CHAPTER 4
JOIN REQUEST KNOWLEDGE BASED MULTICAST PROTOCOL

4.1 INTRODUCTION

One of the major challenges facing multicasting protocols in ad hoc network is the excessive data packet overhead. This problem is more acute in mesh based protocols as compared to tree based protocols. This problem principally occurs because, large number of nodes which are neither interested in receiving nor sending a data packet need to transmit the packet in order to forward it from senders to receivers.

Conventional routing protocols such as Ad hoc Multicast Routing protocol, Core-Assisted Mesh Protocol and Ad hoc Multicast Routing Protocol utilizing Increasing id-numberS are proactive multicast routing protocols. Periodic broadcast of network topology updates are needed to compute the shortest path from the source to every destination. This will incurs a large amount of control overhead.

Protocols like On-Demand Multicast Routing Protocol and Multicast Ad hoc On-Demand Distance Vector protocols have a significant control overhead. Protocols like Robust Multicasting in Ad hoc Networks using Trees (ROMANT) (Vaishampayan & Garcia-Luna-Aceves 2004b) and PUMA have reduced control overhead. Any further reduction in total overhead is possible only by a reduction in data packet overhead. The
Protocol for Multicasting Over Directional Antennas (MODA) for mobile ad hoc networks is the first protocol for MANET’s which uses directional antennas to reduce data packet overhead (Vaishampayan et al 2005). A significant reduction in data packet overhead is achieved in MODA by taking advantage of the longer ranges attainable with directional antenna transmissions. MODA involves only a fewer nodes in the forwarding process. In MODA the nodes try to cover two hops instead of one, while forwarding a data packet.

All nodes in MODA periodically transmits a control packets called as MODA announcement, which contains the details of address of the core, distance to the core, children, parent, grant parent and their locations. A node which is closest to the center of the network is elected as core by the nodes of MODA. Receivers announce them as tree members and the tree establishment process is similar to PUMA. Over wide range of scenarios, MODA provide a packet delivery ratio comparable to ODMRP and PUMA, while incurring far less overhead.

To reduce the signaling overhead incurred in ODMRP, several extensions were presented. ODMRP with Multi-Point Relay (ODMRP-MPR) induct multipoint relay technique to reduce the control overhead (Zhao et al 2003). With high scalability, it effectively solves the unidirectional link problem of wireless communication. ODMRP with ASYMmetric topology (ODMRP-ASYM) presented by Gerla et al (2005) has an advantage of achieving lower control overhead comparable with ODMRP even in highly asymmetric topologies. It handles the unidirectional link without any performance degradation. Narushima et al (2007) enhanced ODMRP by using the node movement patterns adaptively and control packets are transmitted only when it is necessary. The routes are reconstructed in anticipation of topology changes and it eliminates route acquisition latency.
Ant-based Adaptive Multicast Routing protocol (AAMRP) for ad hoc networks exploits group members’ desire to simplify multicast routing and invoke broadcast operations in appropriate localized regions (Sabari & Duraiswamy 2009). By reducing the number of group members which try to participate in the construction of the multicast structure, this protocol significantly lowers the overheads.

This chapter introduces the mobility feature over UFCBMP and proposed a new Join Request Knowledge Based Multicast Protocol (JRKBMP) for mobile ad hoc network. It starts with a description of the junction node selection, source and receiver joining procedure, followed by the description of the mobility management. Then, multicast data delivery procedure of the proposed method is discussed. At the end, the simulation results are presented and concluded.

4.2 EXPERIMENTAL SECTION

The proposed method consists of three important sections.

(i) Cluster establishment section

(ii) Route establishment section and

(iii) Data forwarding section

At the start of the simulation, cluster regions are formed and cluster-heads are selected. This constitutes the cluster establishment section and explained in Section 4.3. The process of source and multicast receiver joining takes place in the route establishment section which is detailed in Section 4.4. Data forwarding process as in Section 4.5 elaborate the method of data flow from source to the multicast receiver.
4.3 CLUSTER ESTABLISHMENT SECTION

For any cluster based protocol, cluster-head election is an important task. In the proposed method, cluster-head elections are based on the Weighted Cluster Algorithm. Using this algorithm, it is possible to have a minimum number of clusters and more cluster members can be accommodated within a cluster. Node weight calculations are done at the start of the simulation. Hello message, which consists of information regarding the node position, its weight, cluster-head address, cluster-head distance is broadcasted by each node to its one hop neighbors. Each node updates its neighbor table on receiving the hello message from other nodes. If the hello message of one node is not received by another neighbor node, then it indicates the node movement away from its neighbor node.

Cluster-heads broadcasts hello message to all the nodes within one hop distance from it. Non-cluster member nodes and non cluster-head nodes, which receive a hello message of the cluster-head forms a cluster. Therefore, all the cluster members are within one hop distance from their cluster-head. A node may discover a change in its neighbor caused by link failure or a new link from the neighbor table. The tables maintained by cluster-heads and cluster member nodes are periodically updated and refreshed.

4.4 ROUTE ESTABLISHMENT SECTION

For proper data delivery, a complete route has to be established between the source and multicast receivers. The multicast receivers of a group may exist in one cluster region or they exist in different cluster regions. The path for data delivery from source to receiver is established in the route establishment phase. On executing the source and receiver join procedure, a
route for data delivery is established between the source and multicast receivers. Source join request and receiver join request messages are not broadcasted or flooded throughout the network. Due to node mobility, the source or receiver may move out of their cluster region. Thus, rejoin procedure for source and receiver are executed during the mobility of the nodes.

4.4.1 Junction Node Selection Procedure

Communication between adjacent cluster-heads or adjacent cluster regions takes place through the junction nodes. Cluster member nodes send the hello message which contains the updated information about their neighbor and cluster-head. At periodic intervals, the cluster-heads, checks their neighbor table to know the status of neighbor nodes. Accordingly, the entries in the cluster-head table are updated. In the proposed method, a cluster-head-m tries to communicate with the other cluster-head-n through the member node of the cluster-head-n’s cluster region. Junction node selection is an important task, which has an influence on the performance of the protocol.

The procedure used for the selection of junction node is shown as a flowchart in Figure 4.1.
Figure 4.1 Flowchart for junction node selection

1. Check the CH table for the first adjacent CH
2. Check the neighbour table for the entry of first member of the adjacent CH
3. Calculate CH-CM-CH distance
4. Is there any other member of this CH available?
   - Yes: Find the next member and calculate CH-CM-CH distance
   - No: New distance < Old distance
     - Yes: Update the CH table
     - No: Junction node identified for all CH?
       - Yes: End of the Process
       - No: Find the next adjacent CH
5. Select the CM as junction node for this CH
Nodes available in the cluster region of the cluster-head-m, which also has an entry in the neighbor table of cluster-head-n, is selected as the junction node and denoted as \( J_{mn} \). Communication from cluster-head-n to the cluster-head-m will take place only through this node. But, the communication from cluster-head-m to the cluster-head-n will not take place through this junction node. For this purpose, a member node of cluster-head-n is selected and denoted as \( J_{nm} \). Detailed description of the junction node selection procedure is given in section 5.2.2 of chapter 5.

If more than one node competes for the junction node selection, then the procedure as explained in section 3.3.5.1 is followed. The notations used in the figures and their description are given for convenience.

\[
\begin{align*}
N & \quad \text{Nodes of the network} \\
CH_i & \quad \text{Cluster-head } i \\
S & \quad \text{Source node} \\
R & \quad \text{Multicast receiver node}
\end{align*}
\]

Figure 4.2 shows the junction nodes and the communication path between adjacent cluster-heads in upstream and downstream directions.
After receiving the next hello message from other cluster members, a new junction node is selected in case of a junction node failure.

### 4.4.2 Source Join Procedure

Every cluster-head maintains a neighbor table, multicast table and cluster-head table. As usual, neighbor table maintains the one hop neighbor nodes detail. Multicast table is used to maintain information regarding the multicast source and receiver nodes. Cluster-head table maintains the details of adjacent cluster-heads and the junction nodes between them. Source node has data packets to send to the multicast receiver nodes. For sending the data, the source node has to initiate the multicast session. This procedure is similar to the procedure explained and used in section 3.3.5.2 of the previous chapter. The source node sends a source join request message \( src_{jr} \), only to its cluster-head. The format of source join request message is shown in Figure 4.3.
The purpose of each field in the source join request message is as follows:

Source IP Address : This field indicates the ID of the source node

Multicast Group Address : This field indicates the Group address of the multicast receivers. All the multicast receivers of a group are identified by a single common address

Next CH to reach Receiver : This field contains the address of the adjacent cluster-head. From the cluster-head table the required address in this field is obtained.

Cluster-head Address : The Cluster-head address of the source node is indicated in this field

Next Hop Address : This field indicates the address of the junction node to reach the adjacent cluster-head, if the join request message emanate from the cluster-head.
Sequence Number : This field is used to find the freshness of the request message received at a node. Duplicate reception and looping of the join request message is avoided by verifying the received sequence number with the already existing sequence number.

Hop count : This count value is incremented, whenever the join request message moves from junction node to cluster-head or cluster-head to junction node. On completion of the source join request procedure this field gives the number of hops between the source cluster-head and other cluster-heads.

Type : It indicates the nature of the node

Figure 4.4 shows the flow chart of source join procedure used in the proposed method. The source cluster-head, forwards the src_jr (source join request) message to all other cluster-heads through proper junction nodes as shown in Figure 4.5. Now the multicast tables of cluster-heads are updated and details about the source node and its cluster-head are stored. Therefore, information regarding the source and the path to reach it is known to all the cluster-heads.
Source Join Request received by a node

Update next hop as cluster-head

Is it a CH?

Forward the join request to cluster-head

Check the multicast table

Drop the request

New sequence number > Old sequence number

Update multicast table and forward the request to adjacent cluster-heads

Check the CH table for adjacent CH

Copy the request packet and update the destination address as adjacent CH address and next hop as junction node

Is all the CH received the request?

End of the Process

Figure 4.4 Flowchart for source join procedure
4.4.3 Receivers Join Procedure

The multicast members intended to receive the multicast data packets are multicast receivers. Multicast receivers send a receiver join request message \textit{recvr\_jr} to their cluster-heads. On receiving this request, the cluster-heads check their multicast table for source entry and identify the source cluster-head. Then, the receiver join request messages are forwarded up to the source cluster-head as shown in Figure 4.6.
A complete path between the source and the receivers is established at the end of source join and receiver join procedure. In the proposed method, src_jr and recvr_jr messages are sent only to the cluster-heads and not flooded. Flow chart shown in Figure 4.7 illustrates the procedure to be followed for receiver joining in the proposed method.

![Flow chart for receiver join procedure](image-url)
4.4.4 Members Leave from Cluster Region

Very simple soft state approach is used in the proposed method for the node leaving a cluster region. Due to mobility, the node movements out of a cluster region and node movement from one cluster region to other cluster region will take place. The node may be a source, multicast group receiver or just a member of the cluster. If a node moves out of a cluster region, then its hello message will not reach the cluster-head. During the next hello period, the cluster-head notice this. In turn, the node’s entry will be removed from the neighbor table of the cluster-head.

4.4.5 Source and Receiver Rejoin Procedure

This section explains the procedure to be followed by the source or receiver nodes when they are moving out of their own cluster region. In the proposed method, during the mobility as well join request messages are not flooded. Instead, source or receiver rejoining messages are sent only to the new cluster-head. Expected and possible movements of the source and receiver nodes are:

(i) Movement of source and receiver within their cluster region.

(ii) Movement of source and receiver from one cluster region to another cluster region.

Movement of source or receiver within their cluster region will not create any problem because they are still under the control of the same cluster-head. Discontinuity or link breakage between source and multicast group receivers takes place only when they are moving out of their cluster region. During this sort of situation, source and receiver have to join to the
cluster-head of the new cluster region. For this purpose, source and receiver rejoining procedure is executed.

Due to mobility, the source or receiver may move from one cluster region to another cluster region. Now, the hello messages of source and receiver will be received by the new cluster-head. The entries for them are updated in its neighbor table and multicast table. After this, the source and the receiver have to send the rejoin request messages to the cluster-head of new cluster region. If the source moves to a new cluster region, then it will send a source rejoin request message $src\_rejn$ to the new cluster-head. Now, the entire path from the source has to be reconstructed. The actual source joining procedure is initiated once again from the new cluster-head and a new path is established.

If the multicast group receiver moves to a new cluster region, it sends a receiver rejoin request message $recvr\_rejn$ to the new cluster-head. Receiver rejoins message contains the group address of the receiver. Now the cluster-head check its multicast table for any other receiver node’s entry. If such an entry is already available, then the cluster-head just add the information regarding new receiver and update the multicast table. Otherwise, a new receiver join procedure is initiated.

4.5 DATA FORWARDING SECTION

Details about the source node and multicast receiver nodes are maintained by the cluster-heads. Source node sent the data packets merely to its cluster-head and not to any other node. The flowchart in Figure 4.8 shows the procedure to be followed by the proposed method during the data forwarding process.
Figure 4.8 Flowchart of data delivery procedure

Multicast table of source cluster-head contains the details of the receiver group nodes and their corresponding cluster-heads. If any of the receiver’s cluster-head is at two hop distance from the source cluster-head, data packets are delivered directly through the junction node and cluster-head. Otherwise, intermediate cluster-heads are used. If the multicast receivers
belong to various cluster regions, then the data packets are sent only to their cluster-heads from the source cluster-head. Data packets are not sent to the cluster-heads, which will not have any receiver entry in their multicast table.

On verifying the multicast table, receiver cluster-heads deliver the data packets to the multicast receivers. Data forwarding phase of the proposed method is shown in Figure 4.9.

![Figure 4.9 Data forwarding paths](image)

**4.6 SIMULATION RESULTS AND DISCUSSION**

To evaluate the effectiveness of the proposed scheme, a simulation is conducted using the network simulator ns-2.32. The scenarios and parameters considered for the simulation are detailed in section 4.6.1. Simulation results for varying hello packet interval, varying node mobility and varying the data transmission over the performance metrics are discussed in sections 4.6.2 to 4.6.4.
4.6.1 Simulation Scenarios

This simulation models a network of 50 nodes randomly placed within a 500 m x 500 m area. One source with ten receivers forms a multicast group. Each simulation is executed for 200 seconds and IEEE 802.11 MAC layer is used. Propagation model used in this simulation is Two Ray Ground propagation model. Nodes are moving as per the random waypoint mobility model. Therefore, each node independently picks a random destination and move towards those destinations with the speed limit specified in the setdest command (0 to maximum speed). When the node reaches the destination, it waits for the specified pause time period and then repeats the process.

Radio propagation range and carrier sensing range are the important parameters to be specified during the simulation, because these ranges decide the coverage area of the transmitted signals from the antenna.

- The Radio propagation range is also called as a transmission range (TX_Range) and it represents the range within which a transmitted packet can be successfully received. The radio propagation range is mainly determined by the transmission power. The large propagation range involves a large amount of power transmission. The battery power of mobile nodes in ad hoc network is one of the scarce resources, and so care has to be taken about it.

- The Physical Carrier Sensing Range (PCS_Range) is the range within which the other node detects a transmission.

- The InterFerence Range (IF_Range) is the range within which nodes in receiver mode will be interfered with a nearby transmitter and thus suffer a loss. In general, \( \text{TX\_Range} \leq \text{IF\_Range} \leq \text{PCS\_Range} \).
In this simulation, the radio propagation range is set at 200 meters with an omni directional link. The PCS_Range is set at 200 meters. The channel capacity is 2Mb/s. The multicast data streams are CBR data packets of 512 bytes. The constant flow can be achieved by generating the data at constant intervals of time. In this simulation, CBR packet interval time is set at 0.1 second.

The metrics such as Control Overhead, Packet Delivery Ratio and Normalized Routing Load are analyzed and compared with PUMA by varying the node mobility and the data transmission rate. Under mobility condition, the Hello Interval (HI) plays an important role in updating the neighbor table. For a hello interval of 0.6s, 1.0s and 1.5s, the required metrics are taken out.

4.6.2 Varying Hello Interval

Under the dynamic condition of the network, the underlying protocols must be able to manage with these dynamics efficiently. In addition to this, they have to provide good performance in terms of bandwidth and packet delivery ratio. Depending on the speed of the node mobility, link breakage may increase or decrease. One node which is neighbor to a node at one instant of time may not be a neighbor node at another instant of time. To update the neighbor table, every node broadcast one hop, hello messages at regular intervals. This time period is called as Hello Interval and controlled by a hello interval timer. If the nodes are moving at a fast rate, the neighbor table has to be updated quickly.

To achieve this quick update, the hello interval has to be reduced or the frequency of hello message broadcast has to be increased. Increasing the number of hello message in turn increases the control overhead and reduces the bandwidth utilization. Thus, a hello interval plays an important role in designing or analyzing the performance of a protocol.
In this simulation, the performance of the proposed method is analyzed by varying the various hello intervals as $HI = 0.6s$, $HI = 1.0s$ and $HI = 1.5s$. In this analysis, the node velocity is varied from 0.1 m/s to 0.9 m/s in steps of 0.1 m/s. Random way point mobility model is adapted as mobility mode.

4.6.2.1 Control overhead analysis

The impact of hello message interval on the control overhead is analyzed in this section. The amount of control overhead incurred by the proposed method for the scenario considered is tabulated in the table 4.1. The impact of hello message interval on control overhead of the proposed method is illustrated in Figure 4.10. On varying the node mobility with $HI = 1$ s, almost same amount of control overhead was used for all the node mobility conditions. When $HI = 0.6s$ is adapted, the control overhead is increased by $28.5 - 29.5 \%$ and when $HI = 1.5$ s is used, control overhead is reduced by $19.7 - 21.3 \%$ with reference to $HI = 1$ s value.

Table 4.1 Control overhead incurred for various HI under mobility in JRKBMP

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Control Overhead (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$HI = 0.6$ s</td>
</tr>
<tr>
<td>0.1</td>
<td>1.02912</td>
</tr>
<tr>
<td>0.2</td>
<td>1.010568</td>
</tr>
<tr>
<td>0.3</td>
<td>1.01008</td>
</tr>
<tr>
<td>0.4</td>
<td>1.01442</td>
</tr>
<tr>
<td>0.5</td>
<td>1.017204</td>
</tr>
<tr>
<td>0.6</td>
<td>1.004092</td>
</tr>
<tr>
<td>0.7</td>
<td>1.011192</td>
</tr>
<tr>
<td>0.8</td>
<td>1.015448</td>
</tr>
<tr>
<td>0.9</td>
<td>1.010604</td>
</tr>
</tbody>
</table>
4.6.2.2 Normalized routing load analysis

The impact of hello packet interval over the normalized routing load is tabulated in Table 4.2 and illustrated in Figure 4.11. In Normalized Routing Load analysis with HI = 1 s, NRL value is almost constant and maintained between 0.34 and 0.38. When HI = 0.6 s is used, it is increased by $28.5 - 34.5\%$. When HI = 1.5 s is used, it is decreased by $19 - 22.3\%$.

Table 4.2 Normalized routing load for various HI under node mobility in JRKBMP

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Normalized Routing Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HI = 0.6 s</td>
</tr>
<tr>
<td>0.1</td>
<td>0.563815</td>
</tr>
<tr>
<td>0.2</td>
<td>0.551485</td>
</tr>
<tr>
<td>0.3</td>
<td>0.515768</td>
</tr>
<tr>
<td>0.4</td>
<td>0.499191</td>
</tr>
<tr>
<td>0.5</td>
<td>0.489462</td>
</tr>
<tr>
<td>0.6</td>
<td>0.513516</td>
</tr>
<tr>
<td>0.7</td>
<td>0.50524</td>
</tr>
<tr>
<td>0.8</td>
<td>0.509976</td>
</tr>
<tr>
<td>0.9</td>
<td>0.528753</td>
</tr>
</tbody>
</table>
4.6.2.3 Packet delivery ratio analysis

The outcome of the packet delivery ratio analysis of the proposed method is tabulated in Table 4.3. With HI = 1 s, the packet delivery ratio analysis of varying mobility is shown in Figure 4.12. The packet delivery ratio is maintained between 92 – 99 %. The standard deviation of control overhead for different value of HI varies between 0.0064 – 0.0066.

Table 4.3 Packet delivery ratio for various HI under mobility in JRKBM P

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Packet Delivery Ratio (%)</th>
<th>JRKBM P [HI = 0.6 s]</th>
<th>JRKBM P [HI = 1.0 s]</th>
<th>JRKBM P [HI = 1.5 s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>87</td>
<td>92</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>89</td>
<td>96</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>95</td>
<td>95</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>97</td>
<td>99</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>99</td>
<td>99</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>98</td>
<td>97</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>96</td>
<td>97</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>94</td>
<td>96</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>93</td>
<td>94</td>
<td>96</td>
<td></td>
</tr>
</tbody>
</table>
4.6.3 Varying Node Mobility

The task of establishing and maintaining routes in mobile ad hoc networks is not so simple. Node mobility creates frequent unpredictable topological changes. In this analysis the value of HI is set at 1 second.

4.6.3.1 Control overhead analysis

Figure 4.13 shows the control overhead analysis of the proposed method. Almost an equal amount of control overhead was used by the proposed method for all the node mobility conditions. When compared with PUMA, only a lower amount of control overhead is used. In JRKBMP, the hello packet is used only to update the node movements and its position. Once the cluster region and cluster-heads are assigned, the data transmission takes place only through the cluster-heads. After this, the number of hello message has little impact on the overall control overhead. That is why the control overhead is maintained with little variation for different value of HI. In PUMA, the announcement signal carries all the information about the node
position as well the data to be delivered. Thus, large amount of control overhead is involved in PUMA. Even though, the number of hello packets transmitted is varying on varying the HI, node mobility with different HI, not mu 1 s, almost same amount of control overhead was used for all the node mobility conditions.

4.6.3.2 Normalized routing load analysis

In the normalized routing load analysis illustrated in Figure 4.14, NRL value with HI = 1 s is almost constant and maintained between 0.34 and 0.38. The NRL value occurred in JRKBMP for different HI value is also indicated in Figure 4.14. They are denoted as JRKBMP [0.6], JRKBMP [1.0] and JRKBMP [1.5]. Only a small variation in NRL is noticed as shown in the Figure 4.14 and given in Table 4.2. Large reduction in NRL value is achieved in the proposed method when compared with PUMA.

4.6.3.3 Packet delivery ratio analysis

The Packet Delivery Ratio analysis on varying the node mobility with HI = 1 s is shown in Figure 4.15. The packet delivery ratio is not reduced to a large extent. Its value is maintained between 94 – 99 %. On average, 1 % reduction in PDR value is noticed when compared with PUMA.

The standard deviation of control overhead for different value of HI varies between 0.0064 – 0.0066 and for PUMA it is 0.853. Table 4.4, gives the value of performance metric for various node mobility. The proposed method uses a reduced number of control overhead than PUMA and it can be verified from the table. Thus, bandwidth is saved and efficiently used without much reduction in PDR value.
Table 4.4 Impact of varying mobility on control overhead, normalized routing load and packet delivery ratio in JRKBMP

<table>
<thead>
<tr>
<th>Mobility (m/s)</th>
<th>Control Overhead (MB)</th>
<th>Normalized Routing Load</th>
<th>Packet Delivery Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JRKBMP PUMA</td>
<td>JRKBMP PUMA</td>
<td>JRKBMP PUMA</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7356 4.9768</td>
<td>0.3814 2.5580</td>
<td>92.78 94.76</td>
</tr>
<tr>
<td>0.2</td>
<td>0.7193 4.9897</td>
<td>0.3609 2.4735</td>
<td>96.12 99.49</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7182 5.7981</td>
<td>0.3626 2.8963</td>
<td>95.77 98.99</td>
</tr>
<tr>
<td>0.4</td>
<td>0.7163 4.4410</td>
<td>0.3527 2.1793</td>
<td>99.40 99.00</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7225 4.1728</td>
<td>0.3450 2.0426</td>
<td>100.00 99.25</td>
</tr>
<tr>
<td>0.6</td>
<td>0.7167 5.7253</td>
<td>0.3609 2.9812</td>
<td>97.22 95.93</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7153 6.7651</td>
<td>0.3608 3.2787</td>
<td>97.04 98.77</td>
</tr>
<tr>
<td>0.8</td>
<td>0.7149 6.1307</td>
<td>0.3609 3.0484</td>
<td>96.97 98.94</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7175 6.0384</td>
<td>0.3722 3.1118</td>
<td>94.13 95.95</td>
</tr>
</tbody>
</table>

Figure 4.13 Node mobility Vs Control overhead
Figure 4.14 Node mobility Vs Normalized routing load

Figure 4.15 Node mobility Vs Packet delivery ratio

4.6.4 Varying Data Transmission Rate

Increase in data transmission rate implies the increase in data packet transmission and delivery. If the data transmission rate increases, control bytes used for proper data delivery also increases. Effectiveness of the proposed method is analyzed by varying the data transmission rate from 10 kb/s to 60 kb/s in steps of 10 kb/s. The proposed method with HI = 1 s is compared with PUMA. The results of this analysis are tabulated and given in Table 4.5.
Table 4.5  Impact of varying data transmission rate on control overhead, normalized routing load and packet delivery ratio in JRKBMP

<table>
<thead>
<tr>
<th>Data Transmission Rate (kb/s)</th>
<th>Packet Delivery Ratio (%)</th>
<th>Control Overhead (MB)</th>
<th>Normalized Routing Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JRKBMP</td>
<td>PUMA</td>
<td>JRKBMP</td>
</tr>
<tr>
<td>10</td>
<td>100.00</td>
<td>98.99</td>
<td>0.42534</td>
</tr>
<tr>
<td>20</td>
<td>99.95</td>
<td>100.00</td>
<td>0.47602</td>
</tr>
<tr>
<td>30</td>
<td>99.66</td>
<td>100.00</td>
<td>0.52993</td>
</tr>
<tr>
<td>40</td>
<td>97.42</td>
<td>99.49</td>
<td>0.58126</td>
</tr>
<tr>
<td>50</td>
<td>90.99</td>
<td>99.59</td>
<td>0.63189</td>
</tr>
<tr>
<td>60</td>
<td>80.60</td>
<td>96.09</td>
<td>0.68110</td>
</tr>
</tbody>
</table>

4.6.4.1  Control overhead analysis

The control overhead analysis is shown in Figure 4.16. As expected, the control overhead is increasing with the increase in data transmission rate for both the proposed method and PUMA. At 60 kb/s, control overhead is increased by a large extent for PUMA and only 60 % for the proposed method from control overhead at 10 kb/s. With the proposed method, control overhead is reduced by 43.6 - 85.1 % of the control overhead used for PUMA. The standard deviation of control overhead of the proposed method is 0.096 and for PUMA it is 1.4.

4.6.4.2  Normalized routing load analysis

In the normalized routing load analysis shown in Figure 4.17, NRL value is decreasing with the increase in data transmission rate. The NRL value of the proposed method is decreased to a large extend from the NRL value of PUMA.
4.6.4.3 Packet delivery ratio analysis

Figure 4.18 illustrates the impact of data transmission rate on the packet delivery ratio. When the proposed method is adopted, PDR is reduced by 1 – 10% compared to PDR of PUMA with the increase in data transmission rate up to 50 kb/s.

![Figure 4.16 Data transmission rate Vs Control overhead](image)

![Figure 4.17 Data transmission rate Vs Normalized routing load](image)
98

Figure 4.18 Data transmission rate Vs Packet delivery ratio

The proposed method uses almost a constant and lower control overhead compared to PUMA for the scenario considered. Even for a larger number of data packet transmission and delivery, the proposed method has a lower value of the normalized routing load.

4.7 CONCLUSION

This chapter proposed a protocol which supports the multicast in mobile ad hoc networks. The proposed Join Request Knowledge Based Multicast Protocol for MANET is a cluster based protocol in which a complete multicast path for data delivery is established on completion of the source and receivers join procedure. Based on the Weighted Cluster Algorithm, cluster formation and cluster-head selection are accomplished in the proposed method. The novelty of the proposed method is its source and receivers joining procedure which avoids the unnecessary flooding of control packets throughout the network. Thus, control overhead is maintained at the same and reduced level. The procedures to be followed during the different sections of the proposed method are illustrated through flowcharts.
Control overhead, normalized routing load and packet delivery ratio are used as a performance metric. The performance of the proposed method is analyzed by varying the hello packet interval, node mobility speed and data transmission rate. The results of the analysis shows, the proposed method is effective in utilizing lower control overhead without much degradation in packet delivery ratio. An extension is made in JRKBMP to have more than one source, and proposed a novel multi-source multicast protocol for MANET in the next chapter.