  $M$ ,  $\overline{F}_M$ . We take the direction of  $\hat{F}_M$  as the forward direction. We compute the effective force on each of the  $k$  neighbouring nodes  $v_1, v_2, \dots, v_k$  in neighbourhood set of the current node,  $N_u$ , by taking into account the cumulative effect of dampening force ( $\overline{E}_{v_i}$ ),  $u$  and  $M$ . By looking at the effective force on the neighbours, we determine the  $p$  neighbours  $w_1, w_2, \dots, w_p$  which form part of the multicast route, and forward the query to them.

### 3.3. System Model

Liu, *et al.* [7] defined Equation 1 for computing the virtual force the current node  $v$  to the destination  $v_d$ , where  $d_{\max}$  is the maximum distance measure, and  $d(v, v_d)$  is a measure of distance between  $v$  and  $v_d$ . This equation is sufficient for dealing with a unicast transmission, and for the computation of simple, point-to-point force between any two nodes. However, for a multicast transmission, we need to expand the notion to incorporate all the multicast group nodes.

$$F_{dest}(v, v_d) = d_{\max} - d(v, v_d) \quad (1)$$

Alternatively, a definition of destination force as given in Equation 2 can also be used, where  $Q$  is a large constant charge value assigned for ease of computation. This is a minor modification of Equation 4 in [7].

$$F_{dest}(v, v_d) = \frac{2 \times Q}{d(v, v_d)}, v \neq v_d \quad (2)$$

At any node in the network, we compute the effective force on the node  $u$  due to the current set of destinations,  $M$ , as shown in Equation 3. Here,  $\overrightarrow{F_{u,M}}$  is effective force on the node  $u$  exerted by all the nodes in the set of current multicast destination,  $M$ .

$$\overrightarrow{F_{u,M}} = \sum_{d \in M} \overrightarrow{F_{u,d}} \quad (3)$$

Apart from this force acting on the packet, there is a force component from the source node  $s$  and a dampening force caused by other parameters determining the choice of the next node. The effective force on the packet  $p$  at node  $u$  is given in Equation 4.

$$\overrightarrow{F_p} = \overrightarrow{F_{s,u}} + \overrightarrow{F_{u,M}} - \overrightarrow{E_u} \quad (4)$$

The dampening force  $\overrightarrow{E_u}$  produced by the node  $u$  depends on the various QoS parameters that are considered. If  $k$  parameters are considered, the weightage of the  $i^{\text{th}}$  parameter is  $\alpha_i$ , the value of the parameter  $i$  in the node  $u$  is  $\delta_i^u$ , the requested value for the QoS parameter is  $\delta_{req}$  and the maximum value allowed for that parameter is  $\lambda_{max}$ , then the value of the dampening force at node  $u$  is given by Equation 5. Here, for each parameter  $i$ ,  $0 \leq \beta_i \leq 1$  and  $\sum \beta_i = 1$ .

$$\overrightarrow{E_u} = \left( \sum_i \frac{\alpha_i \cdot (\delta_{req} - \delta_i^u)}{\lambda_{max}} \right) \cdot \hat{F}_{s,u} \quad (5)$$

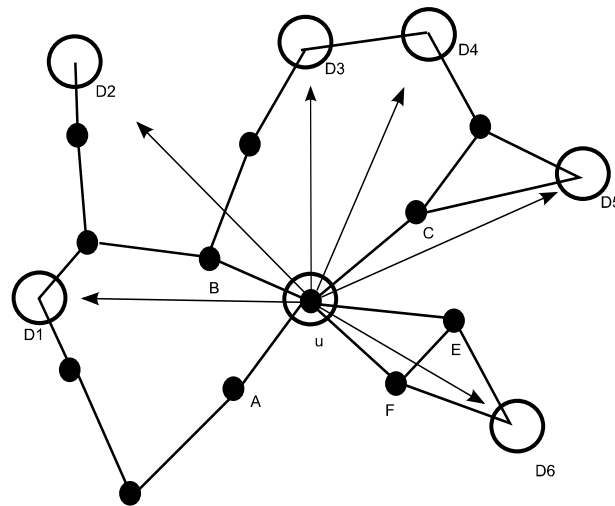


Figure 1. A sample multi-cast split node.

#### 4. MULTICAST ROUTING ALGORITHM USING VIRTUAL-FORCE

In this section, we discuss the design of the Multicast Routing Algorithm using Virtual-force (MRAV) and its pseudo-code.

##### 4.1. A sample working of MRAV

Figure 1 shows the node  $u$  that needs to multicast to six members of the multicast group  $M=\{D1,D2,D3,D4,D5,D6\}$  represented as circles. Other members of the MANET are indicated as black bubbles. Viable communication links are indicated using lines. The node  $u$  has five neighbouring nodes, namely A, B, C, E and F.

Let us consider the example of a multicast from node  $u$  to the set of destination nodes depicted in Figure 1. The packet at node  $u$  experiences virtual force as a result of interactions with all the six destination nodes. The direction of the force is indicated with the help of arrows in the figure. We first compute the effective force on the packet that is currently at node  $u$  to each of the destination nodes. The arrows in the figure indicate the direction of the force vectors due to the six destinations. By computing the effective force on the node, we can compute the general forward direction of the node.

The Figure 2(a) shows the various force values on the current node,  $u$ , of the network shown in Figure 1. Here the resultant force at the node  $u$  is computed as vector addition of all forces at the point corresponding to node  $u$ . For the purpose of illustration, let us assume that the forward direction, which is the direction of the resultant force, is towards  $D3$ . Figure 2(b) shows the cumulative sum of forces along the axes at node  $u$ . Since there are destination forces affecting more than two directions in Figure 2(b), it can be inferred that the node  $u$  is a branching node, also termed as a split node. It now has to decide which among the nodes in its neighbourhood set ( $N_u$ ) will form part of the multicast route.

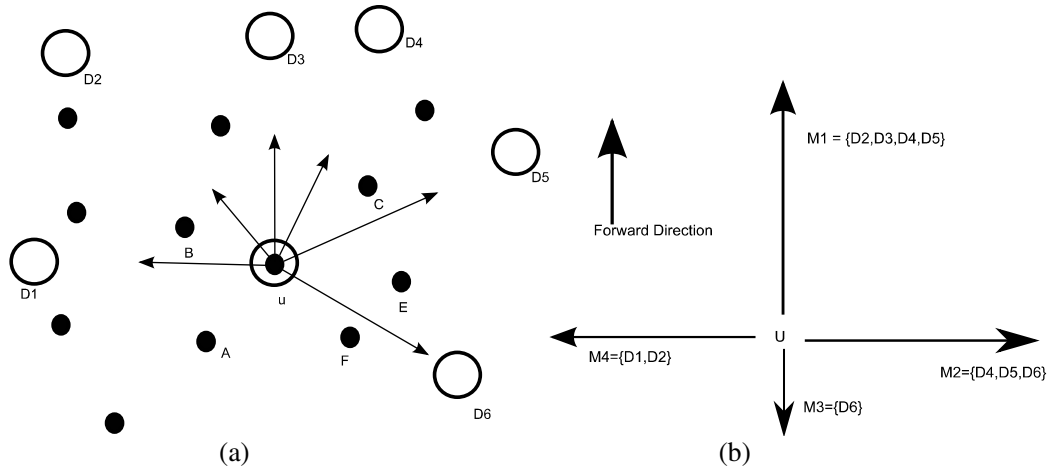


Figure 2(a). The virtual forces exerted by the destinations on node  $u$ . (b) Effecting force along axes.

Let us have four subsets of the destination set  $M$ , named as  $M_1, M_2, M_3$  and  $M_4$ , indicating the set of destination nodes that had contributed to the force vector upwards, rightwards, downwards and leftwards respectively. In Figure 2(b), the elements belonging to the subsets are indicated along with the force component to which they contribute. It can be seen that  $D6$  belongs to the

destination subset  $M_3$ ,  $M_2$  contains {D4, D5} and  $M_4$  contains {D1, D2}.  $M_1$  contains D2, D3, D4, and D5.

From Figure 2(b), it can be inferred that only the destination D6 is contributing to the downward direction, which is the direction opposite to the direction of the effective force at u. Also, in a clockwise direction starting from the left-side, it can be observed that the angle between D1 and D6 is greater than  $180^\circ$ . Because of the factor that it is the only node in subset  $M_3$  as well as being more than  $120^\circ$ , D6 is considered separate from the remaining set of destination nodes. To reach D6, the relevant neighbours of u are the nodes E and F. If E is chosen as a multicast forwarding node for u, then both destination D5 and D6 can be served through it. In our algorithm, all nodes that are within  $\pm 60^\circ$  from the direction of the chosen force vector are considered to be part of the same general path towards multicast set. The Query message is forwarded to node E, with destination set as {D5, D6}

When the multicast route query reaches the node E, it identifies that D6 is its neighbour. So the node E directly communicates the query message to D6. To communicate to D5, however, there is no path in the general direction towards it. Thus choice of E as the next node neighbour for communicating to D5 is a wrong one.

To circumvent this issue of wrongly identifying a neighbour to forward the multicast query, we first evaluate all possibilities first, before deciding on the list of most suitable forwarding nodes. If the node still encounters a void, we choose perimeter routing as discussed in Greedy Perimeter Stateless Routing (GPSR)[13].

In this case, we first try to combine D5 and D6 as was discussed earlier, and then go on to determine the rest of the nodes to forward the query. The nodes D1 and D2 can be considered together, as they form part of the set  $M_4$ . The effective force that acts on the node B is the strongest among the neighbours of u. Hence, the forwarding node for these two multicast destinations will be B. The remaining nodes D3 and D4 can now be considered. The closest neighbour according to our force metric is node C. For the destination node D5, while comparing between the node C and the node E, it can be observed that C feels a stronger force than E. The destination D5 is hence removed from the *Query* message to node E and then attached to node C. The *Query* message is going to be forwarded to the nodes B, C and E with the destination sets marked as {D1, D2}, {D3, D4, D5} and {D6} respectively. Note that the node F can be used to replace E, as the effective force value on both E and F due to node D6 is the same.

## 4.2. The MRAV Algorithm

In this section, we discuss the multicast routing algorithm using virtual-force.

### 4.2.1. Data structures

Data structures relevant for the algorithm are mentioned in this sub section.

$N_u$  : This is typically implemented as a linked list to store the list of neighbours of the current node. The list is updated with the help of responses from the periodic Hello packets transmitted as per the Hello Protocol.

$P(M)$ : This is the power set of M, such that all nodes within each of the subset created are reachable in increasing values of the angles. For example, in the sample network in Figure 1,  $P(M)$  may contain {D1, D2, D3}, but may not contain {D1, D3} as there exists a node D2 direction is in between D1 and D3.

$\Pi$  : This is a priority queue that stores the subset of nodes with highest force value on the front of the queue. It is assumed that this queue also has separate lookup function implemented that can retrieve the force value for any subset given as input to the function. This can be achieved by including an auxiliary data structure in the form of a hash table along with  $\Pi$ .

$v_s$ : The set of possible neighbours to choose from, while marking a split node. This list stores the next node as well as the list of destinations that are reachable from it.

#### 4.2.2. Algorithm

The algorithm for sending a *Query* message is shown in Algorithm 1. Once the *Query* message has reached the destination or a split node, a virtual circuit linking the destination and the previous split node or the source node shall be constructed as the acknowledgement to the *Query* travels back to the source node.

---

#### Algorithm 1: Multicast routing Algorithm using Virtual-force

---

```

1: procedure SendQuery(u,M)
2:   if  $u \in M$  then // The current node is itself a multicast destination
3:      $M \leftarrow M - \{u\}$ 
4:   end if
5:   for each  $v \in N_u \wedge v \in M$  do // A neighbour of the current node is a multicast destination
6:     Call SendQuery(v,M)
7:   end for
8:   Let  $M' = M$ 
9:   Repeat
10:    for each  $s \mid s \in P(M)$  do
11:       $\vec{F}_s \leftarrow \text{calculateVirtualForce}(u,s)$ 
12:      Add  $\vec{F}_s, s$  to  $\Pi(M)$ 
13:    end for
14:    Let  $M_s$  be front( $\Pi(M)$ )
15:    Compute  $w \in N_u, s.t. \angle uw, \hat{M}_s \leq 60^\circ \vee \angle \hat{M}_s, \vec{uw} \leq 60^\circ$ 
16:    if  $M_s = M$  then // all nodes are in the same general direction
17:      Call SendQuery(w,M) and exit
18:    end if
19:    Add (w,  $M_s$ ) to  $v_s$ 
20:    Remove all entries from P(M) which contain the destination nodes  $M_s$ 
21:     $M' = M' - M_s$ 
22:  while  $M' \neq \emptyset$ 
23:  Mark u as split node
24:  for each  $v \in v_s$  do
25:    if  $v_s \neq \emptyset$  then
26:      select  $w \in v_s \mid \vec{F}_{\max} = \max_{w \in v_s} \vec{F}_{w, M_s} - \vec{E}_w$ 
27:      Check for other destination in the direction of uw and verify whether they can also
    
```

---



---

```

                be addressed by the same neighbour, w.
28:             SendQuery(w, Ms)
29:         end if
30:     end for
31: end procedure

```

---

## 5. SECTOR-BASED MULTICAST ROUTING ALGORITHM

In this section, we discuss the sector-based virtual-force-based multi-cast routing algorithm.

### 5.1. Motivation

Though MRAV uses virtual force technique to correctly identify the set of neighbours to whom the query message needs to be forwarded, the message complexity for making the computations is on the higher side. One of the reasons for higher overhead and energy consumption derives from the fact that the algorithm compares all the nodes in the neighbourhood for making the routing decision. However, most of the nodes queried are not in the general direction of communication. The number of nodes that need to be enquired about the effective force values can hence be reduced. At each node  $u$ , we need to identify the subset of  $N_u$  that are least likely to be part of the multicast route.

The primary challenge while trying to decide which nodes can be excluded for path computation comes from the uncertainty regarding the rest of the network.

### 5.2. Basic working

Unlike MRAV and [12], we divide the nodes around  $u$  into  $\alpha$  sectors, with each sector covering an angle of  $\theta = \frac{2 \cdot \pi}{\alpha}$ . Here,  $\alpha$  varies from 3 to 6 depending on the density of the region containing  $u$ . For  $\alpha = 4$ , we have  $\theta = \frac{\pi}{2}$ . For each of the  $\alpha$  sectors containing some destination nodes, we select the appropriate neighbour in that sector to forward the packet. We compute the effective force on each of the  $k$  neighbouring nodes  $v_1, v_2, \dots, v_k$  in the sector  $s$ , by taking into account the cumulative effect of dampening force ( $\overline{E}_{v_i}$ ),  $u$  and  $M_s$ , where  $M_s \subseteq M$ ,  $M_s$  contains all the destination nodes in sector  $s$ . Once the appropriate forwarding node  $v_s$  for the sector  $s$  is determined, the packet is communicated to  $v_s$  along with  $M_s$ , which is the set of destination nodes to be handled in sector  $s$ .

When a source node  $s$  is interested in transmitting a packet to a multicast group  $M$ , it first sends a *Query* message to establish the multicast route. This *Query* message contains a pointer to indicate the multicast group and the set of nodes in  $M$  represented using bit array. The algorithm for sending the *Query* message is given in Algorithm 1.

When a node  $u$  decides on forwarding a packet to the destinations, it determines whether it has to split the multicast packet or not. The node determines whether all destinations are in the same general direction or not by computing the effective force from destinations,  $\overline{F}_{u,M}$ . If  $\overline{F}_{u,M}$  and  $\overline{F}_{s,u}$  are in opposing directions, then the node determines the appropriate neighbour in the forward direction to route the *Query* packet. The best way to determine whether all nodes are in

the same general direction is to check whether the direction of force from each destination,  $\overrightarrow{F_{d_i}}$ , where  $d_i \in M$ , are within  $\pm \frac{\theta}{2}$  angle from  $\overrightarrow{F_{u,M}}$ .

Figure 3 shows a simple illustration on how the force vectors are used in our algorithm. First, by using Equation 3, the effective repulsive force on the node  $u$  due to the destinations in multicast group  $M$ , given as  $\overrightarrow{F_{u,M}}$ , is computed. The next step is to divide the region around the node  $u$  into  $\alpha$  sectors. Here  $\alpha$  is the split parameter that we use. For dividing the region into sectors, we first look at the unit vector corresponding to the effective destination force,  $\hat{F}_{u,M}$ . From the direction of the unit vector, we take all nodes within angle  $\pm \frac{\theta}{2}$  as the first sector, where  $\theta = \frac{2 \cdot \pi}{\alpha}$ . In the example in Figure 3,  $\alpha=4$  and  $\theta = \frac{\pi}{2}$ . The nodes within the next  $\theta$  radians form part of the second sector. Thus,  $\alpha$  sectors will be marked around the current node  $u$ . The four sectors in Figure 3 are divided by the dotted lines as illustrated.

Now, we divide the neighbours of  $u$  into  $\alpha$  sets ( $V_1, V_2, \dots, V_\alpha$ ), such that all neighbouring nodes in sector  $s$  are put inside the set  $V_s$ . We apply the same procedure to divide the nodes in  $M$  into ( $M_1, M_2, \dots, M_\alpha$ ), where each set  $M_s$  indicates the set of destination nodes in the current destination set  $M$ . In the example in Figure 1, the set  $M$  is divided into two sets  $M_1 = \{d_1\}$  and  $M_2 = \{d_2, d_3, d_4\}$  corresponding to sectors 1 and 2. Similarly, the neighbours of  $u$  are divided into two sets;  $V_1 = \{v_{n1}, v_{n2}\}$  and  $V_2 = \{v_{n3}, v_{n4}, v_{n5}\}$ . Here,  $\bigcup_i V_i = N_u$ , where  $N_u$  is the set of neighbouring nodes of the current node  $u$ . Also,  $\bigcup_i M_i = M$ .

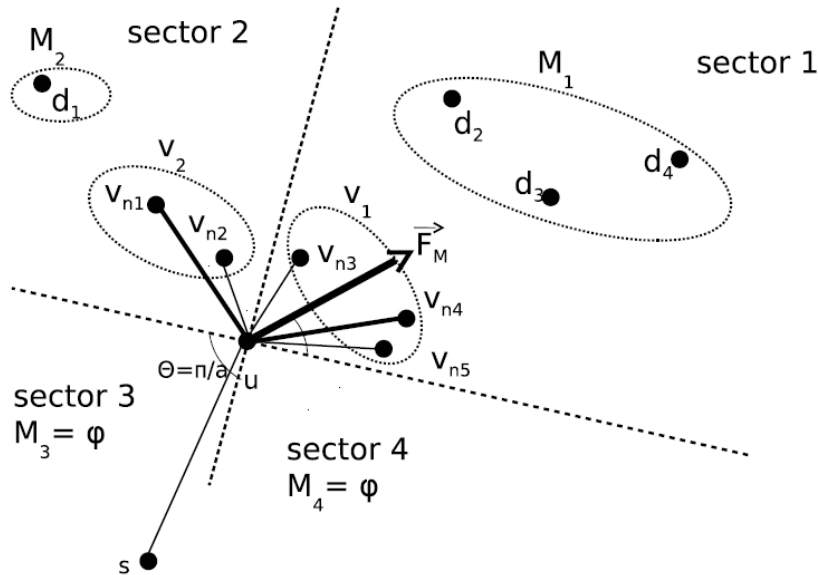


Figure 3. Virtual Force acting on node  $u$  with  $\alpha=4$ .

Once the subsets of  $M$ ,  $M_s$ , has been computed, we can easily determine whether any sector is having destination or not by checking for the condition:  $M_s = \phi$ . If the condition holds true, then

we don't have to explore that sector. If  $M_s \neq \emptyset$ , then we have to determine appropriate neighbour to forward the packet in that sector  $s$ .

To choose the appropriate neighbour in sector  $s$ , first we have to determine whether  $v_s = \emptyset$ . If so, then we have encountered a void in that sector. The simplest way to overcome void in the network is to follow the approach taken by Liu, *et al.* in [7]. Virtual force is used to cross across the void if possible, or else the perimeter rule of greedy perimeter stateless routing (GPSR) algorithm [13] is applied.

If  $v_s \neq \emptyset$ , then the effective force on the packet as given in Equation 4 is computed for each neighbour in  $v_s$ . The neighbour  $w \mid w \in v_s$  with the maximum force on the packet is chosen as the next hop neighbour.

### 5.3. Algorithm

The algorithm that uses the notion of sectors and virtual force, sector-based virtual force-based multicast routing algorithm (VFM), is shown in Algorithm 2.

---

#### Algorithm 2: Virtual Force-based Multicast routing algorithm

---

```

1: procedure SendQuery(u,M)
2:   if  $u \in M$  then
3:      $M \leftarrow M - \{u\}$ 
4:   end if
5:   for each  $v \in N_u \wedge v \in M$  do
6:     Call SendQuery(v,M)
7:   end for
8:    $\overline{F}_{u,m} \leftarrow \text{calculateVirtualForce}(u,M)$ 
9:   Divide region into  $\alpha$  sectors centred at  $u$ , each of angle  $\theta$  such that first sector lies within  $\pm \frac{\theta}{2}$  of  $\hat{F}_{u,m}$ 
10:  for each sector  $s$  do
11:     $v_s \leftarrow \emptyset$ 
12:    for each  $v \in N_u$  do
13:      if  $s = \text{sector}(v)$  then
14:         $v_s \leftarrow v_s \cup v$ 
15:      end if
16:    end for
17:  end for
18:  for each sector  $s$  do
19:     $M_s \leftarrow \emptyset$ 
20:    for each  $d \in M$  do
21:      if  $s = \text{sector}(d)$  then
22:         $M_s \leftarrow M_s \cup d$ 
23:      end if
24:    end for
25:  end for

```

---

---

```

26: for each sector  $s$  such that  $M_s \neq \emptyset$  do
27:     if  $V_s \neq \emptyset$  then
28:         select  $w \in V_s \mid \overrightarrow{F_{\max}} = \max_{w \in V_s} \overrightarrow{F_{w, M_s}} - \overrightarrow{E_w}$ 
29:         SendQuery( $w, M_s$ )
30:     end if
31: end for
32: end procedure

```

---

## 6. SIMULATION RESULTS

We have performed simulation of our algorithm using a simulator written in Python language. The simulator uses the same energy model and Two Ray propagation model as implemented in ns-2. For the simulation runs, we have placed nodes in a fixed area of 2000m x 2000m with maximum transmission range set as 250m. We have compared MRAV and VFM with the QBIECRA algorithm proposed by Rahman, *et al.* [11]. While comparing with the sector-based VFM algorithm, we have taken  $\alpha$  values as 3, 4 and 6. We have chosen QBIECRA since it uses a notion of quadrants for computing the multicast tree. Though Rahman, *et al.* also proposed another version of their quadrant based algorithm in [12], while performing simulations we didn't perceive a major difference between the results of QBIECRA and 4-N Intelligent routing. This is may be due to the fact that these algorithms only differ on the basis of which four neighbours are going to be chosen for computing the next forwarding node in the multicast path. The simulation results obtained are discussed in the rest of this section.

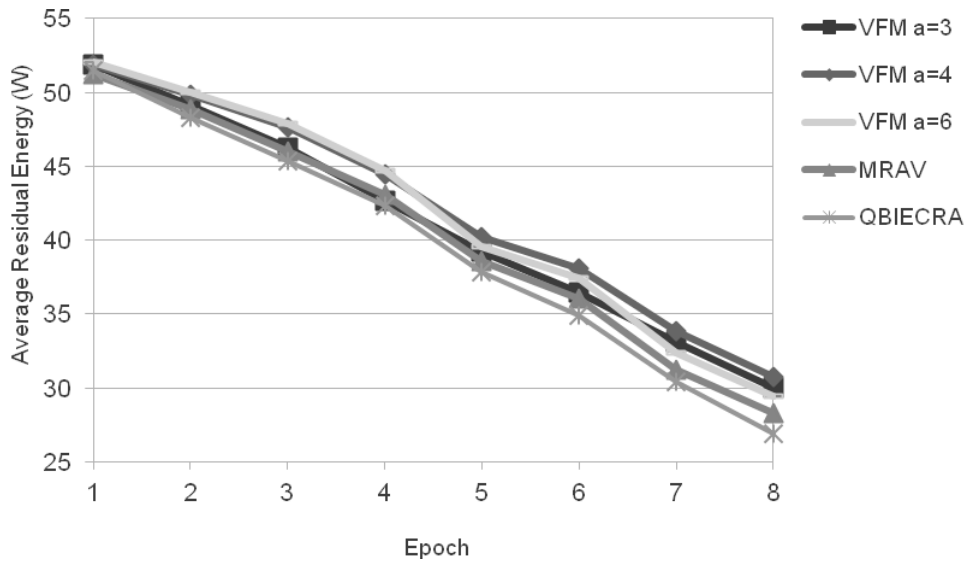


Figure 4. Average normalized residual energy over time

Average normalized residual energy is computed as the average residual energy of all nodes in the network normalized in percentage terms to account for variations in initial energy values of various nodes. We have computed the average of residual energy in each of the nodes after completing one cycle of packet transmissions for a sender node in each epoch, and are showing the normalized value in percentage terms after each epoch. The results for normalized average

residual energy are shown in Figure 4. MRAV performs slightly better than QBIECRA. We observed that in the initial period, the choices made by QBIECRA and MRAV were same in many cases. But the choices began to diverge only later, when there was significant difference between the choices made, primarily due to the effect of the dampening force. For VFM, we observed that for all values of  $\alpha$ , it performed as same as or better than QBIECRA and MRAV.

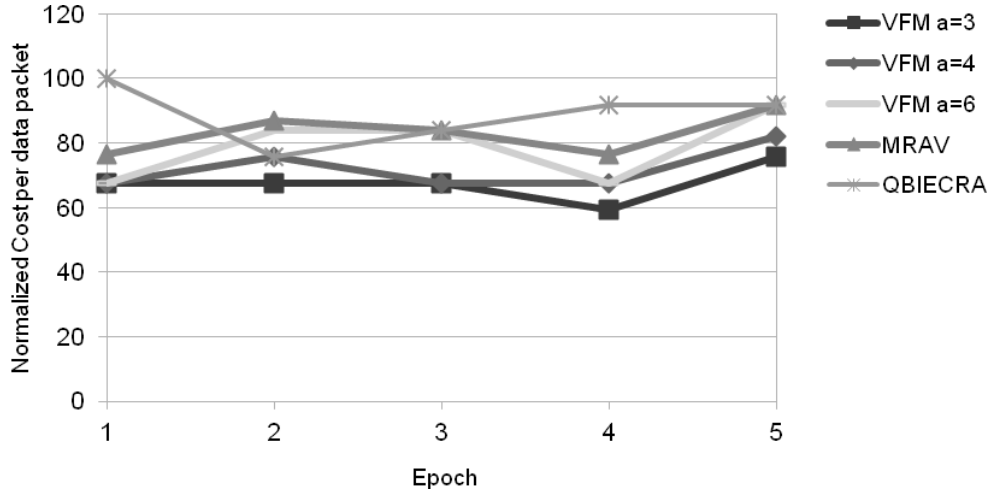


Figure 5. Normalized cost per data packet over time

Normalized cost per data packet is computed as the overall cost for sending a data packet from source node to destinations that is normalized in terms of percentile energy cost for the sake of comparison across multiple simulation runs with varying network sizes. The results for this metric are shown in Figure 5. We observed that VFM performs better than QBIECRA primarily by optimizing the effective energy spent on optimal path computation. Due to the number of comparisons made, MRAV was performing worse than VFM. We couldn't statistically establish a clear difference between MRAV and QBIECRA based on normalized cost per data packet.

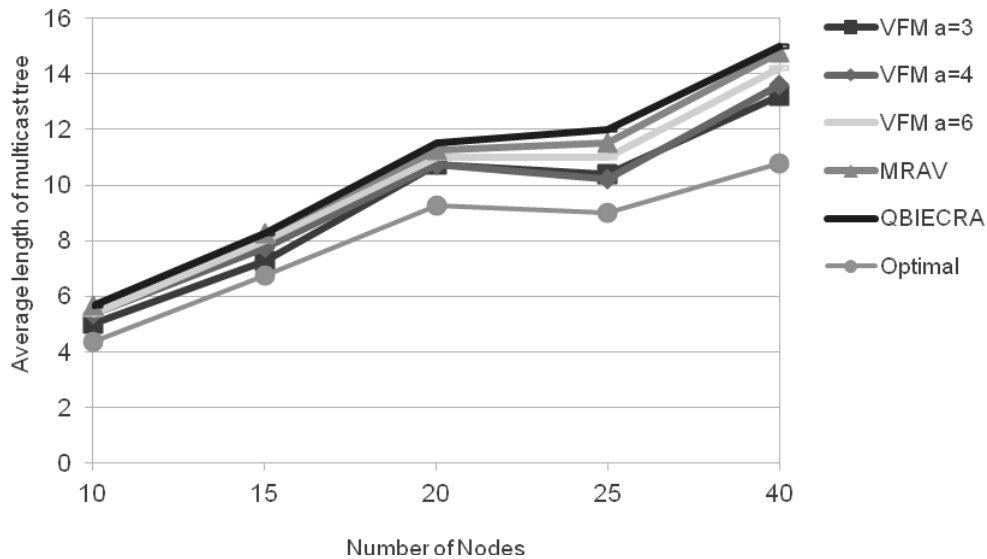


Figure 6. Average length of multicast tree with respect to number of nodes

We also observed the effective length of the multicast tree generated as shown in Figure 6. We observed a variation in the general trend of length of the multicast tree for 20 nodes for both of our algorithms. We attribute this variation to the choice of multicast set, presence of voids and other aspects specific to the test graphs used for simulation. Even with this variation, both MRV and VFM provided shorter length for multicast tree as compared to QBIECRA. MRV performed consistently better than QBIECRA as far as this metric is concerned for a larger network. For very small networks, there was no major difference between the two algorithms. We observed that  $120^\circ$  is the best angle between the branches for getting relatively minimal Steiner trees as defined by Gilbert, *et al.*[2]. As expected, the length of the multicast tree was nearing the optimal values for  $\alpha=3$ . We surprisingly found that we were getting good results for  $\alpha=4$  as well. This was primarily due to the fact that we were not having many out-going edges in the opposite direction of the forward path. When it came to  $\alpha=6$ , our algorithm performed worse as expected, as we were generating too many out-going edges in some nodes resulting in a sub-optimal multicast tree. However, the performance of VFM algorithm was still better than QBIECRA as seen from the figure.

## 7. CONCLUSIONS

We have applied the notion of virtual force for energy efficient multicast routing in MANETs. We have presented two algorithms centred on the virtual-force technique. Our algorithm generates relatively minimal Steiner trees for use in multicast routing. The simulation results indicate that our algorithm performs better than other quadrant/sector based multicast routing algorithm. We have evaluated appropriate choice for the number of sectors to be used, and have found that  $\alpha = 3$  or 4 can be used to generate energy efficient multicast paths. While we were applying the notion of virtual forces for multicasting, we observed that the approach as a whole provides good results. We haven't yet explored the relationship between the value of  $\alpha$  and network density and spread. However, we believe that there is still scope in fine tuning the force model used. Though we had only used life of the network for QoS, the dampening force can easily be extended to include other parameters as well. The proposed algorithms have shown that the virtual force approach can be successfully used in MANETs. To our knowledge, this is the first paper that performs multicast routing algorithm using the virtual force approach in mobile ad-hoc networks. This is also the first paper to perform virtual-force computation on the basis of sectors.

## REFERENCES

- [1] Luo Junhai, Ye Danxia, Xue Liu, and Fan Mingyu. A survey of multicast routing protocols for mobile ad-hoc networks. *Communications Surveys Tutorials, IEEE*, 11(1):78–91, quarter 2009.
- [2] E. Gilbert and H. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.
- [3] Marshall Bern and Paul Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32(4):171 – 176, 1989.
- [4] Glencora Borradaile, Philip Klein, and Claire Mathieu. An  $o(n \log n)$  approximation scheme for steiner tree in planar graphs. *ACM Trans. Algorithms*, 5(3):31:1–31:31, July 2009.
- [5] S. Poduri and G.S. Sukhatme. Constrained coverage for mobile sensor networks. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 165 – 171 Vol.1, april-1 may 2004.

- [6] Y. Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293 – 1303 vol.2, march-3 april 2003.
- [7] Cong Liu and Jie Wu. Virtual-force-based geometric routing protocol in manets. *Parallel and Distributed Systems, IEEE Transactions on*, 20(4):433 –445, april 2009.
- [8] Cong Liu and Jie Wu. Swing: Small world iterative navigation greedy routing protocol in manets. In *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings.15th International Conference on*, pages 339 –350, oct. 2006.
- [9] Martin Mauve, Holger Füssler, Jörg Widmer, and Thomas Lang. Position-based multicast routing for mobile ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):53–55, July 2003.
- [10] S. Fotopoulou-Prigipia and A.B. McDonald. Gcrp: geographic virtual circuit routing protocol for ad hoc networks. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 416 – 425, oct. 2004.
- [11] F.M. Rahman and M.A. Gregory. Quadrant based intelligent energy controlled multicast algorithm for mobile ad hoc networks. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 1298 –1303, feb. 2011.
- [12] F.M. Rahman and M.A. Gregory. 4-n intelligent manet routing algorithm. In *Australasian Telecommunication Networks and Applications Conference (ATNAC), 2011*, pages 1 –6, nov. 2011.
- [13] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00*, pages 243–254, New York, NY, USA, 2000. ACM.

#### Authors

**Murali Parameswaran:** is currently pursuing his PhD in Computer Science and Engineering from Birla Institute of Technology & Science, Pilani. His research interests include: Distributed Algorithms, Wireless Ad hoc Networks, and Network Security. He is a member of IEEE, ACM and IETE.



**Chittaranjan Hota:** did his PhD in Computer Science and Engineering from Birla Institute of Technology & Science, Pilani. He was the founding head at BITS Hyderabad and currently he is the Faculty In-charge of Information Processing Center. He has been a visiting researcher and visiting professor at several universities abroad over past few years. He has current research funding from organizations like UGC, DIT, Intel, and TCS. He has guided PhD students and currently guiding PhD students in the area of P2P Overlays, Information Security, Wireless Networks, and Distributed Scheduling. He is a member of IEEE, ACM, IE, and ISTE.

