CHAPTER 5

MULTI-SOURCE MULTICAST ROUTING PROTOCOL
FOR MANET

5.1 INTRODUCTION

Mobile ad hoc network is a collection of arbitrarily located mobile nodes without any centralized infrastructure. The interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad hoc network routing protocol is to provide an efficient route establishment between a pair of nodes and deliver the messages in a timely manner. Route construction should be done with a minimum of overhead and bandwidth consumption. Multicast is a technique for one-to-many or many-to-many communication. More than one source exists in the case of many-to-many communication.

Most of existing multicast routing protocols in MANETs considers only one source and become inefficient when the protocol is extended to multi-source multicasting. This chapter proposes a Multi-source Multicast Routing Protocol (MMRP) for MANET, which is an extension of the Join Request Knowledge Based Multicast Protocol introduced in the previous chapter. Detailed descriptions of algorithms used in the proposed method for junction node selection, multiple-source join and multicast receiver join are given and illustrated in this chapter. Simulations are carried by varying the
speed, data transmission time and number of sources. Finally the output results and performance are analyzed.

Sun et al (2006) makes proper improvement to ODMRP in the Cluster-based On Demand Multicast Routing Protocol (CODMRP) by adopting an Enhanced Weighted Clustering Algorithm (EWCA). This algorithm is an improvement over the basic WCA. It assures the stable hierarchical topological structure, so as to form the forward groups mainly based on the cluster-heads nodes. This technique reduces the unnecessary flooding of redundant routes query packets and achieves low overhead with high throughput.

An efficient cluster based approach for multi-source multicast routing in mobile ad hoc networks is presented by Pandi Selvam & Palanisamy (2011). In this work, the weight of each node is calculated based on number of multicast member nodes in one-hop, number of multicast member nodes in two-hop. The number of multicast member and cluster member nodes within two-hops is also considered. In this protocol, the node with largest weight is elected as cluster-head and performance results show the efficiency of this protocol in comparison with MAODV and ODMRP.

ADaptive Protocol for Unified Multicasting through Announcement (ADPUMA) is a multicast routing protocol specifically designed to reduce the overhead needed to deliver the multicast packets (Menchaca-Mendez et al 2005a). ADPUMA periodically floods a multicast announcement packet to build the mesh for each multicast group, elect the core of the mesh and get two-hop neighborhood information. It uses mesh’s k-dominating set to further reduce overhead induced by flooding the mesh when forwarding data packets. To adapt the mobility and contention, each
node dynamically selects one of the three different operation modes PUMA, DPUMA-1k or DPUMA-2k.

Garcia-Luna-Aceves & Menchaca-Mendez (2011) use the same mechanism to establish the routing structures for forwarding the unicast and multicast packets in Protocol for Routing in Interest-defined Mesh Enclaves (PRIME). Timely updates are given to only those regions of the network which shows interest in the destinations of flows. The rest of the network regions receives updates about the flows with far less frequency, or not at all. The first source that becomes active for a given unicast or multicast destination sends its first data packet. The data packets are piggybacked in a Mesh Request (MR) packet and are flooded up to a horizon threshold. In the case of a multicast flow, the receivers of the multicast group run a distributed election using Mesh Announcement (MA) packets to elect a core for the group. The core acts also as a receiver and continues to generate and send MAs. Core of the flow stops generating MAs, whenever active sources are not detected. To confine the control traffic to a region, an enclave is defined for an established mesh.

Based on the proactive routing protocol OLSR (Clausen & Jacquet 2003), a cross layer approach is adapted in OBAMPxP by Detti et al (2007). This approach reduces the signaling overhead and does not require the underlying support of flooding as in OBAMP. Ripeanu et al (2007) analyzed a distributed multi-source multicast in Unstructured Multi-Source Multicast (UMM). It is based on a simple approach to extracting source-specific multicast trees from an unstructured overlay. UMM relies on soft-state and passive data collection to adapt to the dynamics of the physical network. This results in low protocol complexity and low overheads. Sunita et al (2013) presented a novel multiobjective algorithm based on Ant Colony Optimization for multiobjective multicast routing problem. This protocol
attempts to optimize the end-to-end delay and total transmitted power simultaneously to obtain the Pareto-optimal solutions.

The simulation results from a wide range of scenarios which consist of a range of node mobility, a variety of traffic load, various sized group members, and various numbers of multicast groups illustrates the incurring of lesser amount of control overhead by MMRP. The remainder of this chapter is organized as follows. The concept of the proposed multi-source multicast routing protocol is explained in section 5.2. The simulation results are analyzed in section 5.3 and concluding remarks are given in section 5.4.

5.2 MMRP: MULTI-SOURCE MULTICAST ROUTING PROTOCOL

This section explains in detail the concepts and implementation procedures of the proposed Multi-source Multicast Routing Protocol. Consider a network of n nodes, consisting of more than one multicast group with one source per group. Let $S = \{S_1, S_2, \ldots, S_i\}$ denotes the set of sources and $G = \{G_1, G_2, \ldots, G_i\}$ denotes the set of groups corresponding to each source. All the multicast receivers belong to a group are identified by a common address. In the proposed method, the source is also considered as one among the group members. In multicasting, each source $S_i \in S$ has to deliver the data to all the multicast receivers of their corresponding $G_i \in G$.

The proposed method is a cluster based one and the communication between adjacent clusters take place through cluster-heads and junction nodes. Therefore, cluster establishment is briefed first in section 5.2.1. Route construction and maintenance procedures of the proposed method are explained in section 5.2.2 to 5.2.5. Finally, the multicast data delivery procedure is described in section 5.2.6.
5.2.1 Cluster Establishment

In this proposed method, the cluster formation and cluster-head elections are done as per the Weighted Clustering Algorithm. According to the procedure explained in section 3.3.3 and 3.3.4, the clusters are formed and corresponding cluster-heads are elected. No modifications are carried out in the fields of hello message or neighbor table.

Each node in the network update the details about the neighbor node by receiving the hello packet broadcasted with every other nodes of the network. The details of one hop neighbors are maintained by the nodes in their neighbor table. For convenience, the description of the notations and join request signals used in the proposed MMRP are given in Figure 5.1 and Figure 5.2.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>$i^{th}$ Cluster</td>
</tr>
<tr>
<td>$CH_i$</td>
<td>Cluster-head of $i^{th}$ cluster</td>
</tr>
<tr>
<td>nbt$_i$</td>
<td>Neighbor table of $i^{th}$ node</td>
</tr>
<tr>
<td>cht</td>
<td>Cluster-head table</td>
</tr>
<tr>
<td>Nnbt$_i$</td>
<td>Number of entries in the neighbor table of $i^{th}$ node</td>
</tr>
<tr>
<td>Ncht</td>
<td>Number of entries in the cluster-head table</td>
</tr>
<tr>
<td>nbt$_i$ (j)</td>
<td>Entry of $j^{th}$ node in the neighbor table of $i^{th}$ node</td>
</tr>
<tr>
<td>JN$_{ij}$</td>
<td>Junction node for communication from CH$_i$ to CH$_j$</td>
</tr>
<tr>
<td>JN$_{ji}$</td>
<td>Junction node for communication from CH$_j$ to CH$_i$</td>
</tr>
<tr>
<td>N$_{ik}$</td>
<td>$k^{th}$ member node belongs to $i^{th}$ cluster</td>
</tr>
<tr>
<td>Si</td>
<td>$i^{th}$ Source</td>
</tr>
<tr>
<td>Ri</td>
<td>Multicast receivers of $i^{th}$ source</td>
</tr>
</tbody>
</table>

Figure 5.1 Description of notations used in MMRP
<table>
<thead>
<tr>
<th>Request Signals</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JR_MSRC</td>
<td>Multi-Source join request message</td>
</tr>
<tr>
<td>JR_MRCVR</td>
<td>Multicast Receiver join request message</td>
</tr>
<tr>
<td>REJN_MSRC</td>
<td>Multi-Source rejoin request message</td>
</tr>
<tr>
<td>REJN_MRCVR</td>
<td>Multicast Receiver rejoin request message</td>
</tr>
</tbody>
</table>

**Figure 5.2 Description of join request signals used in MMRP**

### 5.2.2 Junction Node Selection

Communication between adjacent cluster regions takes place through cluster-heads and junction nodes. The detailed description of the junction node selection procedure is given in this section.

Let CH<sub>i</sub> and CH<sub>j</sub> are two cluster-heads which belongs to adjacent clusters C<sub>i</sub> and C<sub>j</sub>. Any member node N<sub>jk</sub> for k = 1, 2, 3… n of cluster C<sub>j</sub>, which also has an entry in the neighbor table nbt<sub>i</sub> of CH<sub>i</sub> is selected as the junction node. It is denoted as JN<sub>ij</sub>. Communications from CH<sub>i</sub> to CH<sub>j</sub> takes place only through this node. But, the communication from CH<sub>j</sub> to CH<sub>i</sub> will not take place through this junction node. Instead, any member node N<sub>ik</sub> for k = 1, 2, 3… n of cluster C<sub>i</sub>, which also has an entry in the neighbor table nbt<sub>j</sub> of CH<sub>j</sub> is selected as a junction node for communication from CH<sub>j</sub> to CH<sub>i</sub> and denoted as JN<sub>ji</sub>. Thus, for communication between any two adjacent cluster-heads, two hops are required (Cluster-head – junction node – Cluster-head). Figure 5.3 shows the scenario of junction node and communication path between the cluster-heads.
If more than one $N_{jk}$ node for $k=1, 2, 3... n.$ which belongs to cluster $C_j$ has an entry in the neighbor table of $CH_i$, then the distance between adjacent cluster-heads $CH_i$ and $CH_j$ through those $N_{jk}$ nodes are calculated. The $N_{jk}$ node through which minimum distance noticed is selected as the junction node $JN_{ij}$. The junction node selection is done as per the algorithm 1 shown in Figure 5.4.

**Algorithm 1 Junction Node Selection Algorithm**

1. **for** ($j=0 : j < Nnbt_i ; j++$)  
2. \Check other cluster-head members\  
3. **if** ($nbt_i(j).ch_addr != ch_addr$) **then**  
4. \Check whether adjacent cluster-head entry is already in cluster-head table or not\  

**Figure 5.4 (Continued)**
5. if (ch_lookup(nbt_i, ch_addr) == -1) then
6. \Calculate cluster-head to cluster-head distance\\
7. ch_ch_dist = nbt_i(nj) . nb_dist + nbt_i(nj) . nb_ch_dist;
8. \Update CHTABLE\\
9. ch_insert(nbt_i(nj).ch_addr, nbt_i(nj).nb_addr, 2, seq_no, ch_ch_dist);
10. else
11. adjacent cluster-head entry is available
12. \Calculate cluster-head to cluster-head distance\\
13. ch_ch_dist = nbt_i(nj) . nb_dist + nbt_i(nj) . nb_ch_dist
14. \If CH to CH distance in CHTABLE is greater than calculated distance, then update cluster-head table\\
15. if (cht_i(nbt_i(nj) . ch_addr).ch_dist > ch_ch_dist) then
16. \Update CHTABLE\\
17. ch_insert(nbt_i(nj).ch_addr, nbt_i(nj).nb_addr, 2, seq_no, ch_ch_dist);
18. end if
19. end if
20. end if
21. Repeat from step 1 for all neighbor table entries
22. End

Figure 5.4 Junction node selection algorithm

5.2.3 Multiple Source Join Procedure

This section explains the procedure to be followed in the proposed method for joining multiple sources in the multicast session. In JRKBMP of chapter 3, one source is used to deliver data to multicast receivers belongs to one multicast group. In multi source environment of the proposed method,
more than one source is used. Corresponding to each source, there is one multicast group. For three sources, there exist three multicast groups. Nodes of each group are given a unique id. Therefore, all the multicast receivers of a particular group are identified by the id of the group.

Algorithm 2, to be followed to join multiple sources in the multicast session, is shown in Figure 5.5. Similar to JRKBMP, sources send their source join request message JR_MSRC, only to their cluster-heads and are not flooded.

**Algorithm 2** Multiple Source Join Algorithm

1. Source node wants to join as a group member
2. Update JR_MSRC message
   1. daddr() = Cluster Head_
   2. next_hop_ = Cluster Head_
3. Send JR_MSRC message to CH
4. if JR_MSRC received by CH then
5. \Check the seq_no for duplicate JR_MSRC message reception\n6. if (message seq_no <= multicast_table seq_no) then
7. drop the packet
8. else
9. \Add source entry in Multicast Table\n10. mc_table_insert(group_addr, source_addr, ch_addr, next_ch_addr, hop_count, seq. no )
11. Forward copy of JR_MSRC message to adjacent CH
12. Check the CHTABLE
13. for (j=0; j < Ncht ; j ++) {
14. \Forward JR_MSRC to CHs other than previous CH\n
**Figure 5.5 (Continued)**
15. \textbf{if} (cht(j) - > ch_addr != ih - > saddr () ) \textbf{then} \\
16. \textbf{\textbackslash{U}pdate next\textunderscore{ch\_addr as current\textunderscore{node id}}} \\
17. \textbf{if} ( CH_ == 1) \{ \textit{next\_ch\_adr = index;} \} \\
18. \\textbf{Take copy of the received packet to forward} \\
19. Pc = P - > copy() ; \\
20. \\textbf{Update destination address as adjacent CH from} \\
\hspace{1cm} \textbf{CHTABLE}\} \\
21. ch_c- > daddr() = cht(j) - > ch_addr; \\
22. \\textbf{Update Next\_hop address as junction node from} \\
\hspace{1cm} \textbf{CHTABLE}\} \\
23. ch_c- > next\_hop = cht(j) - > ch\_next\_hop; [ junction node] \\
24. \\textbf{Schedule the packet to forward} \} \\
25. scheduler ::instance().schedule (target_,Pc,0.0) \\
26. \textbf{end if} \\
27. \textbf{end if} \\
28. Repeat from step 13 \\
29. Drop the original packet (P); \\
30. \textbf{end if} \\
31. \textbf{else if} \hspace{1cm} JR\_MSRC \textbf{received by CM} \\
32. \\textbf{Update Next\_hop address as destination address} \} \\
33. ch - > next\_hop_ = ih - > daddr(); \\
34. \\textbf{Schedule the packet to forward} \} \\
35. scheduler ::instance().schedule (target_,P,0.0) \\
36. \textbf{end if} \\
37. Repeat from step 4 until JR\_MSRC \textbf{received by all CH} \\
38. \textbf{End}

\textbf{Figure 5.5 Multiple source join algorithm}
The source node which has the data packets to send to the multicast receiver at first joins the multicast session. As shown in the algorithm, the sources first update the JR_MSRC message with proper cluster-head address and the multicast source group address.

The JR_MSRC message packet contains the following fields:

\[
< \text{Source address}, \\
\text{Group address}, \\
\text{Cluster-head address}, \\
\text{Next hop}, \\
\text{Hop-count}, \\
\text{Sequence number} >
\]

The source address and group address of the sources remains constant throughout the simulation time. The sequence number is an indicator of packets freshness. The source cluster-head update its multicast table using the information obtained from JR_MSRC message. After verifying the cluster-head table, it forwards the join request message to its adjacent cluster-heads. The adjacent cluster-heads update their cluster-head table and multicast table with the details available in the fields of JR_MSRC message and forwards this source join request packet to their adjacent cluster-head.

Multiple copies of the same source joining message may reach a cluster-head through different path. Duplicate reception the source join message will create confusion and it has to be avoided. For this purpose, message sequence numbers \((seq_no)\) are used. As given in the algorithm, the \(seq_no\) of the received source join request message is compared with the \(seq_no\) of the join request message already exist in the multicast table. If the received message \(seq_no\) is less than or equal to the existing message \(seq_no\)
then, it is an indication of duplicate message and dropped by the cluster-heads.

At the end of this procedure, multicast table of all the cluster-heads are updated and the path to reach the source is known to all the cluster-heads. In multiple source scenarios, the same procedure is followed by all the other sources. Figure 5.6 illustrates how the details about the multiple sources are received by all the cluster-heads.

![Diagram of Multiple sources join procedure](image-url)

**Figure 5.6 Multiple sources join procedure**

In Figure 5.6, S1 and S2 are two source nodes, which belong to cluster regions of the cluster-heads CH1 and CH6. The source node S1, send its source join request message JR_MSRC to its cluster-head CH1 and not to any other node. The flow of this request message from source cluster-head to all other cluster-head is shown by the solid line. The source S2 send its join request message with its group id, only to its cluster-head CH6. The dotted
line of Figure 5.4 indicates the way by which the join request message of source S2 reaches all the cluster-heads.

### 5.2.4 Multicast Receivers Join Procedure

Receiver nodes are cluster member nodes as well as multicast group member nodes, which are interested in receiving the data sent by the source node. The multicast receivers of each group are allotted with a unique group address. Therefore, the multicast receivers of a particular group which exist in different cluster regions are identified properly. In the proposed method, multicast receivers joining takes place as per algorithm 3 given in Figure 5.7. Multicast receivers send the multicast receiver join request message, JR_MRCVR to their cluster-heads. On receiving this request, the cluster-heads check their multicast table for the receiver’s group address entry. Then the cluster-heads identify the sources for those receivers.

**Algorithm 3 Multicast Receivers Join Algorithm**

1. Receiver wants to join as a group member
2. Update JR_MRCVR message
   3. \( \text{daddr()} = \text{Cluster Head}_\text{}; \text{next\_hop\_} = \text{Cluster Head}_\text{}; \)
3. Send JR_MRCVR message to CH
4. \textbf{if} JR_MRCVR received by CH \textbf{then}
5. \textbf{Check the for duplicate JR_MRCVR message reception}\n
6. \textbf{if} (message seq\_no \(<=\) multicast\_table seq\_no) \textbf{then}
7. \text{drop the packet}
8. \textbf{else}
9. \textbf{Add receiver entry in Multicast Table}\n
Figure 5.7 (Continued)
10. \texttt{mc_table_insert(group_addr, source_addr,ch_addr,}
    
    \texttt{next\_ch\_addr,hop\_count,seq.\,no )}

11. \texttt{\textbackslash Find the next CH address to reach the source from Multicast}
    
    \texttt{Table\textbackslash}

12. \texttt{daddr() = mc_table\, .\,get\_next\_ch ( gaddr );}

13. \texttt{\textbackslash Find the junction node for next cluster head from CHTABLE\textbackslash}

14. \texttt{next\_hop\_ = ch\_next\_hop (mc_table\, .\,get\_next\_ch ( gaddr));}

15. \texttt{Forward copy of JR\_MRCVR message to adjacent CH}

16. \texttt{scheduler::instance().schedule (target_,P,0.0)}

17. \texttt{\textbf{end if}}

18. \texttt{\textbf{else if} JR\_MRCVR \textit{received by CM}}

19. \texttt{\textbackslash Update Next\_hop address as destination address\textbackslash}

20. \texttt{next\_hop\_ = ih \,-\, daddr();}

21. \texttt{\textbackslash Schedule the packet to forward\textbackslash}

22. \texttt{scheduler::instance().schedule (target_,P,0.0)}

23. \texttt{\textbf{end if}}

24. \texttt{Repeat from step 4 until JR\_MRCVR reach the source CH}

25. \texttt{End}

---

**Figure 5.7 Multicast receivers join algorithm**

The multicast receiver join request message, JR\_MRCVR consists of the following fields:

\[
< \text{Receiver address}, \\
\text{Group address}, \\
\text{Cluster-head address}, \\
\text{Next hop}, \\
\text{Hop-count}, \\
\text{Sequence number} >
\]
In the join request message, the receiver address and group address remain constant throughout the simulation time and sequence number indicates the packets freshness. As explained in section 5.2.3, multiple reception of the JR_MRCVR message of a receiver in a cluster-head is avoided on verifying the sequence number.

The multicast table is updated by making an entry for the receiver. Then the receiver cluster-head forwards the JR_MRCVR message to its adjacent cluster-heads. This process continues until the join request message reaches the source cluster-head. If there is more than one receiver belongs to a multicast group exist in one cluster region, then the multicast table is updated by making entry for all the receivers. If the multicast receivers of one group are in different cluster regions, then the corresponding cluster-heads make entry for those receivers and update their multicast table.

When multicast receivers in a cluster region belong to different multicast groups, the multicast table is updated as usual. Based on the group address, proper sources are identified. The details about the receivers join request message are properly forwarded up to their respective source cluster-heads. The complete path between multicast receiver’s cluster-head and their respective source cluster-heads are established at the end of the receiver join request procedure. Figure 5.8 shows the procedure of multicast receivers joining to their respective source cluster-heads.
5.2.5 **Source and Receiver Rejoin Procedure**

During node mobility, the source node or the receiver node may move within their cluster region or move out of their cluster region. Movement within their cluster region will not create any problem because the nodes are still under the control of same cluster-heads. If the source or receiver node is moving to the other cluster region, their entry in the multicast table of the previous source or receiver cluster-head is removed. Then, the source and receiver rejoin procedure is initiated.

The source sends a source rejoin message REJN_MSRC to the cluster-head of the cluster region to where it moves. On receiving this message, entry for the source is made in the multicast table of the cluster-head. Next, a new source join procedure is executed for the source by the cluster-head.
The receiver sends a receiver rejoin message, REJN_MRCVR to the cluster-head of the new cluster region to where it moves. On receiving this rejoin message, the cluster-head check the multicast table for the existence of any or at least one receiver node entry having the same group address as the group address of the new receiver. If such an entry exists, then the detail about new receiver is simply appended to the existing entry in its multicast table. Otherwise, a new receiver join procedure is executed as usual.

5.2.6 Multi-source Multicast Data Delivery Procedure

A complete path detail for data delivery from multiple sources to their corresponding multicast group receivers can be obtained from the multicast table and cluster-head table maintained by the cluster-heads. Cluster-heads are categorized according to the nature of their cluster region as follows.

- **Source / Receiver cluster-heads:** Cluster-heads which contain source / receiver in their cluster region.

- **Forwarding cluster-heads:** Cluster-heads having no sources or receivers in their cluster region belongs to this category. These cluster-heads are used to simply forward the data.

- **Source / Receiver cum forwarding cluster-heads:** These types of cluster-heads contain source / receiver in their cluster region. In addition, they forward the data to adjacent cluster-heads.

The procedure as given in algorithm 4 in Figure 5.9 is executed by the sources for the data delivery to their respective multicast group receivers.
Algorithm 4 Multi-Source Multicast Data Delivery Algorithm

1. DATA is ready in Source
2. Update next_hop as cluster Head_ in Packet Header
3. next_hop_ = Cluster Head_
4. Source node forwards packet to CH
5. if data received by CH then
6. Check the multicast table for any entry of receiver node for the group
7. if no receiver entry for the group then
8. Go to STEP 20
9. else
10. \copy the packet and forward to receiver cluster member\n11. for ( i=0 ; i < mc_table . get_receiver_count() ; i++)
12. Pc = P -> copy();
13. \Update Next_hop address as receiver address\n14. next_hop = mc_table . receiver_id(i);
15. \Schedule the packet to forward\n16. scheduler ::instance().schedule (target_,Pc,0.0)
17. Repeat from step 11
18. end
19. end if
20. Cluster-head forwards the DATA to other forwarding cluster- heads
21. for ( i=0 ; i < mc_table . get_forward_count() ; i++)
22. \Forward DATA to CHs other than previous CH\n23. if (mc_table . forward_ch(i) ! = index & & mc_table . forward_ch(i) ! = ch -> prev_hop_)
24. \copy the packet and forward to those CH\n
Figure 5.9 (Continued)
25. \( P_c = P \rightarrow \text{copy}() \)
26. \( \text{next\_hop}_c = \text{ch\_next\_hop}(\text{mc\_table} . \text{forward\_ch}(i)) \);
27. \( \backslash \text{Schedule the packet to forward}\backslash \)
28. \( \text{scheduler} :: \text{instance}().\text{schedule}(\text{target}, P_c, 0.0) \)
29. \( \textbf{end if} \)
30. Repeat from step 21
31. \( \textbf{end} \)
32. Drop the original packet (P);
33. \( \textbf{else if} \) data received by CM then
34. \( \textbf{if} \) CM is receiver of the group then
35. \( \backslash \text{Send the packet to Application Layer}\backslash \)
36. \( \text{recv}(\text{dmux}, p); \)
37. \( \textbf{else} \)
38. Forward the data to ClusterHead
39. \( \textbf{end if} \)
40. \( \textbf{end if} \)

---

**Figure 5.9 Multi-source multicast data delivery algorithm**

Source nodes sent the data packets, only to their cluster-heads. On receiving the data packet, source cluster-head check its multicast table for any receiver’s entry in its own cluster region. If such an entry found, then the data packet is sent to those receivers. Next, the data packet is forwarded to their adjacent cluster-heads, through proper junction nodes. Data packets are not forwarded to cluster-heads having no entry for the source in their multicast table.

On receiving the data, the receiver cluster-head check its multicast table for the receiver’s entry. The receiver cluster-head identifies the receiver nodes from its group address. Then the data are delivered to the respective
multicast receivers. This process continues until the data is delivered to all the multicast receiver nodes of a source. Figure 5.10 shows the data delivery path between multiple sources and their respective multicast receiver members. CH3 is a receiver cluster-head and CH4 is a forwarding cluster-head. CH2 is a source/receiver cum forwarding cluster-head. It distributes the data packets to the multicast receiver exist in its cluster region. Also, it forwards the data packet to its adjacent cluster-heads.

![Figure 5.10 Multi-source multicast data delivery](image)

5.3 PERFORMANCE EVALUATION

In order to capture the realistic effect of the real system, every simulation experiment done to evaluate the performance of a mobile ad hoc system should consider a variety of modeling conditions like data transmission rate, mobility of node, group size and number of groups. The details about the simulator used and the parameters set for the performance analysis of MMRP are explained in section 5.3.1. The impact of modeling
conditions on the performance metric are shown and explained in section 5.3.2 to 5.3.4.

5.3.1 Simulation Parameters

To evaluate the performance of the proposed multicast routing protocol, Network Simulator ns-2 (Version 2.32) was used. NS-2 is an object-oriented discrete time event simulator written in C++, with an OTcl interpreter and its modular design made it to be extensible. In a discrete-event simulation, the model evolution is defined by instantaneous events. Each event corresponds to a transition in a portion of the model state, composed of state variables, each describing a characteristic of the model. This discrete-event simulator provides substantial support for simulation of TCP, routing, unicast, multicast protocols over wired, wireless, and wireless multi-hop ad hoc networks.

This simulation modeled a randomly placed 50 nodes within 500 m x 500 m area. The MAC layer is IEEE 802.11 and each simulation was executed for 100s of simulation time. Bouncing boundary is the boundary policy of the simulated area. By this policy, the mobile node will bounce back towards the simulated area when they touch the boundary line and try to move outside the boundary. The mobile nodes will not leave the simulation area and the number of mobile nodes within the given area during the entire simulation time remains constant.

The random waypoint mobility model is used for the mobile nodes which are assumed to be moving throughout the network area. The mobility of nodes is set using the modified version of setdest command. At the beginning of a mobility interval, the mobile node selects the new destination coordinates (x, y), uniformly distributed in the simulated area. Speed is uniformly distributed in $[V_{min}, V_{max}]$. To avoid the drawback of random
waypoint model (Yoon 2003), the value of $V_{\text{min}}$ is set to a non-zero value. At this point, mobility is not given to the cluster-heads.

With omni directional antenna, Two-Ray ground reflection model assumes propagation over a circular area centered on the transmitter position. In this simulation, the channel capacity set to 2Mb/s, Radio propagation range is set to 200 meters with omni directional link and Physical carrier sense range is set to 200 meters. Mostly used traffic model for MANET simulation analysis is the Constant Bit Rate (CBR) traffic model. In this model, every source generates a constant flow of data packet. In this proposed method, the multicast data streams are CBR data packets of 512 bytes.

Multicast groups with group size of 4, 6, 8 and 10 are considered for the simulation and, the source is also considered as one of the group member. Receivers join the multicast group at the beginning of the simulation and never leave the group during the entire simulation time.

Two main factors influence the performance metric: rate of changes in mobility and data transfer rate. In addition to the network size, the multicast group size may be a more interesting factor and affects multicast performance. Therefore, the performance of the proposed method is analyzed by varying node mobility, data transmission rate and multicast group size. The packet delivery ratio, control overhead and end-to-end delay is taken as performance metrics.

5.3.2 Performance Analysis in Static Network

Performance analysis of the proposed MMRP is first done with the static network condition. Therefore, mobility is not given to any of the nodes. Each multicast group has one source and the remaining members are considered as multicast receivers.
5.3.2.1 Impact of multiple sources on control overhead

With the increase in number of sources and groups, the numbers of multicast receivers are also increasing. As expected, the control overhead increases for both the proposed method and PUMA. Figure 5.11 clearly shows, the proposed method incurs only a lower amount of control overhead bytes compared to PUMA. The proposed method sends the joint request messages, only to the cluster-heads and not flooded or broadcasted and this is the main reason for the lower control overhead.

![Figure 5.11 Control overhead as a function of number of groups](image)

5.3.2.2 Impact of multiple sources on packet delivery ratio

Packet delivery ratio analysis is shown in Figure 5.12 and the packet delivery ratio of the proposed method is comparable to PUMA.
Figure 5.12 Packet delivery ratio as a function of number of groups

The impact of varying data transmission rate on the performance of MMRP is analyzed in section 5.3.3.

5.3.3 The Impact of Varying Data Transmission Rate

By varying data transmission rate, the number of data packets to be delivered to the multicast receivers can be varied. To evaluate the performance of the proposed method under various data transmission rates, three multicast groups, each having one source and three receivers are considered. The data transmission rate is varied from 10 kb/s to 100 kb/s insteps of 10 kb/s.

The value of simulations having 95% Confidence Interval (CI) is calculated and depict in the output graph itself. Error bars shown in the output graphs represent a CI of 95%. Table 5.1 lists the numerical value obtained during the simulation for the packet delivery ratio, control overhead incurred and end-to-end delay on varying the data transmission rate.
Table 5.1  Impact of data transmission rate on packet delivery ratio, control overhead and end-to-end delay

<table>
<thead>
<tr>
<th>Data Transmission Rate (kb/s)</th>
<th>Packet Delivery Ratio (%)</th>
<th>Control Overhead (MB)</th>
<th>End-to-End Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMRP</td>
<td>PUMA</td>
<td>MMRP</td>
</tr>
<tr>
<td>10</td>
<td>94.48</td>
<td>98.84</td>
<td>0.663</td>
</tr>
<tr>
<td>20</td>
<td>95.74</td>
<td>98.86</td>
<td>0.724</td>
</tr>
<tr>
<td>30</td>
<td>93.93</td>
<td>98.95</td>
<td>0.780</td>
</tr>
<tr>
<td>40</td>
<td>91.79</td>
<td>98.49</td>
<td>0.840</td>
</tr>
<tr>
<td>50</td>
<td>90.97</td>
<td>98.02</td>
<td>0.896</td>
</tr>
<tr>
<td>60</td>
<td>89.05</td>
<td>97.11</td>
<td>0.955</td>
</tr>
<tr>
<td>70</td>
<td>91.00</td>
<td>96.18</td>
<td>1.012</td>
</tr>
<tr>
<td>80</td>
<td>90.71</td>
<td>93.37</td>
<td>1.069</td>
</tr>
<tr>
<td>90</td>
<td>90.04</td>
<td>93.59</td>
<td>1.132</td>
</tr>
<tr>
<td>100</td>
<td>87.97</td>
<td>88.93</td>
<td>1.190</td>
</tr>
</tbody>
</table>

The impact of varying data transmission rate on the control overhead incurred is shown in Figure 5.13. The proposed method uses a lower amount of control overhead compared to PUMA. While PUMA shows a large increase in control overhead, only a small increase in control overhead is noticed in MMRP. For the data transmission rate of 60 kb/s, the control overhead incurred for the proposed method is only 16.36 % of the control overhead incurred by PUMA. The percentage increase in control overhead used at the data transmission rate of 100 kb/s, compared to data transmission rate of 10 kb/s is only 79.44 % for the proposed method, while it is 216.28 % for PUMA.
Figure 5.13 Control overhead under various data transmission rate

Figure 5.14 shows the impact of data transmission rate on the packet delivery ratio. On increasing the traffic load, some reduction in packet delivery ratio is noticed for both the protocol. In addition to this, the packet delivery ratio of the proposed method is slightly lower than PUMA. It is interesting to observe from Figure 5.15, the average end-to-end delay of both the proposed method and PUMA increases beyond the data transmission rate of 60 kb/s.

Figure 5.14 Packet delivery ratio under various data transmission rate
5.3.4 The Impact of Varying Mobility

Ad hoc networks may have different degrees of mobility and it plays an important role in designing a protocol for such networks. In a network with a high degree of mobility, nodes move relatively fast and results in rapidly changing topology. Mobility of mobile ad hoc networks adds complexity in achieving reliability. The performance of the proposed method is evaluated with different mobility speed conditions. For one set of simulation, the speed of the mobile node is varied from 0.1 m/s to 0.9 m/s in steps of 0.1 m/s. For another set of simulation, the speed of the mobile node is varied from 1 m/s to 5 m/s in steps of 1 m/s. All the nodes except the cluster-heads are in mobile condition.

Three multicast groups, each having one source and three receivers with the data transmission rate of 40 kb/s is the scenario used for this simulation. The rate of packet delivery ratio achieved and the amount of control overhead incurred are tabulated in Table 5.2 and Table 5.3.
Table 5.2  Impact of low mobility on packet delivery ratio and control overhead

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Packet Delivery Ratio (%)</th>
<th>Control Overhead (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMRP</td>
<td>PUMA</td>
</tr>
<tr>
<td>0.1</td>
<td>89.81</td>
<td>98.16</td>
</tr>
<tr>
<td>0.2</td>
<td>90.01</td>
<td>99.13</td>
</tr>
<tr>
<td>0.3</td>
<td>89.81</td>
<td>98.95</td>
</tr>
<tr>
<td>0.4</td>
<td>90.68</td>
<td>98.46</td>
</tr>
<tr>
<td>0.5</td>
<td>91.49</td>
<td>97.94</td>
</tr>
<tr>
<td>0.6</td>
<td>90.93</td>
<td>98.02</td>
</tr>
<tr>
<td>0.7</td>
<td>90.47</td>
<td>98.46</td>
</tr>
<tr>
<td>0.8</td>
<td>86.37</td>
<td>99.44</td>
</tr>
</tbody>
</table>

Table 5.3  Impact of high mobility on packet delivery ratio and control overhead

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Packet Delivery Ratio (%)</th>
<th>Control Overhead (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMRP</td>
<td>PUMA</td>
</tr>
<tr>
<td>1</td>
<td>85.20</td>
<td>98.92</td>
</tr>
<tr>
<td>2</td>
<td>83.64</td>
<td>99.21</td>
</tr>
<tr>
<td>3</td>
<td>83.14</td>
<td>96.79</td>
</tr>
<tr>
<td>4</td>
<td>84.58</td>
<td>98.70</td>
</tr>
<tr>
<td>5</td>
<td>90.65</td>
<td>99.34</td>
</tr>
</tbody>
</table>

Figure 5.16 illustrates the overhead associated with the multicast route formation and data delivery of the proposed method as a function of lower mobility speed. Nearly same amount of control overhead is used by the proposed method and is much lower than PUMA.
Figure 5.16 Control overhead as a function of mobility speed (0.1 to 0.9 m/s)

Figure 5.17 demonstrates the lower amount of control overhead used by the proposed method during the high mobility condition. In the proposed method, no control signal is flooded throughout the network. The communication path is re-established based on the detail available in the multicast table and cluster head tables maintained by the cluster-heads. In fact, the hello packet interval value is reduced for higher mobility speed to incorporate the quick changes that take place in the position of high mobile nodes.

Figure 5.17 Control overhead as a function of mobility speed (1 to 5 m/s)
Performance of MMRP in terms of the packet delivery ratio as a function of low and high mobility speed was shown in Figure 5.18 and Figure 5.19. The proposed method shows some reduction in the packet delivery ratio compared to PUMA.

Figure 5.18  Packet delivery ratio as a function of mobility speed (0.1 to 0.8 m/s)

Figure 5.19  Packet delivery ratio as a function of mobility speed (1 to 5 m/s)
5.3.5 The Impact of Multiple Sources (Groups) and /or Variable Group Size

In ad hoc networks, the route construction and multicast data delivery has to be done with lower amount of control overhead because of the limited bandwidth associated with it. The impact of multiple sources (groups) and the increasing number of multicast receivers (group size) on control overhead and the packet delivery ratio are analyzed in this section. In this simulation, up to three groups are considered. Each group consists of one source and numbers of receivers in the group are varied as 3, 5, 7 and 9. The results of simulation for packet delivery ratio and control overhead used are tabulated in Table 5.4 and Table 5.5.

Table 5.4 Packet delivery ratio for variation in group size and number of groups

<table>
<thead>
<tr>
<th>Number of multicast group members</th>
<th>Packet Delivery Ratio (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 Group</td>
<td>2 Group</td>
<td>3 Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PUMA</td>
<td>MMRP</td>
<td>PUMA</td>
<td>MMRP</td>
<td>PUMA</td>
<td>MMRP</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>99.40</td>
<td>99.27</td>
<td>98.78</td>
<td>93.46</td>
<td>98.26</td>
<td>90.08</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>99.72</td>
<td>99.10</td>
<td>98.61</td>
<td>90.71</td>
<td>98.12</td>
<td>87.54</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>99.13</td>
<td>95.23</td>
<td>98.79</td>
<td>89.53</td>
<td>98.18</td>
<td>85.82</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>98.76</td>
<td>91.23</td>
<td>92.76</td>
<td>85.08</td>
<td>91.84</td>
<td>81.61</td>
</tr>
</tbody>
</table>

Table 5.5 Control overhead incurred for variation in group size and number of groups

<table>
<thead>
<tr>
<th>Number of multicast group members</th>
<th>Control Overhead (MB)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 Group</td>
<td>2 Group</td>
<td>3 Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PUMA</td>
<td>MMRP</td>
<td>PUMA</td>
<td>MMRP</td>
<td>PUMA</td>
<td>MMRP</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1.107</td>
<td>0.546</td>
<td>3.090</td>
<td>0.705</td>
<td>4.885</td>
<td>0.837</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1.577</td>
<td>0.547</td>
<td>4.259</td>
<td>0.708</td>
<td>6.177</td>
<td>0.845</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1.846</td>
<td>0.548</td>
<td>4.811</td>
<td>0.712</td>
<td>7.317</td>
<td>0.846</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>2.105</td>
<td>0.550</td>
<td>5.534</td>
<td>0.715</td>
<td>7.649</td>
<td>0.847</td>
</tr>
</tbody>
</table>
Figure 5.20 illustrates the packet delivery ratio for various numbers of source (groups) with various numbers of receivers (3, 5, 7 and 9) in each group. With the increase in the number of sources or the number of multicast receiver, some reduction in packet delivery ratio is observed for both MMRP and PUMA. The rate of decrease in the packet delivery ratio is indicated by the linear trend line.

Figure 5.20 Packet delivery ratio as a function of group size and number of groups

The amount of control overhead utilized under multiple sources scenario is shown in Figure 5.21. As expected, increase in control overhead is observed for both MMRP and PUMA.
The average trend line of Figure 5.21 indicates the control overhead incurred in MMRP. It is maintained almost at a same value and only a small variation is noticed. But, PUMA shows a large increase in control overhead.

5.4 CONCLUSION

This chapter proposed a multi-source multicast routing protocol for mobile ad hoc networks. The key contributions of this work shows the possibility of establishing and maintaining routing structures for multi-source multicasting in an ad hoc network without the need to flood the control packets throughout the network. In this cluster based protocol, mobility is not given to the cluster-heads and two junction nodes are used to have communication on both sides between the adjacent cluster-heads. The algorithms to be followed during various phases of MMRP are explained with illustrations. On completion of the source and receiver joining procedure, a complete path between source and receivers is established. To deal with the mobility of source and receivers, a process of rejoin technique is
implemented. In addition to this, the multicast tables maintained by the cluster-heads are used as a collective structure for route maintenance and data delivery.

The results of a series of simulation experiments confirm that, the proposed method incurs a lesser amount of control overhead without much deprivation in packet delivery ratio. Video transmission and voice based applications consumes more bandwidth. In future, the analysis can be extended with audio and video multicast transmission over a mobile ad hoc network.