A Simulation-based Performance Analysis of Multicast Routing in Mobile Ad hoc Networks

Natarajan Meghanathan
Jackson State University, Jackson, MS, USA
E-mail: natarajan.meghanathan@jsuns.edu
doi: 10.4156/ijipm.vol1.issue1.1

Abstract

In an earlier work, we had proposed an algorithm, referred to as OptTreeTrans, to determine a sequence of long-living stable multicast trees for mobile ad hoc networks (MANETs) such that the number of tree transitions is the theoretical minimum. In this paper, we study the performance of representatives from three different classes of on-demand source-tree based distributed multicast routing protocols vis-à-vis the centralized OptTreeTrans algorithm with respect to metrics such as the multicast tree lifetime, number of links per multicast tree and the hop count per source-receiver path. Appropriately, the distributed routing protocols considered are the minimum-links based Bandwidth-Efficient Multicast Routing Protocol (BEMRP), minimum hop-based Multicast Ad Hoc On-Demand Distance Vector (MAODV) routing protocol and the stability-oriented Associativity-Based Ad hoc Multicast (ABAM) routing protocol. Simulation results reveal a tradeoff between the three metrics: we observe that the three classes of distributed multicast routing protocols discover trees that have significantly smaller lifetime than those discovered using OptTreeTrans; on the other hand, the number of links per multicast tree and the hop count per source-receiver path incurred by the stable mobile multicast trees of OptTreeTrans are relatively larger than those discovered by the three classes of distributed multicast routing protocols.

Keywords: Multicast, Bandwidth efficiency, Stability of trees, Hop count, Mobile ad hoc networks

1. Introduction

A mobile ad hoc network (MANET) is a dynamic distributed system of arbitrarily moving wireless devices with limited battery power. The wireless network bandwidth is limited, shared and prone to interference. MANET routes are often multi-hop in nature due to the limited transmission range of the wireless devices. In the presence of node mobility, routes between nodes frequently change and need to be reconfigured. As a result, on-demand route discovery (discovering a route only when required) is often preferred over periodic route discovery and maintenance, which would involve frequent exchange of control information among nodes [2]. We restrict ourselves to on-demand routing protocols in this paper.

Multicasting is the process of sending a single stream of data from one node to multiple recipients by establishing a routing tree, which is an acyclic connected sub graph containing all the nodes in the tree. While propagating down the tree, data is duplicated only when necessary. This is more advantageous than independent unicast transmissions, from the sender to each receiver, which may lead to network clogging. Multicasting in ad hoc wireless networks has numerous applications in collaborative and distributed computing like civilian operations (audio/video conferencing, corporate communications, distance learning, outdoor entertainment activities), emergency search-and-rescue, law enforcement and warfare situations, where establishing and maintaining a communication infrastructure may be expensive or difficult. A common feature among all these applications is one-to-many and many-to-many communications among the participants [20].

Several multicast routing protocols have been proposed for ad hoc wireless networks [16]. They are mainly classified as tree-based and mesh-based. In tree-based multicast protocols, only one route exists between a source and a destination and hence these protocols are efficient in terms of the number of link transmissions. The tree-based multicast protocols can be further divided into two types: source tree-based and shared-tree based. In source tree-based multicast protocols, the tree is rooted at the source, whereas in shared-tree based multicast protocols, a single tree is shared by all the sources
within the multicast group and is rooted at a node often referred to as the core node. Even though shared-tree based multicast protocols are more scalable with respect to the number of sources, these protocols suffer under a single point of failure, the core node. On the other hand, source tree-based protocols are more efficient in terms of traffic distribution. In mesh-based multicast protocols, there are multiple routes between a source-destination pair. Even though this could provide robustness, these protocols do not efficiently use the network bandwidth and are not efficient in terms of the number of link transmissions. In this paper, we restrict ourselves to studying the on-demand source tree-based multicast protocols.

Within the class of on-demand source tree-based routing protocols, three categories of multicast routing protocols have been identified: One category of multicast protocols called bandwidth-efficient protocols attempt to minimize the total number of links in the tree; the second category of protocols called minimum-hop based protocols attempt to minimize the number of hops in the paths from the source to every receiver and the third of category of protocols attempt to find stable trees. In this research, we study the classical Bandwidth-Efficient Multicast Routing Protocol (BEMRP) [17], the Multicast Ad Hoc On-Demand Distance Vector (MAODV) routing protocol [19] and the Associativity-Based Ad hoc Multicast (ABAM) routing protocol [20] respectively as representatives of the bandwidth-efficient, minimum-hop based and stability-oriented multicast routing protocol categories.

Stability of paths and trees is an important design criterion to be considered while developing MANET multicast routing protocols. A tree is considered to be broken even if one link in the tree is broken or failed. Link failures in ad hoc networks mainly occur when the constituent nodes of the link move away. Frequent attempts to discover a multicast tree could congest the network and also drain out the battery power at the critical nodes. The battery charge available at the nodes needs to be treated precisely as the nodes may be deployed in environments where recharging might be next to impossible. Link failures could also thus occur when at least one of the two constituent nodes of the link ran out of battery power. Stability of the multicast trees is essential from a Quality-of-Service point of view too. For multi-media applications that require packets to be delivered in-order, out-of-order delivery at the receiver may trigger frequent timeouts leading to unnecessary retransmissions at the source side. Eventually, the application layer at the receiver gets overloaded in handling out-of-order, lost and duplicate packets.

In an earlier work [11], we proposed a polynomial-time optimal algorithm called OptTreeTrans to determine a sequence of stable multicast Steiner trees, referred to as Stable Mobile Multicast Tree, such that the number of tree transitions is the theoretical minimum. Given the complete knowledge of the future topology changes, algorithm OptTreeTrans operates based on the following greedy principle: Whenever a multicast Steiner tree is required at a time instant \( t \), choose the longest living Steiner tree from \( t \). The above strategy is repeated over the duration of the multicast session. The sequence of such longest living stable multicast Steiner trees is called the Stable Mobile Multicast Tree. Note that in this paper, we use the terms ‘path’ and ‘route’, ‘edge’ and ‘link’ interchangeably. They mean the same.

The high-level contribution of this paper is a simulation-based performance comparison study of the theoretically optimal algorithm OptTreeTrans with that of the distributed multicast ad hoc routing protocols such as BEMRP, MAODV and ABAM, representing three different categories of MANET on-demand source-tree based multicast routing protocols. This paper is an extension of our earlier work comparing the source tree-based MANET routing protocols with respect to the stability and link efficiency [12]. Several earlier works (e.g., [6][8][15]) in the literature have compared the MANET multicast routing protocols. However, we could not find any work in the literature comparing the performance of the distributed multicast protocols with that of a theoretically optimal routing algorithm. The rest of the paper is organized as follows: Section 2 briefly describes algorithm OptTreeTrans and Section 3 gives brief descriptions of the BEMRP, MAODV and ABAM protocols. Section 4 describes the simulation environment and presents the results. Section 5 concludes the paper.

2. Algorithm for the Optimal Number of Tree Transitions (OptTreeTrans)

The problem of determining the multicast Steiner tree is that given a weighted network graph \( G = (V, E) \) where \( V \) is the set of vertices and \( E \) is the set of edges connecting these nodes, and a subset \( S \subseteq V \) of
vertices called the multicast group or Steiner points, we want to determine the set of minimum-weight edges of $G$ that can connect all the vertices of $S$ and they form a tree. In this paper, we assume the weight of each edge is unity and that all the edges of the Steiner tree are contained in the edge set of the graph. Accordingly, we define the minimum Steiner tree as the tree with the least number of edges required to connect all the vertices in the multicast group (i.e., the set of Steiner points). The problem of determining a minimum Steiner tree in an undirected graph like that of the unit disk graph is NP-complete. Efficient heuristics (e.g., [7]) have been proposed in the literature to approximate a minimum Steiner tree.

Algorithm \textit{OptTreeTrans} uses the notion of a mobile graph and mobile tree which are defined as follows: A \textit{mobile graph} [4] is defined as the sequence $G_M = G_1G_2 \ldots G_T$ of static graphs that represents the network topology changes over some time scale $T$. In the simplest case, the mobile graph $G_M = G_1G_2 \ldots G_T$ can be extended by a new instantaneous graph $G_{T+1}$ to a longer sequence $G_M = G_1G_2 \ldots G_TG_{T+1}$, 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 one second, which could, in reality, be the instants of data packet origination at the source. For simplicity, we assume that all graphs in $G_M$ have the same vertex set (i.e., no node failures).

Given the complete knowledge of future topology changes, the algorithm operates on the following principle: Whenever a multicast tree connecting a given source node to all the members of a multicast group is required, choose the multicast tree that will keep the source connected to the multicast group members for the longest time. The above strategy is repeated over the duration of the multicast session and the sequence of stable multicast Steiner trees obtained by running this algorithm is called the Stable Mobile Multicast Steiner Tree. We use the Kou et. al’s [7] well-known $O(|V||S|^2)$ heuristic ($|V|$ is the number of nodes in the network graph and $|S|$ is the size of the multicast group) to approximate the minimum Steiner tree in graphs representing snapshots of the network topology. We give a brief outline of the heuristic in Figure 1. An $(s-S)$-tree is defined as the multicast Steiner tree connecting a source node $s$ to all the members of the multicast group $S$, which is also the set of Steiner points. Note that $s \in S$.

**Input:** An undirected graph $G = (V, E)$

**Multicast group $S \subseteq V$**

**Output:** A tree $T_H$ for the set $S$ in $G$

**Step 1:** Construct a complete undirected weighted graph $G_C = (S, E_C)$ from $G$ and $S$ where $\forall (v_i, v_j) \in E_C$, $v_i$ and $v_j$ are in $S$, 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 $G_S$ 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 $T_S$ in $G_S$ (If more than one minimal spanning tree exists, pick an arbitrary one). Note that each edge in $G_S$ has weight 1.

**Step 5:** Construct the minimum Steiner tree $T_H$, from $T_S$ by deleting edges in $T_S$, if necessary, such that all the leaves in $T_H$ are members of $S$.

**Figure 1.** Kou et. al’s Heuristic [7] to find an Approximate Minimum Steiner Tree

Let $G_M = G_1G_2 \ldots G_T$ be the mobile graph generated by sampling the network topology at regular instants $t_1, t_2, \ldots, t_T$ of a multicast session. When an $(s-S)$-tree is required at sampling time instant $t_i$, the strategy is to find a mobile sub graph $G(i, j) = G_i \cap G_{i+1} \cap \ldots \cap G_j$ such that there exists at least one multicast $(s-S)$-tree in $G(i, j)$ and none exists in $G(i, j+1)$. A multicast $(s-S)$-tree in $G(i, j)$ is selected using Kou’s heuristic. Such a tree exists in each of the static graphs $G_i, G_{i+1}, \ldots, G_j$. If there is a tie, the $(s-S)$-tree with the smallest number of constituent links is chosen. If sampling instant $t_{i+1} \leq t_T$, the above procedure is repeated by finding the $(s-S)$-tree that can survive for the maximum amount of time since
A sequence of such maximum lifetime multicast Steiner \((s-S)\) trees over the timescale of \(G_M\) forms the Stable Mobile Multicast Steiner Tree in \(G_M\). The pseudo code is given in Figure 2.

**Input:** \(G_M = G_1 \cup G_2 \ldots \cup G_T\), source \(s\), multicast group \(S\)  
**Output:** \((s-S)_{\text{MobileStableTree}}\) // Stable-Mobile-Multicast-Steiner-Tree  
**Auxiliary Variables:** \(i, j\)  
**Initialization:** \(i=1; j=1; (s-S)_{\text{MobileStableTree}} = \Phi\)  

**Begin OptTreeTrans**  
1. while \((i \leq T)\) do  
   2. Find a mobile graph \(G(i, j) = G_i \cap G_{i+1} \cap \ldots \cap G_j\) such that there exists at least one \((s-S)\)-tree in \(G(i, j)\) and \{no \((s-S)\)-tree exists in \(G(i, j+1)\) or \(j = T\}\)  
   3. \((s-S)_{\text{MobileStableTree}} = (s-S)_{\text{MobileStableTree}} \cup \{\text{Minimum Steiner} (s-S)\text{-tree in} \ G(i, j)\}\)  
   4. \(i = j+1\)  
end while  
6. return \((s-S)_{\text{MobileStableTree}}\)  
**End OptTreeTrans**

Figure 2. Pseudo Code for Algorithm OptTreeTrans

### 3. Review of On-Demand Source Tree-Based Multicast MANET Protocols

In this section, we briefly review representative protocols from three different categories of on-demand source tree-based MANET multicast protocols: based on minimum-links per tree; based on minimum-hop count per path from source to receiver and based on stability.

#### 3.1 Bandwidth-Efficient Multicast Routing Protocol (BEMRP)

According to BEMRP [17], a newly joining node to the multicast group opts for the nearest forwarding node in the existing tree, rather than choosing a minimum-hop count path from the source of the multicast group. As a result, the number of links (edges) in the multicast tree is reduced leading to savings in the network bandwidth. On the other hand, the number of hops in the paths from the source to receiver increases leading to increased delay in the delivery of data packets from the source to individual receivers.

The multicast tree construction is receiver-initiated. When a node wishes to join the multicast group as a receiver, it initiates the flooding of *Join control* packets targeted towards the nodes that are currently members of the multicast tree. On receiving the first *Join control* packet, the member node waits for a certain time before sending a *Reply* packet. The member node sends a *Reply* packet on the path, traversed by the *Join control* packet, with the minimum number of intermediate forwarding nodes. The newly joining receiver node collects the *Reply* packets from different member nodes and would send a *Reserve* packet on that path that has the minimum number of forwarding nodes from the member node to itself.

To provide more bandwidth efficiency, the tree maintenance approach in BEMRP is hard-state based, i.e. a member node (including the receiver) transmits control packets only after a link breaks. BEMRP uses two schemes to recover from link failures: *Broadcast-multicast scheme* – the upstream node of the broken link is responsible for finding a new route to the previous downstream node; *Local-rejoin scheme* – the downstream node of the broken link tries to rejoin the multicast group by using a limited flooding of the *Join control* packets.

#### 3.2 Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing Protocol

MAODV [19] is the multicast extension of the well-known MANET unicast routing protocol: Ad hoc On-demand Distance Vector (AODV) routing protocol [18]. Here, a receiver node joins the...
multicast tree through a member node that lies on the minimum-hop path to the source. As a result, the hop count of the paths from the source to the receivers will be low.

A potential receiver wishing to join the multicast group broadcasts a Route-Request (RREQ) message. If a node receives the RREQ message and is not part of the multicast tree, the node broadcasts the message in its neighborhood and also establishes the reverse path by storing the state information consisting of the group address, requesting node id and the sender node id in a temporary cache. If a node receiving the RREQ message is a member of the multicast tree and has not seen the RREQ message earlier, the node waits to receive several RREQ messages and sends back a Route-Reply (RREP) message on the shortest path to the receiver. The member node also informs in the RREP message, the number of hops from itself to the source. The potential receiver receives several RREP messages and it selects the member node which lies on the shortest path to the source. The receiver node sends a Multicast Activation (MACT) message to the selected member node along the chosen route. The member node and all the intermediate nodes in the chosen path update their multicast table with the state information stored in the temporary cache. The route from the source to the receiver in the multicast tree is now setup.

Tree maintenance in MAODV is based on the expanding ring search (ERS) technique using the RREQ, RREP and MACT messages. The downstream node of a broken link is responsible for initiating the ERS to issue a fresh RREQ for the group. This RREQ contains the hop count of the requesting node from the source and the last known sequence number for that group. The RREQ can be replied only by those member nodes whose recorded sequence number is greater than that indicated in the RREQ and whose hop distance to the source is lower (to ensure that there is no loop).

3.3 Associativity-Based Ad Hoc Multicast (ABAM) Routing Protocol

ABAM [20] constructs the multicast tree based on link stability rather than hop distance. The stability of a link with a neighbor is characterized by the associativity ticks, which is the number of beacons periodically received from that neighbor since the link was formed. Each node stores the value of the associativity ticks with its neighbors. The source node initiates the tree construction phase by broadcasting a Multicast Broadcast Query (MBQ) message in the network to inform all potential receivers. When an intermediate node receives the MBQ message, it appends its node ID, the associativity ticks with the upstream node and then rebroadcasts it. Upon receiving several MBQ messages through different paths, a receiver node of the multicast group selects the most stable path and sends a MBQ-Reply packet along the selected path. After receiving MBQ-Reply packets from each receiver of the group, the source sends MC-Setup messages to all receivers in order to establish the multicast tree.

Tree maintenance in ABAM is by using a Local Query Reply cycle. The upstream node of the broken link attempts to fix a route to the receiver by broadcasting a Local Query message with TTL value of 1. When the receiver receives the Local Query message, it responds with a Local Reply message. The upstream node then sends the MC-Setup message to the receiver. If the upstream node could not find a route to the receiver, then it transfers the responsibility of fixing the route to its immediate upstream node on the path from the receiver to the source. This upstream node then initiates a broadcast of the Local Query message with TTL value of 2. This procedure is continued until the timer at the receiver expires and it sends a Join Query message to join the multicast group.

4. Simulations

We implemented the distributed BEMRP, MAODV and ABAM multicast routing protocols in the ns-2 (version 2.28) simulator [3] and the centralized OptTreeTrans algorithm (abbreviated as OTT in the Performance Figures 3 through 11) in a custom-built discrete-event simulator developed by the author in Java. This simulator has been used successfully to implement several centralized MANET routing algorithms recently proposed (e.g., [11][13][14]) by the author. The mobile graph is generated by sampling the network topology (obtained through the mobility trace files) for every 0.25 seconds. We consider a square network of dimensions 800m x 800m. The transmission range of the nodes is
250m. We vary the density of the network by conducting simulations with 35 nodes (low density) and 70 nodes (high density). The simulation time for a multicast session is 1000 seconds.

4.1 Physical Layer and Link Layer Models

The physical, data link and Medium Access Control (MAC) layer models are based on the multi-hop wireless network extension [2] provided by the CMU’s Monarch research group. The MAC layer uses the Distributed Coordinated Function (DCF) of the IEEE Standard 802.11 [5] for Wireless LANs. The radio model uses the standard channel bandwidth of 2 Mbps. The signal propagation model used is the two-ray ground reflection model [2]. The interface queue stores both the routing and data packets sent by the routing layer until the MAC layer is able to transmit them. We use a FIFO-based interface queue of length 100.

4.2 Node Mobility Model

The node mobility model used is the Random Waypoint model [1]. Each node starts moving from an arbitrary location (i.e., waypoint) at a speed uniformly distributed in the range \([v_{\text{min}}, \ldots, v_{\text{max}}]\). Once the destination is reached, the node may stop there for a certain time called the pause time and then continue to move to a new waypoint by choosing a different target location and a different velocity. A mobility trace file generated for a particular \(v_{\text{max}}\) value over the duration of the simulation time is the congregate of the location, velocity and time information of all the waypoints for every node in the network. In this paper, we set \(v_{\text{min}} = 0\). The \(v_{\text{max}}\) values used are 5 m/s (low mobility), 20 m/s (moderate mobility) and 50 m/s (high mobility). The pause time is 0 seconds.

4.3 Traffic Model and Multicast Group Size

The traffic model is constant-bit rate model. We assume there is only one source for the multicast group and the multicast group size is varied using 4 receivers (small), 12 receivers (moderate) and 24 receivers (high) for both the low and high density network scenarios. The receivers join the tree during time instants uniformly distributed from 1 to 50 seconds. During a multicast session, the source sends data packets of size 512 bytes to the multicast group for every 0.25 seconds. Each performance metric listed in Section 4.4 and plotted in Figures 3 through 11 is measured using 5 different multicast groups for each size that are run on five different mobility trace files generated for a particular value of \(v_{\text{max}}\). For each of the five versions of a particular multicast group size, the source is picked randomly from the set of nodes in the network, and the source is not a part of the multicast group.

4.4 Performance Metrics

The performance metrics measured are as follows:

(i) **Lifetime per Multicast Tree:** Whenever a link break occurs in a multicast tree, we establish a new multicast tree. The lifetime per multicast tree is the average of the time between successive multicast tree discoveries for a particular routing protocol or algorithm, over the duration of the multicast session. The larger the value of the lifetime per multicast tree, the lower the number of multicast tree transitions or discoveries needed.

(ii) **Number of links per tree:** This metric refers to the total number of links in the entire multicast tree, time-averaged over the duration of the multicast session. For example, a multicast session uses two trees, one tree with 10 links for 3 seconds and another tree with 15 links for 6 seconds, then the time-averaged value for the number of links per tree for the 9-second duration of the multicast session is \((10 \times 3 + 15 \times 6) / (3 + 6) = 13.3\) and not 12.5.

(iii) **Number of hops per receiver:** We measure the number of hops in the paths from the source to each receiver of the multicast group and average it for the duration of the multicast session. This metric is also a time-averaged value of the number of hops from a multicast source to a receiver and then averaged over all the receivers of a multicast session.
4.5 Lifetime per Multicast Tree

The lifetime per multicast tree is a measure of the stability of the trees. Algorithm OptTreeTrans determines multicast trees whose lifetime is at least 6 times larger than the lifetime of the trees discovered by the distributed multicast routing protocols. The difference in the lifetime per multicast tree between those discovered using algorithm OptTreeTrans and that discovered using the distributed routing protocols increases as the multicast group size increases. For larger group sizes, the lifetime per stable mobile multicast tree can be as large as 20 and 40 times more than the lifetime of the multicast tree discovered by ABAM and MAODV respectively. For a particular multicast group size and node mobility, the lifetime per stable mobile multicast tree increases with increase in network density. This can be attributed to the increase in the connectivity of a mobile graph spanning several static graphs. Algorithm OptTreeTrans makes use of the increase in the number of links in the network to discover long-living stable trees connecting the source to the receivers of the multicast group. On the other hand, the distributed routing protocols post a decrease in the tree lifetime with increase in network density. This can be attributed to the relative insensitiveness (to the increase in the number of links) behind the routing principles of these protocols.

Among the three distributed routing protocols, we see a clear ranking among these protocols with respect to the lifetime per multicast tree. ABAM incurs the least number of tree transitions, which is as expected because ABAM is a stability-based protocol. The interesting observation is that BEMRP trees are almost as stable as that of the ABAM trees, especially in conditions of low and moderate node mobility and for all multicast group sizes. This is because BEMRP attempts to minimize the number of links in the multicast tree. Lower the number of links, maximum is the stability. In networks of moderate and high node mobility, with smaller multicast group size, the lifetime per BEMRP tree is lower than that of ABAM tree by only 10%. On the other hand, when mobility is high, the lifetime per BEMRP tree is about 25% lower than that of ABAM trees.
In the case of MAODV, we end up choosing multicast trees with more links as we give 100% importance to the hop count per path and no priority to the number of links in the trees. Minimum hop paths are found to be less stable due to the edge effect [9][10]. The physical distance between the constituent nodes of a hop is close to the transmission range of the nodes and is bound to break at any time. This coupled with the increase in the number of links per tree makes MAODV trees highly unstable. In other words, larger the number of links in a minimum hop (shortest path) tree, larger is the probability that any link in the tree will break at any time. Edge effect is more prominent in networks of high density as MAODV attempts to choose the farthest lying node in the neighborhood as the next hop. In networks of high density and high node mobility with moderate or larger multicast group size, the lifetime per MAODV tree is about 90-100% lower than that of ABAM trees. In networks of low node mobility and smaller multicast group size, BEMRP trees are stable as that of ABAM trees.

4.6 Number of Links per Tree

As expected BEMRP, proposed with the objective to reduce bandwidth usage, yields trees that have the minimum number of links. For a given simulation condition, we observe that the number of links in the trees determined by BEMRP and ABAM are almost the same, differing by a factor of only 2-3%. For smaller multicast group sizes, algorithm \textit{OptTreeTrans} incurs the maximum number of links; where as, for moderate and larger multicast group sizes, MAODV is incurs the maximum number of links for most of the scenarios. This could be attributed to the relative independence, in MAODV, among the shortest paths chosen from the source to every receiver node of the multicast group. MAODV is designed to optimize the hop count of the paths from a source to each receiver. In pursuit of minimum hop paths, MAODV ends up choosing paths that share relatively less common links when compared to BEMRP. The probability of choosing minimum hop source-receiver paths that do not share common links increases as we increase the network density. This is because, as we increase the network density, the number of links in the network also increases.

![Figure 6.1. $v_{max} = 5$ m/s](image1)

![Figure 6.2. $v_{max} = 20$ m/s](image2)

![Figure 6.3. $v_{max} = 50$ m/s](image3)

**Figure 6.** Average Number of Links per Multicast Tree with 4 receivers in the Multicast Group

![Figure 7.1. $v_{max} = 5$ m/s](image4)

![Figure 7.2. $v_{max} = 20$ m/s](image5)

![Figure 7.3. $v_{max} = 50$ m/s](image6)

**Figure 7.** Average Number of Links per Multicast Tree with 12 receivers in the Multicast Group

![Figure 8.1. $v_{max} = 5$ m/s](image7)

![Figure 8.2. $v_{max} = 20$ m/s](image8)

![Figure 8.3. $v_{max} = 50$ m/s](image9)

**Figure 8.** Average Number of Links per Multicast Tree with 24 receivers in the Multicast Group
As we increase the multicast group size from low to moderate and high, the increase in the number of links per tree is not proportional and is less than linear. When we tripled the number of receivers to increase the multicast group size from low to moderate, the number of links per tree for all the three routing protocols and algorithm OptTreeTrans increased by a factor of $2.2 - 2.5$. On the other hand, when we increased the number of receivers by six times, i.e., when we increased the multicast group size from low (4) to high (24), the number of links per tree for all the four multicast routing protocols increased by a factor of less than 4. This shows that as we increase the number of receivers in the multicast group, the probability of link sharing increases.

4.7 Hop Count per Source-Receiver Path

The number of hops on a source-receiver path is a measure of the end-to-end delay per data packet. Algorithm OptTreeTrans incurs the largest values for the number of hops per source-receiver path, clearly illustrating the tradeoff between tree lifetime and hop count. MAODV has been proposed to reduce the hop count of the paths from the source to every receiver. Among the three distributed routing protocols, the average hop count of the paths in MAODV trees is the minimum for all the simulation conditions. But, the reduction in the hop count with respect to MAODV trees is only by at most 10%. This reduction in hop count is obtained at the expense of a lower tree lifetime and a larger number of links per MAODV tree as explained in Sections 4.5 and 4.6 respectively. The hop count per source-receiver path incurred with the stable mobile multicast trees can be 1.5 to 2.5 times more than that incurred with the three distributed routing protocols.
existence for at least certain time and are thus believed to exist for some more time in the future. BEMRP chooses routes that minimize the number of newly added links in the multicast tree while adding a receiver. Even though a minimum hop count path might exist from the source directly to the receiver, if including the minimum hop count path would result in adding more new links to the multicast tree, BEMRP would prefer a source to receiver path that would add the least number of links to the multicast tree, even though this may result in using a path that has a larger hop count from the source to receiver. We also observe that the average hop count of the paths from the source to the receivers decreases as the number of receivers in the multicast group increases. The reduction is by a factor of 10-20%.

For the three distributed routing protocols, with increase in network density, the average hop count of the paths decreases by 10-15%, especially when the multicast group size is moderate and high. This is because, as we increase the number of nodes in the neighborhood, preference would be given to choose the farthest lying node as the next hop node, particularly in the case of minimum hop count based protocols like MAODV. On the other hand, in the case of algorithm OptTreeTrans, the algorithm attempts to utilize the presence of a larger number of neighbors per node and discover stable trees that will exist for a relatively longer lifetime. This could result in the determination of trees that have a relatively larger number of links and hop count as observed in Figures 6 through 11.

5. Conclusions

The high-level contribution of this paper is a detailed simulation analysis on the three main categories of source-tree based distributed multicast routing protocols for MANETs vis-à-vis a centralized algorithm, OptTreeTrans, to discover a sequence of stable multicast trees such that the number of tree transitions/discoveries is the theoretical minimum. We simulated one classical protocol from each category of distributed MANET routing protocols: BEMRP for protocols that minimize the number of links per tree, MAODV for protocols that minimize the hop count of the paths from the source to every receiver and ABAM for protocols that aim for stable trees. We conducted detailed simulations of these three distributed routing protocols and algorithm OptTreeTrans by varying the multicast group size, network density and node mobility. Simulation results indicate that the lifetime per multicast tree incurred by the three distributed routing protocols is significantly smaller than those discovered by algorithm OptTreeTrans; thus, the stability of the multicast trees discovered by the distributed routing protocols is significantly smaller than what it could be. On the other hand, we notice a tradeoff between tree lifetime vs. the number of links per tree and the hop count per source-receiver path. The number of links per stable mobile multicast tree could be as large as 30% more than the number of links per BEMRP tree. Also, the hop count per source-receiver path in a stable mobile multicast tree could be 1.5 to 2.5 times larger than that discovered using MAODV.

Among the three distributed routing protocols, BEMRP trees are as stable as that of ABAM trees, especially in conditions of low and moderate node mobility with smaller and moderate multicast group size. Even in networks of high node mobility and larger multicast group size, BEMRP trees incur only about 25% more transitions than that of ABAM trees. MAODV trees are the least stable of all the three distributed routing protocols for all simulation conditions. In conditions of high network density, high node mobility and larger multicast group size, the lifetime per MAODV tree could be even half of that incurred for ABAM trees. The MAODV protocol gives 100% importance to minimum hop count at the expense of the increase in the number of links per tree. The larger the number of links in the tree, the lower is the stability of the tree. BEMRP trees attempt to minimize the bandwidth usage by aiming for trees that have minimum number of links. The tradeoff is increase in the number of hops on a source-receiver path, but the increase is within 10-15%. We find BEMRP to be the best distributed multicast routing protocol that not only minimizes the number of links, it incurs a relatively smaller increase in the hop count per path and incurs at most a 25% lower tree lifetime compared to ABAM.

6. References

A Simulation-based Performance Analysis of Multicast Routing in Mobile Ad hoc Networks
Natarajan Meghanathan


