Chapter 6

QGMR: A QoS-aware Gateway-based Multi-path Routing Scheme for Multi-hop WiLD Networks

6.1 Introduction

Routing plays a crucial role in provisioning end-to-end QoS over WiFi-based Long Distance (WiLD) mesh networks. In gateway-based mesh networks, the optimal paths between gateway and other nodes often overlap and hence degrade the overall network performance significantly. This chapter addresses the issue of finding QoS-aware paths for smooth transmission of real-time traffic in multi-hop WiLD networks.

Traditional routing protocols usually maintain a single optimal path between each pair of source and destination nodes. When multiple nodes transmit through the same path or different paths with some overlapped portions, naturally the path gets congested and hence cannot forward the traffic efficiently. Multi-path routing protocols [?, ?, ?] are widely used in WMN for solving this problem and providing some level of end-to-end QoS to various network applications. Routing protocols for WiLD mesh networks need to exploit its relatively static network topology with wireless links and deal with issues like interference and noise in a way that optimizes the overall network performance in terms of throughput and delay. The classical routing protocols are not considered suitable for providing QoS over WiLD mesh networks for various reasons. As a consequence, many new routing protocols have been proposed to support QoS in WMNs. However, discovery of end-to-end paths in accordance with QoS requirements of heterogeneous applications and maintaining them with change of quality of paths is not considered in a holistic manner.

In this chapter, we propose a QoS-aware multi-path routing protocol called *QGMR* which discovers multiple maximally disjoint paths between the gateway and a given node. It uses two end-to-end QoS parameters as routing metric: *expected path bandwidth* and *end-to-end delay.* The gateway node selects a set of suitable QoS paths for a given source node. Before starting a given real-time flow, a path selection scheme chooses the appropriate path or a set of paths from the discovered paths based on the QoS requirements of the flow. To ensure quality of end-to-end paths chosen and distribute the traffic among the chosen paths evenly, an admission control mechanism is employed. In the situation of significant change in the quality of a link which is a part of any QoS paths, a path maintenance procedure is defined which induces the neighbour nodes to trigger a route update process. The proposed protocol enhances the performance of real-time traffic significantly. The simulation results confirm the improvements achieved in terms of provisioning QoS for real-time applications.

The rest of the chapter is organized as follows. Section **??** takes a look on the related works. A comparison of relevant multi-path routing protocols is presented in Section **??**. Details of the proposed protocol has been discussed in Section **??**. It explains the multi-path route discovery, route maintenance and admission control mechanisms of the proposed routing scheme with appropriate examples.

Section **??** presents the performance evaluation of the proposed protocol through simulation studies. Finally, Section **??** gives the conclusion to this chapter.

6.2 Related Works

Many multi-path routing protocols are found in the WMN literature addressing various QoS issues. Some of the relevant multi-path routing schemes are presented in this section.

The goal of a QoS routing protocol is to find a loop-free path satisfying a given set of constraints on parameters like bandwidth and delay. The existing classical routing protocols (DSDV [**?**], AODV [**?**], DSR [**?**], etc.) are not suitable to provide QoS in WMNs. Hence, many new routing protocols ([**?**, **?**, **?**, **?**, **?**, **?**]) have been designed to serve the purpose. A AODV-based routing protocol namely QAODV (QoS-AODV) [**?**] provides QoS by reducing invalid transmission of RREQ packets in the route discovery process. It comprehensively considers bandwidth, delay, hop counts and congestion situation of nodes in selecting routes. In QUORUM [**?**], flooding of control messages has been reduced by using explicit knowledge of the network topology. Similarly, to ensure delay in route discovery phase, it chooses the route on which the first in-time reply arrives at the source. If the route reply does not arrive within the time duration of two times of maximum end-to-end delay, it considers the route discovery phase to be failed. In such case, the source may back-off and initiate a route rediscovery procedure later or turn down the flow.

Many researchers claim that multi-path routing techniques improve the load balancing, QoS, and reliability and also allows to use bandwidth aggregation technique. Multi-path routing protocols are highly used in provisioning QoS. A routing protocol for wireless mesh network known as AOMDV [**?**] offers a multi-path, loop-free extension to AODV. Two additional routing fields- hop count and last hop are stored in the route entry to help in addressing the problems of loop freedom and path disjointness respectively. But, AOMDV routing protocol fails to provide precise QoS guarantees for real-time traffic. Another multi-path routing protocol for WMN is proposed in [?] which discovers multiple acceptable paths by using existing routing metrics (e.g., ETT, ETX, etc.) for any traffic. The gateway nodes periodically broadcast advertisement of connectivity information. After hearing these, the children nodes find some acceptable paths towards the gateway nodes. In providing QoS for real-time traffic using multi-path routing, Shu et al. [?] defines two routing constraints: an interference free link schedule constraint and an interference-free node schedule constraint. Based on these two constraints called interference-aware multi-path selection metric and using AODV like protocol; the protocol finds out multiple candidate paths. Zuo et al. [?] proposes a hybrid multi-path routing algorithm called DAWMNet which works in two phases and uses distance as a routing metric. In the first phase, shortest route is discovered from the gateway to each end node by adopting enhanced Dijkstra's algorithm. Using the chosen route, multiple routes are explored based on the Ant Colony optimization (ACO) algorithm by diffusing pheromone packets. The maintenance of route is also done using ACO technique. Although the above multi-path routing protocols discover multiple paths and find QoS guaranteed paths, but they do not consider path disjointness characteristics.

A routing protocol based on Dynamic Source Routing (DSR) is presented in [?]. Here, the path followed by the first RREQ packet received by the destination node is considered as the primary route (shortest route). The destination node then prepares RREP packets for primary and other backup routes. The destination node sends RREP packets for the primary route and for the backup routes which are link-disjoint to the primary route. On the failure of the primary route, the source node switches to the shortest backup route. Split Multi-path Routing (SMR) [?] splits traffic into two maximally disjoint routes. Intermediate nodes do not reply to route request even if they have available routes to the destination, instead they forward RREQ packets to find maximally disjoint paths.

Different routing metrics are proposed in WMNs to take care of QoS requirements of various applications while taking routing decisions. To ensure throughput for real-time bandwidth greedy traffic, a bottleneck bandwidth metric is proposed in [?]. The bandwidth is estimated by using packet pair technique where probe packets are exchanged between the end-points of a link to characterize link delay and loss. However, packet pair delay faces certain serious shortcomings like out-of-order delivery, limitations due to clock resolution, changes in bandwidth. While addressing these shortcomings, a significantly more robust procedure termed as Packet Bunch Mode (PBM) [?] is proposed. PBM deals with the limitations of packet pair technique by forming a bunch of 2 to 5 packets for estimation of link characteristics. To ensure delay for real-time delay sensitive traffic, a delay metric called end-to-end delay (E2ED) is used in [?] which calculates the end to end delay of a path by adding the link delay and the queuing delay. To improve throughput and delay performance in AODV routing protocol, [?] proposed a new routing metric. In this proposal, the intermediate node checks whether the available bandwidth and accumulated delay is is according to the demand of the application or not. If both the conditions are true, then only traffic is forwarded to its destination. Providing support to multimedia real-time services such as voice and video applications in WMNs with QoS requires a pre-specified bandwidth between the end-points. Protocols such as [?, ?, ?, ?, ?, ?] use admission control mechanism in routing to find a route with sufficient bandwidth for an admitted flow.

The multi-path protocols discussed above do not exploit bandwidth aggregation schemes as they use only a single path to forward traffic. The path maintenance process of existing routing protocols considers only path failure condition, rather than considering path quality change condition. Most of the routing protocols do not have a novel path selection scheme to guarantee QoS for real-time traffic. The multi-path routing protocols compute multiple available paths from source to destination. However, they use the best path on the basis of some routing metrics. While enabling QoS for different real-time traffic classes, using a single path for all all types of traffic may create congestion or overloading in routing. The path maintenance process of many existing routing protocols only considers path failure condition, rather than considering path quality change. While multiple paths are available, a path selection scheme to guarantee end-to-end QoS for real-time traffic is also important. Thus, to ensure various QoS parameters for different traffic classes, we consider an integrated approach considering all these issues.

6.3 Comparative Study of Different Multi-path Routing Protocols

A plethora of multi-path routing protocols are available in WMN literature. Considering the QoS aspect, a few relevant routing protocols are presented in Section ?? (page ??). MP-DSR [?] finds multiple unicast paths considering end-to-end reliability as a routing metric. SMR [?] is a source based on demand routing scheme which employs a highly centralized routing protocol. The gateway node selects best two maximally disjoint paths after receiving all the RREQ packets. On detection of failure, the concerned node sends route error message to the gateway node which in turn informs the respective source nodes. MMESH [?] considers multi-gateway based networks and finds multiple paths to different gateway nodes. However, they consider change in link quality as a metric for route maintenance. QoS-MOLSR [?] is a OLSR-based multi-path routing protocol which uses end-to-end delay as a routing metric to support real-time applications over MANET. Routes are computed by using a modified version of Dijktra's algorithm. AOMDV [?] is a basic multipath routing protocol which discovers and maintains multiple paths between each pair of source and destination nodes. It maintains the path-disjointness and loop-freedom properties while discovering paths. The route discovery mechanism is initiated only when all the paths fail. Although AOMDV does not consider any specific QoS parameter, it provides reliability in data transmission by maintaining multiple paths. A comparison of different multipath routing protocols is provided in Table ?? of Chapter ?? (page ??).

6.4 The Proposed QoS-aware Gateway-based Multi-path Routing (QGMR) for Multi-hop WiLD Networks

The proposed routing protocol, called QGMR aims at provisioning of QoS in multi-hop WiLD networks. QGMR uses a flow-path mapping technique which maps a flow to paths that meet the QoS requirements of a given type of traffic. It establishes multiple paths for a single flow in order to meet the bandwidth requirement which could be provisioned as an aggregate of the available bandwidth of the selected paths. In its venture to provide QoS to real-time traffic, it increases the throughput of the network with multi-path consideration. The working of the proposed routing protocol is as follows.

Initially, each node in the network estimates the expected throughput and link delay to each neighbour nodes. Before starting communication, a node first checks for available paths to the destination. If no path is available, it initiates a multipath route discovery process. In this process, the route request is broadcast on all loop-free paths towards the destination node. Each intermediate node cooperates in the discovery process by appending its identification in the accumulated path and updating the expected path bandwidth and delay. The destination node, on receiving the route requests, selects a set of maximally disjoint feasible paths whose path delay and bandwidth are within the QoS-bounds of the corresponding traffic class and sends route replies only for those paths. On receiving a route reply, the source node registers a flow in one or more available paths to the destination.

In a situation where there exist some paths to a destination, the source node does not initiate route discovery process. Rather, it tries to register the given flow in the available paths that meet the specific QoS requirements. The path maintenance process is distributed in nature and does not involve any end-to-end mechanism for monitoring the quality of the discovered paths. All the nodes periodically monitor their neighbouring links. On detecting a significant change in the link quality or failure of a link, an intermediate node triggers route update packets to all the source nodes those have a path through this node.

Looking at the heterogeneity of Internet traffic, we classify traffic into three categories based on their QoS requirements. In IP-based network, the gateway node of a WMN is responsible for reliable data forwarding to its sub-networks. In such a case, if the gateway node can identify the traffic characteristic, it can find appropriate path by using multi-path selection scheme. Considering delay and throughput as QoS metrics, traffic are classified as Delay sensitive (Class 1), Bandwidth bound (Class 2) and Best-effort (Class 3). VoIP, video conferencing and videophony belong to Class 1 which produces delay sensitive traffic. Applications which require a certain minimum bandwidth guarantees such as video streaming, video broadcasting, and audio streaming are included in Class 2. Class 3 includes best-effort traffic such as HTTP, E-mail etc.

A gateway-based multi-hop WiLD network can be considered as a directed graph, $G(V, E)$ where V is the set of nodes and E is the set of edges or links connecting the nodes. A special node called gateway node (or Root node), $R \in V$ is considered to be connected to a fixed infrastructure with high-bandwidth connectivity. The remaining nodes communicate with the gateway node over multiple hops in order to get themselves connected to the Internet. The quality of a link can improve or deteriorate dynamically based on factors like interference, traffic load, etc. Any link, $j = (u, v) \in E$, between any two neighbouring nodes u and v has an associated bandwidth (Capacity) L_j^b and delay L_j^d. An i^{th} path between any two arbitrary nodes u and v, denoted by π_i, is given as $\pi_i = (u, x_1, x_2, ..., x_n, v)$ having n intermediate nodes $x_1, x_2, ..., x_n$. In WiLD mesh network, there exists multiple paths between a source and the gateway node. The protocol incorporates a multi-path route generation scheme which discovers all loop-free paths from the given source node to the destination node. The set of all possible paths between the nodes u and v, represented as $P(u, v)$ is given as $P(u, v) = \{\pi_1, \pi_2, \pi_3, ..., \pi_m\}$, where m is the number of available paths.

QGMR uses periodic probe messages to check the quality of links in the network. The aggregate quality of all the links in a path gives the end-to-end path quality. Let the path delay and path bandwidth of a path π_i be denoted as $D(\pi_i)$ and $T(\pi_i)$ respectively. The delay and throughput metric discussed in Section **??** are used to estimate the values of $D(\pi_i)$ and $T(\pi_i)$. Let us assume that bandwidth already reserved on link i and path π_i are denoted as $B_{resv}(i)$ and $B_{resv}(\pi_i)$ respectively. Then, the value of $B_{resv}(\pi_i)$ can be calculated by using the following equation.

$$B_{resv}(\pi_i) = max\{B_{resv}(u), B_{resv}(x_1), ..., B_{resv}(x_n), B_{resv}(v)\}$$

Routing real-time traffic with respect to their QoS requirements involves selection of appropriate QoS paths. The QoS requirements of different real-time traffic set certain bounds on throughput and delay. Let, Δd be the maximum delay-bound for delay sensitive traffic class and Δb be the minimum throughput-bound for bandwidth traffic class. In that case, a selected path, π_i must satisfy the following conditions:

(i) for delay sensitive traffic

$$D(\pi_i) \leq \Delta d$$

(ii) for throughput sensitive traffic

$$T(\pi_i) - B_{resv}(\pi_i) \geq \Delta b$$

where $B_{resv}(\pi_i)$ is the reserved bandwidth of a path π_i.

(iii) For delay as well as throughput sensitive traffic, both the conditions ((i) and (ii)) must be satisfied.

Based on the above conditions, the protocol carries out multi-path route discovery and finds paths which meet the QoS requirements of the corresponding traffic class.

6.4.1 Routing Metrics Used

A routing metric is essentially a value associated with each route or path which is used by a routing algorithm to select a subset of routes discovered by the routing protocol. The main objective of using routing metric is to minimize delay, maximize probability of data delivery, maximize path throughput, maximize network throughput, equal traffic distribution, etc. Probing based approaches in selecting routing metric has proven to be promising in the context of wireless mesh networks. They directly measure the quantity of interest rather than inferring it from indirect measurements, and do not rely on any analytical assumptions [?]. In the following subsection, we discuss the routing metrics considered in the proposed protocol.

6.4.1.1 Expected Path Bandwidth (EPB)

The throughput metric sets the upper bound on how fast a sender can deliver network layer data to the receiver. It comes from bandwidth estimation of the slowest forwarding node/link in the end-to-end chain that comprises the network path. Various bandwidth estimation techniques such as Packet Pair Delay (PPD) [?] and Packet Bunch Mode (PBM) [?] exists in literature. In PPD, a node periodically sends two back-to-back probes to each of its neighbours where, the first probe is smaller and the second one is larger in size. Neighbour nodes measure delay between the arrival of the two probes and report back to the sender. The sender averages the delay samples and estimates the bandwidth. PPD faced certain shortcomings like out-of-order delivery, limitations due to clock resolution, changes in bandwidth, etc., which resulted in a significantly more robust procedure termed as PBM. PBM deals with PPD shortcomings by forming estimates for a range of packet bunch size where packet bunch size ranges from 2 to 5 packets. PBM works by stepping through an increasing series of packet bunch sizes. We consider PBM for link bandwidth estimation.

In the link bandwidth estimation process, link delay is calculated first. The link delay accounts for the propagation, transmission, and link scheduling delay. The process involves active probing where k probe packets each of size equal to Maximum Segment Size (MSS) are sent to the neighbouring node. Let us assume that the probe packets are sent during time interval, Δt_s and Δt_r be the interval during which k packets are received, which necessarily satisfies the condition: $\Delta t_r > \Delta t_s$. If C^r is the receiver clock's resolution, then the value of k is increased if $\Delta t_r < C^r$, i.e., all the arrivals occurred without the receiver's clock advancing. Therefore, expected link delay of link i denoted as E$[L_i^d]$ can be estimated as given in Equation (**??**).

$$E[L_i^d] = \frac{\Delta t_r}{(k-1)} \tag{6.1}$$

where Δt_r and k take values as discussed above.

After calculating the value of $E[L_i^d]$, the receiver node sends this value to the sender node. The sender node uses the value of $E[L_i^d]$ to finally estimate the link bandwidth which is denoted as $E[L_i^b]$. If PP_{size} is the size of each probe packet, then Expected Link Bandwidth (ELB) of a link i, $E[L_i^b]$ can be estimated as given in Equation (**??**).

$$E[L_i^b] = \frac{PP_{size}}{E[L_i^d]} \tag{6.2}$$

ELB gives the maximum capacity of a link. The available bandwidth of a link i changes with the changing traffic load. If $B_{resv}(i)$ bandwidth is already reserved in the link i, the available bandwidth for that link, denoted as $B_{avail}(i)$ can be calculated by using the Equation (??).

$$B_{avail}(i) = E[L_i^b] - B_{resv}(i) \tag{6.3}$$

The lowest available link bandwidth decides the capacity of the path to admit a flow. Therefore, the available bandwidth of a path, π_i is given as

$$B_{avail}(\pi_i) = min\{B_{avail}(1), B_{avail}(2), ..., B_{avail}(i)..., B_{avail}(n)\}$$

We use this end-to-end path bandwidth as *Expected Path Bandwidth (EPB).*

6.4.1.2 End-to-end Delay (E2ED)

We use E2ED [?] as routing metric to ensure minimum delay for delay sensitive traffic. E2ED considers both the link transmission delay as well as the queuing delay of each link in the path from the source to the destination. The transmission delay is defined as the period from the instant that a packet begins to be served by the MAC layer to the instant that it is either successfully transmitted or dropped after a predefined maximum number of retransmissions. The queuing delay is the time interval from the instant that a packet enters the queue till it is served. The E2ED metric also implies traffic load-balancing as the path with a smaller E2ED normally consists of the links with fewer packets in the queues, and thus balances the traffic from those congested links.

If the queuing delay and transmission delay of a packet over the link i are represented as D_i^Q and D_i^T respectively, the average delay which is denoted as D_i can be calculated as

$$D_i = E[D_i^Q + D_i^T]$$

where, $E[D_i^Q + D_i^T]$ is the expected cumulative queuing and transmission delay for link i.

To measure the transmission delay of a wireless link, a node needs to record the time when a packet becomes the head of the queue and the time when the same packet is transmitted or dropped. The transmission delay is also be termed as the service time of a packet. Let $T_{i,n}$ denote the service time of n^{th} packet measured over link i. The average transmission delay of n^{th} packet over link i, denoted as $TD_{i,n}$ can be estimated by the Exponential Weighted Moving Average (EWMA) scheme [?] by using the Equation (**??**).

$$TD_{i,n} = (1-\beta)D^{T}_{i,n-1} + \beta T_{i,n} \quad 0 < \beta \leq 1 \tag{6.4}$$

Here, β is a smoothing parameter which can be chosen appropriately. Using Equation (**??**), the expected average delay over link i, $E[D_i^T]$ can be easily calculated for a given number of packets. If there are Q_i number of packets in the buffer when a new packet enters the queue of link i, the average delay for that packet over link i, D_i can be estimated by using the Equation (**??**).

$$D_i = (Q_i + 1)E[D_i^T] \tag{6.5}$$

Thus, total delay over a link is resulted from the service time of $(Q_i + 1)$ number of packets. Considering an end-to-end path which includes h hops, the E2ED of a path is calculated by using the Equation (**??**).

$$E2ED = \sum_{i=1}^{h} D_i \tag{6.6}$$

In our proposed routing scheme, $E2ED$ is used as a routing metric along with EPB.

6.4.2 Tables Used in the Routing Protocol

Five different tables are used to maintain the routing path and flow related information at various nodes which help in implementing the routing mechanism.

(i) **Routing Table**

It is a default table used in the routing protocols for storing generic routing information at any node. In our protocol, this routing table is used for routing best-effort traffic and forwarding route update information during the route maintenance phase. For each destination, an entry in the routing table is made which is updated in case of any change in the path status. The major fields in the routing table are Destination Address, Interface, Sequence Number, Hops and Next-hop.

(ii) **Path Table**

The source node maintains a set of QoS feasible paths for each destination node corresponding to different real-time traffic classes. On the basis of the path information received through the route reply packets, the source node lists out a number of best paths and stores them in the path table. The major fields in the path table are Path Id, Destination Address, Sequence Number, No. of Hops, Path List, Path Delay and Path Bandwidth.

(iii) **Flow Table**

A flow table is maintained by all the nodes through which there is active path. Flow table, which stores flow related information, is primarily used for admission control and load balancing purposes. The major fields of a flow table are Flow ID, Destination Address, No. of Hops, Path Id, Path List, Bandwidth Requirement and Delay Bound.

(iv) **Source List**

Source list is used in route maintenance process. It stores all the source nodes corresponding to a next hop link which is a part of discovered paths from them. The nodes maintain these information to communicate with the source nodes whenever there is a significant change in path quality. The fields of this table are Source Address and Next-hop.

(v) **Neighbour Table**

Neighbour table is used to store information about all the neighbour nodes of a node. The major fields of a neighbour table are Neighbour Address, Interface, Hop Delay, ELB, Reserve Bandwidth, and Expiry Time.

6.4.3 Discovery of Multiple Paths

In WiLD mesh network, traffic is primarily from client to the gateway node and vice-versa. QGMR discovers and maintains multiple QoS-aware paths between a given source and the gateway node. A node having traffic to send looks for available paths to the destination that meet the QoS demands of that traffic class. If no such path exists, the node discovers a new set of paths using the route discovery mechanism which is discussed as follows.

6.4.3.1 Multi-path Route Discovery

This process uses Multi-Path Route Request (MPREQ) and Multi-Path Route Reply (MPREP) packets to discover multiple loop-free QoS-aware paths between a pair of source and destination nodes. All intermediate nodes maintain *hop delay* and *available bandwidth* to each of their neighbour nodes. The MPREQs traversing through different nodes gradually calculate the *path delay* and *path bandwidth. Path delay* is calculated by adding all the *hop delays* of different hops in the path till the destination. *Path bandwidth* is calculated by taking the minimum of the *available bandwidth* among all the hops till the destination. When a source node needs a path to the gateway node, it broadcasts a MPREQ packet to all its first hop neighbours whose *path delay* does not exceed maximum delay bound Δd and *path bandwidth* is greater than 0. On receiving MPREQ packets, every intermediate node takes part in the calculation of *path delay* and *path bandwidth* and forwards the MPREQ packets only on those next hop links that satisfy the given conditions. It also sets the reverse path to the source node. After receiving a MPREQ packet, the destination node checks for the same condition and if found satisfied, it caches the MPREQ packet. The gateway node waits for a certain duration of time to allow all the MPREQ packets to arrive. Algorithm **??** shows the procedure for processing MPREQ packets. Eventually, all the MPREQ packets meeting the given throughput and delay bounds reach the destination node by following loop-free paths.

Algorithm 8 Algorithm to process MPREQ at any node Q

Input:
Pkt_MPREQ: MPREQ packet from source node P
N_List: Neighbour list at any node Q
R_Table: Routing table at any node Q
ctimer: Request cache timer

```
$Dest \leftarrow$ DESTINATION_REQ(Pkt_MPREQ)
$Src \leftarrow$ SOURCE(Pkt_MPREQ)
Path ← PATH(Pkt_MPREQ)
path_delay ← PATH_DELAY(Pkt_MPREQ)
path_BW ← PATH_BW(Pkt_MPREQ)
if $Src =$ Node $Q$ then
    Discard Pkt_MPREQ
else if $Dest =$ Node $Q$ then
    if $ctimer = 0$ then
        start $ctimer$
    end if
    if path_delay $\leq \Delta d \wedge$ path_BW $> \Delta b$ then
        cache Pkt_MPREQ
    end if
else if Node $Q$ not in Path then
    for all Node u in N_List do
        Add hop_delay(Q,u) to path_delay
        path_BW $\leftarrow min\{$path_BW, avail_BW(Q,u)$\}$
        if path_delay $\leq \Delta d \wedge$ path_BW $> 0$ then
            Add Node $Q$ in the Path
            Send Pkt_MPREQ to node u
        else
            Discard Pkt_MPREQ
        end if
    end for
else
    Discard Pkt_MPREQ
end if
```

The destination node initiates a timer and starts collecting the MPREQ packets having recorded path delay and bandwidth within the bound. As the timer expires, the node selects a set of maximally disjoint QoS-paths and sends out MPREP packets to the source node through each of the selected paths. A matrix is used to maintain all the paths following which the MPREQ packets have reached the destination. The matrix stores all the intermediate nodes of a path in a row. Each column of the matrix represents a unique path. Disjointness of a set of paths is determined by calculating similarity index of intermediate nodes from the corresponding rows of the matrix. Lesser similarity index in intermediate nodes indicates more disjointness among the paths. Each intermediate node receiving MPREP packet sets next-hop to the destination in its routing table and stores the source node for the corresponding next-hop in a source list. When the source node receives MPREP packets, it adds the path information obtained from MPREP in the path table. Finally, the route discovery process ends when all the MPREP packet reach the source node as illustrated in Algorithm **??**.

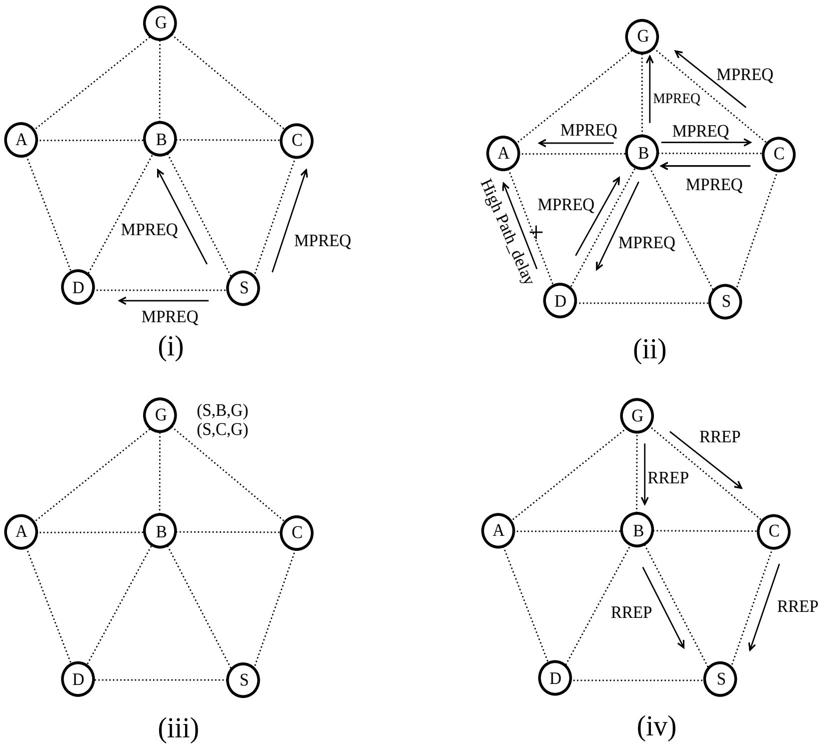
Algorithm 9 Algorithm to process MPREP at any node Q

```
Input:
Pkt_MPREP: MPREP packet received from Node P
S_List: Source list of node Q
P_Table: Path table at node Q

$Dest$ ← DESTINATION(Pkt_MPREP)
$Src$ ← SOURCE(Pkt_MPREP)
Path ← PATH(Pkt_MPREP)
$M$ ← Previous hop of $Q$ in Path
$N$ ← Next hop of $Q$ in Path
if $Dest =$ Node $Q$ then
    Add Path in P_Table corresponding to $Src$
    if Data Pkts buffered at $Q$ then
        Register each real-time flow to Src
        Start Flow
    end if
else
    Add $Dest$ to S_List corresponding to N
    Send Pkt_MPREP to M
end if
```

6.4.3.2 An Example Showing Multi-path Route Discovery Procedure

We now illustrate the multi-path route discovery process with an example. Consider a simple WiLD mesh network with six nodes as shown in Figure **??**. Let us assume that node S wants to establish a path to the gateway node, G. Before sending MPREQ packet to the next-hop, S checks whether the hop delay is less than the tolerable delay bound and the available bandwidth is greater than zero or not. Node S sends MPREQ packets only to those next-hop links for which both the conditions are satisfied. Consider that both the conditions are satisfied for all of its neighbour nodes B, C and D, and hence S sends MPREQ packets to all of them as shown in Figure **??**(i). Nodes B, C and D make a reverse entry in their routing tables and check whether cumulative hop delay (path delay) is less than the tolerable delay and the available bandwidth is greater than zero. Assuming that all the next-hop links of B and C nodes satisfy the given conditions, they send MPREQ packets to all of their neighbours accordingly. However, as $D \rightarrow A$ link is shown not to satisfy the delay requirement (Figure **??**(ii)), D will not forward the MPREQ packet to node A. In this fashion, all the intermediate nodes conditionally broadcast the MPREQ packet step by step. Let us assume that the destination node G receives MPREQ packets through the paths $S-B-G$, $S-C-G$, $S-B-A-G$, $S-B-C-G$ and $S-C-B-G$ by following this procedure. Therefore, node G has a path set $S-B-G$, $S-C-G$, $S-B-A-G$, $S-B-C-G$ and $S-C-B-G$. The gateway node, G selects a set of suitable candidate paths out of the available path set and sends replies through them. As shown in the Figure **??**(iii), the node G selects $S-B-G$ and $S-C-G$ as candidate paths and sends RREP packets to S through them accordingly. G appends the delay and bandwidth value from the corresponding MPREQ packets to the newly generated MPREP packets which are used by the source node in path selection process.



(i)

(ii)

(iii)

(iv)

Figure 6-1: Route Discovery procedure for finding multiple paths from Source to Gateway node

6.4.4 Admission Control and Load Balancing

In this subsection, we have discussed the path selection procedure employed by the source node, flow-level admission control and load balancing scheme used for throughput-bound flows.

6.4.4.1 Path Selection By Source Node

Source node stores the list of maximally disjoint paths to the gateway node in its path table. Before sending a particular class of traffic through the discovered path, it runs Algorithm **??** to choose appropriate path(s) for an i^{th} traffic flow f_i. The algorithm first checks whether the *path delay* is within the tolerable delay bound of the requested flow or not and then checks whether the bandwidth requirement is greater than Δb or not. If both the conditions are satisfied, the reserved bandwidth is updated and the flow is registered in that path. After checking the delay and bandwidth requirements for the path π_i, the requested flow is added to the flow table and *path bandwidth* is updated. Finally, a flow request is sent through the selected path, π_i.

Algorithm 10 Path Selection Algorithm

```
Input:
P_Table: Path table at node Q
F_Table: Flow table at node Q
T_class: Traffic class to which the Flow, $f_i$ belongs to

$Dest \leftarrow$ Destination of the flow, $f_i$
$\Delta d \leftarrow$ Delay Bound of T_class
$\Delta b \leftarrow$ Bandwidth requirement of T_class
Total_BW $\leftarrow 0$
for all Path, $\pi_i$ to Dest in P_Table do
    if $\pi_i$.path_delay $\leq \Delta d$ then
        Total_BW $\leftarrow$ Total_BW + $\pi_i$.path_BW
    end if
end for
if Total_BW $\geq \Delta b$ then
    for all Path, $\pi_i$ to Dest in P_Table do
        if $\pi_i$.path_delay $\leq \Delta d$ then
            Delay $\leftarrow \pi_i$.path_delay
            BW $\leftarrow$ ($\pi_i$.path_BW $\times$ Req_BW)/Total_BW
            Add flow $f_i$ in F_Table with Delay, BW and $\pi_i$
            $\pi_i$.path_BW = $\pi_i$.path_BW $- BW$
            Send Flow Request in the Path, $\pi_i$
        end if
    end for
end if
```

6.4.4.2 Admission Control

The admission control process carries out the task of path reservation and load balancing. All the traffic belonging to Class 1 and Class 2 are required to be registered before they are admitted. However, traffic aggregation technique is applicable to Class 2 traffic. The steps in admission control and load balancing is described in Algorithm **??**. Before starting actual transmission of data, a Flow Request (FREQ) is sent to admit a flow on the selected path. All the intermediate nodes use a flow table to maintain the information about the active flows flowing through them. After finding the *delay bound* and *bandwidth required* of the requested flow, the source node checks whether the *delay bound* is within the *path delay*. If the requested traffic belong to Class 1 and the delay bound can be met, it is admitted in a single path. For Class 2 traffic, it is handled as follows. If m number of paths are found to meet the delay bound, the cumulative path bandwidth for all m paths is calculated as *total bandwidth*. If the *total bandwidth* is greater than or equal to *bandwidth requested*, the flow can be admitted on m paths. The value of m can be carefully chosen based on the requirement. The admission of a flow involves the following activities: *bandwidth share* calculation for different selected paths, adding the flow to the flow table and updating the *available bandwidth*. Finally, the FREQ packet is forwarded to the next-hop node through the selected paths.

Upon receiving a FREQ packet, an intermediate node tries to forward the same in the corresponding next-hop for the specified path. It reserves requested bandwidth for the flow and updates its *available bandwidth*. If an intermediate node can successfully forward a FREQ packet, it enters the flow in its flow table. Otherwise, it sends a Flow Reject (FREJ) packet to the source indicating failure. After receiving FREJ packet, the source node re-initiates the process of flow registration. Finally, on receiving the flow request FREQ, the destination accepts the flow. Upon successful allocation of a new transmission flow, the path status is updated immediately. In that process, based on the availability of resources, the best paths are always selected for transmission.

6.4.4.3 Load Balancing

Bandwidth bound traffic can be distributed over the number of QoS feasible paths available. The proposed protocol forwards delay bound traffic through the best of the available paths even though multiple paths are available for a given real-time flow. However, throughput bound traffic are distributed among a number of QoS feasible paths available between the source and the gateway nodes, and the packets are aggregated at the destination node. Bandwidth aggregation provides greater bandwidth than an individual path can provide.

Algorithm 11 QGMR: Admission Control Algorithm

Input:
flow_req: Flow request at node Q
F_Table: Flow Table at node Q

```
$f_i \leftarrow$ FLOW_ID(flow_req)
$Dest \leftarrow$ DESTINATION(flow_req)
$delay\_bound \leftarrow$ DELAY_BOUND(flow_req)
$bw\_req \leftarrow$ BANDWIDTH_REQ(flow_req)
if flow_req originated at node Q then
    $S \leftarrow$ set of path selected using path selection Algorithm
    $Total\_BW \leftarrow$ Total BW available in $S$
    if $Total\_BW \geq bw\_req$ then
        for all Path, $\pi_i$ in $S$ do
            $N \leftarrow$ Next-hop of Q in $\pi_i$
            $avail\_BW \leftarrow$ BW available from Q to N
            if $\pi_i$.path_delay $\leq delay\_bound$ then
                $bw\_share \leftarrow (\pi_i.\text{path\_BW} \times bw\_req)/Total\_BW$
                Add flow, $f_i$ in F_Table
                $avail\_BW = avail\_BW - bw\_share$
                Send Pkt_FREQ to N
            end if
        end for
    end if
else if $dest =$ Node Q then
    Receive Pkt_FREQ
    Accept the flow request
else
    RECEIVE(Pkt_FREQ)
    $\pi_i \leftarrow$ PATH(Pkt_FREQ)
    $N \leftarrow$ Next-hop of Q in $\pi_i$
    $avail\_BW \leftarrow$ BW available from Q to N
    if $avail\_BW \geq bw\_req$ then
        Add flow, $f_i$ in F_Table
        $avail\_BW = avail\_BW - bw\_req$
        Send Pkt_FREQ to N
    else
        Discard Pkt_FREQ
        Send flow reject to the source node
    end if
end if
```

6.4.4.4 An Example Showing the Working of Admission Control Mechanism

Figure **??** shows the working of the admission control procedure employed in the routing scheme. Let us consider that the node S wants to introduce a bandwidth bound real-time flow to destination G. For this purpose, S selects $S - B - G$, $S - C - G$ and $S - D - A - G$ as probable paths. Node S initiates the admission control process by sending FREQ packets to nodes B, C and D providing them the details about the flow requirements and path information (Figure **??**(i)). The intermediates nodes (B, C and D) then check whether the new flow can be admitted or not. Suppose the nodes B and D can support the given requirements and hence reserve the required bandwidth and forward the received FREQ packets towards G as shown in Figure **??**(ii). Since node C cannot satisfy the given requirements, it does not forward the FREQ packet further. Instead, it sends a FREJ packet to node S. Finally, node G receives the FREQ packets through the paths $S - B - G$ and $S - D - A - G$ and accepts the flow accordingly. Now, the traffic belonging to the said flow can be shared between the paths $S - B - G$ and $S - D - A - G$.

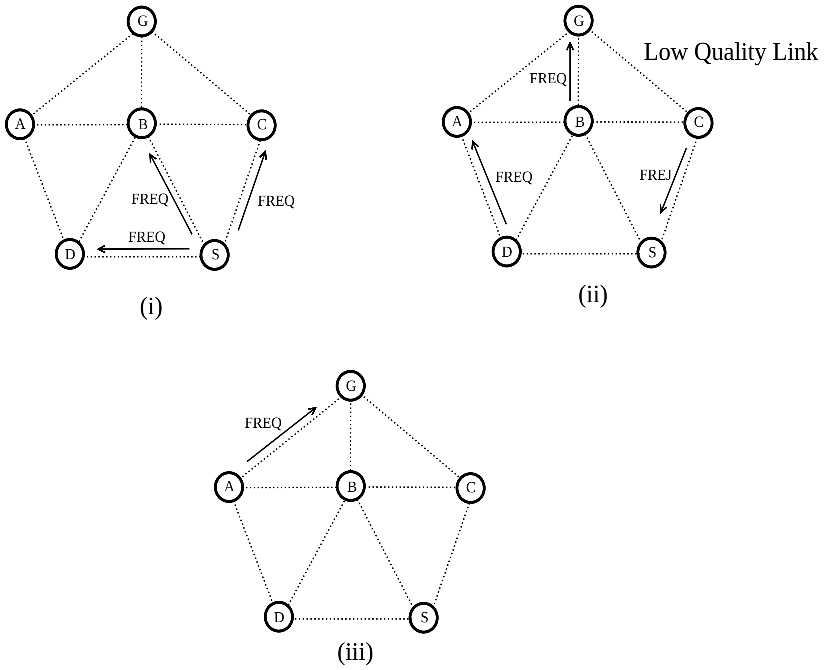


Figure 6-2: Admission Control and Load Balancing in QGMR: An Example

6.4.5 Maintenance of Multiple Paths

QGMR periodically maintains the quality of the paths discovered by the route discovery procedure. The procedure for maintenance of multiple paths is discussed in this subsection.

6.4.5.1 Multi-Path Route Maintenance

Route maintenance process monitors and maintains the various discovered paths. For this purpose, the intermediate nodes use *source list* table to store the list of sources corresponding to paths going through it. Algorithm **??** shows the different steps of route maintenance process. For each MPREP packet traveling through an intermediate node, it creates a source list entry corresponding to the next-hop link through which the MPREP packet reaches the source node. All the nodes periodically monitor the quality of their neighbouring links. The monitoring process checks the *hop delay* and *ELB* for each link. Three types of Route Update (RUPD) packets are triggered on different situations viz., on failure of a link, if path not found while forwarding a packet, and a considerable change in the quality

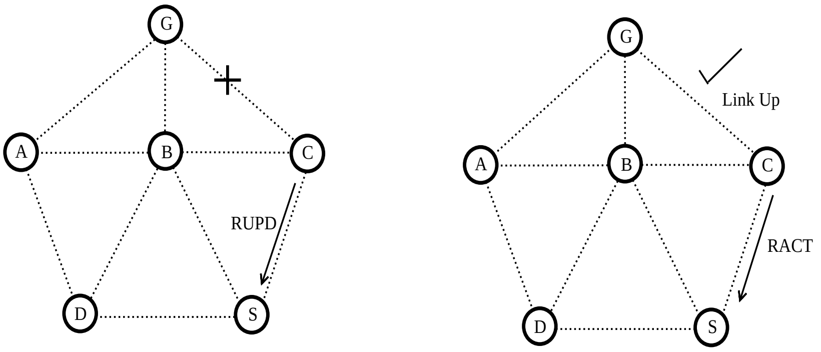
Algorithm 12 Algorithm for Multi-path Route Maintenance

```
Input:
S_List: Source List at node Q
N_Table: Neighbour table at node Q
F_Table: Flow table at node Q
$\delta d$: Threshold in delay change
$\delta b$: Threshold in bandwidth change

Case : 1 (CHANGE IN LINK QUALITY)
for all Neighbour, $u$ in N_Table do
    $ch\_delay \leftarrow |u.cur\_delay - u.prev\_delay|$
    $ch\_bw \leftarrow |u.cur\_bw - u.prev\_bw|$
    if $(ch\_delay \geq \delta d) \vee (ch\_bw \geq \delta b)$ then
        $S \leftarrow$ set of affected sources in S_List
        for each $v$ in $S$ do
            Generate RUPD packet
            Add $u$ in the RUPD packet
            Add $ch\_delay$ and $ch\_bw$ in the packet
            Set flags to PATH_UPDATE
            Send_Pkt RUPD packet to node $v$
        end for
    end if
end for

Case : 2 (BROKEN LINK)
if Neighbour, $u$ deleted in N_table then
    $S \leftarrow$ set of affected sources in S_List
    for each $v$ in $S$ do
        Generate RUPD packet
        Add $u$ in the RUPD packet
        Set flags to BROKEN_LINK
        Send_Pkt RUPD packet to node $v$
    end for
end if

Case : 3 (NO PATH FOUND)
if Flow entry not found in F_Table for source, $v$ then
    Generate RUPD packet
    Set flags to PATH_NOT_FOUND
    Send_Pkt RUPD packet to node $v$
end if
```



(i)

(iii)

Figure 6-3: Multi-path Route Maintenance: An Example

of a path. RUPD packets are sent to all the sources corresponding to the affected next-hops in the source list.

When a link failure is detected, the adjacent nodes inform the source nodes by sending *broken link* message through RUPD packet. On receiving this message, the source nodes update their paths in their respective path lists. Secondly, when an intermediate node does not find any discovered path to forward packets of a flow, then it responds the source node by sending a *path not found* message through RUPD packet. If the monitored delay and throughput metric are found to be changed considerably from the assumed threshold value, the adjacent nodes send RUPD packet to the source nodes with *path update* flag set. The RUPD packet contains information about the change which has recently occurred. On receiving RUPD packet, a source node incorporates the corresponding changes in the appropriate paths.

6.4.5.2 An Example Showing Multi-path Route Maintenance Procedure

On detection of link failure, the adjacent nodes update their routing tables by setting the rank field to ∞ and sends a RUPD message immediately to the relevant neighbour nodes to update them regarding the recent failure. In Figure **??**(i), node C sends RUPD message to node S. On receiving the RUPD message, the neighbour nodes update their own routing tables and forward the message towards the source node. The source node finally receives the RUPD message and updates the path table accordingly. When a failed link comes up, the connected nodes update their routing tables and inform their neighbour nodes about the path activation by sending RACT (Route Activation) message as shown in Figure **??**(ii).

6.5 Simulation and Performance Evaluation

In this section, we evaluate the performance of QGMR protocol and compare with AOMDV protocol. From the literature survey, it can be seen that the existing protocols are either designed for different settings or consider different QoS parameters individually. MMESH [**?**] is designed for multi-gateway based networks whereas we consider only a single gateway. This protocol has considered link quality change as a metric for route maintenance. QoS-MOLSR [**?**] is a Link State Routing (LSR) based protocol which considers end-to-end delay as a routing metric whereas the proposed protocol takes a reactive approach. MP-DSR [**?**] discovers multiple paths considering end-to-end reliability as a routing metric. SMR [**?**] is a source based routing in which the route maintenance procedure is highly centralized. AOMDV [**?**] is a simple multi-path routing protocol without having any specific QoS feature except reliability delivered through its multi-path property. While discovering multiple paths, it maintains path-disjointness and loop-freedom properties which are also considered in QGMR. With these fundamental similarities and some clear cut differences in QoS features, we have compared the performance of QGMR with AOMDV. Although AOMDV discovers multiple paths, it uses only one path at a time. Whereas QGMR discovers multiple paths based on QoS requirement of various flows and uses those paths for transmitting throughput-bound traffic using packet aggregation technique. We have evaluated and compared the performance of QGMR and AOMDV protocols considering the performance metrics and simulation parameters as discussed in the next section.

6.5.1 Performance Metrics and Simulation Parameters

For performance evaluation, we have considered the following metrics-

(i) Control Overhead (CO): Control overhead is a cost of discovery and maintenance of paths for a routing protocol. It is measured in terms of number of control packets required to perform this task.

(ii) Throughput (TP): Throughput refers to the average number of successfully delivered bytes at the destination per second. It is an important metric to provide minimum level of service in a network.

(iii) Delay (D): It is the time difference between the time a packet was delivered at the destination and it was sent by the source. Delay is a very important parameter for delay sensitive real-time traffic.

(iv) Protocol Reliability (PR): This parameter measures the reliability of routing protocol. It verifies how a routing protocol tackles path failure situation and continues providing the ongoing services.

We have performed an extensive simulation in ns-2 to evaluate the performance of the proposed protocol. A modified version of ns-2.34 [**?**] has been used for this purpose. We have augmented multiple interfaces support to AOMDV to support traffic forwarding over multi-hop WiLD networks. A mesh network topology as shown in Figure **??** is used for the simulation work where nodes are connected by long distance point-to-point links. The distance covered by each link is about $9kms$. The simulation is carried out for a duration of 300 $seconds$.

Half-duplex WiLD links with $11Mbps$ bandwidth are used for establishing communication between adjacent nodes. VoIP flow with packet size of $160bytes$ generated at an interval of $20ms$ is used for the simulation of *Class 1* traffic. To simulate *Class 2* traffic, we have used video streaming traffic with $1250bytes$ packet size with arrival rate of $33ms$. *CBR* traffic has been used to simulate best-

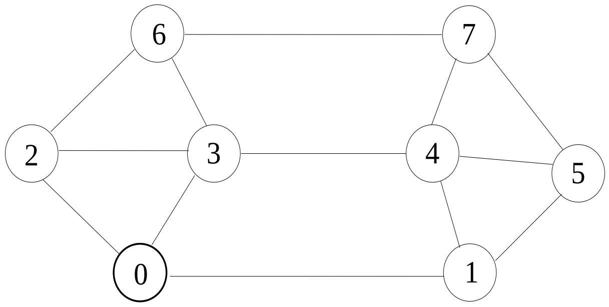


Figure 6-4: Simulation Topology for QGMR

effort (*Class 3*) traffic. Table **??** gives the simulation parameters considered in this simulation study.

Table 6.1: Simulation Parameters for QGMR

Parameters	**Values**
Simulation Area	$50kms \times 50kms$ Flat-grid Area
Hop Distance	$9kms$
No. of Nodes (Max.)	8
Antenna	Directional
Packet Size	160-1400$bytes$
Link Bandwidth	$11Mbps$
MAC Protocol	$2C$
Routing Protocol	$AOMDV, QGMR$
Application	$CBR, VoIP, VBR, FTP$

6.5.2 Results Analysis

Performance of the proposed protocol is evaluated and results are compared with AOMDV routing protocol. At first, we have compared the control overhead (CO) of QGMR and AOMDV protocols. Then, throughput and delay characteristics are presented without considering any path failure. Finally, the impact of path failure on QGMR and AOMDV protocols are analyzed and compared through the evaluation of throughput characteristics.

6.5.2.1 Control Overhead of QGMR and AOMDV

In this experiment, we aim to find the control overhead of QGMR and AOMDV protocols in route discovery and maintenance process. This experiment is con-

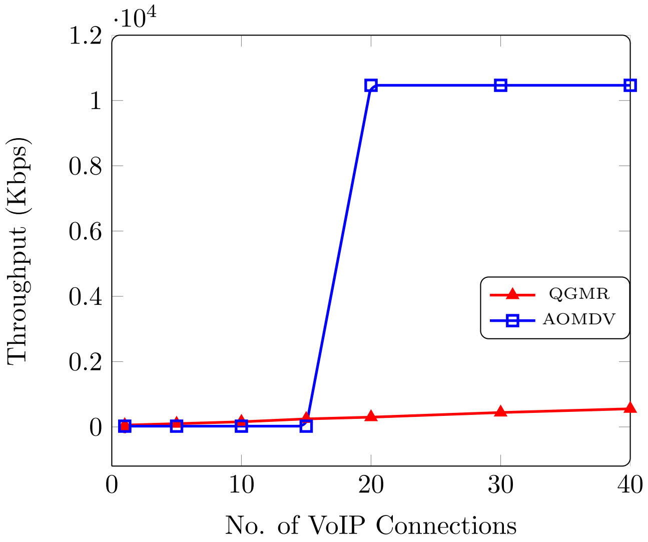


Figure 6-5: Control overhead of QGMR and AOMDV protocols

ducted by increasing the number of video streaming traffic gradually in the considered simulation topology. The different control packets used in AOMDV include RREQ, RREP, and RERROR packets. Whereas QGMR uses MPREQ, MPREP, FREQ, FREJ, and RUPD packets.

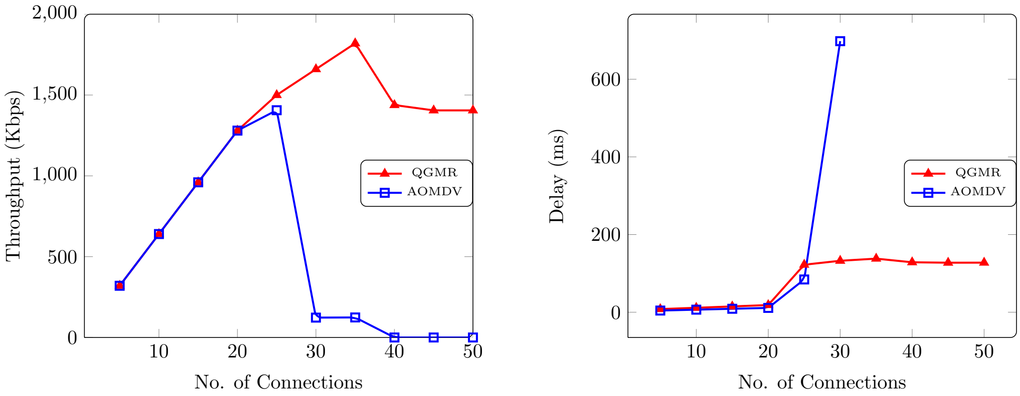
At low load, number of control packets used in AOMDV is smaller than that of QGMR. Up to 15 number of connections, this figure is constant for AOMDV whereas QGMR needed more control packets with increasing number of connections. The reason for this is twofold. First, to find node-disjoint paths, QGMR does not restrict the broadcasting of packets as it explores all the possible paths. Second, it uses additional FREQ and FREJ packets to confirm quality of the discovered paths. The FREQ and FREJ packets attribute to the total number of control packets significantly. Just after crossing 20 numbers of connections, the number of control packets in AOMDV reaches a very high value. Actually, this is the saturation level of the network after which congestion occurs at different nodes and RERR messages start generating in AOMDV. It shows that at high load situation, the control overhead of AOMDV is much higher than QGMR. The results are shown in Figure **??**.

6.5.2.2 Throughput and Delay Characteristics of QGMR and AOMDV

Here, we have evaluated the throughput and delay performance of VoIP and video conferencing for QGMR and AOMDV protocols separately.

VoIP Performance

Figure **??** shows throughput achieved by Class 1 traffic with gradual increase in number of connections. In both the protocols, the throughput increases with the increase in number of connections. But, in overloaded situation, i.e., after 25 numbers of connections, the throughput achieved by AOMDV routing protocol degrades drastically. On the other hand, QGMR maintains a stable throughput level even beyond the saturation point due to its integrated admission control mechanism.



(a) Throughput

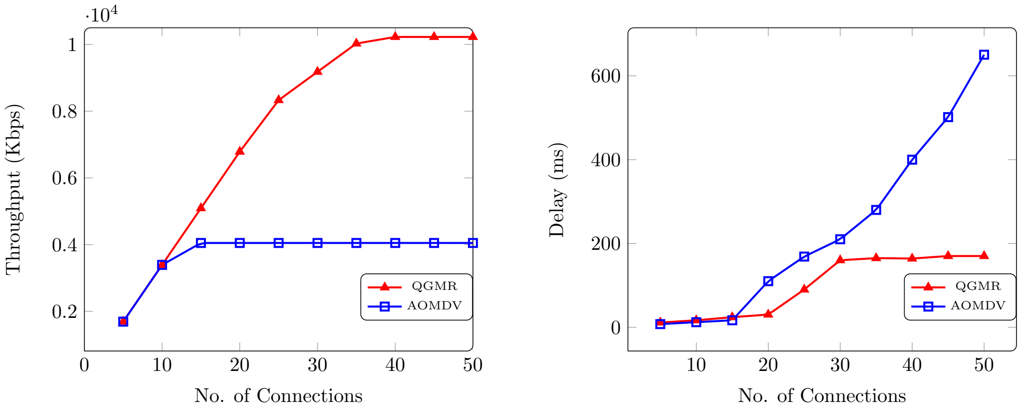
(b) Delay

Figure 6-6: VoIP Performance: QGMR vs. AOMDV

As shown in Figure **??**, delay of VoIP traffic is well within the bounds i.e., 150ms. But in AOMDV, VoIP delay increases drastically after crossing its saturation point.

Video Streaming Performance

The results of this experiment are shown in Figure **??**. For Class 2 traffic, with a gradual increase in number of connections, both the protocols display a constant increase in throughput until reaching 10 connections. After that point, the throughput of AOMDV protocol does not increase and get restricted around a certain value. QGMR shows further increase in throughput up to $10Mbps$. It is observed that the throughput achieved by QGMR exceeds the theoretical achievable network throughput. Use of more than one path for the same flow has attributed throughput of higher magnitude. The reason for getting additional increase in QGMR throughput is due to the parallel use of multiple disjoint paths to achieve aggregated throughput performance. The above results prove the suitability of QGMR routing protocol for bandwidth-greedy traffic.



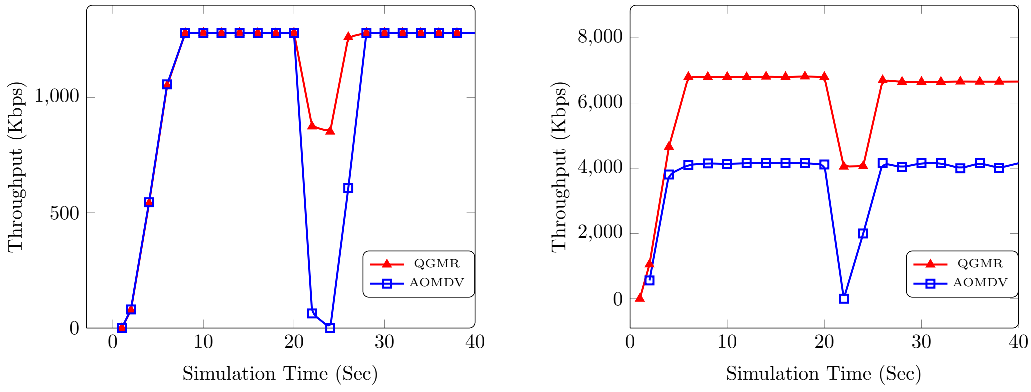
(a) Throughput **(b)** Delay

Figure 6-7: Video Streaming Performance: QGMR vs. AOMDV

In Figure **??**, delay of video streaming traffic in QGMR is also observed to be better than AOMDV and remains within the delay bound for video streaming traffic up to saturation level. Due the packets splitting and aggregation, QGMR shows a very small performance lip with AOMDV with a few connections. However, these values are small and hardly have any impact in the quality of transmission.

6.5.2.3 Throughput Characteristics of QGMR and AOMDV in Path Failure Situation

In this part of evaluation, our aim is to observe the impact of path failure over transmission. All three traffic classes are introduced in this experiment. In this scenario, throughput of VoIP, video streaming and best-effort traffic are observed with respect to simulation time. To simulate path failure, node 4 is turned off



(a) VoIP

(b) Video Streaming

Figure 6-8: Throughput Performance of VoIP and Video Streaming using QGMR and AOMDV with Path Failure

for instance during simulation. The experiments are completed in two phases. In the first phase, we evaluate the throughput performance of different traffic classes individually for QGMR and AOMDV routing protocol considering 20 connections. In the second phase, we evaluate the performance by giving traffic of all three classes simultaneously.

Throughput performance of VoIP, Video Streaming and Best-effort Traffic

From the Figure **??**, it can be observed that VoIP throughput in our proposed protocol recovers itself after a small declination during path failure. The throughput is computed taking the average of every $2ms$. Since QGMR recovers within less than $2ms$ time from failure, it did not touch 0. But, throughput in AOMDV goes close to zero as it takes about 4 *seconds* to recover from failure.

Video streaming also shows fall in throughput but has not become zero as the packets are distributed among multiple paths and aggregated in the destination. This is shown in Figure **??**.

In Figure **??**, best-effort traffic performance in path failure situation is shown. AOMDV throughput has been reduced drastically during path failure. However, the throughput performance of QGMR is not even adjusted a little. It is interesting to note that due to which path failure VoIP and video streaming traffic have suffered, the same path was not used by the best-effort traffic. This is how the impact of path failure is reduced by our protocol.

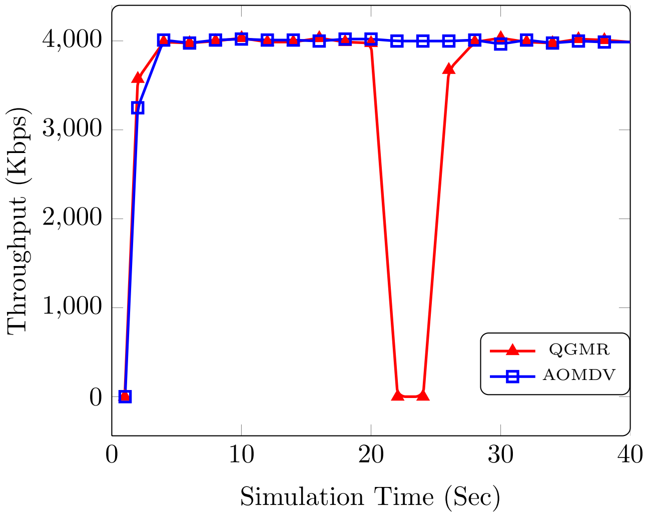
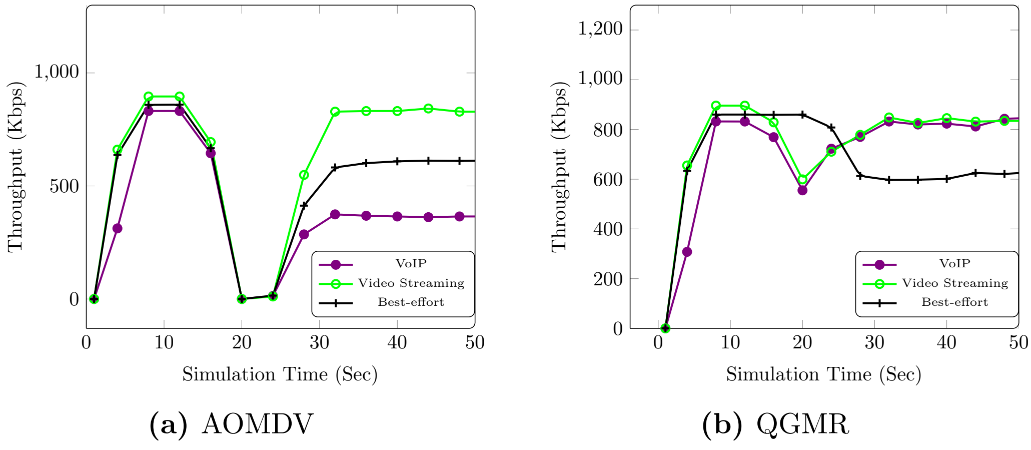


Figure 6-9: Throughput Performance of Best-effort traffic using QGMR and AOMDV protocols with path failure

Finally, the throughput performance of VoIP, video streaming and best-effort traffic in both QGMR and AOMDV protocols are observed by combining all of them together in path failure situation as shown in Figures **??** and **??**. In AODV, all traffic classes suffer equally due to path failure. However, QGMR shows a diminished affects of failure by using traffic aggregation technique.



(a) AOMDV

(b) QGMR

Figure 6-10: Throughput Performance of VoIP, Video Streaming and Best-effort Traffic in AOMDV and QGMR protocols considering Path Failure

6.6 Conclusion

In this chapter, we have proposed a QoS-aware gateway-based multi-path routing protocol in supporting smooth real-time flow over multi-hop WiLD networks. The key contributions of this chapter are as follows.

- An integrated approach has been used in determining delay and bandwidth metric on each hop in order to meet the delay and throughput requirements of real-time traffic on end-to-end basis. Traffic flows are classified based on their characteristics, and delay and bandwidth bounds are determined accordingly.

- A novel multi-path route discovery process has been proposed using which multiple maximally disjoint QoS-feasible paths are discovered for QoS-aware traffic.

- A flow based admission control and load balancing scheme has been introduced in the routing protocol which enhances reliability on QoS provisioning. The proposed path selection scheme finds appropriate paths for a particular class of traffic.

- It also incorporates a route maintenance process to handle path quality change and path failure update. With an additional control overhead in terms of flow request/reply and maintenance of multiple paths, this protocol achieves much higher aggregate end-to-end bandwidth and significantly improved delay which ensure some assured level of QoS for real-time traffic.

- Link failure situations are efficiently handled by the proposed protocol. Impact of path failure has been significantly reduced.

- Support-ability of bandwidth-bound applications over multi-hop WiLD networks has been improved manifold through the use of multiple paths simultaneously for a single flow.