CHAPTER 4

QoS DRIVEN ONLINE MULTICAST ROUTING PROTOCOL

Recent work on multicast routing protocols in the Internet has introduced new notions of building online multicast tree without prior knowledge of the future requests. Online multicast routing protocols are aimed to accommodate more number of multicast requests (scalability) into the network by preserving bandwidth on frequently used links and generate long-hop multicast trees compromising the minimum hop length.

On the other hand, there is an upsurge to develop QoS models with service level differentiation for group applications in order to better utilize the network resources. In service level differentiation, instead of treating the multicast requests in the similar manner, different priorities for requests have been assigned based on QoS requirements while routing in a network.

In this research, a QoS driven Online Multicast Routing Protocol (QOMRP) has been proposed and developed, wherein alternate path rerouting of multicast session is incorporated instead of preserving bandwidth on frequently used links in order to satisfy maximum number of requests. This ensures the scalability in number of multicast sessions and hence increases the overall throughput. The path for rerouting is selected based on the QoS priorities of the ongoing multicast session in the path. The present work focuses on classifying the multicast requests into three classes namely, Class A, Class B and Class C in order to provide high, medium and normal service levels respectively. The QoS priorities are assigned based on the bandwidth and end-to-end delay requirements of multicast requests by the application in the source. This ensures efficient utilization of the network resources.
and accepts more multicast requests. The subsequent sections of this chapter deal with the online multicast routing protocol in detail.

4.1 Motivation

In existing multicast routing protocols, as and when the network receives the multicast request, it will be accommodated by constructing the shortest path multicast tree. The multicast request normally contains multicast source and set of multicast receivers. If more number of multicast requests is received in the network, links that falls in the shortest path of most of the requests will be occupied with more multicast sessions. Due to this, the most preferred link in the network runs out of bandwidth and thereafter no new multicast requests are accepted due to non-availability of bandwidth in the shortest path link to construct the multicast tree.

To address this problem, M. Kodialam, et al., [27] have suggested a heuristic approach to identify the critical links that are heavily loaded in most of the multicast requests and preserve them for future requests thereby accommodating more multicast sessions. An online routing algorithm [27] has been designed specifically for Virtual Private Network (VPN) and Multi Protocol Label Switching (MPLS) applications. Though the algorithm accommodates more multicast sessions, it unnecessarily chooses a long path avoiding most used critical path. Moreover, the algorithm may not yield always better performance, as the persevered links may not be used at all.

D. Zappala [28] has proposed an alternate path routing protocol. In this protocol, if the network is not able to accommodate a multicast request in the shortest path, meeting the requirements, it can choose an alternative path routing to construct the multicast tree. However, these protocols [27, 28] do not address the QoS issues with service level differentiation in order to increase the multicast requests accepted. Most of the current research on QoS guaranteed multicast routing focus on either delay/bandwidth-constrained multicast routing or on satisfying more number of requests [64, 84, 85]. This forms the basis for developing a QoS aware...
online multicast routing protocol, which maximizes the number of multicast sessions in wired network.

4.2 Overview of the Protocol

Generally multicast routing protocol constructs a multicast tree and reserves bandwidth along the tree. When there is not enough bandwidth on a link in the tree, the protocol rejects the multicast request even if there is abundant bandwidth available on other paths leading to the receiver. In order to overcome such shortcomings, alternate path techniques were proposed which would utilize these alternate paths in order to satisfy the request. However, the alternate path approach generates long-hop trees since they by-pass the links which fall on the shortest path. Though this method makes efficient use of the bandwidth, it generates undesirable long-hop trees which do not prove to be suitable for time critical applications.

To overcome this drawback, an improvement has been proposed in alternate path routing approach. The new approach makes use of the alternate paths available in the network and takes wiser decisions to identify which class of request needs to be rerouted along the alternate path. In the present approach three levels of priorities are assigned to the multicast requests and based on the priority levels different treatment will be provided to the requests. Thus, multicast applications which demand a strict time constraint in terms of bandwidth and delay (Class A) are always guaranteed a shortest path tree. Other applications with lower priorities (Class B and Class C) are provided the shortest path tree when there are enough resources available otherwise they are rerouted along the alternate path in the network. The algorithm and the protocol operation are explained with proper illustrations in the following sections.

4.3 The Protocol

The proposed QoS driven online multicast routing protocol assumes that the routing requests are generated with varied bandwidth and delay tolerance.
requirements These requests are classified into three different classes. Once a request arrives, the first step is to construct the Minimum-Hop multicast tree and the next step is to route the request along this tree i.e., reserve bandwidth for the tree. If there is no bandwidth on any link in the tree, then a search is initiated to find an alternate path and the request is routed along the alternate path. If no such alternate path is found, then the request is rejected.

4.3.1 QOMRP Algorithm

Multicast routing request arrives one by one in the network and is defined by a set of values \( \{s, R, b, c, m\} \), where \( s \) is source of the multicast, \( R \) is set of receivers, \( b \) is bandwidth, \( c \) is class of the multicast request and \( m \) is minimum number of receivers that must present in the multicast tree in order to continue the multicast session. In the protocol, the Search_AltPath is a technique that uses the Local Search technique [61, 64], which tries to search for neighbouring on-tree nodes by flooding Request (REQ) messages and connects using Response (RES) messages. The procedure terminates unsuccessfully, if the node does not receive any replies before the expiration of a TTL value. A probe message is sent to the receiver for which the alternate path is to be found and the receiver that receives the probe message starts a Local Search Process to find an alternate path to any on-tree node. The algorithm for the QOMR protocol is given below.

```c
/* R = \{set of multicast receivers\}
   s = Source of multicast
   b = Bandwidth requirement by the source
   c = Class of the multicast request
   m = Minimum number of receivers that should remain on the tree
   P = Set of nodes along the path
   L = Number of receivers in set R \& n(R) */
```
QOMRP \((s, R, b, c, m)\)

\[
\{
\text{for all receivers } R_i \in R, \text{ where } i = 0 \text{ to } L \\
\text{P} = \text{Compute\_ShortestPath} (s, R_i = 0 \text{ to } L) \\
// P = \{S, n_1, n_2, n_3 \cdots R_i\} \text{ i.e., P represents a path with intermediate nodes} \\
\text{n} = 0 \\
\text{for all the nodes in P /* P contains multi-hop path from s to R_i */} \\
\{
// Intermediate links between pair of nodes in P for instance (n) and \(n+1\)
\text{Result} = \text{Allocate} ((P(n), P(n+1)) \\
\text{if (Result} = \text{false}) \\
// P_{\text{alt}} \text{Contains nodes along the alternate path}
\text{P}_{\text{alt}} = \text{Search\_AltPath}(s, R_i, P(n), P(n+1)) \\
\text{if (P}_{\text{alt}} = \{\}) \text{ then}
\text{write "Cannot Allocate"}
\text{else}
// * Alternate path nodes are assigned to P for \(n\) and \((n+1)\) */
\text{P} = \text{P}_{\text{alt}} \\
\text{break}
\text{end if}
\text{end if}
\}
\}
\]

The algorithm incorporates a shortest path procedure, \("\text{Compute\_ShortestPath()}\"\) to find the shortest path satisfying the QoS requirements of the requested multicast session. The algorithm takes the multicast request as an input in the form \((s, R, b, c, m)\) For instance, \((n_3, \{n_1, n_0, n_8, n_7\}, 5, B, 3)\) where \(n_3\) is the multicast source node, \(\{n_1, n_0, n_8, n_7\}\) is the set of receivers, \(5\) is the bandwidth requirement, \(B\) is the service Class B and \(3\) is the minimum number of receivers required to continue the multicast. Using the input, multicast tree is
constructed. During the computation of multicast path between pair of nodes, in case if the required QoS is not able to meet, the procedure ‘Allocate()’ returns false as a result. It means that along the intermediate path, the required QoS cannot be met due to want of resources. In such a case, the procedure ‘Search_AltPath()’ is invoked to find an alternate path based on QoS class to either new multicast session or for the ongoing low priority multicast session along the path. This procedure finds an alternate path between any pair of links in the path between source s and receiver R. If the Search_AltPath() procedure returns an empty path, it means that the new multicast request cannot be allocated due to non-availability of QoS path even with alternate path rerouting. More elaborate execution of the intermediate steps required to complete alternate path rerouting process is explained hereunder.

In the algorithm, as soon as a multicast request is received the shortest path tree is constructed which connects the source and all the destinations to the tree. In case for a specific path if the shortest path is not available due to lack of network resources, the algorithm gets into the alternate path search mode. Alternate search is designed cost-effectively as it is the key for scalability. Single path and multi path search modes have been implemented for alternate path search. These search phases have been incorporated in QMRP [58]. Though the process of searching multiple paths is inefficient, the chances for best possible QoS path could be available with multiple paths. Multi path search is necessary in order to increase the chance of successful search.

In QOMRP, when an alternate path search is required, REQ messages are sent to the neighbour nodes. The search process starts with REQ messages, attempting to find alternate paths. The REQ message carries the QoS requirements, e.g., a bandwidth and end-to-end delay. As it travels, it checks the resource availability of every intermediate node and proceeds only when the node has the required resources. If every node on the path has the required resources, the path is selected as an alternate path between the source and the receiver. Once a feasible path is detected, an RES message is sent back along the branch toward the receiver.
The RES message collects the dynamic QoS metrics along the alternate path. The alternate path search mode works differently for three classes of multicast requests.

When routing a Class C request, the main focus is to accommodate a predefined number of receivers from the set of receivers. The minimum number of receivers, which have to remain on the tree, is defined at the time of making the multicast request. During the multicast tree construction for a Class C request, if there is no bandwidth on a particular link, then a search is made for an alternate path and the request is routed along the alternate path. If the network is not able to accommodate the minimum agreed number of receivers, the multicast request is simply rejected.

When routing a Class B request, if there is no bandwidth on a particular link, then a search is made for an alternate path to bypass the link. If such an alternate path exists, then the request is routed along this path. Otherwise, a Class C application currently using the link is rerouted in an alternate path to give way to Class B request. If Class C is not finding an alternate path, it is freed on the link to accommodate the new Class B request subject to satisfying the minimum assured number of receivers in Class C.

For Class A request, if there is no bandwidth on a particular link in the tree, then a search is made for a Class B or a Class C application, which is already using this link. The sum of the bandwidth of the Class B or Class C request and the remaining bandwidth in the link is greater than or equal to the bandwidth requirement of the Class A, the request will be allocated. If the bandwidth cannot be met, the chosen Class B or Class C application is redirected along an alternate path. Then, this freed up bandwidth is used for the Class A application. When there is no alternate-path bypassing the congested link, then a Class C application currently using the link is chosen and the receiver that obtains data through this link is cut-off from the tree. This freed up bandwidth is used for the Class A request. This Class C receiver is removed only if the number of remaining Class C receivers in the tree does not fall below a certain percentage as negotiated earlier. In any of the above...
case does not arise, then the Class A request is rejected. Thus, it is not possible to accommodate the new request but also provide the desired QoS for the Class A request without degrading the service provided for the existing users. The several scenarios of alternate path rerouting are explained with illustrations in the next section.

4.3.2 Illustrations

Typical scenarios with three different classes of multicast requests and alternate path routing procedures are illustrated with examples. For illustrative purposes, a standard National Scientific Foundation (NSF) network with 14 nodes labeled with n0, n1, ..., n13 has been considered.

In Scenario 1, it is assumed that a Class C request arrives to the network and there is no bandwidth available in the shortest-path tree. The snapshot of the requests arriving at a network is given as (n3, {n1, n0, n8, n7}, 6, A, -) and (n10, {n1, n0, n4, n9}, 5, C, 2). It is also assumed that the residual bandwidth on all the links to be 10 units.

When the first request arrives, the shortest-path tree for the source n3 and its set of receivers {n1, n0, n8, n7} is generated and a bandwidth of 6 units is allocated along the path. When the next Class C request arrives, the bandwidth on the link n1-n0 (which falls in the shortest path of the request) is not sufficient to allocate the request. As the request is Class C, an alternate-path is found which bypasses this link n1-n0 and the shortest such alternate path available is the path n10-n12, n12-n2, n2-n0. The Class C request is then routed along this path. Figure 4.1 (a) shows the shortest-path trees, when the requests would have taken, if there were enough bandwidth on the links. Figure 4.1 (b) shows the rerouted path allocated for the Class C request, which arrives later and the shortest path tree allocated for the Class A request.
In Scenario 2, a Class B request is given to the network and enough bandwidth is not available on the shortest-path tree. The snapshot of the requests arriving to the network is given as (n3, \{n1, n0, n8, n7\}, 6, C, 3) and (n10, \{n1, n0, n4, n9\}, 5, B, -). It is assumed that the residual bandwidth on all the links is 10 units before routing the two new requests.

When the Class C request arrives to the network, the shortest path tree for the source n3 and its set of receivers \{n1, n0, n8, n7\} is generated. A bandwidth of 6
units is allocated along the path. For the next Class B request, there is not enough bandwidth available on the link n1-n0, which falls in the shortest path tree. As the request is Class B, search for an alternate path, which bypasses the link n1-n0 is started. Assume that there is no such alternate path available for the Class B request with the required bandwidth. Then, Class C request that uses this link is identified and an alternate path search has to be made by the receiver of the Class C request, i.e., node n0. If the search is successful, the Class C is rerouted along the alternate path. If the search is unsuccessful due to lack of bandwidth, the only option left is to remove the Class C session using this link. Before doing this, the number of remaining Class C receiver is checked to see if it falls below 3, i.e., the minimum number of receivers for the multicast session. If the number of receivers is not below the minimum required level, the receiver is released.

Figure 4.2 (a) shows the shortest path trees, which the requests would have taken, if there were enough bandwidth on the links. Figure 4.2 (b) shows the Class C receiver cut-off from the multicast tree and the shortest path is allocated to the Class B request.

Scenario 3 illustrates the situation when a Class A request arrives to the network and enough bandwidth is not available on the generated shortest path tree. The requests (n3, {n0, n5, n11, n12, n13}, 6, B, -) and (n9, {n0, n5, n6, n7}, 5, A, -) arrive to the network and at that time, it is assumed that the residual bandwidth on all the links to be 10 units before routing the two multicast requests are arrived.

For scenario 3, the first Class B request arrives to the network and the corresponding shortest path tree is constructed for the source n3 and its set of receivers {n0, n5, n11, n12, n13}. In the multicast tree a bandwidth of 6 units is allocated along the path. When a new Class A request arrives to the network, sufficient bandwidth is not available on the link n1-n0 which falls in the shortest path tree. As the request is Class A, a search is made to find Class B or Class C applications on the link. In this example, it happens to be the Class B request just
allocated. Now an alternate path has to be searched by the receiver of the Class B request which uses this link i.e., node n0.

![Figure 4.2 (a) Shortest path tree for the multicast requests](image)

The multi path mode search at node n0 floods an REQ message in the neighbourhood. On receiving, this message node n1 and node n11 respond back with their RES message. Node n0 then selects the best route from among the responses based on the QoS metrics. In this case, it rejects the message from node n1 because the path from node n1 traverses the congested link n0-n1, instead it sends JOIN message to node n2 (n0-n2, n2-n5). Then, it sends a REMOVE message to node n1 to which it was previously connected. Node n1 propagates this message until the
source node is reached. Once the Class B request is rerouted along this new alternate path, the freed up bandwidth is used to allocate the Class A request.

Figure 4.3 (a) shows the shortest path trees when the requests would have taken, if there were enough bandwidth on the links. Figure 4.3 (b) shows the rerouted path for the Class B request that arrives first and the shortest path tree allocated for the Class A request, which arrives later.

![Figure 4.3 (a) Shortest path tree for the multicast requests](image)

![Figure 4.3 (b) Rerouted path for the first Class B request due to lack of bandwidth on link n1-n0](image)
The proposed QOMRP has been simulated to test the effectiveness in terms of multicast session scalability, QoS and alternate path rerouting. The next section discusses the simulation environment and experimental results in detail.

4.4 Simulation

The simulation results for the proposed QOMRP using alternate path rerouting are analyzed in this section. The experimental results are compared with the min-hop shortest path multicast routing algorithm [27] which is common in use and the multicast network flow-based algorithm [27]. All the above protocols were simulated and an exhaustive simulation was carried out to analyze the performance in terms of multicast request rejection. The succeeding sections discuss the simulation environment and performance analysis.

4.4.1 Simulation Environment

The simulation environment models a standard NSF network size of 14 nodes and another network with 100 nodes. Using these networks, several simulations with a maximum of 2000 multicast requests have been experimented. In the simulation, networks were randomly generated. The average node degree in the network has been assigned between 3 and 6. The bandwidth on all links in the network was initially assumed to be uniform. The receivers and the source nodes that make multicast session request were selected randomly. The number of receivers for each of the multicast request was varied from 30% to 70% of the total nodes in the network. The reserved bandwidth on a link reflects the network load and hence higher value for reserved bandwidth indicate higher network load.

4.4.2 Performance Metrics

The main aim of the protocol is to accept more multicast sessions or reduce the number of multicast request rejection in order to increase the scalability. Hence, the multicast request rejection ratio is considered as one of the performance metrics.
to evaluate the scalability. As the proposed protocol is QoS driven, delay bound in terms of number of hops is also considered as another performance metric for comparison. In this protocol, service level differentiation is used for providing QoS. Therefore, priority-wise multicast rejection ratio is also analyzed to ensure high priority multicast sessions are always accepted more than any other class. A multicast request is said to be *accepted* if a tree connecting the source and the receivers is constructed and the bandwidth requirement is satisfied.

### 4.4.3 Simulation Results and Discussion

The results obtained from the simulation were compared with the *Minimum Hop Tree (MHT)* and *Multicast Network Flow (MNF)* algorithms. Several experiments were conducted with varying network size, node degree and bandwidth on all the links. The average number of multicast request rejected in these varying conditions was taken for comparison. The effect of network size, multicast size and bandwidth on the number of requests rejected has not been dealt individually. The network size and bandwidth do not have an effective influence in terms of multicast request rejections because the behaviour of the protocol remains the same as the number of multicast request also grows along with the network size and increase of bandwidth. Therefore, the results were obtained by varying the network node degree.

The main objective is to accommodate more multicast requests or to decrease the multicast rejections. Figure 4.4 compare the number of requests rejected by the three protocols. The multicast requests rejection is decreased on an average of 51.96% in QOMRP than MHT algorithm and 48.66% than the MNF algorithm. Between MHT and MNF, MNF performs better by about 6.00%, as it preserves critical links for future requests to come. In terms of multicast request rejection, QOMRP outperforms the MHT and MNF protocols as QOMRP uses the alternate path rerouting.
Initially all the requests will be accepted in the network, as there are more resources available. When the network load increases, it starts rejecting the multicast requests. This is the case with all the three protocols. However, the difference lies in
how many requests are rejected at peak load. The MHT and MNF algorithms allocate the bandwidth along the shortest path tree and when there is no bandwidth on any one of the links then the request is rejected. In QOMRP, using more requests is accepted using alternate path rerouting.

QOMRP provides the QoS through service level differentiation. The multicast requests are classified into Class A, Class B and Class C with high, medium and normal QoS requirements. The proposed protocol aims at satisfying more high priority requests in the order of preferences. For the experimental results shown in Figure 4.5, 4.6 and 4.7, the Class A, B and C request are uniformly distributed among 200 requests. Figure 4.5 shows the number of Class A multicast request rejections for the three protocols. QOMRP rejects only 10 to 13 Class A requests out of 200 requests made. The rejection is around 87.06% and 89.67% less than MNF and MHT protocols. It is obvious that the high priority multicast sessions are almost accommodated in the network.

![Figure 4.6 Request rejection graph for Class B multicast requests](image-url)

Figure 4.6 shows the number of Class B multicast request rejections for the three protocols. In QOMRP, the rejection is around 52.38% and 59.82% less than
that of MNF and MHT protocols respectively for the of Class B requests made. It is obvious that QOMRP provides better performance even for medium level multicast requests.

Figure 4.7 shows the number of Class C multicast request rejections for the three protocols. QOMRP rejects higher Class C requests out of 200 requests made. However, the rejection is around 20.93% and 29.08% less than MNF and MHT protocols respectively. Therefore, in any case the multicast rejection is certainly less in the QOMRP irrespective of the QoS classes.

![Figure 4.7 Request rejection for Class C multicast requests](chart)

The QoS parameters considered in QOMRP are bandwidth and end-to-end delay. The bandwidth is ensured while allocating the link for a multicast request. To study the performance delay of the proposed protocol, it is compared with MNF protocol. The MHT protocol is not considered for this analysis because MHT always generates shortest path tree and does not provide QoS. Figure 4.8 shows the delay characteristics of Class A multicast request. The delay characteristic is measured using the number of hops used for the multicast request over a period of time. From the experimental results, it is measured that the average delay in terms of number
hops for QOMRP is 14.4, whereas for MNF, the average delay is 15.7. The other service classes also provide similar performance of MNF.

Figure 4.8 Delay characteristics of the trees generated for Class A requests

Figure 4.9 Percentage of class-wise multicast request rejection

Figure 4.9 shows the percentage of class-wise multicast request rejection. The data have been taken over several experiments of QOMRP. In each execution approximately 1000 requests were generated by uniformly distributing three
different classes of service. The data for each experiment has been averaged over 20 to 30 execution of same number of requests. From the figure 4.9, it is obvious that the rejection of Class A request is relatively less small. The Class B and Class C request rejections are in the subsequent levels respectively.

4.5 Summary

In this chapter, a QoS aware online multicast routing protocol with bandwidth and delay guarantees using alternate path rerouting is presented. The protocol provides service level differentiation with three service classes based on bandwidth and end-to-end delay requirements. The protocol is aimed at accommodating many multicast requests without priori knowledge of future requests. The primary design objectives of the protocol are scalability and QoS through service level differentiation. The experimental results obtained show significant improvement in resource utilization. The number of multicast requests accommodated was significantly larger than the MHT and MNF protocols. Moreover, high priority multicast sessions were given importance and their rejection ratio is less in the new protocol.