QMBF: a QoS-aware multicast routing protocol

Zhi Li, Prasant Mohapatra*

Department of Computer Science, University of California at Davis, Davis, CA 95616, USA

Received 26 February 2002; revised 22 May 2002; accepted 25 June 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 this requirement. However, few of them can achieve high success ratios while maintaining good scalability. In this paper, we propose a new QoS-aware multicast protocol based on a bounded flooding technique (QMBF). In this protocol, every network node maintains the knowledge of a local network cell (LNC) topology as well as QoS state information (collected from the bounded flooding messages). QMBF utilizes this knowledge to increase the probability and accuracy of finding a feasible branch which connects a new member to an existing multicast tree. The protocol exploits two approaches to find feasible branches: repeatedly computing partial feasible branches using the LNC information and searching multiple paths. The design of QMBF allows it to operate on top of any unicast routing protocol or cooperate with a QoS-based unicast routing protocol. Results obtained using NS-2 simulator demonstrate high success ratio for locating feasible branches for multicast paths.

Keywords: Bounded flooding technique; IP multicasting; Local network cell; QoS-aware multicasting; QMBF

1. Introduction

Existing applications, such as 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 routing protocols are QoS-oblivious. They are based on the available best-effort unicasting routing protocols to find paths connecting the members without consideration of their quality of service (QoS) requirements. On the other hand, it is known that most multicast applications are inherently QoS-sensitive. They usually have pre-defined requirements for QoS measures, such as delay, bandwidth, packet 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 a multicast tree that not only covers all the group members but also meets 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 finding an optimal multicast routing tree that satisfies heterogenous QoS requirements is an NP-hard problem [11], because of which, most QoS-based multicast routing protocols utilize heuristic methods while searching the feasible branch (QoS-satisfied branch). In source-based 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 node 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 multicast protocol 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 reachability information (derived from unicast routing information) when either of them changes exceeding some threshold. These broadcast messages are valid for a bounded number of ‘hops’. Using its neighbors’ broadcast messages, every node can have the knowledge of a local network cell’s (LNC) topology, routing information and QoS states. When a multicast join

* Corresponding author.

E-mail addresses: prasant@cs.ucdavis.edu (P. Mohapatra), lizhi@cs.ucdavis.edu (Z. Li).

0140-3664/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(02)00147-0
request arrives at a node (joining node), it can take at most two steps to finish the local partial branch search. First, if some edge nodes of the LNC 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 enhance the robustness and accuracy of finding feasible branches that can connect the new members to the available multicast tree. Through simulation experiments, we have demonstrated the feasibility and performance benefits of QMBF.

This paper is organized as follows. The definition of the QoS-based multicast routing and the formalization of problem statement is outlined in Section 2. Section 3 describes the related work. Section 4 introduces the basic idea of QMBF. Section 5 describes the details of this QoS-aware multicast protocol. The protocol analysis and simulation are presented in Section 6. Finally, we draw the conclusion and present the future work in Section 7.

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 $V_i$ to $V_j$ while $Q(i, j)$ is the QoS-metric of the link from $V_i$ to $V_j$. A multicast group can be defined as a set of nodes $M \subset V$ that are connected with each other to form a directed multicast tree $T$. In multicast communications, the multicast source $s$ needs to send the same copy of information along the multicast tree to all the other nodes belonging to $M \setminus \{s\}$. Multicast routing is the process of constructing such a minimum cost directed acyclic sub-graph $T$ (multicast tree) that connects all the nodes belonging 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 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 $M = \{v_s, v_1, v_2, \ldots, v_m\}$, while $v_1, v_2, \ldots, v_m$, respectively, have the QoS requirement of $q_1, q_2, \ldots, q_m$. Our goal is to find a tree $T_Q$ that meets the following requirements:

1. $\forall i \in m, \forall e(j, j + 1)(0 \leq j < k - 1)$ on the path from $v_s$ to $v_i$ in the tree $T_Q$.
2. If $q_i$ and $Q(j, j + 1)$ are additive QoS-metric,
   $\sum_{j=0}^{k-1} Q(j, j + 1) \geq q_i$ (\(\sum\) means the QoS in the link meets the QoS requirement $q_i$).
3. If $q_i$ and $Q(j, j + 1)$ are multiplicative QoS-metric,
   $\prod_{j=0}^{k-1} Q(j, j + 1) \geq q_i$.
4. If $q_i$ and $Q(j, j + 1)$ are concave QoS-metric, $\min(Q(j, j + 1)) \geq q_i$.

(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 multicast group, we can begin to search a new branch that connects the new member to the existing tree as follows. The search process begins from $v_i$ along the least-cost path toward the sender $v_s$ until some link $e(j, k)$ does not meet $v_i$’s requirement $q_i$. Then, we can sequentially try all the other $v_j$’s links and continue the search process. If we fail after trying all the links of $v_i$, it can backtrack to $v_i$’s parent node (the node next to $j$ on the partial successful path) and continue the above work. 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. However, because the number of nodes in the Internet is so huge, it will be impractical for us to traverse all the nodes before finding a branch for a multicast group.

3. Related work

A few multicast routing protocols have been proposed [2–4]. Some of these protocols use multiple branch searching method to increase the probability of a successful search [1,5,15]. Their branch searching processes are based on least-cost, which may not always satisfy the QoS requirements.

In spanning-joins [1], a new member broadcasts join request in its neighborhood to find on-tree routers. The reply message will collect the QoS properties along its traversing 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 [15]. 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 send unicast bid messages toward 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 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.

Our protocol shares the same merit as QMRP, using distributed QoS-aware mechanism to search for a feasible branch. Deviating from QMRP, we use the $M$-hops bounded flooding mechanism to make the feasible branch searching process more focused. Using this mechanism, every router has the knowledge of LNC information, which can increase the probability and the speed of finding feasible branches.

4. QMBF: basics

In this section, we discuss the basic concept of bounded flooding and the feasible branch searching process. An overview of the QMBF technique using the bounded flooding and the feasible branch searching 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 connects 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 QoS information prevent it from wide implementation.

QMBF is based on the TM algorithm [10] and the depth-first search process. 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 routing tree). The path from the multicast source to the new member should meet the new member’s QoS requirement.

To achieve scalability while maintaining high success rate (number of success in finding a feasible branch/total number of searching attempts), in QMBF, we use $M$-hop bounded flooding to increase the probability of finding feasible branches. 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 (the nodes it can reach through each of its neighboring nodes, computed from the unicast routing information). These messages are valid for at most $M$-hops (we can choose different value of $M$ based on the 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 reachability status. We call this small domain of the network as LNC. The node itself is the center of its LNC with a radius of $M$-hops. These nodes that are at the edge of the current node’s LNC are called edge nodes, which are the last hop valid nodes for the broadcast messages of the center node. Using this information, every node can accurately and quickly direct the QoS-based multicast routing request.

We assume that every on-tree node knows the QoS state from the multicast source 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 state of the path from the sender to the new member meets its QoS requirement. During the new branch searching 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 member’s QoS requirement, the searching process terminates.

Let us consider the example shown in Fig. 1. It shows the LNC of node A after it receives its neighbors’ broadcast message (in this example, the valid hops of flooding messages are 2.). In A’s LNC, X, C, E, G, and W are edge nodes which are the farthest nodes to which A’s broadcast messages can reach. For an LNC, all the edge nodes’ reachability 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 can lead A to the target router. Suppose E can connect A to the target router in this example. 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).

QMBF uses the knowledge of LNC to search for feasible branches. This mechanism makes the request message always travel one step of $M$-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 branch 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 state 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 (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. It then duplicates the JOIN message and sends them to all the edge nodes. The process is repeated until a JOIN message arrives at one of the on-tree routers satisfying the QoS requirements.

Fig. 2 depicts an example of a feasible branch searching process. The thick lines in this figure are the track of JOIN messages that have passed. The valid number of hops for flooding messages is 2. Suppose the join request message has successfully discovered a partial branch from the new router (new group member) to X, and X successfully gets PFB (successful in searching within its LNC). When the join request message arrives at A along the path X–Y–A, A begins the first step of PFB searching. It computes the feasible paths (PFBs) from itself to edge nodes which have the least-cost path 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 the 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 when L receives the JOIN message, 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 technique. Therefore most part of the branch is in the proximity of the least-cost path. This ensures high success ratio and quality of the branch searching process.

5. Detail description

In this section, we describe the network model, control messages and the details of the tree formation and state maintenance of QMBF.

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 measure); (b) The least-cost unicast routing protocol runs at every nodes; (c) Every on-tree node knows the QoS state along the path from the multicast source to itself (mQoS); (d) The bounded flooding function runs at every node. (e). Every node can have some mechanism to get the target address (multicast source or core router’s address) of the multicast group (using some 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 (taddr), group address (gaddr), required QoS (rQoS), accumulated QoS (aQoS, the QoS state along the path 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 been finished, confirming the branch and 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 which belong to the multicast tree. Each of the entries is denoted as M{G, in, out, mQoS, fix, rQoS, num, state}. M.G is the address of the multicast group. M.in is the list of the parents or grand-parents nodes of the multicast tree, which only used during the feasible branch searching process. M.out is the set of interfaces, through which the node can be connected to its children nodes. M.mQoS is the QoS property for this entry from the source to the current node meeting the new member’s QoS requirement (when the 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.num is the number of JOIN messages that have been sent out from the node which are used for searching feasible branch for M.G. M.state refers to the step the current node has done during feasible branch searching period for group M.G. If M.state equals to 1, it means the node has undergone the ﬁrst step of PFB. If M.state is 2, the node has ﬁnished the second step of PFB searching.

To handle the losses of multicast control messages, the incomplete multicast routing entries are maintained as soft-state based. It has an time-out value. If an entry has not been conﬁrmed and it is timed out, it will be removed from the multicast routing table and the reserved QoS resource are released.

5.2. Join request process

When the JOIN message arrives at a node, the node ﬁrst checks whether it already has multicast routing entry for this group. If it has and the mQoS of this group plus the aQoS meets the new member’s QoS requirement (rQoS) (it is represented as ‘mQoS + aQoS ≥ rQoS’), it will add the incoming interface to this group’s children interfaces’ 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, incoming interface, unfixed, mQoS (the required QoS from the source the current node meeting the new member’s requirement, which is represented as ‘rQoS - aQoS’), num = 1 and state = 0. It then updates the aQoS of the message and then 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 of PFB computing process, it ﬁnds the edge nodes which have the least-cost path (not via the nodes in its LNC) information toward the target router. Then, it computes the feasible paths (PFBs) from these edge nodes to itself. If the feasible paths do not pass 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 the reverse direction of these paths. At the same time, it will also reserve the QoS resource and update multicast routing table: gaddr, parent nodes list (those edge nodes which meet our requirement), incoming interfaces list, mQoS (the required QoS property from the source to this to meet the new member’s requirement), unfixed, rQoS, num, state = 1.

If the current node cannot ﬁnd an edge node meeting the above requirement during the ﬁrst step of the PFB computing process, it will initiate the second step. In this step, it computes the feasible paths from all the edge nodes to it (except those paths which uses the incoming node). It then duplicates the JOIN message, updates the PFB information and aQoS, and sends the JOIN messages along these paths. At the same time, it will also reserve the QoS resource, and add an entry to the multicast routing table denoted as: gaddr, parents nodes list, incoming interface, mQoS (the required QoS property from the source the current node meeting the new member’s requirement), unfixed, rQoS, the number of JOIN messages that have been sent out.

If the router cannot ﬁnd a PFB from its LNC information after the M number of hops, 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.

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 has been sent out. If some JOIN messages have not been conﬁrmed, it decreases the number of unconfirmed JOIN messages. Then, it checks the M.state of this multicast group entry. If M.state equals to 0, it means that the node has yet to start the ﬁrst step of PFB searching. So, it begins the ﬁrst step of PFB searching, ﬁnding those PFBs which do not pass the nodes that are in the M.in. It updates the M.in ﬁeld with the list of edge nodes of ﬁrst step searching result. Finally, it sets up several JOIN messages, ﬁlls corresponding ﬁelds and sends them along the PFBs. If M.state equals to 1, it begins the second step of the PFB searching: ﬁnding PFBs linking edge nodes to this node, sends JOIN request messages. Otherwise, it means that the current node has failed in the all the two steps of PFBs searching. It removes the entry of the multicast group. Then, it duplicates the
UNACK messages and sends to the children nodes of the multicast group (listed in the receivers’ interfaces list).

When a node receives a CONFIRM message, it first checks whether any entry of the same group has been confirmed with a better QoS property. If so, 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 entries that are not ‘fixed’, it also duplicates the CONFIRM message and sends them to the children nodes. For the confirmed items, it will send UNACK messages to their parent nodes.

Suppose node \(i\) receives a message from \(j\), the join-response process can be formalized as shown in Algorithm 2.

### 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 earlier 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 have least-cost path toward the target router or multiple PFBs to these edge nodes. If the node sends out multiple copies of JOIN messages, there will be more and more branches along the feasible branch searching paths. To control the number of branches, we only select the edge node that is the nearest one to the least-cost path from the current node to the target node.

2. In the second step of ‘PFB computing process’, QMBF finds PFBs for every edge nodes, duplicates the JOIN message and sends them along these partial branches. If this situation happens too frequently, there will be more and more branching nodes causing excessive overhead. We use the concept ‘maximum branch level’ 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. We use ‘bounded hops’ to control the flooding traffic, which decides the valid hops the flooding messages can travel.

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

### 5.6. Inter-domain routing operation

QMBF is designed to be used in both intra-domain and inter-domain levels. For inter-domain case, the QoS-based multicast routing operation works among border routers. We assume that each border router periodically sends probing messages to its neighboring border routers. This probing message can be sent separately or piggybacked with other messages. A probing message collects the QoS state information between one border router and one of its neighboring border routers. When a border router gets a
probing messages, it responds back the probing message to its sender.

After collecting the responses of the probing messages, a border router can have the QoS state information between itself and its neighboring border routers. When QMBF runs for inter-domain routing, the bounded flooding is deployed between border routers. Every border router broadcasts the QoS state information between itself and its neighbor border routers. The flooding messages are valid for $M$ hops (here, $M$-hops mean the message is valid until it arrives at the $M$th border router).

Currently, the inter-domain routing protocol is BGP, which is a path vector routing protocol. So, for inter-domain routing case, the border routers do not need to broadcast their reachable information. A border router can utilize the BGP routing information to get the domain reachable information of its edge nodes (in LNC).

The routing process for inter-domain case is the same as the intra-domain case as we described in the previous sections.

6. Simulation results and analysis

In this section, we conduct a series of simulations to study and compare the performance of QMBF-$mn$ ($m$ is the flooding hops while $n$ is maximum branch level.) with several multicast routing protocols.

In the simulation, we simulate four other algorithms: single-path joining protocol, directed spanning-joins protocol, QMRP2, and QMRP3. The single-path joining routing is based on current least-cost routing protocols. When a new member wants to join a multicast group, it sends join request toward the multicast source or core router. Whenever an on-tree node gets this request, it will reply a response message. The response message will collect the QoS state along the path from the on-tree node to this new member. If the QoS state along the path meets the new member’s QoS requirement, it will send confirm message along this path. Otherwise, it will give up this join request. Spanning-joins protocols was proposed in Ref. [1]. The directed spanning-joins protocol is an enhanced version of regular spanning joins. Its goal is to minimize the impact of source initiated joining spacing trees and to direct the join request messages toward the target router [1]. QMRP-$n$ is the QoS-aware multicast routing algorithm which is proposed in Ref. [2] ($n$ is the maximum branching degree). We simulate two versions of QMRP-$n$; QMRP2 and QMRP3. The simulations from Ref. [2] show that QMRP2 and QMRP3 can achieve higher successful ratio with good scalability.

For QMBF-$mn$, we simulated six schemes: QMBF-12, QMBF-13, QMBF-22, QMBF-23, QMBF-32, and QMBF-33.
6.1. Simulation environment

We implement single-path joining protocol, directed spanning-joins protocol QMRP and QMBF based on network simulator-2.1 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 \((a \times a)\) grid. A link exists between the nodes \(u\) and \(v\) with the probability \(p(u, v) = a \exp(-d(u, v)) / b \sqrt{2a}\), 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. To the best of our knowledge, this kind of topology does not give any advantage to any specific multicast routing protocol. The network topologies we used in the simulation have a fixed size of 100 nodes in a 100 \(\times\) 100 grid.

Based on above topology generation method, for each simulation, a network topology, a multicast tree, and a new join member out of the tree are randomly generated. When searching for a feasible branch to connect a new member to the existing multicast tree, different tree sizes will affect the protocols’ performance.2 We use two different tree size: 10 and 40 nodes. In each simulation, the new member has a randomly generated QoS requirement. The QoS state of each link is also randomly generated which meets the new member’s QoS requirements or not, based on a random probability. The link success ratio is defined as the ratio of links that meet the new member’s QoS requirement. The default least-cost unicast routing used in the simulation is distance vector routing (Bellman–Ford algorithm) protocol, which is implemented in NS-2.

For QMBF, the bounded flooding messages for QoS state and least-cost reachable information is sent separately. Every node does bounded flooding only when the QoS state is changed (in practical, we can minimize the bounded flooding traffic by doing bounded flooding when its QoS state is change over some threshold.). Every node compute its least-cost reachable information based on its least-cost routing table. A node does bounded flooding only when its routing table is changed. In reality, if the network is using link-state routing protocol (such as OSPF), every node does not need to do a least-cost reachable bounded flooding. Instead, the node can compute its edge nodes’ least-cost reachable information based on the link state messages it receives.

For each situation (different protocols, multicast group sizes, link success ratios), 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

Fig. 3 depicts the success ratio of different modes of QMBF with two different size of multicast groups. We observe that QMBF-\(mn\)'s success ratio increases with increase in \(m\) and \(n\). The reason is that \(m\) decides 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 QoS requirement. The variable \(m\) is the width of the path the join request messages will pass during the feasible branch searching process. With the increase of \(m\), the path will cover more nodes, which also means that the success ratio will increase. The variable \(n\) is the maximum branch level, which is the times that a join request message will be duplicated. That means, with the increasing in \(n\)’s value, the probability for a multicast routing activity to enter multiple path searching process is higher, which in turn increases the probability of finding a feasible branch.

Fig. 4 compares the success ratios of all the simulated protocols with the two different sizes of multicast groups. For the two sizes of multicast tree, The protocol’s relative success ratio comparison have the similar trend: success ratios of QMBF-22 and QMBF-23 are better than QMRP2, QMRP3, spanning-joins protocol, and single-path join.

1 http://www.isi.edu/nsnam/ns/
2 The tree size is the number of nodes a multicast tree covers. It includes both the internal nodes and the end nodes, which usually are directly connected to the multicast group members.
protocol, QMRP2 and QMRP3 are better than spanning-joins and single-path routing.

The reason behind the above simulation results can be explained using Fig. 5. Assume a new member \( o \) wants to join an existing multicast group, the nearest on-tree node is \( i \). Suppose the least-cost path from \( o \) to \( i \) is \( o-l-k-j-i \), 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 \( o-l \) link does not meet the QoS requirement, it will enter the multiple path search period. For example, \( o \) 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 cannot 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, \( o \) will compute a PFB \( o-n-k \) at once when it receives the user’s JOIN message.

6.3. Discussions about simulation

(i) **Network size does not affect the performance advantage of QMBF.** In our simulation, we select the network size with 100 nodes. In reality, the network size maybe much bigger than 100. However, the performance of flooding messages is valid for the same number of hops, the per-router burden to process bounded flooding is the same. Second, as the size of network increases, the feasible branch searching path length may increase. However, the join request always traverses along a \( M \)-hop width path toward the target router. As shown in Fig. 5, even in a larger network environment, the advantage of QMBF still exists. Third, as we have discussed in Section 6.2, QMBF can be used as inter-domain QoS multicast routing protocols, which also enhance its advantage over other protocols.

(ii) **The number of multicast sessions does not affect the performance of QMBF.** In the simulation, we only simulate the case in which there is only one multicast session in the network. In QMBF, the routing protocol deals with routing activity separately for different multicast sessions. Also, when there are two join requests for the same session arriving at the same node, the node will combine these two join requests. When multiple sessions co-exist, they only affect the per-link success ratio. So, when multiple sessions co-exist, it does not affect the relative performance advantages of QMBF.

(iii) **QMBF does not incur loops in multicast tree.** A loop means that one node has more than one parents nodes in the multicast routing tree. The M.in field of QMBF routing table may have more than one parent node. These parent nodes are only useful for the feasible branch searching period. As shown in Algorithm 2, if one node receives more than one CONFIRM messages, it will select the path that has better QoS service. Consider Fig. 6 in which \( A-B, B-C \) and \( B-D \) are the on-tree links. In (a), \( E \) sends out two JOIN messages to \( C \) and \( D \). After a while, it may get two CONFIRM messages from \( C \) and \( D \) as shown in (b). Suppose the path from \( D-E \) can provide better QoS service than the path \( C-E \), it will keep \( D-E \) and send PRUNE message to \( C \) and remove the link \( C-E \). Using this method, QMBF can make sure that every node at any time has only one parent node in the multicast tree, which avoids the formation of loops.
QMBF does not incur any concurrency problems. QMBF is a distributed routing algorithm. However, it does not have the concurrency problems that exist in most distributed systems. First, as we have discussed in (ii) different feasible branch searching processes do not affect each other. They are processed by nodes separately and thus cannot incur any concurrency problems. For a specific feasible branch searching process, the PFB searching activities at different nodes occur sequentially. That means, a node sends JOIN message to another node only after it finishes its PFB searching. When the second node does PFB searching, the first node would have already finished its searching process. Thus QMBF can avoid any concurrency and deadlock problems.

7. Conclusion and future work

In this paper, we propose a new QoS-aware multicast routing protocol called QMBF. QMBF is based on a bounded flooding technique, in which every network node is assumed to have the knowledge of the LNC topology as well as the 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 show that QMBF can achieve better success ratio than other QoS-based multicast routing protocols.

There are still several open problems which will be addressed in our future research work. In QMBF-\( mn \), \( m \) is the valid number of hops of bounded flooding messages. It is also the width of the path the join request messages will pass, which in some sense decides the searching efficiency and the success ratio. In this paper, we assume that all the bounded flooding messages have the same value of \( m \). It will be an interesting research topic if different nodes select the value of \( m \) adapting to the dynamic network environment. When multiple feasible branch searching process happen at the same time, the effective reservation and utilization of network is another future research direction. Furthermore, when QMBF is used for inter-domain routing, how to efficiently integrate it with BGP and evaluate its performance is another topic for future investigation.

Acknowledgments

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

References


