CHAPTER 3

OPTIMIZED DELAY CONSTRAINT ROUTING PROTOCOL

3.1 INTRODUCTION

In this chapter, the system architecture and the delay model for heterogeneous network are analyzed. A new Quality of Service (QoS) protocol called QAOMDV, which is an extension of AOMDV that considers delay as a QoS parameter is also proposed.

3.2 HETEROGENEOUS NETWORK

The proposed system enables content dissemination over heterogeneous network consisting of wired network, base station (gateway) and infrastructure-less wireless networks. In wired-cum-wireless networks, Mobile Hosts (MHs) (Figure 3.1) can roam between different Basic Service Sets (BSSs), which are connected to a wired backbone (infrastructure). The wired infrastructure is an IEEE 802 style Ethernet LAN, and it is connected to the Internet through a gateway router. Wired and wireless networks are interconnected through an Access Point (AP) or base station. During the node mobility, end-to-end Quality of Service (QoS) guarantee in the context of wireless Internet means that if an MH moves between different cells, its bandwidth should be allocated in the new cell and freed from the old cell. In a wireless LAN, it is assumed that all MHs are within the broadcast region of the AP. If a self-organizing ad hoc network is attached to the wired infrastructure through the AP, then MH sends message to AP through several
hops, incurring larger signaling overhead. The proposed work is more appropriate for infrastructure based Wireless Local Area Networks (WLAN).

![Diagram of a heterogeneous network](image)

**Figure 3.1 Heterogeneous Network**

### 3.3 EXAMPLE NETWORK

A graph $G(N, E, D)$ is used for representing the heterogeneous network where $N$ is the set of wired and mobile nodes in the network, $E$ is the set of edges and $D$ is the delay. It is denoted by $(n_i, n_j) \in N$, where the link is formed by the two nodes $n_i$ and $n_j$. The proposed work considers delay QoS parameter $D$. It represents the estimated time during which the communication between the two nodes forming the link is always maintained. Thus, the delay QoS constraint $D(n_i, n_j)$ is incorporated for every link $(n_i, n_j)$. Figure 3.2 illustrates a typical network. If the route discovery
process starts then the source node (S) constructs the special RREQ packet and broadcasts it through the network. Nodes B, F and C check their delay value to the QoS constraints. Let be 25 seconds. When the packet is transmitted, the source node checks its neighbours for their delay value. The nodes B and C satisfy the condition, but in node F delay value is 3, which is less than the threshold value of 25s. Hence the bandwidth constraint is not satisfied and the source node sends the packets only to nodes B and C. No RREQ packet is forwarded to node F. The delay and bandwidth field of the RREQ is updated in the routing table. Node B now checks its neighbours and only when the QoS constraints are satisfied, it forwards the packet to node E and F. The packet is forwarded by node E only. The delay is computed as the difference between two nodes receiving time. Every node is selected, based on this condition.

![Diagram of QoS Routing Network](image)

**Figure 3.2 Example of QoS Routing Network for Delay**

The major goal of this proposed protocol is to maximize the chance of finding an optimal path. The route selection information is collectively
utilized to make hop-by-hop selection. This protocol not only focuses on the QoS requirement, but also considers the optimality of route selection in terms of delay.

### 3.4 NETWORK MODEL

A heterogeneous network is represented by a directed graph \( G = (V, E, D) \), where \( V \) is the set of wired and wireless nodes and \( E \) is the set of links between different nodes. The parameter of cost \( c(i,j) \in E \) is considered for each link \( (i,j) \in E \). Also, delay \( d_{ij} \) is considered, which is additive QoS metric for multiple path \( P = 1, 2, \ldots, p \). The problem is to find the shortest path from source to destination with minimum delay.

In this work the delay for heterogeneous network is modeled using the M/M/1 queuing model. Some challenges posed include:

- Maximizing network resource utilization, and minimizing operational costs, delay and bandwidth, and
- Identifying the QoS associated with fixed, mobile and core networks.

The logical modeling framework is illustrated in Figure 3.3, where multiple paths are established from source to destination nodes. Fixed networks are connected to a wireless network through a router and access point. To model each multiple path, M/M/1 tandem network is selected. Data packets arrive according to a Poisson process and must be routed to parallel queues with exponential service time distributions. The objective is to minimize the expected average delay of end-to-end processing.
**Figure 3.3 Wired-cum-Wireless Network**

This network model has $k$ paths (i.e. $k = 1, 2, \ldots, N$). The network is composed of $S_k$ source node, $H_k$ intermediate nodes and $D_k$ destination nodes. The $k$-th path is modeled as a network where $S_k$ queues are connected in tandem.

**Definition 1**: A Heterogeneous network is a collection of wired and mobile networks, where $i = 1$ represents wired network and $i = 2$ represents mobile network. $N$ represents the number of paths. Each network consists of $M$ resources. The response time of a node is given by

$$
\mu = \sum_{j=1}^{M} \sum_{k=1}^{N} \mu_{i,j,k} + \sum_{j=1}^{M} \sum_{k=1}^{N} \mu_{2,j,k}; i = 1, 2, 3, 4, 5
$$

(3.1)
\( \mu_{i,j,k} \) represents the response of resource \( j \) in a network \( i \) along the path \( k \).

The set \( \mu_i = \mu_1, \mu_2 \) represents the wired and mobile networks respectively. Different mobility models including Random Waypoint (MRW), Reference Point Group Mobility (MRPM), Tactical Indoor Mobility Model (MTIMM), Manhattan (M_M) and Self similar Least Action Walk model (MSLAW) were used in mobile part of heterogeneous networks. It is represented by the second parameter in \( \mu_2 \), namely, \( j (l) \) as \( j_1 = MRW, j_2 = MRPM, j_3 = MTIMM, j_4 = M_M \) and \( j_5 = MSLAW \). In wired network no mobility model is used and hence \( l \) need not be considered.

**Definition 2**: The set \( \lambda_j = \lambda_1, \lambda_2, \lambda_3, ..., \lambda_M \) represents the distribution of the load across a set of the resources. Total traffic sent from the source is \( \lambda_{i,k} = \lambda_k, \forall j = 1, 2, ..., S_k \). If the traffic flow with average Poisson arrival rate \( \lambda \) exists between the source to destination pair, then sub flows are poisson distributed among the \( N \) disjoint paths in parallel.

\[
\lambda = \sum_{l=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} \quad (3.2)
\]

Assume that the delay model for heterogeneous network is a M/M/1 Queuing model. According to the central limit theorem, the end-to-end path delay is calculated by a large number of independent delays in the intermediate queues which is normally distributed. The expected delay \( D_{i,j,k} \) for completion of tasks of the network \( j \) of resource \( j \) at path \( k \) is

\[
D_{i,j,k} = \frac{1}{\mu_{i,j,k} - \lambda_{i,j,k}} \quad (3.3)
\]

The mean value of the end-to-end delay of path \( k \) is
\[ D_{\text{min}} = \frac{1}{\lambda} \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} D_{i,j,k} = \frac{1}{\lambda} \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \frac{\mu_{i,j,k}}{\lambda_{i,j,k} - \lambda_{i,j,k}} \]  

(3.4)

Note that \( \mu_{i,j,k} \) is set of resources from j-th server along with path-k of i-th network and the total traffic is denoted by \( \lambda_{i,j} = \lambda_j, \forall j = 1, 2, 3 \ldots S_k \).

The delay is optimized by minimizing the average end-to-end delay \( D_{\text{min}} \) for the computation task of M resources in a heterogeneous network. Minimizing optimal resource allocation is denoted by \( \lambda^0 \).

### 3.4.1 Delay Modeling For Multipath Routing

The main goal of this research work is to find a minimum of the average delay \( D_{\text{min}} = D(\lambda/\mu) \) from equation (3.4). The solution for multipath routing is obtained from Karush-Kuhn-Tucker theory (Mostafavi 2010).

The optimal traffic dispersion according to Lagrangian multipliers is

\[ L(\lambda, \gamma) = D_{\text{min}} - \gamma_0(\sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} - \lambda) - \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \gamma_i \lambda_{i,j,k} \]  

(3.5)

Substituting \( D_{\text{min}} \) value from Equation (3.4)

\[ L(\lambda, \gamma) = \frac{1}{\lambda} \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \frac{\lambda_{i,j,k}}{\mu_{i,j,k} - \lambda_{i,j,k}} - \gamma_0 \left( \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} - \lambda \right) - \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \gamma_i \lambda_{i,j,k} \]  

(3.6)
The optimal traffic dispersion is estimated according to Lagrange Multiplier. \( \gamma = (\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_N) \) is a set of Lagrange Multipliers. Here \( \gamma_0 \) is the Lagrange Multiplier which deals with the constraint \( \sum_{k=1}^{N} \lambda_k = \lambda \). The multiplier \( \gamma_k, 1 \leq k \leq N \) is used to find the inequality constraints \( \lambda_k \geq 0 \) The Karush Kuhn Tucker equations are applied to find the optimum value \( \lambda_{\min} \).

Differentiating \( L(\lambda, \gamma) \) with respect to \( \lambda_{i,j,k} \),

\[
\frac{dL(\lambda, \gamma)}{d\lambda_{i,j,k}} = \frac{1}{\lambda} \left[ \frac{1}{\mu_{i,j,k} \cdot \lambda_{i,j,k}} - \frac{\lambda_{i,j,k}}{(\mu_{i,j,k} \cdot \lambda_{i,j,k})^2} \right] - \gamma_0 - \gamma_j \tag{3.7}
\]

The global constraint for complete load distribution is denoted as

\[
\frac{dL(\lambda, \gamma_0)}{d\gamma_0} = \lambda - \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} = 0 \tag{3.8}
\]

\( \gamma_i \lambda_{i,j,k} = 0, \lambda_{i,j,k} \geq 0, \) where \( i = 1,2, j = 1,2, M \) and \( k = 1,2, \ldots, N \) (3.9)

The Equation (3.9) is called inequality constraint.

Here \( \lambda_j = 0 \) is called active constraint for the optimal solution and the Lagrange multiplier \( \gamma_j \neq 0 \)

where \( \lambda_j > 0 \) is called inactive constraints and hence \( \gamma_j = 0 \)

The Equation (3.7) can be written as

\[
\mu_{i,j,k} \cdot \lambda_{i,j,k} = \frac{\mu_{i,j,k}}{\lambda_{\gamma_0} \cdot \gamma_0}, 1 \leq i \leq 2, 1 \leq j \leq M, 1 \leq k \leq N \tag{3.10}
\]

Equation (3.10) gives explicit versions for inactive and active constraints.
\[ \mu_{i,j,k} = \frac{\mu_{i,j,k}}{\lambda \gamma_0}, \text{when } \lambda_i > 0, \quad (3.11) \]

\[ \mu_{i,j,k} = \frac{\mu_{i,j,k}}{\gamma_i \gamma_0}, \text{when } \lambda_i = 0, \quad (3.12) \]

If \( \gamma_0 \) is known, Equation (3.12) solves the unknown Lagrangian multipliers \( \gamma_i \) in terms of computational resources \( \mu_{i,j,k} \).

If \( \gamma_0 \) is unknown, then sum up all \( i, j \) and \( k \) terms in Equation (3.10) and use the Equations (3.11) and (3.12) to calculate active and inactive constraints,

\[ \mu - \lambda = \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \mu_{i,j,k} - \lambda_{i,j,k} \quad (3.13) \]

By solving the above equation, the unknown value of the Lagrangian multiplier \( \gamma_0 \) is obtained as a function of the collection of resources \( \mu \) and total traffic \( \lambda \).

From obtained Equation \( \gamma_0 = \gamma_0 (\lambda / \mu) \) substituted in Equations (3.11) and (3.12) to get the optimal load distribution

\[ \lambda_{i,j,k}(\min) = \max \left( \mu_{i,j,k} \cdot \frac{\mu_{i,j,k}}{\lambda \gamma_0}, 0 \right) \quad (3.14) \]

By substituting \( \lambda_{i,j,k} = \lambda_{i,j,k}(\min) \) in Equation (3.14), the minimal delay per computational task is obtained as

\[ D_{\min} = \frac{1}{\lambda} \sum_{i=1}^{2} \sum_{j=1}^{M} \sum_{k=1}^{N} \lambda_{i,j,k} \quad (3.15) \]

The valid solution \( \mu_{i,j,k} \cdot \frac{\mu_{i,j,k}}{\lambda \gamma_0} > 0 \)

if and only if \( (\sqrt{\lambda \gamma_0 \mu_{i,j,k}} - 1) > 0 \)
3.5 **ROUTING PROTOCOLS FOR WIRED NETWORK**

The heterogeneous environment uses separate protocols for wired and wireless set up. The protocols used for wired set up include Dijkstra Transverse algorithm and Multipath Dijkstra Transversal algorithm.

These protocols are analyzed in the following subsections:-

### 3.5.1 Dijkstra Transverse Algorithm

Dijkstra Transverse (DT) algorithm is based on the link state algorithm and employs periodic exchange of messages to maintain topology information of the networks. In the mean time, it updates the routing table in an on-demand scheme and forwards the packets in multiple paths which have been determined at the source. If a link failure is detected, the algorithm recovers the route automatically.

Most backbone networks still use unipath routing such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS) or their Equal Cost Multi- Path feature (ECMP). With these routing protocols, the forwarding only changes upon topology variations and not upon traffic variations. Dynamic multipath routing is able to provide several services such as load balancing, reduction of delays and improvement of throughput, and availability of fast rerouting schemes in case of failures. The reliability of an IP network against failures and congestion depends on the reaction time necessary for the convergence of the underlying routing protocol. A simple variant of Dijkstra is selected where equal cost paths are inherited along the Shortest Path Tree (SPT). The optimality condition of sub-paths computed with ECMP restricts the number of loop free paths and so reduces potential advantages of multipath routing. A simple hop by hop scheme does not require a signaling protocol to validate loop free paths. If the validation procedure, whose goal is to verify the absence of loops, is local
(without exchanging any message) and does not involve all routers, then the deployment can be incremental. Here, the approach is equivalent to ECMP in terms of time, space and message exchange complexity, but it allows computing for a greater diversity of forwarding alternatives.

### 3.5.2 Multipath Dijkstra Transverse (MDT) Algorithm

Multipath Dijkstra Transverse (MDT) algorithm is the improvement of Dijkstra Transverse (DT) algorithm. However, the basics of DT and MDT are similar. The summation of DT and MDT computes a multipath cost matrix on a given root node. A multipath cost matrix $M_c$ contains an overestimation of the best costs for all $(n - 1)$ destinations and via all possible outgoing neighbours denoted by $k^+(s)$ of source node $s$. The goal of this algorithm is to calculate a set of candidate next hops incorporating path costs via each neighbour. The calculation consists of two main stages:-

- Computing the best path tree and transverse edges, and
- Computing backward and forward transverse paths.

The best transverse paths are computed by iteration based on the first hop. Without an optimized structure to implement the best cost vector (the priority queue, denoted by $T_c$), the complexity of calculating DT for each node is the worst.

$$O(|N|^2 + |E| + |N| \times k^+(s)) = 0(|N|^2) \quad (3.16)$$

DT adds a time complexity, proportional to the outgoing degree of the given root node $s$ compared to Dijkstra.

The MDT computation is based on the order of node exploration which depends on the rank of costs stored in $T_c$. In MDT, the first computation phase calculates all candidates next hops corresponding to
ECMP alternatives. Recursively, the cost inheritance takes into account all the set of equal, best cost paths for all marked nodes. The complexity of MDT is slightly greater than the one of the DT

The general principle of this algorithm is to look for the shortest path $P_i$ to the destination $d$ at step $i$. Then the edges in $P_i$ or those pointing to $P_i$ have their cost increased in order to prevent the usage of similar path in the forthcoming steps. Path cost matrix ($f_p$) is used to increase the costs of arcs belonging to the previous path $P_i$ (or the opposite arcs belonging to it). This encourages future paths to use different edges, but not different vertices. Future edge cost matrix ($f_e$) is used to increase the costs of the edges which lead to the vertices of the previous path $P_i$. Different $f_p$ and $f_e$ are chosen to get link-disjoint path or node-disjoint routes as necessary.

**Algorithm** MultipathDijkstra ($f, e, S, N$)

1. Initialize cost $c$ and set $S$ as $M_{c_1} = M_c$ and $S_1 = S$
2. For $i = 1$ to $N$ do
   a. Source $i = \text{Dijkstra} \ (S_i, f)$
   b. $P_i = \text{Path} (\text{Source}_i, d)$
   c. If $e$ or Reverse($e$) is in $P_i$ then $M_{c_{i-1}}(e) = f_p(M_{c_i}(e))$
   d. else if Head($e$) is in $P_i$ then $M_{c_{i-1}}(e) = f_e(M_{c_i}(e))$
   e. else $M_{G-1}(e) = M_{c_i}(e)$
3. $S_{i-1} = (V, E, M_{c_{i-1}})$
4. Return multiple shortest path $(P_1, P_2, \ldots, P_i)$

**End** MultipathDijkstra

**Figure 3.4 Multipath Dijkstra Traversal Algorithm**
3.6 **ROUTING PROTOCOL FOR WIRELESS NETWORK**

3.6.1 **Ad hoc On-demand Multipath Distance Vector Routing (AOMDV)**

Ad hoc On-demand Multipath Distance Vector Routing (AOMDV) (Marina & Das 2001) belongs to on-demand and reactive routing protocol of ad hoc wireless networks. The main goal is to compute multiple loop-free and link-disjoint paths between source and destination pair. The merits of AOMDV are estimated in terms of the increased packet delivery ratio, throughput and reduced average end-to-end delay and normalized control overhead. The average end-to-end delay is reduced by introducing multiple loop-free paths in this scheme. In multiple routes, the destination contains the list of next-hops along with the corresponding hop counts in routing table entries. If all the next hops have the same sequence number, the advertised hop count is defined as the maximum hop count for all the paths. Route advertisement is effectively sent to the destination by using this hop count value. If any duplicate route advertisement is received by a node, it forwards the packet through an alternate path to the destination. If it is less than the advertised hop count for that destination, then the loop freedom is ensured by selecting the alternate path for destination on the basis of the hop count value. The destination node is to sort out all the paths by maximum hop count value.

3.6.2 **Proposed QoS Multipath Routing (QAOMDV)**

In this section, an extension of the AOMDV protocol is proposed in order to support certain mechanism and technique to improve its performance. AOMDV allows finding of many routes between source and destination during the same route discovery procedure, but only one path is used to transmit data. When the source receives one or many RREP packets from many disjoint paths, it decides the best path in which the packet is to be broadcast. In QAOMDV a feasible path which satisfies the bandwidth
constraint is identified. In contrast to the flooding based algorithms, QAOMDV search only a small number of paths, which reduces the routing overheads. In order to maximize the chance of finding a feasible path, the information is collectively utilized to make hop by hop selection. This protocol not only considers the QoS requirement, but also considers the optimality of the routing path in terms of optimal delay. In QAOMDV, a simple delay constraint model has been used to calculate the optimal delay values of different nodes at different times.

This routing protocol has three phases:-

(i) Route Discovery,

(ii) Route Selection, and

(iii) Route Maintenance.

3.6.3 Route Discovery

In the route discovery procedure, the QAOMDV find multiple routes using route request and route reply query cycle. When a source node wants to send a packet to a destination for which it does not already have a route, it forwards a Route Request (RREQ) packet to all the neighbours across the network. The extended route request packet of QAOMDV is shown in Figure 3.5.

<table>
<thead>
<tr>
<th>SA</th>
<th>DA</th>
<th>Seq.No.</th>
<th>Hop count</th>
<th>Timeout</th>
<th>Recv Time</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>-</td>
<td>Source IP Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>-</td>
<td>Destination IP Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq.No.</td>
<td>-</td>
<td>Destination Sequence Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hop count</td>
<td>-</td>
<td>Number of hops needed to reach the destination</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeout</td>
<td>-</td>
<td>Lifetime of the route</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recv Time</td>
<td>-</td>
<td>Receive Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>-</td>
<td>Minimal Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.5 Extended Route Request (RREQ) Message
Nodes receiving this RREQ packet update their information for the source node and set up backward pointers to the source node in the routing tables. The QAOMDV protocol adds two additional fields to verify QoS constraint. RREQ contains the source node's IP address, destination address, hop count, timeout value, the received timing of the packet (RT) and the transmission delay of the packet (Delay). When a RREQ message is received by an intermediate node (Node$_{i}$) from the source node, it updates the current time in (RT) field and broadcasts to its neighbours. When the RREQ message arrives at node$_2$, RT field is updated and the delay is calculated by subtracting RT of node$_2$ from RT of node$_1$. This value is stored in a Delay field in RREQ message. The delay is measured in terms of queuing time and the transmission time. A threshold delay (D$_{th}$) is computed by the number of nodes n, collection of resources μ and total traffic λ. ($D_{th} = \frac{n}{(\mu - \lambda)}$). If the computed delay value exceeds the threshold value, then the link between node$_1$ and node$_2$ is unavailable. Therefore, node$_1$ will not forward any RREQ message to the node$_2$; but the RT and the Delay fields will be updated in the RREQ message. When the RREQ message arrives at the destination, it contains the cumulative sum of delay of all nodes.

A node sends a request to all its neighbours. Maximum of (n – 1) neighbours are possible for a node in the network. QAOMDV calculates the average distance between the source node and all its neighbours based on the location information of the nodes and then checks their delay value. Finally, it selects the neighbour having delay value below the threshold value. The structure of routing table entries for AOMDV and EEAOMDV is given in Table 3.1.
Table 3.1 Routing table entries for AOMDV and QAOMDV

<table>
<thead>
<tr>
<th>AOMDV</th>
<th>QAOMDV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Address</td>
<td>Destination Address</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>Sequence Number</td>
</tr>
<tr>
<td>Advertised Hopcount</td>
<td>Advertised Hopcount</td>
</tr>
<tr>
<td>Route List</td>
<td>Route List</td>
</tr>
<tr>
<td>(nexthop₁, hopcount₁)</td>
<td>(nexthop₁, hopcount₁, Delay₁)</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>(nexthopₙ, hopcountₙ)</td>
<td>(nexthopₙ, hopcountₙ, Delayₙ)</td>
</tr>
<tr>
<td>Time out</td>
<td>Time out</td>
</tr>
</tbody>
</table>

The following algorithm illustrates the Route Discovery process:-

Algorithm: Route Discovery

Step 1. Source node S initiates the following:-

a) Creates the Route Request (RREQ) packet with field values
   Source Address (SA), Destination Address (DA), Sequence
   Number (Seq.No.), Timeout, Hop Count, Delay (D) initialized
   as SA = S, DA = D, Seq.No. = 1, TTL = T, Hops = H, D = 0 ,
   and

b) Broadcasts the RREQ packet to next neighbour node whose
   \( D_{th} \geq D \).

Step 2. If the intermediate node receives the RREQ packet then

a) The Delay is calculated as the difference between current
   node’s received time with previous node’s received time,

b) The RREQ packet is forwarded to the next neighbour node,
d) The route is selected if $D_{th} \geq D$. Otherwise the link between node$_1$ and node$_2$ is unavailable, and

e) If the above QoS constraint is satisfied, then the Delay value is updated in the routing table.

**Step 3.** If the node is receiving the RREQ packet in the destination node D, then

a) Destination node D generates the RREP packet and broadcasts it to the source. The delay field of the RREP packet is updated with the cumulative delay of the path, and

b) D broadcasts all the node disjoint paths back to the source node S.

### 3.6.4 Route Selection

When the RREQ is received at the neighbour node, it forwards a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Suppose the node receives a processed RREQ, then it discards the RREQ packet. If a RREP of multiple paths are received at source node, it is stored by the hop count value. In AOMDV, the route is selected on the basis of minimum number of hops. But the QAOMDV protocol selects the optimal path by sorting multiple routes in descending order of delay of the data packets. Suppose multiple paths have the same delay, the path with minimum number of hops is selected. In case of any conflict in the number of hops, then the source node follows ‘First come, first served’ policy.

### 3.6.5 Route Maintenance

If the delay value exceeds the threshold value $D_{th}$, the link is broken. A Route Error (RERR) message is sent back to the previous node to
indicate the route breakage. If a node receives this RERR message, it informs
the source node to start the route discovery procedure again.

3.7 SIMULATION SETUP

NS-2 provides support for wired-cum-wireless scenarios, where
several wireless networks are connected through wired nodes. The idea is to
define different wireless domains and one or more base-station nodes in each
domain. The packets are transmitted to or received from destinations outside
the domain. In reality, the base station node is nothing but a mobile node
connected to one or more wired links. The simulation parameters for
experiment results are illustrated in Table 3.2.

### Table 3.2 Simulation Parameters

<table>
<thead>
<tr>
<th>Simulator</th>
<th>NS-2.34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Protocols</td>
<td>AODV, AOMDV, QAOMDV</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>500 s</td>
</tr>
<tr>
<td>Simulation Area</td>
<td>800 * 800 m</td>
</tr>
<tr>
<td>Number of Wired Nodes</td>
<td>10 (max)</td>
</tr>
<tr>
<td>Number of Mobile Nodes</td>
<td>500 (max)</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>250 m</td>
</tr>
<tr>
<td>Mobility Models</td>
<td>Random waypoint, RPGM, Manhattan, TIMM, SLAW</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>50 m/sec (max)</td>
</tr>
<tr>
<td>Pause Time</td>
<td>500 s (max)</td>
</tr>
<tr>
<td>Connection Rate</td>
<td>5 packets /sec</td>
</tr>
<tr>
<td>Data Payload</td>
<td>512 B</td>
</tr>
<tr>
<td>Traffic Type</td>
<td>CBR</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>1000 J</td>
</tr>
</tbody>
</table>
3.7.1 Mobility Models

The mobility models considered for the simulation are Radom Waypoint (RW), Reference Point Group Mobility (RPGM), Tactical Indoor Mobility Model (TIMM), Manhattan and Self similar Least Action Walk model (SLAW) mobility model. The field configurations used is 800m * 800m with 100 to 500 nodes. Each packet starts its transmission from source location to desired destination according to the mobility model. After the destination is reached, it selects another destination after a pause. The pause time affects the relative speed of the mobile node.

Radom Waypoint model

The random waypoint model is a random model for mobility of the ad hoc networks. It is an elementary model which describes the movement pattern of independent nodes by simple terms. This mobility model is used for simulation purpose when new network protocols are evaluated in MANET. In random-based mobility simulation models, the mobile nodes move randomly and freely without restrictions. To be more specific, the destination, speed and direction are all chosen randomly, independent of other nodes.

Each node begins by pausing for a fixed number of seconds. The node then selects a random destination in the simulation area and a random speed between 0 and maximum speed. The node moves to this destination and again pauses for a fixed period before selecting another random location and speed. This behaviour is repeated for the length of the simulation (Camp et al 2002).

The measure of the relative speed between node i and j at time t is

\[
RS (i, j, t) = | \vec{V}_i(t) - \vec{V}_j(t) | \quad (3.17)
\]
The mobility metric $M$ is calculated as the measure of the relative speed averaged over all node pairs and over all time. The formal definition is as follows:

$$M = \frac{1}{|i-j|} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{T} \int_{0}^{T} RS(i,j,t)dt$$  \hspace{1cm} (3.18)

**Reference Point Group Mobility model**

The Reference Point Group Mobility (RPGM) model represents the random motion of a group of mobile nodes as well as the random motion of each individual mobile node within the group. In this model (Camp et al 2002), each group has a centre, which is either a logical centre or a group leader node. For the sake of simplicity, the centre is assumed as the group leader. Thus, each group is composed of one leader and a number of members. The movement of the group leader determines the mobility behavior of the entire group. Initially, each member of the group is uniformly distributed in the neighbourhood of the group leader. Subsequently, at each instant, each node has a speed and direction that are derived by randomly deviating from that of the group leader. The movement in group mobility can be characterized as follows:

$$|V_{member}(t)| = |V_{leader}(t)| + random() \times SDR \times \text{max speed}$$  \hspace{1cm} (3.19)

$$|\theta_{member}(t)| = |\theta_{leader}(t)| + random() \times ADR \times \text{max angle}$$  \hspace{1cm} (3.20)

SDR is the Speed Deviation Ratio and ADR is the Angle Deviation Ratio. SDR and ADR are used to control the deviation of the velocity (magnitude and direction) of group members from that of the leader. Therefore, model parameters will be ADR, SDR, initial distance (members
initial distance from the leader node) and group size which determine the number of group nodes.

**Manhattan mobility model**

The Manhattan mobility model uses a grid road topology. This mobility model was mainly proposed to emulate the movement pattern of mobile nodes on the streets in urban areas, where the streets are in an organized manner and the mobile nodes move in horizontal or vertical direction on an urban map. This model employs a probabilistic approach in the selection of node movements. At each intersection, a mobile node chooses to keep moving in the same direction. The probability of going straight is 0.5 and taking a left or right is 0.25 each. Thus the Manhattan mobility model is also expected to have high spatial dependence and high temporal dependence. It too imposes geographic restrictions on node mobility (Bai et al 2003).

**Tactical Indoor Mobility Model (TIMM)**

Tactical Indoor Mobility Model (TIMM) (BonnMotion 2011) is used to model a group of soldiers in an urban warfare scenario. A graph representation of the building is used to restrict the movements of valid paths. Rooms and doors are represented as vertexes while edges represent valid paths between the vertexes. Each time a group reaches its destination, a new destination is chosen.

**Self-similar Least Action Walk (SLAW) model**

Self-similar Least Action Walk (SLAW) can produce synthetic mobility traces in various mobility settings, including user created virtual ones for which no empirical information is available. Lee et al (2009) have created synthetic traces for virtual environments. It is important for the
performance evaluation of mobile networks. Network designers test their networks in many diverse network settings. This tool emulates human walk behaviors in diverse application scenarios that can be applied to accurate urban planning, traffic forecasting and biological and mobile virus spread analysis.

### 3.7.2 Energy Model

The energy model uses transmission power, receiving power and idle power for simulation and emulation, as illustrated in Table 3.3.

<table>
<thead>
<tr>
<th>Transmission Power</th>
<th>Receiving Power</th>
<th>Idle Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4 W</td>
<td>1.0 W</td>
<td>0 W</td>
</tr>
</tbody>
</table>

### 3.7.3 Traffic Model

Random traffic connection of User Datagram Protocol (UDP) and Continuous Bit Rate (CBR) can be set up between mobile nodes using a traffic scenario generator script. The source to destination pairs are spread randomly over the mobile and fixed network. The number of clusters and the packet transmission rate in each cluster is varied by changing the traffic load and pause time.

### 3.7.4 Performance Metrics

Performance comparison has been done using normalized routing overhead, packet delivery ratio, throughput, end-to-end delay and energy consumption metrics. A description of these metrics is given in this section.
Packet Delivery Ratio (PDR)

It is calculated as the ratio between the total numbers of data packets received by the destination node to the number of data packets sent by the source node during the time period of the simulation.

\[
PDR = \frac{\text{Number of Data Packets Received}}{\text{Number of Data Packets Sent}} \quad (3.21)
\]

Normalized Routing Overhead (NRO)

It is defined as the ratio of the number of routing packets sent to the destination node to the number of data packets actually received at the source node.

\[
NRO = \frac{\text{Number of Routing Packets Sent}}{\text{Number of Data Packets Received}} \quad (3.22)
\]

Average Throughput

The average throughput is defined as the total number of data packets delivered divided by the total duration of simulation.

\[
\text{Throughput} = \frac{\text{Number of bytes received} \times 8}{\text{Simulation Time} \times 1000} \text{ kbps} \quad (3.23)
\]

Average End-to-End Delay

The average end-to-end delay is a measure of the average time taken to transmit each packet of data from the source to the destination. Network congestion is indicated by higher end-to-end delays.
Average Delay = \frac{\text{Transmission time} - \text{Sent time}}{\text{Total Number of connection pairs}} \quad (3.24)

**Energy Consumption**

Consumed energy is defined as the ratio of global consumed energy to the number of data packets received. The transmitted and received energy is defined as

\[ E_{\text{tx}} = \text{Transmitted Power} \times \text{Transmission Time} \quad (3.25) \]

\[ E_{\text{rx}} = \text{Received Power} \times \text{Transmission Time} \quad (3.26) \]

Total Consumed Energy (T_{\text{con}}) is represented as

\[ T_{\text{con}} = \text{Number of Nodes} \times \text{Initial Energy} - \text{Residual Energy} \quad (3.27) \]

The lifetime of the network extends when the energy consumption of the nodes are less.

### 3.8 EMULATION

#### 3.8.1 Introduction

Heterogeneous network can be simulated, emulated or set up physically. Physical testbed captures detailed network transactions. However, its scalability is limited by the availability of physical resources. Emulation abstracts the networking hardware using virtualization. The major advantage is to overcome some of the scaling limitations of physical testbeds. In emulation, the applications and protocols are real implementations whereas in simulation, they are virtual. Real-time network simulation is somewhere between emulation and simulation. It aims to combine the fidelity and realism
offered by network emulation with the flexibility, control and scale of simulation.

3.8.2 Related Works

Most of the network simulators are discrete event simulators that measure the level of network behavior in detail. Network Simulator (NS-2) is widely used in the current Internet world. There are other universal network simulators like OPNET (2008), Omnitnet++ (Andras Varga 2005), SWiMNet (Boukerche et al 2001) and GloMoSim (2000). They provide an important method to test network protocols and topologies, but share some major problems. They use their own protocol implementations, and hence, they cannot be used to test the interaction and behavior of actual protocol implementations used in operating systems as well as network devices used in live networks. They cannot be directly used by the traffic generated by different applications. In order to study actual protocol implementations under realistic conditions, network emulators are required (Mahrenholz & Ivanov 2004). Emulators can work in different ways. EMUNET (Meng 1996) is a software library that simulates a communication network inside a user application. The network simulator NS-2 is a widely accepted discrete event network simulator, actively used for wired and wireless network simulations. It has an emulation feature, with the ability to introduce the simulator into a live network using a soft real-time scheduler, which tries to tie the event execution within the simulator with the real-time. Heterogeneity problems (Majumder et al (2010); Jahir et al (2010); Kumar et al (2007)) cannot provide a complete solution for future Internets. The simulation and emulation of different routing protocols for wireless and heterogeneous network provide a bridge between infrastructureless and infrastructure based
networks (Nordstrom et al 2006, Calarco & Casoni (2012), Van Vorst (2012)). This work modifies MANET routing protocols to select the optimal path during data delivery and also compare the performance of different routing protocols in simulation and emulation environment. The proposed framework is composed of Linux based testbed and NS-2 simulator.

### 3.8.3 Emulation Setup

HETEROTB testbed allows the real-time emulation of heterogeneous network composed of the Wireless Local Area Network (WLAN) with wired Ethernet LAN. Emulation is a technique to introduce the simulator into a live network. The special object in the simulator introduces live traffic into the simulator and also injects traffic from the simulator into the live network. The collection of objects, tap agents and network objects should be used to interface between simulator and live network. Tap agents integrate live network data into simulated packets and vice-versa. Network objects have been installed in tap agents to provide an entry point for sending and receiving live data. The real-time scheduler should always be used by the emulation facility. This approach enables an accurate end-to-end evaluation of real user applications over a realistic and complete heterogeneous network with advanced algorithms and end-to-end Quality of Service (QoS) management policies.

HETEROTB is an integration of wired / wireless, reconfigurable, secure, scalable and heterogeneous architecture testbed. In a building, a testbed was constructed for the proposed study, which consists of 10 wired nodes, two gateways and wireless nodes scalable from 5 to 50. Emulation
environment was used to test and demonstrate the combination of a discrete event simulation with real applications.

Figure 3.6 Emulation set up

Figure 3.6 illustrates the emulation set up for wired-cum-wireless network. This testbed has the tap agents and network objects in the simulator to introduce live traffic into the simulator and vice versa. The proposed AOMDV and EELSRP protocols have been emulated using this setup.
3.9 EXPERIMENTAL RESULTS

This section presents an experimental study on the simulation results of protocols used in wired side (DT, MDT and ECMP) and wireless side (AODV, AOMDV and QAOMDV)

3.9.1 Wired Network

This simulation work is carried out for the wired networks. The candidate protocols considered are Equal Cost Multiple Path routing (ECMP), Dijkstra Transverse (DT) routing and Multipath Dijkstra Transverse (MDT). Figure 3.7 illustrates the simulation set up of 50 wired nodes. The performance of these protocols is measured in terms of overhead, packet delivery ratio and end-to-end delay.

![Simulation of Wired Network](image)

Figure 3.7 Simulation of Wired Network
The performance of ECMP, DT and MDT is compared by varying the number of nodes and pause time. Figure 3.8 shows the effect of varying the number of nodes on routing overhead. The overhead of MDT is higher than the ECMP and DT routing protocols, because multipath routing mechanism is used in MDT.

![Graph showing routing overhead vs number of nodes]

**Figure 3.8 Routing Overheads Vs Number of Nodes**

The effect of end-to-end delay as a function of number of nodes is shown in Figure 3.9. Average end-to-end delay is minimized in MDT protocol. This delay includes the queue delay and the propagation delay from the source to the destination. Multipath routing effectively reduces the queue delay because the traffic is distributed in different routes. Delay constraint is verified in every node to select the optimal path.
Figure 3.9 End-to-End Delay Vs Number of Nodes

Figure 3.10 Throughput Vs Number of Nodes for MDT
Figure 3.10 shows the throughput analysis by varying the number of nodes with bandwidth reservation and without bandwidth reservation for MDT protocol. Bandwidth reservation improves the throughput of the network.

3.9.2 Heterogeneous Network

This section presents the results from simulation experiments carried out using Network Simulator (NS-2.34) software. The Network Animator (NAM) tool output for the wired-cum-wireless network is illustrated in Figure 3.11.

Wired-cum-wireless simulation allows both wired and wireless nodes. A base station plays the role of a gateway for the wired and wireless domains. It is responsible for delivering packets into and out of the wireless domain.

Figure 3.11 Simulation set up
Average end-to-end delay, packet delivery ratio, normalized routing overhead, throughput and energy consumption metrics have been measured for AODV, AOMDV and proposed QAOMDV protocols. Multipath Dijkstra Transverse (MDT) routing protocol is used in the wired side of heterogeneous network.

Performance Analysis based on Nodes

Figure 3.12 indicates the average end-to-end delay of AODV, AOMDV and QAOMDV. It includes the queuing delay in every node and the propagation delay from the source to the destination. Multipath routing effectively reduces the queue delay, because, the traffic is distributed in different routes. QAOMDV maintains delay constraint.

![Figure 3.12 End-to-End delay Vs Number of Nodes](image-url)
Figure 3.13 shows the number of nodes versus normalized routing overhead. The overhead of QAOMDV is higher than AOMDV and AODV protocols as it uses a large number of control packets to search and maintain multiple routes.

![Graph showing NRO vs Number of Nodes]

*Figure 3.13 Normalized Routing Overhead Vs Number of Nodes*

Figure 3.14 shows the tradeoff between packet delivery ratio and number of nodes. The packet delivery ratio of QAOMDV is better than AODV and AOMDV protocol. By splitting single path into multiple paths, the bandwidth is effectively utilized between source and destination. Limited bandwidth usage reduces the packet loss during transmission. Hence the PDR of QAOMDV is increased when compared to AODV and AOMDV.
Figure 3.14 Packet Delivery Ratio Vs Number of Nodes

Figure 3.15 Energy Consumed Vs Number of Nodes
Figure 3.15 illustrates the average energy consumption of different routing protocols. From the figure, it is evident that QAOMDV protocol utilizes less energy when compared to AODV, AOMDV. QAOMDV uses delay constraint while selecting optimal path.

![Graph showing End-to-End Delay vs Pause Time](image)

**Figure 3.16 Average End-to-End delay Vs Pause Time**

Figure 3.16 shows the performance of various protocols on varying the pause time from 100 to 700 second. From the figure 3.16, it is seen that the delay of QAOMDV is reduced when compared to AODV and AOMDV.

3.10 CONCLUDING REMARKS

This chapter presented a mathematical model for delay in heterogeneous network with multipath routing. Based on the assumptions made for the delay in multipath routing, the approximated mathematical
model has been derived. The proposed model was used for simulation and emulation purpose.

The performance of the proposed QAOMDV protocol has been compared with AOMDV for the wireless segment of heterogeneous network. Multipath routing protocols that compute multiple paths during route discovery reduces overhead, latency and bandwidth. It is observed that the performance of QAOMDV is more efficient when compared AODV and AOMDV.

Simulation results show that the performance of heterogeneous networks using QAOMDV on wireless side and Multipath Dijkstra Transverse (MDT) routing protocol on wired side. This research attempts to prove that multipath routing algorithm provides low delay, high throughput, better bandwidth utilization, low packet loss and energy consumption during data transmission. Different routing protocols are compared by average end-to-end delay along with pause time. On an average QAOMDV has on an average 10.6% less delay, 3.3% better PDR and 27.6% less energy consumption when compared to AOMDV protocol.