

Impact of Route Selection Metrics on the Performance of On-Demand Mesh-based Multicast Ad hoc Routing

Natarajan Meghanathan (Corresponding Author)

Department of Computer Science, Jackson State University

P. O. Box 18839, 1400 John R. Lynch Street, Jackson, MS 39217, USA

Tel: 01-601-979-3661 E-mail: natarajan.meghanathan@jsums.edu

Srilakshmi R. Vavilala

Department of Computer Engineering, Jackson State University

E-mail: srilakshmi.vavilala@gmail.com

Abstract

The main objective of this paper is to study the stability and energy consumption issues of mesh-based multicast routing for mobile ad hoc networks (MANETs). This has been accomplished as follows: (i) The well-known mesh-based on-demand multicast routing protocol (ODMRP) is modified to choose routes based on two different route selection metrics: (a) hop count, as chosen by the Dynamic Source Routing (DSR) protocol and (b) predicted link lifetime, as chosen by the Flow-Oriented Routing Protocol (FORP). The modified ODMRP is referred to as ODMRP_DSR and ODMRP_FORP respectively; (ii) We propose an algorithm called *OptMeshTrans* to determine the sequence of stable multicast meshes connecting a set of sources to a set of receivers, such that the number of mesh transitions is minimal. Simulation results indicate that the multicast meshes determined using ODMRP_FORP are more stable than those of ODMRP_DSR. There is no appreciable difference between these two ODMRP implementations with respect to hop count per source-receiver path, number of edges and energy consumption per node. The meshes determined using *OptMeshTrans* are the most stable with relatively fewer edges and incur lower energy consumption per node when compared to the meshes determined using the other two protocols.

Keywords: Mesh, Multicast, Optimality, Stability, Energy Consumption, Simulations

1. Introduction

A mobile ad hoc network (MANET) is an autonomous system of mobile nodes connected through wireless links. Each node in the network, apart from being a source or destination, also routes packets for other nodes in the network. Routing protocols for MANETs are of two types: proactive and reactive (on-demand). Based on their route selection principle, on-demand unicast MANET routing protocols can be classified into two types (Meghanathan, 2009): minimum-weight based and stability-based. Dynamic Source Routing protocol (DSR – Johnson, Maltz & Broch, 2001) and Flow Oriented Routing Protocol (FORP – Su, Lee & Gerla, 2001) are respectively one of the best performing protocols under the minimum-weight and stability-based categories (Meghanathan, 2008). DSR discovers minimum hop routes and FORP discovers long-living stable routes based on the predicted lifetime of links.

Multicasting has emerged as a desirable and essential technology for several distributed applications in wireless networks such as audio/ video conferencing, distance learning, collaborative and groupware applications and etc. The mobility of nodes, with the constraints of limited battery charge and bandwidth, makes multicast routing a very challenging problem in MANETs (Murthy & Manoj, 2004). It is advantageous to use multicast rather than multiple unicast, especially in ad hoc environments, where bandwidth comes at a premium. MANET multicast routing protocols can be classified into two types based on the multicast topology (Murthy & Manoj, 2004): *tree-based* and *mesh-based*. In tree-based multicast routing protocols, there exists only a single path between a source-receiver pair, whereas in mesh-based multicast routing protocols there are multiple paths between a source-receiver pair. The presence of multiple paths adds to the robustness of the mesh-based protocols at the cost of multicast efficiency. We focus on mesh-based multicast routing in this paper.

The On-Demand Multicast Routing Protocol (ODMRP – Lee, Gerla & Chaing, 1999) is a well-known and classical mesh-based multicast protocol that uses a forwarding group concept. ODMRP is well suited for ad hoc wireless networks with mobile hosts where bandwidth is limited, power is constrained and topology changes frequently and rapidly. In this protocol, a mesh is formed by a set of nodes called forwarding nodes which are

responsible for forwarding data packets between a source-receiver pair. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. It uses shortest routes, exhibits robustness to host mobility and maintains multiple redundant paths.

Our contributions in this paper are two fold: First, we have implemented ODMRP with respect to the minimum hop metric of DSR and the predicted link lifetime metric of FORP. The modified implementations of ODMRP are referred to as ODMRP_DSR and ODMRP_FORP respectively. In ODMRP_DSR, a mesh is constructed by selecting minimum hop paths; whereas, in ODMRP_FORP, a mesh is constructed by selecting the routes that have the largest predicted lifetime. The predicted lifetime of a route is the minimum of the predict lifetime of its constituent links. Our hypothesis in this paper is that the performance of the ODMRP protocol may be influenced by the underlying route selection metrics used to form the mesh. Our second contribution is the development of the *OptMeshTrans* algorithm that determines a sequence of stable multicast meshes connecting the sources to all the multicast receivers such that the number of mesh transitions is minimal.

We evaluate the performance of the three implementations ODMRP_DSR, ODMRP_FORP and *OptMeshTrans* and compare them. We simulate the protocols/algorithms under different conditions of node mobility, network density, number of sources and receivers per multicast group. The performance metrics measured are the lifetime per mesh, number of edges per mesh, hop count per source-receiver path and energy consumed per node. Our simulations indicate that *OptMeshTrans* discovers multicast meshes that have a significantly larger lifetime compared to those discovered by the two ODMRP-based protocols. Also, the meshes discovered using *OptMeshTrans* have fewer edges per mesh and also incur lower energy consumption per node. On the other hand, meshes formed using ODMRP_DSR and ODMRP_FORP have relatively lower hop count. Throughout the paper, the terms ‘link’ and ‘edge’, ‘node’ and ‘vertex’, ‘protocol’ and ‘algorithm’, ‘source’ and ‘sender’, ‘receiver’ and ‘destination’ are used interchangeably. They mean the same.

The rest of the paper is organized as follows: In Section 2, we briefly review the three routing protocols DSR, FORP and ODMRP used in this paper. Section 3 reviews the related literature work on performance comparison studies involving ODMRP. Section 4 describes our *OptMeshTrans* algorithm to determine sequence of stable meshes. Section 5 describes simulation results involving ODMRP_DSR, ODMRP_FORP and *OptMeshTrans* and highlights the tradeoff between these three mesh-based implementations with respect to different performance metrics. Section 6 concludes the paper.

2. Review of DSR, FORP and ODMRP

2.1 Dynamic Source Routing (DSR) Protocol

The unique feature of DSR (Johnson, Maltz & Broch, 2001) is source-routing: data packets carry the route from the source to the destination in the packet header. As a result, intermediate nodes do not need to store up-to-date routing information as the data packets themselves have all the information required to make a routing decision. This also eliminates the need for the beacon control neighbor-detection packets – a common feature in the stability-oriented routing protocols. Route discovery in DSR is by means of the broadcast query-reply cycle. A source node s wishing to send a data packet to a destination node d , broadcasts a *Route-Request* (RREQ) packet throughout the network. The RREQ packet reaching a node contains the list of intermediate nodes through which it has propagated from the source node. A *Route-Reply* (RREP) packet to the source node is generated either at the destination or at a node that has a cached route to the destination. Each node maintains a route cache in which it stores all the source routes learned or overheard. DSR uses route caching to limit the frequency and propagation of RREQ packets. Also, the destination node sends multiple RREP packets to the source, each time it learns a better s - d route (a minimum hop route) through the RREQ packets reaching it from the source. As a result, it is possible for the source node to learn multiple routes to the destination, which again is aimed to reduce the number of network-wide flooded RREQ packets.

2.2 Flow-Oriented Routing Protocol (FORP)

FORP (Su, Lee & Gerla, 2001) utilizes the mobility and location information of nodes to approximately predict the expiration time (LET) of a wireless link. The minimum of the LET values of all wireless links on a path is termed as the route expiration time (RET). The route with the maximum value of the RET is selected. Each node is assumed to be able to predict the LET values of each of its links with its neighboring nodes based on the information regarding the current position of the nodes, velocity and the direction of movement, and transmission ranges. FORP assumes the availability of location-update mechanisms like Global Positioning System (GPS – Hofmann-Wellenhof, Lichtenegger, & Collins, 2004) to identify the location of nodes and also assumes that the clocks across all nodes are synchronized. Route discovery is similar to the broadcast

query-reply cycle described in Section 2.1, with the information propagated in the RREQ packet being the predicted LET of each link in a path.

Given the motion parameters of two neighboring nodes, the duration of time the two nodes will remain neighbors can be predicted as follows: Let two nodes i and j be within the transmission range of each other. Let (x_i, y_i) and (x_j, y_j) be the co-ordinates of the mobile hosts i and j respectively. Let v_i, v_j be the velocities and Θ_i, Θ_j , where $(0 \leq \Theta_i, \Theta_j < 2\pi)$ indicate the direction of motion of nodes i and j respectively. The amount of time the two nodes i and j will stay connected, D_{i-j} , can be predicted using the following equation:

$$D_{i-j} = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)r^2 - (ad - bc)^2}}{a^2 + c^2}$$

where, $a = v_i \cos\Theta_i - v_j \cos\Theta_j$; $b = x_i - x_j$; $c = v_i \sin\Theta_i - v_j \sin\Theta_j$; $d = y_i - y_j$

2.3 On-Demand Mesh-based Routing Protocol (ODMRP)

ODMRP (Lee, Gerla & Chaing, 1999) is a mesh-based multicast protocol using a forwarding group concept. ODMRP suits well for ad hoc networks that have limited bandwidth, frequently changing topology and nodes operating with limited battery charge. A mesh is formed by a set of nodes called forwarding nodes which are responsible for forwarding data packets between a source-receiver pair. ODMRP operates in two phases: Mesh Initialization Phase and Mesh Maintenance Phase.

In the *Mesh Initialization Phase*, a multicast mesh is formed between the sources and receivers. The source node floods the *Join-Request* control packets in order to create a mesh. The intermediate nodes that receive a *Join-Request* control packet broadcast the packet again after storing the upstream node identification number (ID). The receivers on receiving the *Join-Request* control packets respond with a *Join-Reply* along the shortest reverse path. The *Join-Reply* packet contains a source ID and the corresponding next node ID. When a node receives the *Join-Reply* control packet from the receiver, the node sets a forwarding flag and becomes the forwarding node for that multicast group. The node after waiting for a specific amount of time composes a new *Join-Reply* packet and forwards it to its upstream node in the selected path. In this way, a route is established by forwarding subsequent *Join-Reply* packets by the intermediate nodes. In the *Mesh Maintenance Phase*, attempts are made to maintain the multicast mesh formed with sources, forwarding nodes and receivers. Here, a soft state approach is used to maintain the mesh wherein the source node periodically floods the *Join-Request* control packet to refresh the route between source and receiver. The multicast mesh protects the session from being affected by the mobility of the nodes.

As ODMRP uses the soft state approach for maintaining the mesh, it exhibits robustness. But this robustness is at the expense of high control overhead. The duplicate data packets and *Join-Request* control packets are detected by the forwarding nodes using a *message-cache*. The multicast efficiency is reduced when the destination node receives the same data packet through more than one path (i.e., increase in data packet transmissions).

3. Related Work on Performance Comparison Studies involving ODMRP

The following are some of the performance comparison studies involving ODMRP that we identified in the literature:

- (i) Zhang & Jacob (2003) propose a Multicast Zone Routing Protocol (MZRP) and compare its performance with that of ODMRP. MZRP uses a zone routing mechanism that proactively maintains routing information for a local neighbor within its routing zone and reactively acquires routes to destinations beyond the routing zone. MZRP exhibits better normalized multicast routing overhead than ODMRP; whereas, ODMRP exhibits a higher packet delivery ratio that increases with the number of multicast senders. The packet delivery ratio of ODMRP is also relatively higher as the mobility of the nodes is increased.
- (ii) Vasililiou and Economides (2005) compare the performance of ODMRP with the multicast extension to the Ad hoc On-demand Distance Vector routing protocol (MAODV – Royer & Perkins, 1999). As traffic increases, MAODV incurs lower end-to-end delay per packet than ODMRP. In networks of smaller density, ODMRP achieves lower end-to-end delay per data packet, but also a lower packet delivery ratio. In networks of larger density, MAODV incurs lower end-to-end delay per data packet, but also a lower packet delivery ratio. The difference in node speed does not affect the packet delivery ratio of ODMRP; but, MAODV incurs higher packet delivery ratio at higher node speeds.
- (iii) Chen, Wang & Lee (2005) propose a Hybrid Overlay Multicast Routing Protocol (HOMRP) and compare with ODMRP and Ad hoc Multicast Routing Protocol (AMRoute – Xie, Talpade, McAuley & Liu, 2002). HOMRP uses an overlay approach to combine the advantages of both tree-based and mesh-based

protocols to achieve better performance. ODMRP achieves the highest packet delivery ratio and lower end-to-end delay per data packet, closely followed by HOMRP. The packet delivery ratio of AMRoute is only about 50% of that achieved by HOMRP and ODMRP. The control overhead of HOMRP is 18% less compared to ODMRP and 12% less compared to AMRoute.

(iv) Begdillo, Asadi & Haghigat (2007) observed that the packet delivery ratio of ODMRP was improved using multi-coded paths. According to this scheme, three multi-paths (multiple paths) are maintained on a mesh between a source and destination. If two different data packets p_1 and p_2 are sent on two of the three paths, a data packet that is nothing but an XOR of p_1 and p_2 is sent on the third path. If one of the three data packets gets lost, the destination could retrieve the lost packet by using the XOR relationship between the three packets.

(v) A performance comparison study (Mazinan, Arabshahi & Adim, 2008) of ODMRP and the source-initiated, shared-tree based Ad hoc multicast routing Protocol utilizing increasing ID-numbers (AMRIS – Wu, Tay & Toh, 1998) reveals that ODMRP shows a significantly better performance in terms of packet delivery ratio as one increases node mobility and/or the offered traffic load. AMRIS shows a near constant value of the control overhead, but ODMRP shows an increase in control overhead as the number of senders increases.

(vi) Lee, Su, Hsu, Gerla, & Bagrodia (2000) conduct a comprehensive study by comparing the performance of AMRoute, AMRIS, Core-Assisted Mesh Protocol (CAMP – Garcia-Luna-Aceves, & Madruga, 1999), ODMRP and flooding. Simulation results illustrate that under varying node mobility, the packet delivery ratio of ODMRP is close to that of flooding – the strategy that incurs the largest packet delivery ratio. AMRIS incurs the lowest packet delivery ratio, even in the absence of node mobility. AMRoute incurs the highest number of data packets transmitted per data packet delivered because of loops in its forwarding structure. AMRIS incurs lower control overhead compared to other protocols and CAMP incurs a higher control overhead than ODMRP at high mobility conditions. The performance of ODMRP and CAMP improved with increase in number of senders. The size of the multicast group does not affect the performance of ODMRP and flooding, whereas, the performance of CAMP improves with increase in the number of receivers.

The above performance comparison studies in the literature illustrate that, in general, mesh-based protocols perform better than tree-based protocols in mobile environments and, in particular, ODMRP is very effective and efficient in most of the scenarios and with respect to most of the metrics. Hence, it is worth studying the impact of the route selection metrics on the performance of ODMRP and comparing the performance of ODMRP with respect to the theoretical algorithm *OptMeshTrans* that determines the sequence of stable meshes.

4. Algorithm to Determine Sequence of Stable Meshes

The *OptMeshTrans* algorithm to determine sequence of long-living stable meshes uses the following greedy strategy: Whenever a mesh connecting a set of source nodes to a set of receiver nodes is required, we choose the mesh, called the Stable-Static-Mesh, which exists for the longest time. In this pursuit, we determine a long-living minimum edge Steiner tree connecting the source nodes to the receiver nodes and the Stable-Static-Mesh is an extension of this Steiner tree by including in the mesh, all the edges that exist between the constituent nodes of the tree. We use the Kou et al.'s heuristic (Kou, Markowsky, & Berman, 1981) to approximate a multicast Steiner tree connecting the set of source nodes to the set of receiver nodes using the minimum number of edges. When such a Stable-Static-Mesh gets disconnected, leading to the absence of a path from any source to any receiver node, we use the above greedy principle to construct another long-living mesh. The sequence of long-living Stable-Static-Meshes determined over the duration of the multicast session time is called a Stable-Mobile-Mesh.

4.1 Mobile Graph

A *mobile graph* (Farago & Syrotiuk, 2003) is defined as the sequence $G_M = G_1G_2 \dots G_T$ of static graphs that represents the network topology changes over a time scale T . In the simplest case, the mobile graph $G_M = G_1G_2 \dots G_T$ can be extended by a new instantaneous graph G_{T+1} to a longer sequence $G_M = G_1G_2 \dots G_T G_{T+1}$, where G_{T+1} captures a link change (either a link comes up or goes down). But such an approach has very poor scalability. In this research work, we sample the network topology periodically for every 0.25 seconds, which could, in reality, be the instants of data packet origination at the source.

4.2 Kou et al.'s Heuristic

We use the Kou et al.'s well-known $O(|V||SR|^2)$ heuristic ($|V|$ is the number of nodes in the network graph and $|SR|$ is the size of the multicast group comprising of the source nodes and the receiver nodes) to approximate the

minimum edge Steiner tree in graphs representing snapshots of the network topology. An $(S-R)$ -Steiner-tree is defined as the multicast Steiner tree connecting the set of source nodes, S , to the set of receiver nodes, R . We give a brief outline of the heuristic in Figure 1.

4.3 OptMeshTrans Algorithm

We now describe the *OptMeshTrans* algorithm proposed to determine the sequence of multicast meshes connecting a set of sources (S) to a set of receivers (R), such that the meshes exist for the longest possible time and the number of mesh transitions is minimal. The pseudo code is given in Figure 2. Algorithm *OptMeshTrans* operates according to the following greedy strategy: Whenever a multicast mesh connecting all the source nodes (S) to all the receiver nodes (R) of a multicast group is required, the multicast mesh, called the Stable-Static-Mesh, represented as $(S-R)_{\text{Stable-Static-Mesh}}$, which exists for the longest time is selected. An illustration of a Steiner tree and the corresponding multicast mesh on a sample network topology is shown in Figure 11. The *OptMeshTrans* algorithm can run for any scenario (such as network density, number of sources and etc) and yield the optimal result, i.e., the sequence of longest living meshes, for that scenario.

A mobile graph $G_M = G_1 G_2 \dots G_T$ is generated by sampling the network topology at regular time intervals $t_1, t_2 \dots t_T$. At time instant t_i , when a multicast mesh is required, a mobile sub graph $G(i, j) = G_i \cap G_{i+1} \cap \dots \cap G_j$ is constructed such that there exists at least one mesh connecting every source $s \in S$ to every receiver $r \in R$ in $G(i, j)$ and no mesh exists in $G(i, j+1)$. A minimum edge $(S-R)$ -Steiner-tree connecting every source $s \in S$ to every receiver $r \in R$ is constructed based on the Kou's heuristic and the Steiner tree is extended to a mesh by including all the edges (represented by the set *Additional-Edges* in the pseudo code, Figure 2) that exist between the constituent nodes of the tree in the mobile sub graph $G(i, j)$. The above procedure is repeated until time instant $j+1 \leq T$, where T is the duration of the multicast session. The Stable-Mobile-Mesh, represented as $(S-R)_{\text{Stable-Mobile-Mesh}}$, is a sequence of such maximum lifetime Stable-Static-Meshes and will undergo the minimum number of mesh transitions.

4.4 Time Complexity of OptMeshTrans Algorithm

If T is the duration of the multicast session and k is the sampling rate (k samples of static graphs collected per unit time) used to form the mobile graph, the Kou et al.'s heuristic has to be run $T*k$ times, each time on a graph of $|V|$ nodes. During each such iteration, we will also have to form the set of *Additional-Edges* to extend the minimum edge Steiner tree to a multicast mesh. At the worst case, there would be $O(|V|)$ vertices in the minimum edge Steiner tree and it would take $O(|V|^2)$ time to determine whether an edge between every pair of vertices in the Steiner tree exists in the mobile sub graph for inclusion in the set of *Additional-Edges*. Hence, the run-time complexity of *OptMeshTrans* would be $O((|V||SR|^2 + |V|^2) T*k)$, where SR is the union of the set of sources and receivers of the multicast group.

4.5 Proof of Correctness of OptMeshTrans Algorithm

Given a mobile graph $G_M = G_1 G_2 \dots G_T$, set of sources S and the set of receivers R , let the number of mesh transitions generated by *OptMeshTrans* in the Stable-Mobile-Mesh, $(S-R)_{\text{Stable-Mobile-Mesh}}$ be m . We use the proof by contradiction technique to prove the correctness of the *OptMeshTrans* algorithm. To show that m is optimal, we assume the contrary as the hypothesis for our proof, i.e., there exists another Stable-Mobile-Mesh $(S-R)'_{\text{Stable-Mobile-Mesh}}$ with m' number of mesh transitions, such that $m' < m$.

Let $\text{epoch}^1_{(S-R)}, \text{epoch}^2_{(S-R)}, \dots, \text{epoch}^m_{(S-R)}$ and $\text{epoch}^1_{(S-R)'}, \text{epoch}^2_{(S-R)'}, \dots, \text{epoch}^{m'}_{(S-R)'}$ be the set of sampling time instants in $(S-R)_{\text{Stable-Mobile-Mesh}}$ and $(S-R)'_{\text{Stable-Mobile-Mesh}}$ respectively, wherein no mesh transitions exist. Let $t^{init,j}_{(S-R)}, t^{init,k}_{(S-R)'}$ be the initial and $t^{end,j}_{(S-R)}, t^{end,k}_{(S-R)'}$ be the final sampling time instants of $\text{epoch}^j_{(S-R)}$ where $1 \leq j \leq m$ and $\text{epoch}^k_{(S-R)'}$ where $1 \leq k \leq m'$ respectively. Since $(S-R)_{\text{Stable-Mobile-Mesh}}$ and $(S-R)'_{\text{Stable-Mobile-Mesh}}$ exist over the same time period T , the initial and final sampling time instants are same (i.e., $t^{init,1}_{(S-R)} = t^{init,1}_{(S-R)'}$ and $t^{end,m}_{(S-R)} = t^{end,m'}_{(S-R)'}$). As our hypothesis is that the number of transitions in $(S-R)'_{\text{Stable-Mobile-Mesh}}$ is less than that of $(S-R)_{\text{Stable-Mobile-Mesh}}$, there should exist a mesh in $(S-R)'_{\text{Stable-Mobile-Mesh}}$ that has longer lifetime than that in $(S-R)_{\text{Stable-Mobile-Mesh}}$, $m' < m \Rightarrow \exists j, k$ where $1 \leq j \leq m$ and $1 \leq k \leq m'$ such that $\text{epoch}^j_{(S-R)} \subset \text{epoch}^k_{(S-R)'}$, i.e., $t^{init,k}_{(S-R)'} < t^{init,j}_{(S-R)} < t^{end,j}_{(S-R)} < t^{end,k}_{(S-R)'}$. In other words, there should exist a $(S-R)'_{\text{Stable-Static-Mesh}}$ in $[t^{init,k}_{(S-R)'}, \dots, t^{end,k}_{(S-R)'}]$. But, in *OptMeshTrans* algorithm, a transition was made at $t^{end,j}_{(S-R)}$ as the mesh that started to exist at $t^{init,j}_{(S-R)}$ does not exist beyond $t^{end,j}_{(S-R)}$. So, $t^{end,k}_{(S-R)'}$ should be less than or equal to $t^{end,j}_{(S-R)}$ and cannot be greater. There is no common $(S-R)'_{\text{Stable-Static-Mesh}}$ in $[t^{init,j}_{(S-R)'}, \dots, t^{end,k}_{(S-R)'}]$ and hence there is no common $(S-R)'_{\text{Stable-Static-Mesh}}$ in $[t^{init,k}_{(S-R)'}, \dots, t^{end,k}_{(S-R)'}]$. Therefore, the lifetime of all the meshes in $(S-R)'_{\text{Stable-Mobile-Mesh}}$ has to be less than or equal to that of $(S-R)_{\text{Stable-Mobile-Mesh}}$ i.e., $m' \geq m$. This is in contradiction to the hypothesis. Hence, m , the number of transitions resulting from *OptMeshTrans* algorithm is optimal (minimum) and $(S-R)_{\text{Stable-Mobile-Mesh}}$ is the Stable-Mobile-Mesh connecting the set of sources S to the set of receivers R .

5. Simulations

5.1 Simulation Environment

We implemented ODMRP_DSR, ODMRP_FORP and *OptMeshTrans* in a discrete event simulator developed by us in Java. This simulator has also been successfully used in some of our recent studies (e.g., Meghanathan, Thomas & Addison, 2009; Meghanathan, 2009) on MANET multicast routing. The network dimensions are 1000m x 1000m. The transmission range of each node is 250m. We vary the density of the network by conducting simulations with 50 nodes (low density) and 100 nodes (high density). The Medium Access Control (MAC) layer model used is the IEEE 802.11 model (Bianchi, 2000) and link bandwidth is 2 Mbps. The simulation time is 1000 seconds.

5.1.1 Energy Consumption Model

In ad hoc wireless networks, energy consumption at a node can be divided into three components: (i) Energy utilized for transmitting a message, (ii) Energy utilized for receiving a message and (iii) Energy utilized in idle state. It has been shown by Kim, Garcia-Luna-Aceves, Obraczka, Cano & Manzoni (2003) that in the presence of overhearing, no real optimization in the energy consumption or node lifetime can be achieved. Therefore, in this paper, we do not focus on the energy lost in the idle state and focus only on the energy consumed for transmitting and receiving a message (data packets and MAC layer RTS-CTS-ACK packets) and energy consumed due to transmission and reception of broadcast control messages for mesh formation. The overhearing model considered is the reduced overhearing model (Meghanathan, 2007) according to which neighbors of a node listen to the channel only to receive the header of the data packets and then turn themselves off if the packet is not addressed to them. Nodes are however charged for receiving the packets belonging to the Medium Access Control (MAC) layer. The transmission power is 1.4W and the reception power is 1W. These are the standard values used in literature (e.g., Feeney, 2001; Park & Sivakumar, 2002) for transmissions without any power control. Each node is provided an initial energy of 500 Joules, large enough to avoid any node failure due to loss of energy in transmission and reception. Also, since the routing protocols studied are not power-aware, there would not be any effective influence of the available battery charge at a node on the choice of the routes being selected.

5.1.2 Mobility Model

The mobility model used is the Random Waypoint model (Bettstetter, Hartenstein & Perez-Costa, 2004), wherein the velocity of a node is uniform-randomly selected from $[0, \dots, v_{max}]$ every time the node incurs a direction change to travel to a randomly selected location within the network. The v_{max} values used are 5 m/s and 50 m/s characteristic of low and high mobility respectively. The pause time is 0 seconds. For each v_{max} value, we generated five mobility profiles of the nodes for simulation time of 1000 seconds.

5.1.3 Traffic Model

The values for the number of sources used are: 2, 4 and 8; the number of receivers used is: 3, 6 and 9. The data packet size is 512 bytes and the packet sending rate from each source to the set of receivers is 4 packets per second. For each value of the number of sources and receivers, we created one list of source nodes and five lists of receiver nodes. All the node lists are generated randomly; but, we made sure that a node acts at most only as a source or a receiver; not both. Simulations for a given number of sources were run for each of these five lists using the five mobility profiles generated for each v_{max} value. Each data point obtained for the protocols/algorithms in the performance figures 3 through 10 is the average value obtained from these 25 experiments for a given value of the number of sources and v_{max} .

5.2 Performance Metrics

The following performance metrics are measured for each of ODMRP_DSR, ODMRP_FORP protocols and the *OptMeshTrans* algorithm under the different simulation conditions described above.

- (i) Lifetime per Mesh – average of the lifetimes of the sequence of multicast meshes discovered over the duration of the entire multicast session.
- (ii) Edges per Mesh – time-averaged value of the number of edges per mesh connecting the set of sources to the set of receivers, computed over the entire multicast session.
- (iii) Hop Count per Source-Receiver Path – time-averaged hop count of the paths from the source to each receiver, considering all the source-receiver pairs, computed over the entire multicast session.
- (iv) Energy Consumed per Node – average of the energy consumed across all the nodes in the network. The energy consumed at a node is the sum of the energy lost due to the transmission and reception of data

packets, transmission and reception of the broadcast control messages for mesh formation and the transmission and reception of the control messages (RTS, CTS and ACK) at the MAC layer.

5.3 Average Lifetime per Mesh

It is imperative to form multicast meshes with larger lifetime because each time a new mesh is to be formed, a global network-wide broadcast of the control messages from each source node is initiated. The larger the value for the mesh lifetime, the lower will be the number of times such resource-consuming global broadcast of control messages will happen in the network. One can see a clear ranking for mesh lifetime in Figures 3 and 4. The meshes formed using the *OptMeshTrans* algorithm are relatively more stable (have larger lifetime) than compared to the meshes formed using ODMRP_DSR and ODMRP_FORP. Between the two ODMRP based protocols, for a majority of the scenarios, the meshes formed using the link lifetime prediction approach of ODMRP_FORP are more stable than the meshes formed using the minimum hop approach of ODMRP_DSR.

On average, the meshes formed using ODMRP_FORP have 20%-25% longer lifetime than the meshes formed using ODMRP_DSR. In the worst case, the meshes formed using ODMRP_FORP can have a lifetime as large as 70% and 90% more than that of the meshes formed using ODMRP_DSR in low node density and high node density networks respectively. On the other hand, on average, the meshes formed using algorithm *OptMeshTrans* have 400%-450% longer lifetime than the meshes formed using ODMRP_FORP. In the worst case, the meshes formed using *OptMeshTrans* can have a lifetime as large as 830% and 1450% more than that of the meshes using ODMRP_FORP. These observations indicate that there is a still lot of scope to improve the stability of the multicast meshes.

For a given node mobility and network density, we observe that the lifetime of meshes formed using ODMRP_DSR and ODMRP_FORP increases with increase in the number of sources. This can be attributed to the increase in the number of edges to ensure connectivity in the mesh with increase in the number of sources. With an increased number of edges, there is an increase in the number of alternate paths between a source-receiver pair in a mesh, resulting in larger lifetime between two successive mesh transitions. On the other hand, the lifetime of meshes formed using *OptMeshTrans* does not relatively change much with increase in the number of sources. This can be attributed to the working principle of the algorithm to look into the future topology changes and consider a mobile sub graph that consists of the minimum number of edges that will exist for a longer time as well as constitute a mesh.

For fixed node mobility and number of sources, with increase in the node density, the average lifetime per mesh discovered using ODMRP_DSR and ODMRP_FORP decreases. This is due to the decrease in the hop count per source-receiver path with increase in node density, leading to an increase in the probability of a path break in the near future. But, with algorithm *OptMeshTrans*, as node density increases, the mesh comprises of relatively better stable paths in which the physical distance between the end nodes of the constituent links is close to only 50-60% of the transmission range of the nodes. For fixed node density and number of sources, with increase in node mobility, the neighbors of each node move very fast, leading to a larger probability of link break in the near future. Hence, the lifetime per mesh for all the three implementations studied would naturally be lower with increase in node mobility. Similarly, for a fixed node density, mobility and number of sources, the average lifetime per mesh is more likely to decrease with increase in the number of receivers as it becomes difficult to maintain the connectivity of a mesh involving more receiver nodes, but a fixed number of source nodes.

5.4 Average Number of Edges per Mesh

The meshes formed using the *OptMeshTrans* algorithm have relatively few edges compared to those discovered using ODMRP_DSR and ODMRP_FORP. This can be again attributed to the decrease in the number of edges in the mobile sub graph which is an intersection of the static graphs of the network in the future. The ODMRP_DSR and ODMRP_FORP are respectively focused on discovering minimum hop path and stable path between a source-receiver pair available at the current instant and the congregation of such locally optimal paths forms the mesh. There is no inclination to reduce the number of edges in the mesh when the individual paths are discovered in these two ODMRP-based protocols. On the other hand, *OptMeshTrans* looks at the future and focuses on discovering a mesh that will exist for a longer time with a reduced number of constituent links (which is also the objective of the underlying Kou et al.'s minimum Steiner heuristic).

As node mobility increases, the number of edges per mesh decreases for each of the three protocols. For fixed node mobility, the number of edges increases for each of the three protocols with increase in node density. For different node mobility and density scenarios, the number of edges per mesh for each of the three protocols increases with increase in the number of sources. We can also observe an increase in the number of edges with increase in the number of receivers with different node density and node mobility values. As the number of

sources and receivers increases, the number of edges per mesh increases to maintain connectivity in the mesh. However, for all the above scenarios, the rate of increase in the number of edges per mesh discovered using the *OptMeshTrans* algorithm is lower than that observed with ODMRP_DSR and ODMRP_FORP.

On average, the meshes formed using ODMRP_FORP have 5%-7% more edges than that of the meshes formed using ODMRP_DSR. In the worst case, the meshes formed using ODMRP_FORP can have as large as 15% and 30% more edges than that of the meshes formed using ODMRP_DSR in low node density and high node density networks respectively. But, on average, the meshes formed using algorithm *OptMeshTrans* have 13%-20% fewer edges than the meshes formed using the ODMRP-based protocols. In the worst case, the number of edges in the meshes formed using *OptMeshTrans* can be 45%-48% less than the number of edges in the meshes formed using the ODMRP-based protocols.

5.5 Average Hop Count per Source-Receiver Path

The average hop count per source-receiver path is a measure of the end-to-end delay per data packet sent from a source node to a receiver node. The source-receiver paths that are part of the meshes discovered using algorithm *OptMeshTrans* have a larger hop count compared to those discovered using ODMRP_DSR and ODMRP_FORP. This can be attributed to the relatively fewer number of edges in the meshes discovered using algorithm *OptMeshTrans*. With fewer edges, some of the paths between a particular source node and receiver node in the mesh could be relatively longer (i.e. more hops). Algorithms ODMRP_DSR and ODMRP_FORP look at the current network topology and determine minimum hop paths and stable paths between individual source and receiver nodes.

For fixed node mobility, the average hop count per source-receiver path in the meshes discovered using ODMRP_DSR and ODMRP_FORP increases with increase in the number of sources and receivers, but the increase is below 25%. For all the simulation scenarios, the average hop count per source-receiver path in meshes discovered using ODMRP_DSR and ODMRP_FORP decreases with increase in node density, but the decrease is below 20%. For different node mobility and node density, the average hop count per source-receiver path in the meshes discovered using *OptMeshTrans* is not affected much by the number of sources and receivers.

For a given simulation scenario, even though DSR is a minimum-hop based unicast routing protocol, there is no appreciable difference in the average hop count per source-receiver path between ODMRP_DSR and ODMRP_FORP. This illustrates when the ODMRP_FORP mesh is formed as a congregate of all the individual stable paths between each source and all the receivers, the average hop count per source-receiver path for the whole mesh is not much different than that of the mesh formed using ODMRP_DSR. On average, the difference is within $\pm 5\%$. In the worst case, the hop count per source-receiver path in the meshes formed using ODMRP_FORP can be as large as 15%-23% more than that of the hop count per source-receiver path in the meshes formed using ODMRP_DSR and the hop count per source-receiver path in the meshes formed using ODMRP_DSR can be as large as 18%-22% more than that of the hop count per source-receiver path in the meshes formed using ODMRP_FORP.

The difference in the hop count per source-receiver path in the meshes formed using *OptMeshTrans* and the meshes formed using the two ODMRP-based protocols increases with increase in node density. For a given node mobility, the hop count per source-receiver path in the meshes formed using *OptMeshTrans* can be as large as 87% and 130% more than that of the hop count per source-receiver path in the meshes formed using the ODMRP-based protocols. On average, considering all the scenarios, the hop count per source-receiver path in the meshes formed using *OptMeshTrans* can be about 65%-78% more than that of the hop count per source-receiver path formed using the two ODMRP-based protocols.

5.6 Average Energy Consumption per Node

Algorithm *OptMeshTrans* consumes relatively less energy in the network compared to the two ODMRP-based protocols. This can be attributed to the relatively less number of edges per mesh and longer lifetime per mesh. The meshes formed in the two ODMRP-based protocols get frequently disconnected, leading to an increase in the number of network-wide global broadcasts. In addition, the ODMRP-based meshes are formed without any consideration for energy-efficiency. Since fewer links are involved in the transmission and reception of messages in the meshes formed using algorithm *OptMeshTrans*, the energy consumed per node in the Stable-Mobile-Meshes is relatively low.

For fixed node mobility, the energy consumption per node for each of the three protocols increases with increase in node density. This can be attributed to the increase in the number of edges per mesh with increase in node density. For fixed node mobility and node density, the energy consumption per node increases with increase in

the number of sources as well as with increase in the number of receivers. This can be again attributed to the increase in the number of edges (and hence more transmissions and receptions) to maintain mesh connectivity with increase in the number of sources or receivers.

For a given simulation scenario, there is no appreciable difference in the energy consumption per node between ODMRP_DSR and ODMRP_FORP. On average, the difference is within $\pm 5\%$. In the worst case, the energy consumed per node using ODMRP_FORP can be as large as 16% more than that of ODMRP_DSR and the energy consumed per node using ODMRP_DSR can be as large as 27% more than that of the energy consumed per node using ODMRP_FORP. On the other hand, the difference in the energy consumption per node between *OptMeshTrans* and the two ODMRP-based protocols increases with increase in node density. For a given node mobility, the energy consumed per node for the two ODMRP-based protocols can be as large as 46% and 80% more than the energy consumed per node due to *OptMeshTrans* in low density and high density network respectively. On average, considering all the scenarios, the energy consumed per node for the two ODMRP-based protocols can be about 35%-40% more than that of the energy consumed per node for *OptMeshTrans*.

6. Conclusions

Our high-level contributions in this paper are the investigation about the impact of route selection metrics on the performance of the mesh-based ODMRP protocol and a comparison of ODMRP performance with that of a theoretically optimal algorithm *OptMeshTrans* that forms a sequence of stable meshes. We observe an impact of the route selection metrics on the performance of ODMRP with respect to mesh lifetime and to a certain extent on the number of edges per mesh. We do not observe any significant impact of the route selection metrics on the energy consumption per node and hop count per source-receiver path for the two ODMRP-based protocols. While comparing the two ODMRP-based implementations with algorithm *OptMeshTrans*, we observe the *OptMeshTrans* algorithm to yield relatively long-living stable meshes with lower energy consumption per node and less number of edges per mesh. However, on average, the hop count per source-receiver path in the Stable-Mobile-Meshes is significantly larger (65%-78%) compared to the two ODMRP-based protocols. This indicates the tradeoff between ODMRP and *OptMeshTrans* and future research will involve developing a distributed version of *OptMeshTrans* that can minimize this {mesh lifetime, edge count, energy consumption} vs. {hop count} tradeoff and further enhance mesh-based multicast routing. Future work will also involve studying the performance of *OptMeshTrans* in more dynamic and/or less reliable mesh networks.

References

- Begdillo, S. J., Asadi, M., and Haghigat, A. T. (2007). Improving Packet Delivery Ratio in ODMRP with Route Diversity. *International Journal of Computer Science and Network Security*, 7, 12, 146 – 151.
- Bettstetter, C., Hartenstein, H., and Perez-Costa, X. (2004). Stochastic Properties of the Random Waypoint Mobility Model. *Wireless Networks*, 10, 5, 555 – 567.
- Bianchi, G. (2000). Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal of Selected Areas in Communications*, 18, 3, 535 – 547.
- Chen, C.-K., Wang, K., and Lee, L.-S. (2005). *A Hybrid Overlay Multicast Routing Protocol for Mobile Ad hoc Networks*. Paper presented at the International Conference on Wireless Networks, Communications and Mobile Computing, USA.
- Farago, A., and Syrotiuk, V. R. (2003). MERIT: A Scalable Approach for Protocol Assessment. *Mobile Networks and Applications*, 8, 5, 567 – 577.
- Feeney, L. M. (2001). An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad hoc Networks. *Journal of Mobile Networks and Applications*, 3, 6, 239 – 249.
- Garcia-Luna-Aceves, J. J., and Madruga, E. L. (1999). *A Multicast Routing Protocol for Ad hoc Networks*. Paper presented at 18th International Conference on Computer and Communications, USA.
- Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (2004). *Global Positioning System: Theory and Practice*. (5th rev. ed.) Springer.
- Johnson, D. B., Maltz, D. A., and Broch, J. (2001). *Ad hoc Networking*, Addison-Wesley.
- Kim, D., Garcia-Luna-Aceves, J. J., Obraczka, K., Cano, J.-C., and Manzoni, P. (2003). Routing Mechanisms for

- Mobile Ad hoc Networks based on the Energy Drain Rate. *IEEE Transactions on Mobile Computing*, 2, 2, 161 – 173.
- Kou, L., Markowsky, G., and Berman, L. (1981). A Fast Algorithm for Steiner Trees. *Acta Informatica*, 15, 141 – 145.
- Lee, S.-J., Gerla, M., and Chaing, C.-C. (1999). *On-Demand Multicast Routing Protocol*. Paper presented at the IEEE Wireless Communications and Networking Conference, USA.
- Lee, S.-J., Su, W., Hsu, J., Gerla, M., and Bagrodia, R. (2000). *A Performance Comparison Study of Ad hoc Wireless Multicast Protocols*. Paper presented at the 19th IEEE International Conference on Computer and Communications, Israel.
- Mazinan, E., Arabshahi, Z., and Adim, J. (2008). *Comparing AMRIS and ODMRP in Ad hoc Networks by Qualnet*. Paper presented at the 7th International Conference on Networking, Mexico.
- Meghanathan, N. (2009). Survey and Taxonomy of Unicast Routing Protocols for Mobile Ad hoc Networks. *The International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks*, 1, 1, 1 – 121.
- Meghanathan, N. (2008). Exploring the Stability-Energy Consumption-Delay-Network Lifetime Tradeoff of Mobile Ad hoc Network Routing Protocols. *Journal of Networks*, 3, 2, 17 – 28.
- Meghanathan, N., Thomas, D., and Addison, E. S. (2009). *Multicast Extensions to the Flow-Oriented Routing Protocol and Node Velocity-based Stable Path Routing Protocol for Mobile Ad hoc Networks*. Paper presented at the 1st IEEE International Workshop on Mobile Computing and Networking Technologies, Russia.
- Meghanathan, N. (2009). Multicast Extensions to the Location Prediction Based Routing Protocol for Mobile Ad hoc Networks. *Lecture Notes of Computer Science*, 5682, 190 – 199.
- Meghanathan, N. (2007). Energy Consumption Analysis of the Stable Path and Minimum Hop Path Routing Strategies for Mobile Ad hoc Networks. *International Journal of Computer Science and Network Security*, 7, 10, 30 – 39.
- Murthy, C. S. R., and Manoj, B. S. (2004). *Ad hoc Wireless Networks: Architectures and Protocols*, Prentice Hall.
- Park, S.-J., and Sivakumar, R. (2002). *Load-Sensitive Transmission Power Control in Wireless Ad hoc Networks*. Paper presented at the IEEE Global Communications Conference, Taiwan.
- Royer, E., and Perkins, C. E. (1999). *Load-Sensitive Transmission Power Control in Wireless Ad hoc Networks*. Paper presented at the IEEE Global Communications Conference, Taiwan.
- Su, W. Lee, S.-J., and Gerla, M. (2001). Mobility Prediction and Routing in Ad hoc Wireless Ad hoc Networks. *International Journal of Network Management*, 11, 1, 3 – 30.
- Vasiliou, A., and Economides, A. A. (2005). Evaluation of Multicasting Algorithms in MANETs. *World Academy of Science, Engineering and Technology*, 5, 94 – 97.
- Wu, C. W., Tay, Y. C., and Toh, C. K. (1998). Ad hoc Multicast Routing Protocol Utilizing Increasing ID-Numbers (AMRIS) Functional Specification. *IETF Internet Draft*.
- Xie, J., Talpade, R. R., McAuley, A., and Liu, M. (2002). AMRoute: Ad hoc Multicast Routing Protocol. *Mobile Networks and Applications*, 7, 6, 429 – 439.
- Zhang, X., and Jacob, L. (2003). *Multicast Zone Routing Protocol in Mobile Ad hoc Wireless Networks*. Paper presented at the 28th IEEE International Conference on Local Computer Networks, Germany.

Input: An undirected graph $G = (V, E)$
 Multicast group $SR \subseteq V$

Output: An $(S-R)$ -Steiner-tree for the set SR in G

Step 1: Construct a complete undirected weighted graph $G_C = (SR, E_C)$ from G and SR where $\forall (v_i, v_j) \in E_C, v_i$ and v_j are in SR , and the weight of edge (v_i, v_j) is the length of the shortest path from v_i to v_j in G .

Step 2: Find the minimum weight spanning tree T_C in G_C (If more than one minimal spanning tree exists, pick an arbitrary one).

Step 3: Construct the sub graph G_{SR} of G , by replacing each edge in T_C with the corresponding shortest path from G (If there is more than one shortest path between two given vertices, pick an arbitrary one).

Step 4: Find the minimal spanning tree T_{SR} in G_{SR} (If more than one minimal spanning tree exists, pick an arbitrary one). Note that each edge in G_{SR} has weight 1.

Step 5: Construct the $(S-R)$ -Steiner-tree, from T_{SR} by deleting the edges in T_{SR} , if necessary, such that all the leaves in the $(S-R)$ -Steiner-tree are members of SR .

Figure 1. Heuristic to find an Approximate Minimum Steiner Tree

Input: $G_M = G_1 G_2 \dots G_T$, Set of source nodes - S , Set of receiver nodes - R

Output: $(S-R)_{\text{Stable-Mobile-Mesh}}$ // Stable-Mobile-Mesh

Auxiliary Variables: $i, j, \text{Additional-Edges}$

Initialization: $i=1; j=1; (S-R)_{\text{Stable-Mobile-Mesh}} = \emptyset, \text{Additional-Edges} = \emptyset$

Begin $OptMeshTrans$

1. **while** ($i \leq T$) **do**
2. Find a mobile sub graph $G(i, j) = G_i \cap G_{i+1} \cap \dots \cap G_j$ such that there exists at least one $(S-R)$ -Steiner-tree connecting every source $s \in S$ to every receiver $r \in R$ in $G(i, j)$ and {no such $(S-R)$ -Steiner-tree exists in $G(i, j+1)$ or $j = T$ }
3. **if** \exists a $(S-R)$ -Steiner-tree in $G(i, j)$ **then**
4. **for** (every vertex u and v in $(S-R)$ -Steiner-tree)
5. **if** (edge $(u, v) \in G(i, j)$ and edge $(u, v) \notin (S-R)$ -Steiner-tree) **then**
6. Additional-Edges = Additional-Edges U $\{(u, v)\}$
7. **end if**
8. **end for**
9. $(S-R)_{\text{Stable-Static-Mesh}}$ in $G(i, j) = \{(S-R)$ -Steiner-tree in $G(i, j)\} \cup$ Additional-Edges
10. $i = j + 1$
11. Additional-Edges = \emptyset
12. **end if**
13. **end while**
14. **return** $(S-R)_{\text{Stable-Mobile-Mesh}}$

End $OptMeshTrans$

Figure 2. Pseudo code for $OptMeshTrans$ algorithm

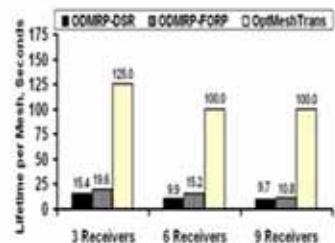
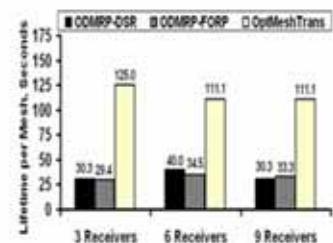
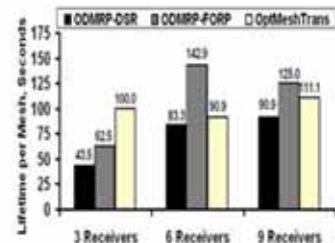
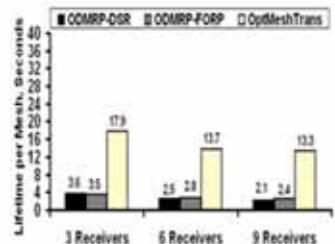
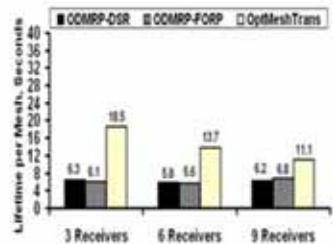
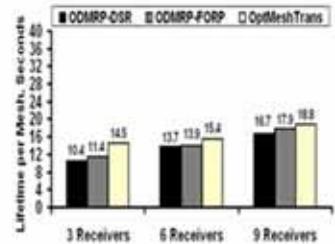
**Figure 3.1.** 5 m/s, 2 sources**Figure 3.2.** 5 m/s, 4 sources**Figure 3.3.** 5 m/s, 8 sources**Figure 3.4.** 50 m/s, 2 sources**Figure 3.5.** 50 m/s, 4 sources**Figure 3.6.** 50 m/s, 8 sources

Figure 3. Average Lifetime per Mesh (50 Node Network)

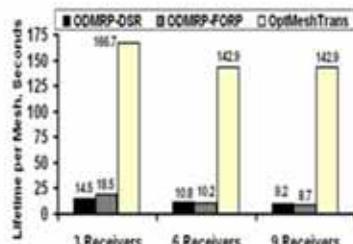
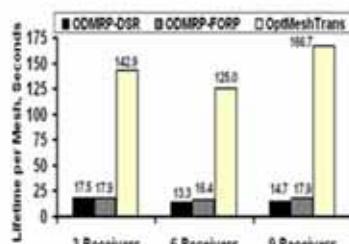
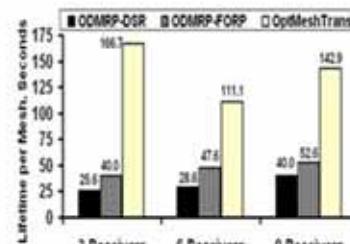
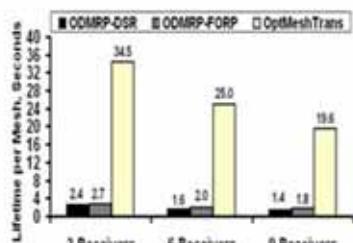
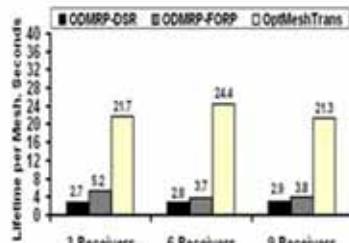
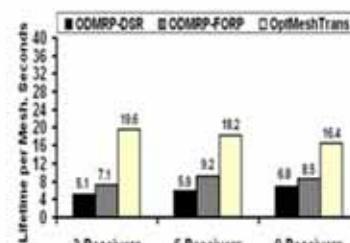
**Figure 4.1.** 5 m/s, 2 sources**Figure 4.2.** 5 m/s, 4 sources**Figure 4.3.** 5 m/s, 8 sources**Figure 4.4.** 50 m/s, 2 sources**Figure 4.5.** 50 m/s, 4 sources**Figure 4.6.** 50 m/s, 8 sources

Figure 4. Average Lifetime per Mesh (100 Node Network)

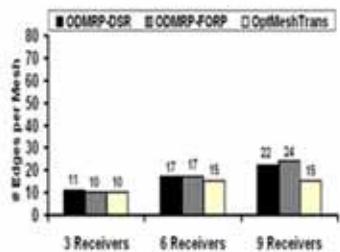
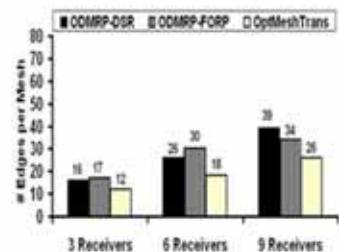
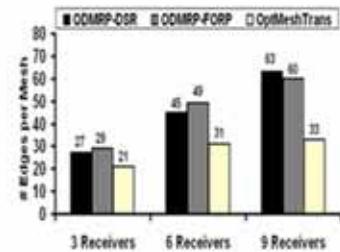
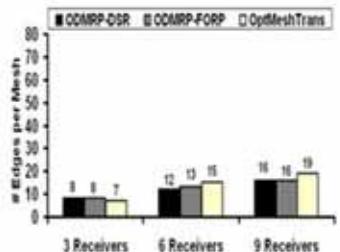
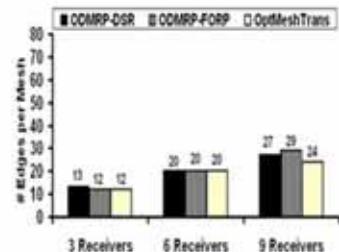
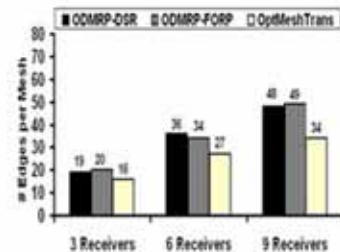
**Figure 5.1.** 5 m/s, 2 sources**Figure 5.2.** 5 m/s, 4 sources**Figure 5.3.** 5 m/s, 8 sources**Figure 5.4.** 50 m/s, 2 sources**Figure 5.5.** 50 m/s, 4 sources**Figure 5.6.** 50 m/s, 8 sources

Figure 5. Average Number of Edges per Mesh (50 Node Network)

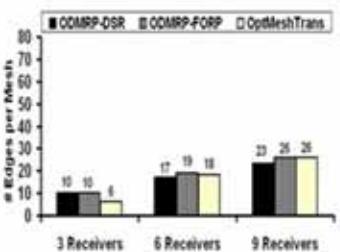
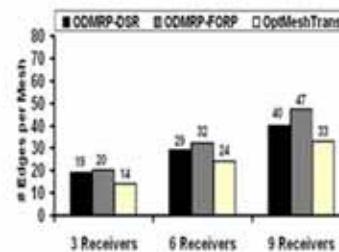
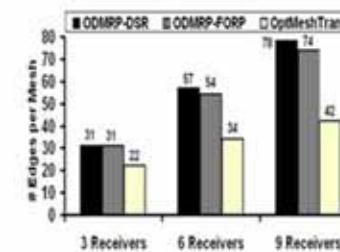
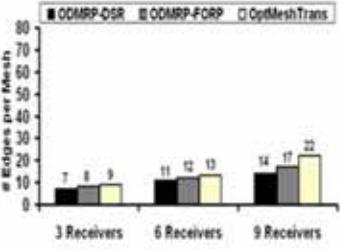
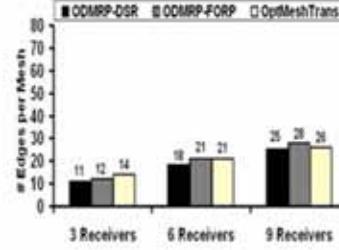
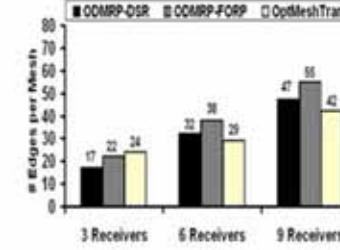
**Figure 6.1.** 5 m/s, 2 sources**Figure 6.2.** 5 m/s, 4 sources**Figure 6.3.** 5 m/s, 8 sources**Figure 6.4.** 50 m/s, 2 sources**Figure 6.5.** 50 m/s, 4 sources**Figure 6.6.** 50 m/s, 8 sources

Figure 6. Average Number of Edges per Mesh (100 Node Network)

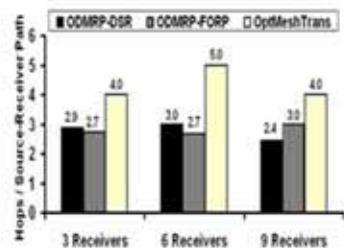
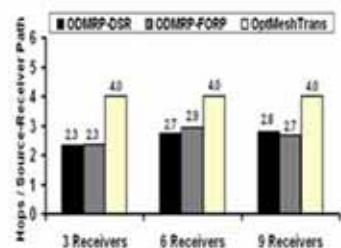
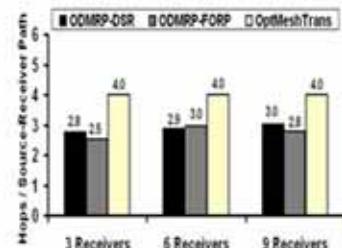
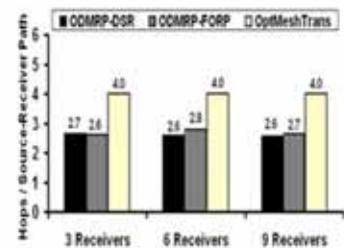
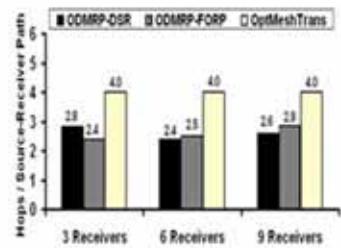
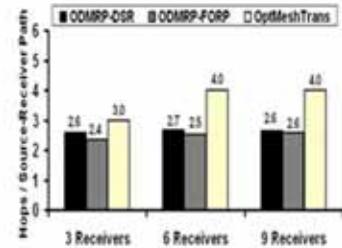
**Figure 7.1.** 5 m/s, 2 sources**Figure 7.2.** 5 m/s, 4 sources**Figure 7.3.** 5 m/s, 8 sources**Figure 7.4.** 50 m/s, 2 sources**Figure 7.5.** 50 m/s, 4 sources**Figure 7.6.** 50 m/s, 8 sources

Figure 7. Average Hop Count per Source – Receiver Path (50 Node Network)

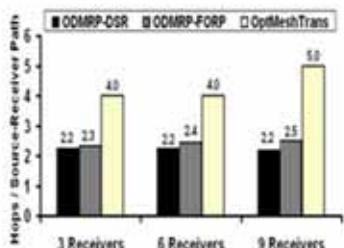
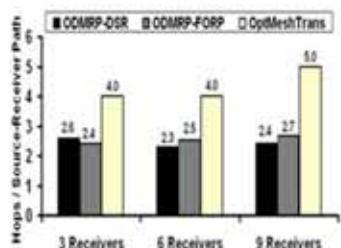
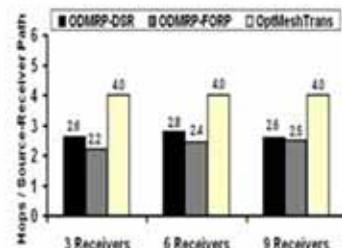
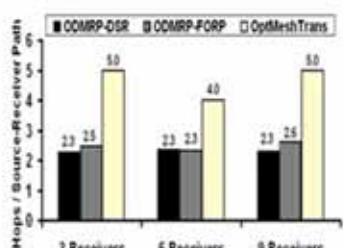
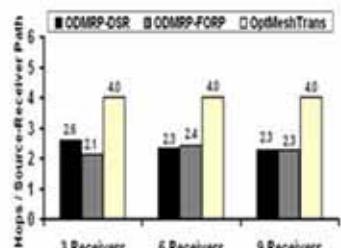
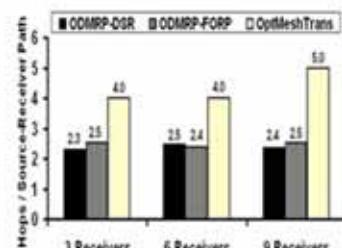
**Figure 8.1.** 5 m/s, 2 sources**Figure 8.2.** 5 m/s, 4 sources**Figure 8.3.** 5 m/s, 8 sources**Figure 8.4.** 50 m/s, 2 sources**Figure 8.5.** 50 m/s, 4 sources**Figure 8.6.** 50 m/s, 8 sources

Figure 8. Average Hop Count per Source – Receiver Path (100 Node Network)

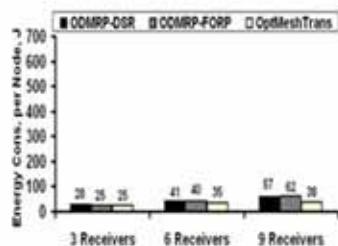
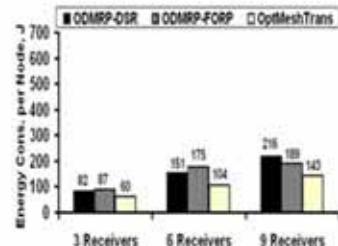
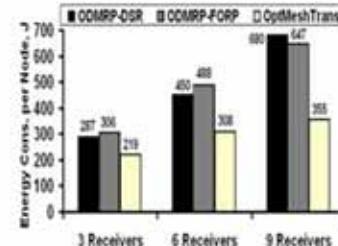
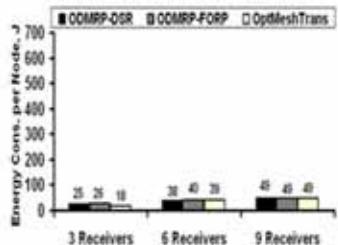
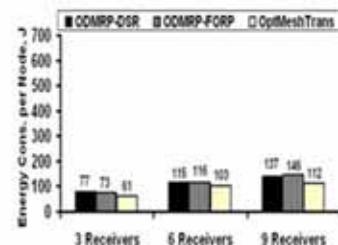
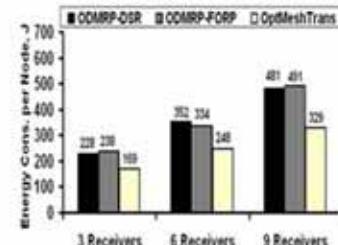
**Figure 9.1.** 5 m/s, 2 sources**Figure 9.2.** 5 m/s, 4 sources**Figure 9.3.** 5 m/s, 8 sources**Figure 9.4.** 50 m/s, 2 sources**Figure 9.5.** 50 m/s, 4 sources**Figure 9.6.** 50 m/s, 8 sources

Figure 9. Average Energy Consumed per Node (50 Node Network)

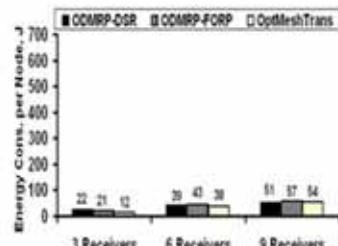
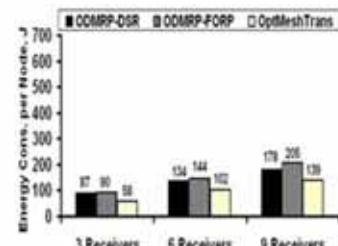
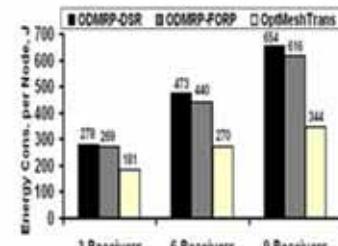
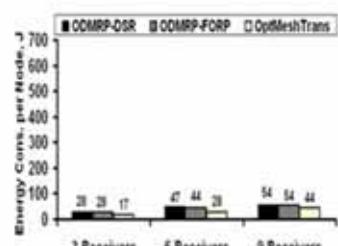
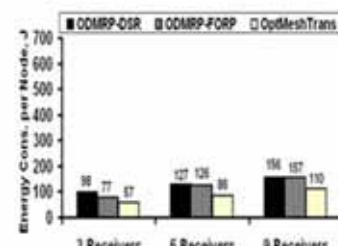
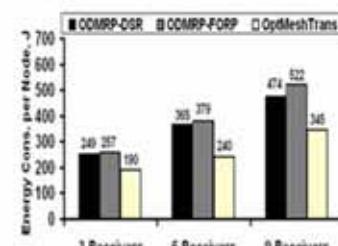
**Figure 10.1.** 5 m/s, 2 sources**Figure 10.2.** 5 m/s, 4 sources**Figure 10.3.** 5 m/s, 8 sources**Figure 10.4.** 50 m/s, 2 sources**Figure 10.5.** 50 m/s, 4 sources**Figure 10.6.** 50 m/s, 8 sources

Figure 10. Average Energy Consumed per Node (100 Node Network)

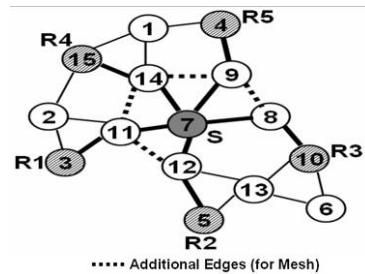
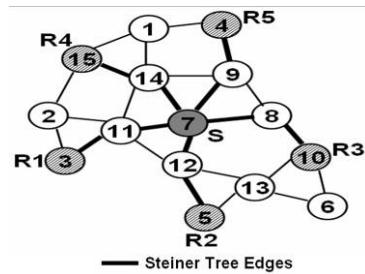
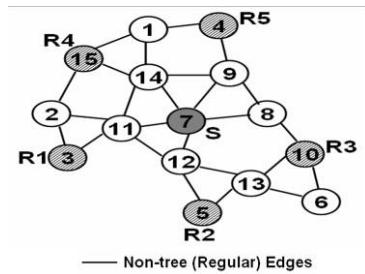


Figure 11.1. Sample Topology

Figure 11.2. Steiner Tree

Figure 11.3. Multicast Mesh

Figure 11. Steiner Tree and the Corresponding Multicast Mesh on a Sample Network Topology