QMBF: A QoS-Aware Multicast Routing Protocol[*]

Zhi Li and Prasant Mohapatra[†]
Department of Computer Science
University of California, Davis
One Shields Avenue
Davis, CA, 95616
Email: {lizhi, prasant}@cs.ucdavis.edu

February 26, 2002

Abstract

Many multicast applications, such as video-on-demand and tele-education, desire quality of service(QoS) support from the underlying network. Recently, many QoS-based multicast protocols have been proposed to meet these requirements. However, few of them can achieve high success ratios while keeping good scalability. In this paper, we propose a new QoS-aware Multicast Protocol using Bounded Flooding (QMBF) technique. In this protocol, every network node has the knowledge of local network cell topology as well as QoS state information(collected from bounded flooding messages). QMBF utilizes this knowledge to increase the probability of finding a feasible branch that connects a new member to the multicast tree. It bases on two methods to find such feasible branch: computing partial feasible branches using local network cell information and multiple path searching. The design of QMBF allows it to operate on top of any unicast routing protocol or cooperate with a QoS-based unicast routing protocol.

Key Words: Bounded Flooding Technique, IP Multicasting, Local Network Cell, QoS-aware Multicasting, QMBF.

---

[*] This research was supported in part by the National Science Foundation through the grants CCR-0296070 and ANI-0296034.

[†] Corresponding Author.

1 Introduction

Existing applications such as audio and video on-demand services, distance-learning, tele-conferencing need the support of efficient multicasting techniques. Several multicast routing protocols have been proposed in the literature with varying performance, cost, and implementation [6][12]. However, most of the proposed multicast protocols are QoS-oblivious as they only use the available best-effort unicast routing protocols to find paths from the sender to the receivers without considering the members' QoS requirements. On the other hand, it is known that most of the multicast applications are inherently QoS-sensitive. They usually have pre-defined requirements for QoS measures, such as delay, bandwidth, loss rates, etc. In order to provide satisfactory QoS to the applications, different mechanisms have been proposed, such as RSVP (resource reservation protocol), differentiated services, MPLS, etc [14]. The goal of QoS-based multicast routing schemes is to search and construct multicast trees that not only cover all the group members but also meet their QoS requirements.

The basic requirements of QoS-based multicast routing schemes are: (a) Scalability to large groups; (b) Support for dynamic changes in membership; (c) Support for receiver-initiated, heterogeneous resource reservations [7]. It is known that the design of QoS-based multicast routing schemes that satisfy heterogeneous QoS requirements is a NP-hard problem [11]. Therefore, most QoS-based multicast routing protocols utilize heuristic methods for the *feasible branch* (QoS-satisfying branch) searching process.

In source-based multicast routing, every node has the knowledge of the whole network's topology as well as its QoS properties. If a feasible path exists, it is easy for every router to find a QoS-satisfied path toward the target router. However, the scalability problem and obsolescence of the QoS state information prevent it from wide adoption.

QoS-aware Multicasting using Bounded Flooding (QMBF) technique combines the merits of source-based routing and the QoS-aware routing [2] in the feasible branch searching process. Every node broadcasts its local QoS state and least-cost unicast path information (derived from unicast routing information) when either of them changes over some threshold. This broadcast information is valid for a bounded number of hops. Every node can gather its neighbors' broadcast message to have the knowledge of the local network cell's topology, routing information, and QoS states. When a multicast join request message arrives at a node (joining node), it can take two steps to finish the local partial branch search. First, if some edge nodes of the local network cell have least-cost paths toward the target router (not via the nodes in this local cell), the joining node computes feasible paths toward these edge nodes and forwards the request to them along the feasible paths. If this step fails, the node will compute feasible paths toward all the edge nodes and forward the request to all the edge routers along the feasible paths. Using this mechanism, the routing protocol can greatly increase the robustness and accuracy of finding QoS-satisfied (feasible) paths that can connect the new members to the available multicast tree. Through simulation experiments, we have demonstrated the feasibility and performance benefits of the QMBF technique.

This paper is organized as follows. The definition of the QoS-based multicast routing and the formalization of the problem statement is outlined in Section II. Section III describes the related work. Section IV introduces the basic idea of QMBF. Section V describes the details of the QoS-aware multicast protocol. The protocol analysis and simulation are presented in Section VI. Finally, we draw the conclusions in Section VII.

2 Problem Statement

Internet can be modeled as a connected directed graph $G = (V, E)$, where V is the set of nodes and E is the set of communication links. We use $C(i, j)$ to refer the cost to transfer one packet on the link from the node i to j while $Q(i, j)$ denotes the QoS status of the link from i to j. One multicast group can be seen a set of nodes M $\subset$ V that are connected with each other to form a directed multicast tree T. In multicast communications, the mulitcast source v_s needs to send the same copy of information along the multicast tree to all the other nodes belonging to M ($\{M - v_s\}$). Multicast routing is the process of constructing such a minimum cost directed acyclic sub-graph T (multicast tree) that connects all the nodes belong to M.

Finding a minimum-cost spanning tree, called Steiner Tree problem, has been proved to be an NP-complete problem [9][12]. Many heuristic algorithms have been proposed to address this problem. Many of the current multicast routing protocols are based on the TM algorithm [10]. It begins with the multicast tree T with only one group member (usually the multicast source). Then, step by step, it adds new members in M to the tree T that has the shortest path to any of the nodes in T.

When considering the receivers' QoS requirement, the routing problem becomes more complicated because of the resource requirements and management issues. QoS-based multicast routing can be formalized as follows:

Suppose the multicast group member set M = $\{v_s, v_1, v_2, ..., v_m\}$, while $v_1, v_2, ..., v_m$ have the QoS requirements of $q_1, q_2, ..., q_m$, respectively. Our goal is to find a tree T_Q that meets the following requirements:

1) $\forall$ i $\in$ m, $\forall$ e(j, k) on the path from v_s to v_i in the tree T_Q,

Q(j,k) $\geq q_i$ ($\geq$ means the QoS in the link meets the QoS requirement q_i).

2) At the same time, $\forall$ e(j, k) $\in T_Q$, minimize $\sum$ e(j,k).

Whenever a new member v_i wants to join a mulitcast group, we search a new branch that connects the new member to the available tree. We begin the search process from v_i along the least cost path toward the sender v_s until some link $e(j,k)$ does not meet i's QoS requirement q_i. We then can sequentially try all the other j's links and continue the search process. If we fail after trying all the links of j, it can backtrack to j's parent node (the node next to j on the partially successful path) and continue the above process. Using this method, if there is at least one feasible path that can connect the new node to the multicast tree meeting its QoS requirement, we should be able to discover it.

3 Related Work

A few QoS-aware mutlicast routing protocols have been proposed in the literature [2]-[4]. Some of these protocols use multiple branch searching method to increase the probability of a successful search [1, 3, 5]. Their branch searching processes are based on least-cost, which may not always satisfy the QoS requirements. Chen et al have integrated QoS-awareness idea into the branch searching process [2], where the feasible branch searching is based on unidirectional broadcasting if the least-cost path can not satisfy the QoS requirements. In spanning-joins [1], a new member broadcasts join request in its neighborhoods to find on-tree routers. The reply message will collect the QoS properties along its traveling path, which is one of the candidate paths. When the new member receives multiple reply messages, it selects the best candidate path as a multicast tree branch to join the group.

The QoS sensitive multicast Internet protocol (QoSMIC) uses a “manager router” to construct shared multicast trees [3]. The new router has two ways to find an on-tree router to join a multicast group: local search and multicast-tree search. The local search period is the same as the spanning-joins. In multicast-tree search period, the host router sends a join request to the manager. The manager selects some on-tree routers as candidate routers, which will unicast bid messages towards the host router. After receiving the bid messages, the host router selects the best path to connect to the multicast tree.

The QoS-aware multicast routing protocol (QMRP) [2] constructs a shared tree by unicasting a request message from the host router toward the core router (or source router). If a router in the unicast path does not satisfy the QoS requirements, the request message is replicated and sent out to all other neighbors of the router. It introduces the idea of QoS-awareness into the path selection period, which increases the ability of finding a feasible branch. However, it requires temporal state in the network routers for each join request. It is only applicable for applications with non-additive QoS requirements such as bandwidth and buffer space, and cannot be used for additive requirements such as delay or packet loss.

Our protocol shares the same merit as QMRP, using distributed QoS-aware mechanism to search for a feasible branch. Deviating from QMRP, we use M-hops bounded broadcasting mechanism to make the feasible branch searching process more focused. Using this mechanism, every router has the knowledge of local network cell information, which can increase the chance of finding feasible branches. Another advantage of the proposed technique is that the protocol also supports additive QoS properties without involving the backtracking process and maintaining temporary states.

4 QMBF:Basics

In this section, we discuss the basic concept of bounded flooding and the search process of the feasible path. An overview of the QMBF technique using the bounded flooding and the feasible path search process is also presented.

4.1 Bounded Flooding and Partial Feasible Branch

When a new member wants to join a multicast group, QoS-based multicast routing protocols should have the ability to find a feasible (QoS-satisfied) branch that connect the new member to the available multicast tree if it exists. Source-based QoS routing can find the possible feasible branch if it exist because every node has the knowledge of the global network topology and QoS state information. However, the scalability problem and obsolescence of the QoS information prevent it from wide implementation.

QMBF is based on the TM algorithm [10] and depth first-search algorithm. It begins with a multicast group with one group member. Whenever a new node joins a multicast group, QMBF finds a branch that can connect the new member to one of the *on-tree nodes* (the nodes that are already in this group's multicast tree). The path from the multicast source to the new member should meet the new member's QoS requirement.

To achieve good scalability while maintaining high sucess rate (Number of successes in finding a feasible branch/Total number of search attempts) in QMBF, we use *M-hop bounded flooding* to increase the chance of finding feasible paths. We assume that every node has the QoS state information of itself and its outgoing links (*local QoS state information*), which includes the available bandwidth, average delay, etc. *M-hop bounded flooding* requires every node to periodically broadcast its local *QoS state information* and *unicast reachable information* (which nodes it can reach through each of its neighboring nodes, computed from the unicast routing information) for at most M hops (we can choose different value of M based on network size and status as discussed later). From these messages, every node can have a view of the partial local network topology, QoS state information as well as the route reachablity status. We call this small domain of the network as *local network cell (LNC)*. The node itself is the center of its LNC whose radius is M hops. Those nodes that are at the edge of the current node's LNC are called *edge nodes*, which are the last hop valid nodes of A's broadcast messages. Using this information, every node can accurately and quickly direct the QoS-based multicast routing request.

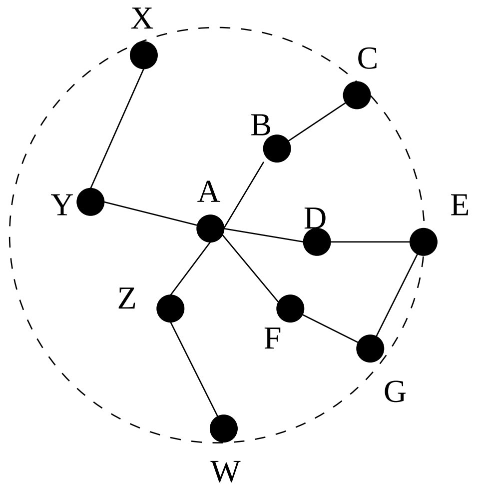

Figure 1: Illustration of bounded flooding technique.

We assume that every on-tree node knows the QoS property from the multicast message sender to itself. So, our goal is to find an on-tree node by which the new member can connect to the multicast tree and the QoS property from the sender to the new member meets its QoS requirement. During the new branch finding process, the join request is forwarded toward target router (multicast source for source-based multicast tree, core router for shared-multicast tree). In the process of forwarding the join request toward the target router, whenever the join message meets an on-tree node which fulfill the new router's QoS requirement, the search process terminates.

Let us consider the example shown in Figure 1 which shows the LNC of node A after it gets neighbors' broadcast messages (M=2 in this example). In A's LNC, X, C, E, G, and W are edge nodes which are the last valid hop nodes of A's broadcast messages. All the edge nodes' reachable information constitute the LNC's least-cost routing information. When a join request message arrives at A, A can check which of the edge nodes have the least-cost path that leads A to the target router (multicast source or core router of this mutlicast group). Suppose E can connect A to the target router. Then, A can locate a feasible path from itself to E based on its LNC knowledge. This kind of feasible path from a node to one of its LNC edge nodes is called a *partial feasible branch (PFB)* during the feasible branch searching process.

QMBF uses the knowledge of LNC to search for feasible branches. This mechanism makes the request message always travel one step of M-hop along a M-hop wide path toward the target router, which ensures that it can quickly locate one feasible branch and greatly increase the success ratio. Because QMBF utilizes edge nodes' least-cost reachability and LNC information, most part of the successful path is closer to the least-cost path.

4.2 Overview of QMBF

When a new member wishes to join a multicast group, it sends a JOIN message toward the target router (source router or core router). This JOIN message carries the user's QoS requirement, target address, group address and the accumulated QoS information of the path it has traversed. When one node receives the JOIN message, it first checks whether there are edge nodes with least-cost paths toward the target router. Next, it will use the LNC information to find feasible paths (or the PFBs) from itself to the edge nodes. Then, the node duplicates and forwards the JOIN message toward these edge nodes and reserves QoS resource along the PFBs. If there is no such feasible path or no such edge node, the current node will compute feasible paths from itself to all the edge routers. Then, it makes multiple copies of the JOIN message and sends them to all these edge nodes. The process is repeated until a JOIN message arrives at one on-tree router satisfying the QoS requirement.

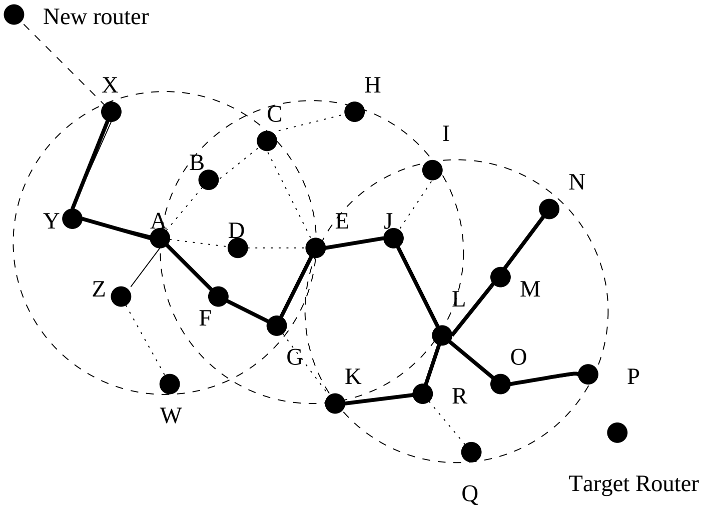

Figure 2: Example of QoS-satisfied branch searching process.

Figure 2 depicts an example of feasible branch searching process (the thick lines are the tracks that the of JOIN messages have traversed, M=2). Suppose the join request message has sucessfully discovered a branch from the new router (new group member) to X, and X sucessfully gets the PFB (successful in searching within its LNC). When the join request message arrives at A along the path X-Y-A, A first begins the first step of PFB searching. It computes the PFB from itself to edge nodes which have the least-cost toward the target router. Suppose it finds E with least-cost path toward the target router. It then computes a feasible path from A to E using LNC information, and sends the JOIN message and reserves resources along the path:A-F-G-E. When the request message arrives at E, it will repeat the same process and find another PFB: E-J-L. Suppose L receives the JOIN message, and it fails in the first step of PFB search; no edge router has least-cost path toward the target router or no feasible path from the current node to these edge nodes (PFBs). At this time L will begin the second step of PFB search; it computes feasible paths to the edge nodes (except the incoming edge node E). Then, it duplicates and forwards the JOIN messages along the PFBs.

From the above example, we can observe that QMBF utilizes the edge nodes' least-cost information and the M-hop bounded flooding method. Therefore most part of the branch is in proximity of the least-cost path, which ensures high success ratio of branch searching.

5 Detail Description

In this section, we describe the details of the tree formation and maintenance of the tree using the QMBF approach.

5.1 Network Functions

QMBF is based on the current network infrastructure where the network nodes are connected by duplex, asymmetric links. We assume the network could provide the following functions: (a) each nodes can realize the available local QoS state (such as the available bandwidth, delay or other QoS measures); (b) The least-cost routing protocol runs at every nodes; (c) Every on-tree node knows the QoS property from the multicast source to itself (mQoS); (d) The bounded flooding functions run at every network nodes; (e) Every node can use some mechanism to get the target address (multicast source or core router's address) of the multicast group (using technique such as SDP (Session Directory Protocol)[13]).

There are four types of messages involved in QMBF:

1) JOIN (taddr, gaddr, rQoS, aQoS, FPB): The join request message is initiated by a member and sent toward the target router requesting to find a feasible branch which can link the new member to the multicast group tree satisfying the QoS requirements. It includes the multicast target address (tadrr), group address (gaddr), required QoS (rQoS), accumulated QoS property (aQoS, the QoS property from the current node to the new member), and available PFB (which is computed based on the bounded flooding messages).

2) CONFIRM (gaddr, rQoS, mQoS): It is initiated by an on-tree router. It traverses along the newly-found feasible branch toward the new member. It indicates that the feasible branch searching process has finished, confirming the branch and the reserved resources. The message includes the group address (gaddr), the accumulated QoS from the multicast source node to the sender of the message (mQoS), etc.

3) UNACK (gaddr, rQoS): It is sent by one router along the reverse direction of the JOIN message. It indicates that the branch searching process has failed and it removes the multicast

routing information while releasing the reserved resources. It includes group address, requested QoS (rQoS).

4) PRUNE (gaddr): It is sent by one end user or a router, removing unnecessary branches.

The multicast tree is recorded by the multicast entries residing at the nodes of the multicast tree. Each of the entries is denoted as M{G, in, out, mQoS, fix, rQoS, num}, where M.G is the address of the multicast group, M.in is the parent of the multicast tree, M.out is the set of child nodes, M.mQoS is the QoS property for this entry from the source (when this entry is confirmed) or the required QoS from the source to the current node to meet the new member's QoS requirement (when the entry is unconfirmed), M.fix denotes whether the entry has been confirmed or not. M.num is the number of JOIN messages that have been sent out from the node for the rQoS.

5.2 Join-Request Process

When a JOIN message arrives at a node, the node first checks whether it already has multicast routing information for this group. If it has the information and the mQoS property of this group plus the aQoS meets the new member's QoS requirement (rQoS) (it is represented as "mQoS $+$ aQoS $\geq$ rQoS"), it will add the incoming interface to this group's receivers' list. Then, it sends a CONFIRM message toward the new member. Otherwise, it will check whether there are some available PFP information within the JOIN message. If available, it will reserve the QoS resource and add an entry to the multicast routing table: gaddr, JOIN message sender id, unfixed, mQoS (the required QoS property from the source the current node meeting the new member's requirement, which is represented as "rQoS $-$ aQoS"), num=1. It then updates the aQoS of the message and forwards it along the PFB to the next node. If no such path information is found within the message, the node will begin the "PFB computing" process using the LNC information.

In the first step, it finds the edge nodes which has the least-cost path (not via the nodes in this LNC) information toward the target router. Then, it computes the PFBs from itself to these edge nodes. If the next hop node of the feasible paths is not the incoming node (the node which had sent JOIN message to the current node), the current node will duplicate the JOIN message, update the PFB information and aQoS and send the JOIN message along these paths. At the same time, it will also reserve the QoS resource and update multicast routing table: gaddr, message sender list, unpinned mark, QoS, mQoS (the required QoS property from the source the current node meeting the new member's requirement), num.

If the current node can not find a node satisfying the above requirement during the first step of "PFB computing process," it will begin the second step, where it computes the feasible paths from itself to all the edge nodes. If the next hop of the feasible paths is not the same as the incoming node, it will then duplicate the JOIN message, update the PFB information and aQoS and send the JOIN messages along these paths. At the same time, it also will reserve the QoS resource and add an entry to the multicast routing table: gaddr, JOIN message sender, unfixed, mQoS (the required QoS property from the source the current node meeting the new member's requirement), the number of JOIN messages that have been sent out.

If the router cannot find a PFB from its LNC information after the above two steps, it means that the feasible branch searching process has failed. It then sends an UNACK message back to the sender of the message.

Suppose node i receives a JOIN (taddr, gaddr, rQoS, aQoS, FPB) message from j, its functionalities are formalized in Algorithm 1.

Algorithm 1 Join-Request Process

```
if i is an on-tree node of the group G
  if (i is the source of the group or core router)
    send CONFIRM(G,rQoS) to j
    exit
  retrieve the entry M of the group G
    if ($M.mQoS + aQoS \geq rQoS$)
      add j into the M.out
      if (M.fix==TRUE)
        send CONFIRM(G,rQoS,mQoS) message to j
    exit
create an entry M
M.out = j
M.mQoS = rQoS - aQoS
M.fix = FALSE
if FPB includes the next hop information
  M.in = next hop
  update the aQOS field of the JOIN message
  send out the JOIN message to the next hop
else
   retrieve the edge nodes whose next hops toward taddr is not within the LNC.
  for each of these edge nodes:
    compute a feasible path from i to the edge node according to rQoS using LNC.
    if (There is such a path and the next hop of the feasible path != j)
      update aQoS, sent the JOIN message along the path.
      increase the M.num
  if (M.num == 0)
    for all edge nodes of the LNC:
      compute out a feasible path from i to the edge node which meet rQoS using LNC.
      if (There is such a path and the next hop of the path != j)
        update the aQoS field of the message and sent out the JOIN along the path.
        increase the M.num
if (M.num == 0)
  send UNACK(G,rQoS) to j
```

5.3 Join-Response Process

When a node receives an UNACK message, it first checks how many JOIN messages of this group with the rQoS it has been sent out. If there are some JOIN messages that have not been confirmed, it decreases the number of unconfirmed JOIN message. If not, it removes the entry of the multicast group of this rQoS. Then, it duplicates the UNACK messages and sends to the children nodes of the multicast group (listed in the receivers' list).

When a node receives a CONFIRM message, it first checks whether any entry of the same group has been confirmed with better QoS property. It sends back a PRUNE message to the incoming node. Otherwise, it will set the "fix" field of the routing entry as "TRUE", update the parent node and mQoS of this entry. Then, it will duplicate the CONFIRM message, update the mQoS field and send a CONFIRM message to the children nodes of the multicast group. It will also check whether there exists the same group's routing information with lower QoS support than the confirmed one. If so, it adds the children nodes of the routing entry to this entry. For the unfixed entries that are not "fixed," it also duplicates the CONFIRM message and sends them to the children nodes. For the confirmed items, it will sent PRUNE messages to their parent nodes.

Suppose node *i* receives a message from *j*, the above Join-Response process can be formalized as shown in Algorithm 2.

Algorithm 2 Join-Response process

```
If the message is CONFIRM(gaddr, rQoS, aQoS):
    retrieve the multicast entry M1,M2,M3...Mn of gaddr (assuming that M1.rQoS = rQoS)
    if (exits Mk where Mk.fix = TRUE and Mk.rQoS > rQoS)
      send PRUNE(gaddr) to j
      add M1.out to Mk.out
      delete M1
      exit
    set M1.fix = TRUE
    for those Mk(k=2..n)
      if (Mk.rQoS < M1.rQoS)
        if (Mk.fix != TRUE)
          sent CONFIRM(gaddr, Mk.rQoS, aQoS) to Mk.out
          sent PRUNE(gaddr) to M.in
        add Mk.out to M1.out
        delete Mk
If the message is UNACK(gaddr,rQoS):
    retrieve the multicast entry M with gaddr and rQoS
    update M.num
    if (M.num=0)
      sent UNACK(gaddr,rQoS) to M.out
      delete M
```

5.4 Pruning Process

Whenever an end router becomes the leaf node of a multicast tree, it will send a PRUNE message up the multicast tree and remove itself from the multicast tree. When a node receives a PRUNE message, it will first update the multicast routing information. Then, it will check whether there exists other children nodes of the multicast group. If not, then it will also send a PRUNE message to its parent node.

5.5 Optimization of QMBF

To minimize the traffic overhead of the above QMBF algorithm, we proposed the following methods for optimization.

1) In the first step of the “PFB computing process,” when one JOIN message arrives at one node, the node perhaps can locate multiple edge nodes which has least-cost path toward the target router or multiple PFBs to these edge nodes. If this node sent out multiple copies of JOIN messages, there will be more and more branches along the JOIN paths. To control the number of branches, every time we only select the edge node that is the nearest one to the least-cost path leading the current node to target node.

2) In the second step of “PFB computing process,” QMBF finds PFBs for every edge nodes, duplicates the JOIN message and send out along these partial branches. If this situation happens too frequently, there will be more and more branch nodes causing excessive overhead. We use the concept “Maximum Branch Degree” from [2] to control the traffic, which decides the number of nodes that will send multiple copies of join request messages during the feasible branch searching process.

3) Because QMBF is based on bounded flooding technique, if the flooding messages travel too many hops, it also burdens the network. So, we use “bounded hops” to control the flooding traffic, which decides for how many hops the flooding message is valid.

Based on above methods, we optimize QMBF into QMBF-mn (m means the flooding hops, n means maximum branch degree). We can use different values of m and n to meet our requirements under various network load and configuration.

6 Simulation Results & Analysis

In this section, we study and compare the performance of QMBF-mn (m is the flooding hops while n is maximum branch degree used during the second step of PFB search) with other QoS-based multicast protocols. Four other algorithms were simulated: single-path joining protocol, directed spanning-joins protocol, QMRP2, and QMRP3. For QMBF-mn, we simulated six schemes: QMBF-12, QMBF-13, QMBF-22, QMBF-23, QMBF-32, and QMBF-33.

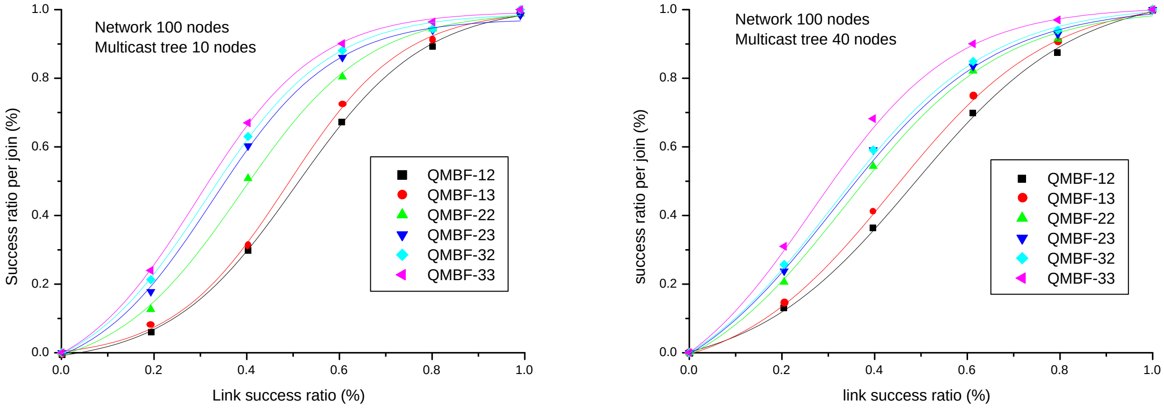

Figure 3: Success ratio of QMBF-mn with two multicast group size.

6.1 Simulator Environment

The simulations were conducted on the Waxman network topology[8]. We use the following approach to randomly generate network topology: network nodes are randomly chosen in a square ($\alpha*\alpha$) grid. A link exists between the nodes u and v with the probability p(u,v) =a*exp(-d(u,v))/b*$\sqrt{2}$*α), where d(u,v) is geometric distance between u and v, a and b are constants that are less than 1. In the simulation, a and b are randomly chosen so that the average degree of nodes is between 4 and 5. The networks have a fixed size of 100 nodes over a 100*100 grid.

Based on above topology generation method, for each simulation, a network topology, a multicast tree, and a new member of the tree are randomly generated. We use two types of tree size (including the internal nodes): 10 and 40 nodes. A new member has a randomly generated QoS requirement. The QoS property of each link is also randomly generated. For each situation (different protocols, multicast group sizes, link success ratios (what percent of the links meet the new user's QoS requirement.)), we run the simulation 200 times. We have mainly focused on the success ratio as the measure of the performance, which is defined as the ratio of the number of successful search attempts to the total number of searches.

6.2 Results and Analysis

Figure 3depicts the success ratio of different modes of QMBF with the two different size of multicast groups. We can see that QMBF-mn's success ratio increases with the increase in m and n. The reason is that m decides the the scope of every node's LNC. As m increases, the nodes can have a higher probability of finding PFBs when some part of the least-cost paths do not meet the new member's requirement. The parameter n decides on the instances of the multiple path searching process, which also increases the probability of finding a feasible branch.

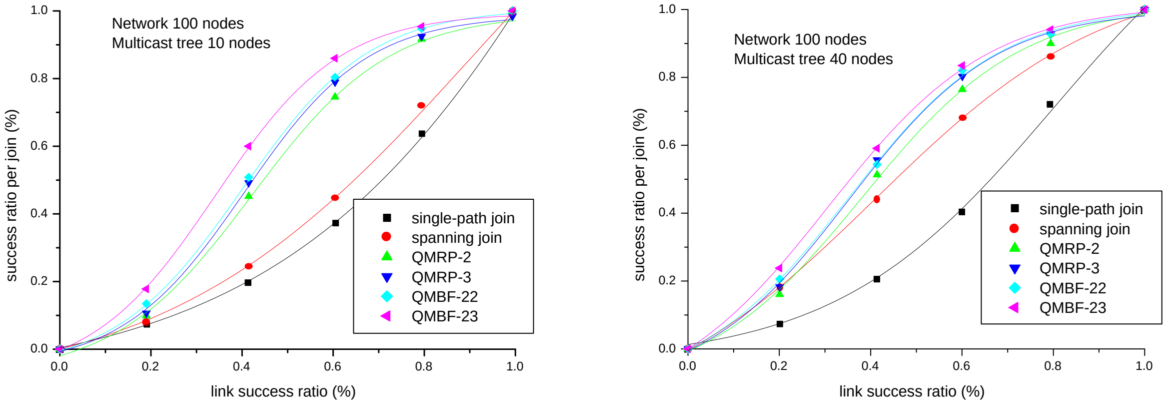

Figure 4: Success ratio of different multicast protocols with two group size.

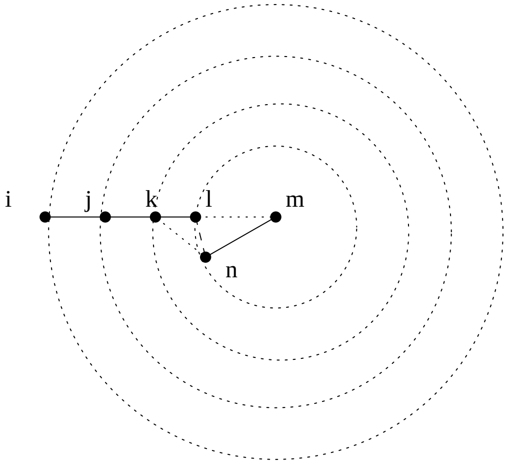

Figure 5: Explanation the advantage of QMBF.

QMRP2 and QMRP3 are two kinds of QoS-aware multicast routing protocols defined in [2], which can achieve higher success ratio. Figure 4 compares the success ratios of all the simulated protocols with the two different sizes of the multicast group. The figure shows that the success ratios of QMBF-22 and QMBF-23 are better than QMRP2, QMRP3, spanning-joins protocol, and single-path join protocol.

The reason behind the above simulation result can be explained using Figure 5. Assume a new member m wants to join an existing multicast group, the nearest on-tree node is i. Suppose the least-cost path from m to i is m-l-k-j-i, if any of the link on the path does not meet the new member's QoS requirement, both the spanning-joins protocol and single path routing protocol searching methods will fail. For QMRP, when m-l link doesn't meet the QoS requirement, it will enter the multiple path search period. For example, m will send the request to n to continue the search process. Suppose the least-cost path from n to i is n-l-k-j. If link n-l again can't satisfy the new member's QoS requirement, n will have to enter the multiple path searching process. If this situation happens too frequently, QMRP will have to enter multiple-path searching process many times to find a feasible branch. However, for QMBF, m will compute a PFB m-n-k at once when it receives the user's JOIN message.

7 Summary

In this paper, we propose a new QoS-aware multicast routing protocol called QMBF. QMBF is based on bounded flooding technique, in which every network node is assumed to have the knowledge of the LNC topology as well as QoS state information. QMBF uses two methods to find a feasible branch: Computing PFB using LNC information (collected from bounded flooding) and using multiple path searching. QMBF can either operate on top of any unicast routing protocol or cooperate with QoS-based unicast routing protocols. The protocol requires no intermediate routers to keep the temporal searching states and does not flood the whole network to find the multicast path. The simulation results shows the feasibility and performance of QMBF. It is also shown that QMBF can achieve better success rate than other QoS-based multicast routing protocols.

References

[1] K. Carlberg and J. Crowcroft, “Building Shared Trees Using a One-to-Many Joining Mechanism,” Computer Communication Review, Jan. 1997.

[2] S. Chen, K. Nahrstedt. and Y. Shavitt, “A QoS-Aware Multicast Routing Protocol,” IEEE INFOCOM, May 2000.

[3] M. Faloutsos, A. Banerrjea, and R.Pankaj, “QoSMIC: Quality of Service Sensitive Multicast Internet Protocol,” SIGCOMM, Sep. 1998.

[4] A. Fei, M. Gerla, “Receiver-initiated Multicasting with Multiple QoS Constraints,” IEEE INFOCOM, May 2000.

[5] Y. Jia, I. Nikoladis, P. Gburzynski, “Multiple Path Routing,” Proceedings of ICC’01, June 2001.

[6] S. Deering, “Multicast Routing in Internetworks and Extended LANs,” SIGCOMM’88, Aug. 1988.

[7] S. Chen and K. Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions,” IEEE Network, Special Issue on Transmission and Distribution of Digital Video, Nov./Dec. 1998.

[8] B. M. Waxman, “Routing of Multipoint Connections,” IEEE Jornal on Selected Areas in Communications, Vol. 6, No. 9, Dec. 1988.

[9] P. Winter, ”Steiner Problem in networks: a survey”, IEEE Networks, volume 17, pages 129-167, 1987.

[10] H. Takahashi and A. Matsuyama, ”An approximate solution for the Steiner Tree problem in graphs”, Math. Jap., pages 573-577, 1980.

[11] C. Diot, W. Dabbous, and J. CrowCroft, “Multipoint Communication: a survey of protocols, functions, and mechanisms,” IEEE Journal on Selected Areas in Communications, Vol.15(3), April 1997.

[12] B. Wang, C. Hou, “A survey on mulitcast routing and its QoS extension: problems, algorithms, and protocols,” IEEE Network Magazine, Vol.14, No.1, Jan./Feb. 2000.

[13] M. Handley and V. Jacobson, “SDP:Session Directory Protocol (draft2.1),” Internet Draft, Feb. 1996.

[14] X. Xiao and L. M. Ni, “Internet QoS: the Big Picture,” IEEE Network, Mar. 1999.