CHAPTER 5

AGENT ASSISTED FRAMEWORK FOR MCQMR

5.1 INTRODUCTION

In MANETs, most of the multicast routing protocols establish routes on-demand where a considerable amount of time is spent in finding the routes from source to all the group members. This introduces additional delay to the overall end-to-end delay. Also, in the event of link failure, the route rediscovery mechanism takes a considerable amount of time and leads to violation of delay constraint in packet delivery. Moreover, in MANETs, the mobile nodes are resource constrained. They have very limited amount of bandwidth, battery power and memory. In multicast routing, for the multicast tree initialization and maintenance, the mobile nodes need to exchange a lot of control information. These control packets consume a large portion of useful bandwidth, resulting in violation of bandwidth constraints in the network. The mobile agent technology offers a very promising solution to reduce the route setup latency and control overhead. Mobile agents have the ability to support asynchronous communication and flexible query processing. This is because user tasks can be delegated to mobile agents, and whenever the agent feels that there is communication availability it will roam the network and fulfill the task assigned by its user (Griss et al 2001). In this way, a mobile node requires less communication connectivity than the traditional client/server approaches.
Another important advantage of mobile agents in wireless networks is that they can reduce the network traffic. The mobile nodes running on battery power do not have enough power to run complex routing protocols necessary in ad hoc networks. Hence an alternative is to use mobile agents to discover the topology with QoS routing metrics and generate a routing table with those metrics. This routing table is used for proactive routing operations, thus saving battery power of mobile nodes and reducing routing complexity and network traffic.

The key idea in developing an agent assisted multi-constrained QoS multicast protocol is based on the agent based multicast routing strategies proposed by Wang et al (2001), Shekar et al (2005), Manvi et al (2008, 2011), Krishna Deshpande (2012). In these routing protocols, the mobile agents roam around the network and collect the QoS metrics and update the routing table of all mobile nodes. This proactive mechanism reduces the control overhead to some extent. Hence, in this research work an agent assisted framework has been devised to improve the efficiency of multi-constrained QoS multicast routing scheme in terms of reducing the route setup latency and minimizing the control overhead.

5.2 AGENT ASSISTED MULTI-CONSTRAINED QoS AWARE MULTICAST ROUTING PROTOCOL

The Agent-MCQMRP is a hybrid protocol which uses mobile agents for route discovery between source and multicast receivers. This protocol pre-computes the paths based on the QoS routing metrics required by the applications. It uses a routing table with reachability information to forward the packets from the source to multicast receivers. The goals of the Agent-MCQMRP are the following.
To utilize the mobile agents to discover the topology and QoS status of each node in the network and distribute/updates them periodically to other nodes,

- To discover the route in a definite path instead of flooding the RREQ packets, thereby reducing control overhead.

The agent based QoS routing scheme identifies a set of multiple paths from the source to multicast receivers and selects a path which meet the multiple QoS requirements of a particular application. The QoS constraints considered here are the same as described in section 3.3 in chapter 3. \(B_T, D_T, J_T\) and \(L_T\) are the QoS constraints of bandwidth, delay, jitter and packet loss rate respectively of a multicast tree.

### 5.2.1 Routing Agency

Every node in a network maintains a routing agency. It employs a set of static and mobile agents for QoS routing. The mobile agents are simple packets that are generated at selected nodes in the network. The mobile agents move across the network and collect information such as node id, and QoS values of the nodes. This information is used in updating the reachability table. The data structure of the mobile agent comprises of the fields as shown in Figure 5.1.

<table>
<thead>
<tr>
<th>agent.ID</th>
<th>agent.type</th>
<th>source_id</th>
<th>destination_id</th>
<th>hopcount</th>
<th>QoS values</th>
<th>max-hop</th>
<th>history-list</th>
</tr>
</thead>
</table>

**Figure 5.1 Data Structure of the Mobile Agent**

- **agent.ID**: a unique id of the agent.
- **agent.type**: the type of agent in the route discovery and maintenance process. It distinguishes between
Node Manager Agent, Route Discovery Agent and Link Management Agent.

- **source_id**: source node address.
- **destination_id**: multicast receiver address.
- **hopcount**: calculates the number of hops, the agent has traversed from the source.
- **QoS values**: the residual bandwidth, delay, jitter and packet loss values for each link.
- **max-hop**: the maximum number of hops allowed to be traversed for a mobile agent.
- **history-list**: the nodes-visited-stack, contains the IDs of nodes through which the agent passes.

To perform the topology discovery with QoS metrics, each node in a network maintains a routing agency which consists of a set of static and mobile agents as shown in Figure 5.2. An agency consists of a static Node Manager Agent (NMA), a static Link Management Agent (LMA), a mobile Route Discovery Agent (RDA) for setting up a feasible path, and a Knowledge Base. The above mentioned agents are defined as follows.
Figure 5.2 Agency for QoS Routing

**Knowledge Base (KB)** contains the information about the neighbours and QoS status information contained within it, such as residual bandwidth, delays and jitters to neighbours and percentage of packet losses on links that connect a node. This information can be read and updated by the agents. The knowledge base also contains a routing table with a list of nodes reachable from the current node, QoS routing metrics, time of update etc.

**Node Manager Agent (NMA)** is a static agent running at each node to serve the QoS applications and also supports route finding operations when the node is acting as an intermediate node. This agent is responsible for creation of Link Management Agent and Route Discovery Agent and the knowledge base in the agency. It synchronizes the activities of all other agents. All the operations, like communication, updating or reading the knowledge base and so on take place with the permission of NMA.
**Link Management Agent (LMA)** is a static agent that monitors the links of one-hop neighbours of a node by exchanging hello packets periodically. LMA agents are created for each link by the NMA. This LMA agent computes the residual bandwidth, delays, jitters and packet losses for each link and updates the knowledge base at regular intervals as shown in Table 5.1. All the parameters are computed within a given continuous time window.

<table>
<thead>
<tr>
<th>Link</th>
<th>Bandwidth (Mbps)</th>
<th>Delay (ms)</th>
<th>Jitter (ms)</th>
<th>Packet Loss Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Route Discovery Agent (RDA)** is a mobile agent invoked by the node manager agent to find QoS routes on demand for an application. This mobile agent migrates from one mobile node to another in a loop-free manner to discover the connectivity of the ad-hoc networks as well as fuse QoS routing metric information at the visited node to update the routing tables.

5.2.2 **Agent based Topology Discovery with QoS Metrics**

Agent-MCQMRP uses mobile agents that can move in the ad hoc network to discover the topology, and collect/update the QoS routing metrics at every node as they move towards the targeted destination. At each node, the NMA creates the LMA and RDA agents to gather the topology information. The mobile RDA migrates from one node to another in a loop-free manner. It collects the neighbourhood connectivity and the QoS status information of neighbour nodes from the knowledge base of the visited nodes and marks the nodes it has visited. Nodes marked as visited will not be considered for migration, thus preventing loop movement of RDAs. To avoid
unnecessary wastage of bandwidth consumed by agent movements, agent migrations are limited to a certain number of hops. It migrates to the unvisited neighbour node of a residing node that has the bandwidth above the threshold bandwidth, $B_T$ value. During migration, the agent updates the collected information of the already visited nodes in the knowledge base of the visiting node with the permission of NMA. The mobile agent updates the routing table with the reachability information, i.e. the nodes that are reachable from the current visited node. Figure 5.3 shows the topology discovery performed by the mobile agents.

![Figure 5.3 Mobile Agent based Topology Discovery](image)

In the above Figure 5.3, the static NMA creates the static LMA and the mobile RDA at the source node ‘S’. Static LMA collects the neighbourhood information and updates the QoS values of neighbourhood in the knowledge base. RDA collects the neighbourhood information from the knowledge base of mobile node ‘S’. If the available bandwidth of the neighbour node is above the threshold bandwidth $B_T$, then it migrates to the unvisited node. In the above topology, the neighbour node ‘A’ satisfies the bandwidth constraint and hence it migrates to mobile node A. During migration it updates the routing table of the visited node. The structure of the routing table is shown in Table 5.2.
Table 5.2  Routing Table with Reachability Information in Agent-MCQMRP

<table>
<thead>
<tr>
<th>Visited node</th>
<th>Update time</th>
<th>List of reachable nodes</th>
<th>QoS values</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>t_S</td>
<td>S</td>
<td>TD_{SA}, TJ_{SA}, TL_{SA}</td>
</tr>
<tr>
<td>A</td>
<td>t_A</td>
<td>S, A</td>
<td>TD_{SA} + TD_{AB}, TJ_{SA} + TJ_{AB}, TL_{SA} * TL_{AB}</td>
</tr>
<tr>
<td>B</td>
<td>t_B</td>
<td>S, A, B</td>
<td>TD_{SA} + TD_{AB} + TD_{BD}, TJ_{SA} + TJ_{AB} + TJ_{BD}, TL_{SA} * TL_{AB} * TL_{BD}</td>
</tr>
<tr>
<td>D</td>
<td>t_D</td>
<td>S, A, B, D</td>
<td>TD_{SA} + TD_{AB} + TD_{BD}, TJ_{SA} + TJ_{AB} + TJ_{BD}, TL_{SA} * TL_{AB} * TL_{BD}</td>
</tr>
</tbody>
</table>

The routing table includes the information of visited nodes; the routing table updates time and a list of nodes that are reachable from the current node and their QoS values. The route update time is an important metric and it ensures the freshness of the route information brought by the mobile agent. TD, TJ and TL are the accumulated values of the delay, jitter and packet loss rate respectively from the source to the current node visited by a mobile agent. This table is used for proactive routing and the process of route establishment using this route table is as follows.

**Route Establishment**

When a multicast session starts at the given source node, route request takes place. The source node checks the routing table for reachability of multicast group members, identified by the multicast destination address. If a route is found in the routing table to a multicast receiver, then it checks for other QoS constraints such as delay, jitter and packet loss rate. The QoS constraints \( D_T, J_T \) and \( L_T \) are compared with the TD, TJ and TL values available in the routing table for a route. If all the QoS constraints are
satisfied, then the source node generates the RREQ packets and sends to only those neighbour nodes through which destination nodes are reachable. When the RREQ reaches an intermediate node, it also checks reachability of destination nodes from its routing table and forwards the RREQ packet to only neighbours through which destination nodes are reachable. This process continues until RREQ packet reaches all destination nodes. Thus, the RREQ packet takes a definite path from the source to all destinations, thereby reducing the route discovery latency and hence over all end-to-end latency. As the route request packets takes a definite path provided by mobile agents, the resource reservation can be made during this phase. This essentially avoids the use of two passes: QoS route determination and then resource reservation.

The route establishment of Agent-MCQMRP is explained with the following example shown in Figure 5.4. Consider the QoS requirements of an application \( Q = \{4\text{Mbps, 8ms, 2ms, 0.2}\} \). i.e. the minimum bandwidth requirement \( B_T \) is 4 Mbps, maximum delay requirement \( D_T \) is 8ms, maximum jitter requirement \( J_T \) is 2ms and maximum packet loss requirement \( L_T \) is 0.2 respectively. The maximum number of hops (max-hop) for agent migration is restricted to 3. In Figure 5.4, the QoS values are specified in each link of a mobile node.
Here, the mobile agents RDA$_1$ and RDA$_2$ are generated by NMA at nodes 5 and 7 respectively. The mobile agent RDA$_1$ collects the neighbourhood information from the knowledge base of node 5. It checks the available bandwidth of neighbour nodes and finds that node 3 has the bandwidth above the threshold bandwidth of 4 Mbps and hence it migrates to node 3. It updates the routing table of node 3 by making an entry that node 5 is reachable through node 3 and also updates the QoS values. The number of hops visited is set to 1 in the data structure of the mobile agent. Now it checks the bandwidth availability of node 2 and it is satisfied. Hence the RDA$_1$ migrates to node 2 and updates the routing table of node 2 by making an entry that nodes 3 and 5 are reachable through node 2 and number of hops visited is given an increment of one and set to 2. The QoS metrics delay, jitter and packet loss rate are summed up and updated in the routing table. Similarly it migrates to node 1 and updates the routing table of node 1. The...
list of reachable nodes is updated as 2, 3, and 5 and the QoS values are also updated in the routing table of node 1. Now the number of hops travelled by the RDA_1 is reached to the maximum limit of 3 and hence disposes itself at node 1. The routing table updates by RDA_1 and RDA_2 are shown in the Table 5.3 and Table 5.4 respectively.

### Table 5.3  Routing Table Updates by RDA_1

<table>
<thead>
<tr>
<th>Visited node</th>
<th>Update time</th>
<th>List of reachable nodes</th>
<th>QoS values</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>t_1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>t_2</td>
<td>3, 5</td>
<td>2, 0.5, 0.1</td>
</tr>
<tr>
<td>2</td>
<td>t_3</td>
<td>2, 3, 5</td>
<td>3, 0.7, 0.01</td>
</tr>
<tr>
<td>1</td>
<td>t_4</td>
<td>1, 2, 3, 5</td>
<td>5, 0.9, 0.001</td>
</tr>
</tbody>
</table>

### Table 5.4  Routing Table Updates by RDA_2

<table>
<thead>
<tr>
<th>Visited node</th>
<th>Update time</th>
<th>List of reachable nodes</th>
<th>QoS values</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>t_1</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>t_2</td>
<td>4, 7</td>
<td>3, 0.2, 0.01</td>
</tr>
<tr>
<td>1</td>
<td>t_3</td>
<td>4, 1, 7</td>
<td>6, 0.3, 0.001</td>
</tr>
</tbody>
</table>

In the Figure 5.4, mobile node 1 is the source and hence it starts the multicast session with the nodes 3 and 7 as the multicast receivers. The mobile node 1 checks the routing table for reachability of these nodes. It is learnt from the routing table that node 3 is the neighbour of node 2 and node 7 is the neighbour of node 4 and hence these nodes can be reached through the nodes 2 and 4. The node 1 then sends RREQ packet to neighbour nodes 2 and
4 with multicast address for nodes 3 and 7 added to IP header field. When node 4 receives RREQ message with destination 7 and 3, it checks the routing table and learns that node 7 is the neighbour to node 4 and forward the RREQ packet to node 7. Similarly, node 2 checks the routing table and learns that node 3 is the neighbour to node 2 and hence forwards the RREQ to node 3. When RREQ reaches nodes 3 and 7, the nodes check the destination IDs and mark as multicast receivers. Both the nodes 3 and 7 send back the RREP packets to the source via intermediate nodes, which update their multicast table for multicast routing to take place.

On receiving the RREP packets, the intermediate nodes reserve the required resources and the source starts the data transmission. In Agent-MCQMRP, since the RREQ packets travel on a definite path instead of flooding, the control traffic is very much reduced compared to on demand multicast routing protocol MAODV that uses flooding as basis. If no entry exists in the routing table for the destination node, then the protocol follows broadcast route discovery mechanism of MAODV. If a node movement is detected by any of the mobile nodes due to link failure, it will send a broadcast message to all the nodes to initiate a route rediscovery procedure.

5.3 Agent-MCQMRP ALGORITHM

The following Agent-MCQMRP algorithm presents the pseudo code for the agent based topology discovery with QoS metrics.

**Input:** N = Number of mobile nodes, nhop = Number of hops allowed for mobile agent migration, BT = Threshold Bandwidth requirement at a visited mobile node.

**Output:** List of reachable nodes and their QoS metrics for each node.
Variables: \( RDA_i \) = RDA generated at mobile node \( i \), where \( 1 \leq i \leq N \),
\( \text{visited}_\text{nodes}[0 \ldots \text{nhop}] \) = List of mobile nodes visited by \( RDA_i \)

Begin

For \( i = 1 \) to \( N \) do

// Discovery of one-hop neighbours and their QoS values

Begin

NMA at node ‘i’ triggers LMA to discover one-hop neighbours and their available QoS values by sending HELLO messages periodically and store them in the knowledge base.

End

For \( j = 1 \) to \( N \) do

// Initialization of mobile agent’s data structure

Begin

NMA at node ‘j’ generates RDAs to roam in the network by initializing their knowledge base such as \( \text{nhop} \) = maximum hops, \( \text{hop} = 0 \), \( \text{visited}_\text{nodes}[..] = 0 \), \( \text{QoS values}[..] = 0 \).

End

For each \( RDA_i \) Do

// Routing table updates and mobile agent migration

Begin

\( \text{Visit}_\text{nodes}[0] = i; \)

For \( \text{hop} = 1 \) to \( \text{nhop} \) Do

Begin
// RDA<sub>i</sub> searches the knowledge base of node ‘i’ for one-hop neighbour node ‘j’ with Bandwidth<sub>Avail</sub> > B<sub>T</sub>

If neighbour node’s Bandwidth<sub>Avail</sub> > B<sub>T</sub>, Begin

RDA<sub>i</sub> migrates to neighbour node ‘j’

visited_nodes [hop] = j;

hop = hop + 1; RDA<sub>i</sub> = RDA<sub>j</sub>;

TD [j] = TD [j] + TD [i];

TJ [j] = TJ [j] + TJ [i];

TL [j] = TL [j] * TL [i];

// update the knowledge base of node ‘j’ with
// reachability information

End

else

RDA<sub>i</sub> disposes itself at node ‘i’

End

End

5.4 SIMULATIONS RESULTS AND DISCUSSION

To evaluate the performance of Agent-MCQMRP protocol, two sets of simulations are conducted. The first set measures the performance by varying the number of multicast group size whereas the second set measures the performance by varying the mobility speed.

Simulation Model

To study the performance of Agent-MCQMRP, the protocol is simulated in a discrete event simulator namely, Network Simulator 2 (NS-2.26). The mobile agent population and the history size play an
important role in balancing the network load and information gathering capability of the mobile agents. The maximum number of hops an agent can travel, that is Hops/Agent (HPA) is set to 15. A new agent will be generated if a node does not receive agent packet within the specified interval of the time or if the agent has traveled its maximum hop distance. The mobile agent hopping time also has an impact on the network load. A hopping time of 10ms is used in the simulation. The other parameters considered for simulation are shown in Table 5.5.

**Table 5.5 Simulation Environment for Agent-MCQMRP**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Space</td>
<td>1500 m * 300 m</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>900s</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>60</td>
</tr>
<tr>
<td>Number of runs</td>
<td>10</td>
</tr>
<tr>
<td>MAC layer</td>
<td>802.11</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250m</td>
</tr>
<tr>
<td>Mobility speed</td>
<td>0-20 m/s</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random way point</td>
</tr>
<tr>
<td>Multicast Group size</td>
<td>5, 10, 15, 20, 25, 30, 35, 40, 45</td>
</tr>
</tbody>
</table>

**Results and Discussion**

The proposed Agent-MCQMRP algorithm is simulated to test the effectiveness in terms of connectivity, end-to-end latency, agent overhead and agent response time.

**Average End-to-end delay**

The simulation results of Agent-MCQMRP obtained for end-to-end delay is compared with the MAODV and Agent Based Multicast Routing
Scheme (ABMRS) proposed by Manvi et al (2008). ABMRS makes use of agent technology to find the multicast routes and to create the backbone for a reliable multicast routing. All three multicast routing protocols, Agent-MCQMRP, ABMRS and MAODV are compared by varying the mobility and the multicast group size in the network. The end-to-end delay includes buffering delay and queuing delay at each node’s interface queue and delay due to retransmission and propagation. The average end-to-end latency for Agent-MCQMRP is less compared to ABMRS and MAODV. The reduced latency is less due to the availability of reachability information to other nodes. This is brought about by the mobile agents roaming around the network. The route discovery delay is also reduced as the route request is sent on the predetermined path from the source to destinations. This is possible by using the reachability information available in the routing table, which in turn reduces the overall latency of the protocol. In ABMRS, the mobile agents are used to find the one-hop neighbour reliable node in terms of higher bandwidth only and a complete list of reachability information is not available in each node. Hence, the route request takes place by flooding which in turn increases the average end-to-end delay.

The average end-to-end delay increases as the mobility speed increases as shown in Figure 5.5. This is because of the possibility of more link failures due to high speed movement of the mobile nodes. Agent-MCQMRP performs better than ABMRS and MAODV. On an average, Agent-MCQMRP shows 33.47% and 57.7% reduction in end-to-end latency compared to ABMRS and MAODV protocols respectively. This is due to reachability and QoS status information is made to be available in all nodes before the route request takes place in Agent-MCQMRP scheme.
The end-to-end latency increases as the group size increases when the mobility speed is kept constant as shown in Figure 5.6 and Figure 5.7. When the mobility speed is 1m/s, on an average the Agent-MCQMRP gives 6.43% and 17.3% improvement in end-to-end latency compared to ABMRS and MAODV protocol respectively. When the group size is increased beyond 35 nodes the average end-to-end latency decreases, because the same nodes may be used for forwarding the data to more than one destination. When the mobility is increased to 10 m/s, still the proposed Agent-MCQMRP performs better than MAODV and ABMRS due to the mobile agents’ topology discovery mechanism.
Figure 5.6  End-to-end Latency with varying Group Size (Mobility: 1m/sec)

Figure 5.7  End-to-end Latency with varying Group Size (Mobility: 10m/sec)
Connectivity

The connectivity of MANET is defined as the ratio of the number of QoS applications reachable (that have the paths) to the total number of QoS applications generated. The percentage of connectivity rises with increase in number of RDAs generated at each mobile node and large number of hops traveled by a RDA. With high mobility of nodes, the mobile agent movement will become low which decreases the percentage of connectivity. For 50 number of QoS requests, the connectivity of Ad hoc network ranges from 40% to 100% with different number of RDAs and hops/agent. 100% connectivity indicates complete topology discovery and availability of routing information at every node. It is observed from the Figure 5.8, the best choice of RDAs and number of hops travelled by an agent (HPA) is 6 and 12 respectively to achieve 100% of routing information.

Figure 5.8 Percentage of Connectivity in Agent-MCQMRP
Agent Overhead

Agent overhead is defined as the average bandwidth required by a RDA during topology discovery and route information fusion, i.e. it is defined as ratio of bandwidth required by each RDA to the number of RDAs generated. The agent overhead increases with increase in the number of RDAs generated and number of hops traveled by a RDA. In Agent-MCQMRP, even if there is no communication, the agents would still be traversing the network and update the routing tables. The bandwidth consumption is increased when more number of mobile agents generated per node and the number of hops travelled by the agent increase as shown in Figure 5.9.

![Figure 5.9 Agent Overhead of Agent-MCQMRP](image-url)
Topology Discovery Time

Agent topology discovery time is defined as the time taken by mobile agents to roam the ad-hoc network to discover the topology and fuse QoS routing information. It includes migration and execution times at each of the nodes. The topology discovery time increases with increase in the number of RDAs and the number of hops traveled by each RDA. Here the HPA is set to 15, and hence the mobile agents travel to more number of hops and update the reachability information in the routing table at each node. It is observed from the Figure 5.10, when mobility is kept at 1m/s, maximum time is taken by 14 HPAs to roam the ad hoc network and to discover the topology compared to lesser numbers. When mobility is increased, the topology discovery time also increases due to more frequent link breaks. When mobility is 10 m/s, the topology discovery time increases as shown in Figure 5.11.

![Figure 5.10 Topology Discovery Time of Agent-MCQMRP with Mobility 1m/sec](image)
A hybrid agent assisted multi-constrained QoS multicast routing protocol is developed and presented in this chapter. In this protocol, the mobile agents move around the network and provide the current topology information and other QoS values, which helps the mobile nodes for taking efficient routing decision. The protocol is aimed at reducing the flooding of route request packets by using the pre-computed paths between the nodes. The primary objective of the agent based model is to make the mobile nodes to be topology aware. The results obtained for connectivity show a significant improvement in finding the QoS routes. Although this method requires an extra cost for processing the agents, the benefits would be gained in terms of better end-to-end latency.

Figure 5.11  Topology Discovery Time of Agent-MCQMRP with Mobility 10 m/s

5.5 SUMMARY