

# 支持 QoS 的分层数据传输的动态组播路由算法\*

陆慧梅<sup>†</sup>, 向勇, 史美林, 杨敏

(清华大学 计算机科学与计算机系, 北京 100084)

## A QoS-Based Dynamic Multicast Routing Algorithm for Streaming Layered Data

LU Hui-Mei<sup>†</sup>, XIANG Yong, SHI Mei-Lin, YANG Min

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-62785609, E-mail: lhmei@csnet4.cs.tsinghua.edu.cn, <http://cscw.cs.tsinghua.edu.cn>

Received 2003-06-09; Accepted 2003-07-24

Lu HM, Xiang Y, Shi ML, Yang M. A QoS-based dynamic multicast routing algorithm for streaming layered data. *Journal of Software*, 2004,15(6):928-939.

<http://www.jos.org.cn/1000-9825/15/928.htm>

**Abstract:** To support different QoS requirements in terms of bandwidth and delay constraints imposed by heterogeneous and dynamic joining receivers, this paper proposes an algorithm named QDMR-LD (QoS-based dynamic multicast routing for streaming layered data). When a new receiver joins, receiver-oriented path searching heuristic is used to find a feasible path with minimum cost from the multicast tree to the receiver. RBMF (reverse best metric forwarding) mode proposed in our previous work is adopted in this paper to increase the joining success ratio. When a receiver leaves, corresponding part of the multicast tree is pruned. Simulation results show that QDMR-LD increases the success ratio and lowers the multicast tree cost compared with other related schemes.

**Key words:** QoS-based multicast routing; dynamic multicast routing; layered multicast; heterogeneous

**摘要:** 为了支持组成员的异构性和动态性,满足组成员不同的带宽和时延约束要求,提出了支持 QoS 的流式分层数据传输的动态组播路由算法(简称:QDMR-LD)。当新的组成员加入时,使用面向接收者的启发式搜索方法找到一条位于组播树和组成员之间的可行的并具有最小代价的路径。为了提高组成员的加入成功率,采用了以前工作中提出的 RBMF 转发算法。当组成员离开时,位于组播树上多余的部分将被剪裁。仿真结果表明,与其他相关工作相比,QDMR-LD 在提高了加入成功率的同时,降低了组播树的代价。

**关键词:** 支持 QoS 的组播路由;动态组播路由;分层组播;异质性

中图分类号: TP393 文献标识码: A

\* Supported by the National Natural Science Foundation of China under Grant No.90104002 (国家自然科学基金)

LU Hui-Mei was born in 1974. She is a Ph.D. candidate at Department of Computer Science and Technology, Tsinghua University. Her current research area is QoS-based multicast routing. XIANG Yong was born in 1967. He is an associate professor at Department of Computer Science and Technology, Tsinghua University. His current research areas are multicast routing and ad hoc networks. SHI Mei-Lin was born in 1938. He is a professor and doctoral supervisor at Department of Computer Science and Technology, Tsinghua University. His current research areas are CSCW and networks. YANG Min was born in 1978. He got his BS degree at Department of Computer Science and Technology, Tsinghua University. His research area is networks.

Multicast employs a tree structure in the network to efficiently deliver the same data stream to a group of receivers. In general, receivers differ greatly in their capabilities of accessing the Internet. To accommodate heterogeneous receivers, a multicast source may change the data transmission rate by using rate-adaptive data or combining a layered compression algorithm with a layered transmission scheme<sup>[1]</sup>. However, the former approach tends to be suboptimal because there is no single target rate for a group of heterogeneous receivers. Layered multicast is an effective method, in which a stream is encoded into a base layer and several successive enhancement layers, and then receivers in a multicast session can receive different layers according to their bandwidth capabilities. In this area, great efforts can be classified into two different ways: (i) Receiver-driven schemes<sup>[2-5]</sup>, in which a source transmits each layer of a session on a separate IP multicast group, and each receiver decides which layers it can receive. These researches focus on designing an adaptive congestion control mechanism. (ii) Router-supported schemes<sup>[6-11]</sup>, in which a source sends all the layers of a session on a single IP multicast group and the involved routers decide which layers to forward. These researches aim to construct a multicast tree subject to certain constraints or optimization objectives. In receiver-driven schemes, each layer is associated with a different multicast group. As a result, several multicast addresses are allocated for a single multicast session and several corresponding multicast trees will be set-up and maintained. It is not scalable because of the limited multicast addresses in IPv4. If a sparse-mode multicast routing protocol is used<sup>[12]</sup>, or the QoS requirements of receivers in different layers are considered when multicast trees are built<sup>[6]</sup>, data from different layers may be routed via different paths, and hence results in different delays. The size of data buffer at the receiver has to be large enough to ensure that data from different layers can be decoded synchronously. In router-supported schemes, only a single multicast tree is constructed and maintained during a multicast session. All the layers of a data to a receiver are delivered via the same path. To indicate which layers should be forwarded, a router usually adds an item to an out interface of a multicast routing entry. The main difference of those two approaches is illustrated in Fig.1. Our method belongs to the router-supported scheme.

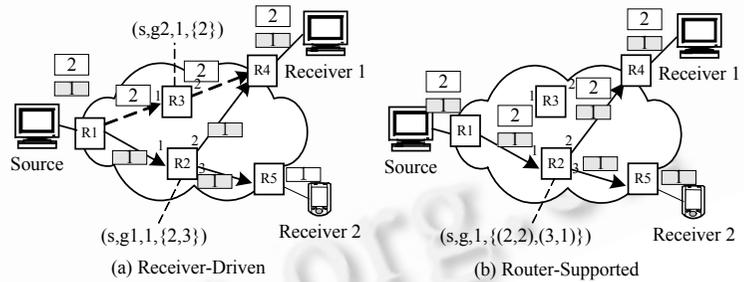


Fig.1

According to the membership behavior of the receivers, multicast routing problems can be classified into two types: (i) the static problem, where all receivers are known in advance, and (ii) the dynamic problem, where receivers can join and leave dynamically throughout the multicast session. Static multicast routing algorithms have been studied in Refs.[7,8] for streaming layered data. However, for some applications, like live audio or video broadcast, a dynamic routing algorithm is mandatory for receivers' frequent tuning in and tuning out<sup>[13,14]</sup>.

In order to support QoS for multicast with dynamic member joining, we have presented QDMR (QoS-based distributed Dynamic Multicast Routing)<sup>[15]</sup>, in which RMBF (Reverse Best Metric Forwarding) mode is proposed to increase the success ratio of routing. However, QDMR does not consider QoS requirement with heterogeneous bandwidth. To address this problem, QDMR is expanded to QDMR-LD in this paper. There are three major goals in the design of QDMR-LD. 1) When a new receiver joins, a feasible path is found through which the QoS requirements imposed by the receiver both in terms of heterogeneous bandwidth and delay are satisfied. 2) When a receiver leaves, corresponding part of the multicast tree can be pruned. 3) The amount of total network resources used is minimized.

The rest of the paper is organized as follows. In Section 1, the related work is summarized. In Section 2, the

network model, the data model and the problem formulation are presented. Section 3 gives a detailed description of the QDMR-LD. Simulation results are demonstrated in Section 4. Section 5 draws our conclusions.

## 1 Related Work

The objective in this section is to briefly touch upon the most important contributions that are directly relevant to this paper. Existing dynamic multicast routing schemes considering QoS and/or heterogeneous receivers are reviewed in the following.

YAM<sup>[16]</sup>, QoSMIC<sup>[17,18]</sup>, QMRP<sup>[19,20]</sup> and RIMQoS<sup>[13]</sup> are among the work on QoS-based dynamic multicast routing. They build a multicast tree in a distributed fashion and allow a dynamic joining and leaving behavior of the receivers. YAM and QoSMIC provide multiple candidate paths, from which a best path can be chosen as a branch of the multicast tree. QMRP starts with a single path but, when necessary, it can expand to multiple path searches. YAM, QoSMIC and QMRP adopt RPF (Reverse Path Forwarding) mode to forward the search messages toward on-tree nodes. Only the message is forwarded if received on an interface used to send unicast data to the new receiver. RIMQoS assumes link-state information and a QoS-based unicast routing protocol are available. A receiver computes a path from the multicast source to itself. Then it sends a join-request along the path to join the multicast session. Just as QDMR, all of them do not aim at supporting the heterogeneous bandwidth requirements by receivers.

Research works that support dynamic and heterogeneous receivers can be found in Ref.[9] and QMRH<sup>[10]</sup>. The contribution in Ref.[9] is the use of the number of layers of the source signal rather than the hop as the cost of a link. But no explicit QoS in terms of bandwidth and delay bound is considered. In QMRH, every node maintains an auxiliary routing table that records an  $h$ -hop path with the maximum residual bandwidth, where  $1 \leq h \leq H$ , and  $H$  is the maximum hop count and be set implicitly as the diameter of the network. So, a node can easily calculate a path from itself to another node with a given bandwidth and delay requirement. The algorithm works by growing the tree from the source and recursively adding the next receiver that has the highest receiving capability among the remaining static receivers that have not been connected to the tree. When a new dynamic receiver intends to join an existing multicast session, the source multicasts a message to all on-tree nodes to find a join path. The problem is that when an on-tree node computes a join path, it excludes paths without enough residual bandwidth from consideration even though part of the path lies on the multicast tree and bandwidth has already been reserved. Thus, if the quality level of a new receiver is higher than that of on-tree nodes, it maybe fails to find an existing feasible path. QMRH is more appropriate to multicast applications with static receivers than to those with dynamic receivers.

## 2 Problem Formulation

In this section, we describe the network model and traffic model. On the basis of the two models, our problem is formulated.

### 2.1 Network model

A network can be modeled as a weighted digraph  $G = (V, E)$ , where  $V$  and  $E$  are the sets of nodes and links, respectively. For any link  $\bar{e} \in E$ , we define a link-delay  $d(\bar{e}): E \rightarrow R^+$ , available bandwidth  $b(\bar{e}): E \rightarrow R^+ \cup \{0\}$ . Here  $d(\bar{e})$  is a measure of the delay that packets experience on link  $\bar{e}$ , including the queuing delay, transmission time, and propagation delay. A multicast tree  $T$  is a sub-graph of  $G$ , which can be represents as  $T=(V_T, E_T)$  ( $V_T \subseteq V, E_T \subseteq E$ ), where  $V_T$  is the set of nodes, including not only the receiver nodes but also the relay nodes on the

multicast tree, and  $E_T$  is the set of the edges connecting the nodes belonging to  $V_T$ .  $V_g (V_g \subseteq \{V - \{s\}\})$  is the set of receivers whose group address is  $g$ , here  $s \in V$  is the multicast source. We use  $P_T(s, v_d)$  to denote the path from a source node  $s$  to a receiver node  $v_d (v_d \in V_g)$  along the tree  $T$ .

## 2.2 Traffic model for layered data

There are two kinds of layering schemes: cumulative and non-cumulative<sup>[1]</sup>. In this paper, we suppose traffic is encoded cumulatively. The signal is encoded hierarchically into  $L$  layers with the first layer containing the most essential information, and Layer  $i$  contains bits less significant than those in Layer  $i-1$ , but more significant than those in Layer  $i+1$ . Thus, a signal with higher layers has a better quality and at the same time requires more bandwidth for transmission. Only layers that a given link can manage are forwarded.

## 2.3 Problem formulation

We assume that data are encoded into different layers and are distributed from a source to dynamic receivers with heterogeneous QoS in terms of bandwidth and delay constraints. We take the assumption proposed in Ref.[8] that all the layers of data to a receiver must be delivered via the same path.

Suppose any on-tree node  $u (u \in V_T)$  keeps the quality level of the session denoted as  $l_u$ , the corresponding reserved bandwidth  $br(\bar{e})$  on every downstream link  $\bar{e}$ , and the delay  $d(P_T(s, u))$  from the source along the on-tree path. Suppose the bandwidth required by each layer is characterized by  $R_i, 1 \leq i \leq L$ , and  $R^k = \sum_{i=1}^k R_i, 1 \leq k \leq L$ , is the aggregated (up to layer  $k$ ) bandwidth of the data.

**Problem formulation:** When a new receiver  $v_d$  intends to join a group, it proposes that  $R_d^r$  is the maximum rate it can receive and the delay from the source to  $v_d$  should be less than  $D_d$ . If  $R^k \leq R_d^r < R^{k+1}$ , then receiver  $v_d$  is capable of receiving up to layer  $k$  of the data. We name it a layer- $k$  receiver. The problem is to search a path  $P(u, v_d)$  from any on-tree node  $u$  to the receiver which satisfies the bandwidth and delay constraints, and meanwhile the total bandwidth used by the session is tried to be minimized.

1) For any link  $\bar{e} \in P_T(s, u)$ , from the source to the added node, the bandwidth reserved for the group plus the available bandwidth on the link must be greater than the bandwidth constraint of node  $v_d$ ,  $\min_{\bar{e} \in P_T(s, u)} (br(\bar{e}) + b(\bar{e})) \geq R^k$ .

2) Bandwidth constraint:  $b(P(u, v_d)) \geq R^k$ , where  $b(P(u, v_d)) = \min_{\bar{e} \in P(u, v_d)} \{b(\bar{e})\}$ .

3) Delay constraint:  $d(P_T(s, u)) + d(P(u, v_d)) \leq D_d$ , where  $d(P(u, v_d)) = \sum_{\bar{e} \in P(u, v_d)} d(\bar{e})$ .

4) Cost optimization:  $c(P(s, v_d)) = h(P(u, v_d)) * R^k + \sum_{\bar{e} \in P_T(s, u)} \max(R^k - br(\bar{e}), 0)$ . Here  $h(P(u, v_d))$  means the hop counts

of  $P(u, v_d)$ . The cost includes two parts: one is the cost associated with the links along  $P(u, v_d)$  which are not on the current multicast tree; the other is the cost increment associated with the links already on the multicast tree in order to satisfy the quality level requirement of  $v_d$ . From the set of feasible paths, one with the minimum cost is chosen as the branch. Hence, QDMR-LD can construct a near-optimal multicast tree<sup>[15]</sup> and lead to efficient uses of the network resources.

### 3 Proposed Dynamic Algorithm

#### 3.1 Overview

QDMR-LD is designed for multicast applications with a single source and dynamic join/leave of receivers. Routes from the source to any receiver must satisfy the receiver’s heterogeneous bandwidth and delay requirements while inter-receiver paths have no QoS guarantees. Before providing a detailed description of the operation of QDMR-LD, we briefly describe the messages exchanged between routers and the state information kept in routers. After that, the difference of RBMF and RPF is illustrated.

Table 1 contains a list of the protocol messages exchanged among the multicast routers to establish and maintain the multicast tree. Similar to PIM<sup>[21]</sup>, we assume a router keeps the forwarding information such as source/group, in-interface and the quality level of in-interface, out-interface set and the quality level of each out-interface. The quality level of in-interface equals to the maximum level of out-interfaces. In the previous section, the quality level kept in a node refers to that level of in-interface. Additionally, in a forwarding entry, each out-interface is labeled as active or pending. Multicast traffic is only forwarded to out-interface(s) marked as active.

**Table 1** An explanation of messages used in QDMR-LD

Message	Explanation
Join-Request	A new receiver searches for an on-tree node. The message is forwarded in RBMF mode which is more prone to satisfy the receiver’s QoS requirements.
Bid	An on-tree candidate tells the new receiver a feasible path between them and the cost of the path.
Bid-Reply	The new receiver orders one of the on-tree candidates to establish the connection.
Set-Up	An on-tree candidate sets up the connection and reserves the bandwidth.
Accept	The new receiver informs the success of the establishment.
Deny	An intermediate node informs the failure of the establishment.
Probing	A node sends it to probe a node whether the path between them satisfies the QoS requirement.
Prune	A leaving node tears down the unwanted part of the multicast tree.

RBMF<sup>[15]</sup> is a forwarding mode where a data packet is accepted for forwarding to other interfaces except the incoming, if and only if the packet travels from a path whose reverse path is the BMP (Best Metric Path) from the current node to the new receiver. Here BMP means that the path can satisfy the bandwidth and delay requirements by the receiver, meanwhile its delay is less than the minimum delay from the current node to the receiver kept at the current node so far. In Ref.[15], we have proven that RBMF can improve the success ratio of routing for prior

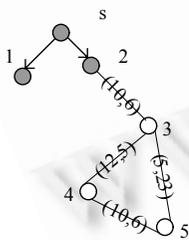


Fig.2 RPF vs. RBMF

consideration of satisfying a new receiver’s QoS requirements. Figure 2 illustrates how RBMF differs from RPF. Node 5 intends to join the group with a bandwidth of 9 Mbps and delay bound of 25 ms. The tuple of (5,23) means the available bandwidth is 5 Mbps and the delay is 23 ms on the link  $e(3,5)$ . Suppose node 3 receives two join-request messages. One is from node 5 directly to node 3. Another is from node 5, through node 4 and finally to node 3. In RPF mode, the first one is accepted to forward. In fact, the path can not satisfy node 5’s requirement because the bandwidth of  $e(3,5)$  is 5 Mbps which is less than 9 Mbps. While in RBMF, the second one is accepted to forward because  $b(3→4→5)$  is enough for 9 Mbps and  $d(3→4→5)$  is less than both 25ms and  $d(3→5)$ .

#### 3.2 Detailed description of QDMR-LD

QDMR-LD uses receiver-oriented mechanisms: (1) to find and set up routes that can support a given receiver’s QoS requirements; (2) to prune unwanted parts of the multicast tree left by a leaving receiver. In the following we will describe the joining and leaving procedure separately.

3.2.1 The search for candidates

We assume the address of the multicast group is known to potential receivers through some kind of advertisement or query mechanism. When a node  $v_d$  receives a join-request from a local host, it floods a Join-request to identify neighboring on-tree nodes if it is not already on the tree. If an on-tree node receives a Join-request from an off-tree node, we name it a first-on-tree node; otherwise we name it an intermediate-on-tree node.

3.2.1.1 An off-tree node receives a join-request

When an off-tree node receives a Join-request, it will forward or discard the message based on route information carried in the request. Detailedly, suppose two neighboring off-tree nodes  $v_x, v_y \notin V_T$ , and the link  $(v_x, v_y) \notin E_T$ . A three-tuple of  $(s, g, v_d)$  records the minimum delay and the cost from  $v_x$  to  $v_d$ . When  $v_x$  receives the join-request from  $v_y$ , it decapsulates  $R^k, P(v_y, v_d), d(P(v_y, v_d))$  and  $c(P(v_y, v_d))$  from the message. Then it does as follows:

Step 1. Initialize  $(s, g, v_d)$  if it does not exist: set  $Min\_Delay = \infty, Min\_Cost = \infty$ . Then go to Step 2 to identify whether the reverse route traveled by the join-request is the BMP from  $v_x$  to  $v_d$ .

Step 2. If  $b(v_x, v_y) \geq R^k$ , go to Step 3; else abandon join-request message.

Step 3. Set  $d(P(v_x, v_d)) = d(v_x, v_y) + d(P(v_y, v_d))$ . If  $d(P(v_x, v_d)) \leq D_d$ , go to Step 4; else abandon join-request message. End.

Step 4. if  $d(P(v_x, v_d)) < Min\_delay$ , go to Step 5; else abandon join-request message. End.

Step 5. Set  $c(P(v_x, v_d)) = c(P(v_y, v_d)) + R^k$ . Encapsulate  $P(v_x, v_d), d(P(v_x, v_d))$  and  $c(P(v_x, v_d))$  into join-request

and forward the message to other interfaces except the incoming interface. Update the value kept at  $v_x$  node. Set the parameters of  $(s, g, v_d)$ :  $Min\_Delay = d(P(v_x, v_d)), Min\_Cost = c(P(v_x, v_d))$ . End.

3.2.1.2 A first-on-tree node receives a join-request

An on-tree node  $u$  receives a join-request from its neighbor off-tree node  $v_z$ , it will check the QoS information for that source/group. Firstly, it checks whether the reverse path of Join-Request is the BMP or not. Then it checks the delay along the path currently used. Finally it examines the quality level. Suppose the delay from  $s$  to  $u$  kept at  $u$  is  $d(P_T(s, u))$ .

Step 1. Check whether the path is the BMP from  $u$  to  $v_d$  or not. Do the same as Steps 1~4 of off-tree node in the above section A.1. If it is the BMP, go to Step 2, else go to Step 6.

Step 2. Set  $d(P(s, v_d)) = d(P_T(s, u)) + d(P(u, v_d))$ . If  $d(P(s, v_d)) \leq D_d$ , go to Step 3, else go to Step 6.

Step 3. Set  $c(P(u, v_d)) = c(P(v_z, v_d)) + R^k$ . Update the value kept at  $u$  node. Set the parameters of  $(s, g, v_d)$ ,  $Min\_Delay = d(P(u, v_d)), Min\_Cost = c(P(u, v_d))$ . Go to Step 4.

Step 4. Examine the quality level. If the quality level of the new receiver is higher than that of  $u$ , the join-request is forwarded toward the upstream. End; else go to Step 5.

Step 5. Encapsulate  $P(u, v_d)$  and  $c(P(u, v_d))$  into Bid message and unicast the message to  $v_d$ . End.

Step 6. Abandon join-request message. End.

Let's consider Fig.3 and assume the node in question is node 3 and its quality level is layer 2. The quality level

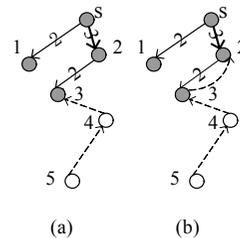


Fig.3 Join-Request received by on-tree node

requirement of node 5 is  $k$ . In Fig.3(a), if  $k \leq 2$ , it sends a Bid message towards node 5 with the route  $(3 \rightarrow 4 \rightarrow 5)$  and its cost  $(2 \times R^k)$  specified in it. Node 3 considers node 5 as a tentative dependent, and cannot leave the tree unless the tentative status is timed out. However in Fig.3(b), if  $k > 2$ , it needs to forward the request to the intermediate upstream node 2.

3.2.1.3 An intermediate-on-tree node receives a join-request

If an on-tree node  $u'$  receives a Join-request from neighbor on-tree node  $u$ , it extracts the cost  $c(P(u, v_d))$  and path  $P(u, v_d)$  from the message. Then it will do as follows:

Step 1. Check whether the bandwidth reserved for the group plus the bandwidth available on the downstream link is greater than the bandwidth constraint of  $v_d$ .  $br(e(u', u)) + b(e(u', u)) \geq R^k$ . If so, go to Step 2 otherwise go to Step 4.

Step 2. Set  $c(P(u', v_d)) = c(P(u, v_d)) + (R^k - br(e(u', u)))$ . Then start examination of the quality level. If the quality level of  $v_d$  is higher than that of  $u'$ , join-request is forwarded toward the upstream node. End; else go to Step 3.

Step 3. Encapsulate  $P(u', v_d)$  and  $c(P(u', v_d))$  into bid request message and unicast the message to  $v_d$ . End.

Step 4. Abandon join-request message. End.

3.2.2 Connection establishment

If there is no feasible branch, node  $v_d$  will not receive a bid message at all; otherwise it may receive  $N$  Bid messages. It extracts the route and its cost from the message, and then selects a path with minimum cost as a branch from the multicast tree to  $v_d$ . Suppose  $P(u, v_d)$  is the final minimum cost path. A bid-reply is sent along  $P(v_d, u)$  to inform  $u$  to establish the connection. Once  $u$  receives the bid-reply, it sends back a set-up.

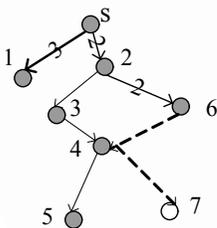


Fig.4 Set-Up conflicts

When an on-tree node receives a set-up and the interface from which it receives the set-up is the in-interface of the multicast routing state, it reserves the bandwidth value of  $\max((R^k - br(e)), 0)$  and updates the quality levels of the in-interface and out-interface. Then it forwards the set-up to the next hop. It is worth noting that the interface from which an on-tree node receives a set-up may not be the in-interface of the multicast routing state. As shown in Fig.4, the set-up path is  $6 \rightarrow 4 \rightarrow 7$ . Node 4 is a new on-tree node and receives a set-up from node 6. The situation is a little more complicated. Assume  $d4 = d(s \rightarrow 2 \rightarrow 3 \rightarrow 4)$ ,  $l4$  is the current quality level of node 4.  $d4' = d(s \rightarrow 2 \rightarrow 6 \rightarrow 4)$ ,  $l4'$  is the possible quality level. There are four possibilities:

The first case is that  $d4 < d4'$  and  $l4 > l4'$ . In this situation, node 4 can simply add the out-interface towards node 7 to the existing multicast routing entry for that source/group, mark it as pending, and start a timer. Then it forwards the set-up to node 7. The pending state will be pinned when an accept message is received, or will be flushed out if a deny message is received, or the timer times out.

The second possibility is that  $d4 > d4'$  and  $l4 > l4'$ . Node 4 will do the same as in the first case. But because the delay on the path of  $(s \rightarrow 2 \rightarrow 3 \rightarrow 4)$  is greater than that on  $(s \rightarrow 2 \rightarrow 6 \rightarrow 4)$ , node 4 needs to send a probing message to node 7 to check the delay constraint. If the delay bound can not be satisfied, node 7 will send back a deny message.

The third case is that  $d4 > d4'$  and  $l4 < l4'$ . Then, node 4 will forward the set-up to node 7. It also adds a pending routing entry with node 6 as the upstream neighbor and marks it as the preferred upstream node for that source/group. When node 4 receives an accept message from node 7, it will enable the new routing entry and forward the accept message to node 6. It also sends a prune message to node 3, but only after the multicast traffic is received from node 6. This causes a switch from the old path to the new one. By construction, delay on the new path

from  $s$  to node 4 is less than that on the old path. Thus delay constraint for all existing downstream of node 4 is still satisfied, and the quality level is not degraded.

The last case happens when  $d_4 < d_4'$  and  $l_4 < l_4'$ . In this case, if the old path is switched to the new one, delay on the new path from  $s$  to node 4 is greater than that on the old path. Thus delay constraint for all existing downstream of node 4 can not be guaranteed. Node 4 sends a deny message to node 6 to inform the set-up failure.

When an off-tree node, not the new receiver, receives a set-up, it reserves the bandwidth on the downstream link. If the reservation fails, it sends a deny message towards upstream. Otherwise, it creates a new multicast routing entry (source/group, in-interface and its quality level, out-interface and its quality level). The out-interface is the interface to which it should forward the set-up based on the route information carried in the message, and the in-interface is the interface from which it receives the set-up. It then forwards the set-up to the next hop, and marks the newly created routing entry as "pending". The routing entry will be marked as active when an accept message is received. Such an accept message is then forwarded upstream. A pending routing entry is flushed if a deny message is received or its timer goes off. When the new receiver receives a set-up, it replies an accept message to announce that it has been connected to the multicast tree successfully.

### 3.2.3 Leaving a group

When a node receives a leaving request from a local host, it sends a prune message upstream with zero quality level and zero bandwidth specified in it. The quality level and bandwidth in a prune message indicate respectively the level that the downstream nodes are still going to receive and the level's corresponding bandwidth. For simplicity, we use  $l$  and  $b$  to represent them respectively. On receiving a prune message, a node does the following operations:

Step 1. Check the value of  $l$ . If the value is zero, the reserved bandwidth  $br(e)$  on downstream link is released and the out-interface is erased from the out-interface set. Go to Step 2; otherwise the quality value of out-interface is updated to  $l$ , and the reserved bandwidth of  $(br(e)-b)$  on the downstream link  $e$  is released with  $br(e)$  updated to  $b$ . Go to Step 3.

Step 2. Check whether out-interface set is null. If not, go to Step 3; otherwise, there is no out-interface, the routing entry and other states (e.g. delay from the source to itself and the bandwidth reserved) will be demolished and a prune message with zero quality level and zero bandwidth is sent upstream further. End.

Step 3. Compare the quality level of in-interface with the maximum level of out-interfaces. If the former is higher than the latter, it updates the former with the latter and sends a prune message towards the upstream node with the maximum level of out-interfaces and its corresponding bandwidth. End.

The prune message travels upstream along the on-tree branch to release the needless bandwidth until it reaches a node where the quality level of in-interface is what the downstream nodes still want to receive. While in Refs.[10,21], a prune procedure ceases when the prune message reaches a fork node (i.e., a node with more than one out-interface and/or with receivers on its directly attached subnet).

## 4 Simulation Results

Three algorithms, SPT (shortest path tree), QMRP<sup>[19,20]</sup> and YAM<sup>[16]</sup>, are used to evaluate the performance of QDMR-LD. These three algorithms are expanded to support heterogeneous receivers. Four algorithms differ mainly in how to search for an on-tree node, which is summarized in Ref.[15]. The algorithm in Ref.[10] is also suitable for comparison but not included because it uses a customized network simulator named NetSim<sup>Q</sup> that is not available for the public use.

#### 4.1 Simulation environment

The Network Simulator (NS-2)<sup>[22]</sup> is used as the basic simulation platform. The topologies used in the simulation are the well-known MCI-vBNS Internet topology (Fig.5)<sup>[10]</sup> and Mbone topology that has 84 nodes<sup>[23]</sup>.

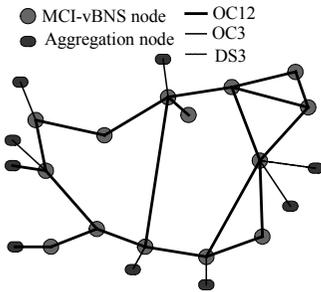


Fig.5 The vBNS topology

The link capacities for the OC12, OC3, and DS3 links are 622 Mbps, 155.5 Mbps, and 45 Mbps, respectively in the MCI-vBNS Internet topology. In the Mbone topology, the link capacity is distributed between 6.312Mbps and 44.736Mbps. The delay of each link follows a normal distribution with a mean of 30 ms. In our simulation, the bandwidth of the video data is 2.43 Mbps. The video data is encoded into three layers: a base layer, a most significant layer, and an enhancement layer with the bit rates being 30%, 50%, and 20% of the total video bit rate, respectively. Each group has one randomly selected source. The receiving capability,  $R_d^r$ , of each receiver  $v_d$  is uniformly distributed among 1, 2, 3 layers.

#### 4.2 Performance parameters

In general, four performance metrics, joining success ratio, tree cost, average message overhead and computation complexity, are paid more attention when a QoS-based multicast routing algorithm is evaluated. Computation complexity is the computation needed to find paths and construct the tree. In QDMR-LD, nodes do not perform complex computation and usually do some message processing. Therefore, the evaluation of computation complexity is omitted. Here the other three parameters are evaluated. They are defined as follows:

success ratio = number of new receivers accepted / total number of joining requests

bandwidth used per member = total bandwidth used by successful receivers / number of new successful receivers

avg. msg. overhead = total number of join-request messages sent / total number of joining members

Success ratio quantifies how well a multicast tree is constructed with respect to QoS required by receivers. Bandwidth quantifies the resources, that is, the cost of a multicast tree. Average message overhead quantifies how many messages have been sent out when a new member emerges. Sending a message over a link is counted as one message. Hence, for a message traversing a path of  $l$  hops,  $l$  messages are counted. In vBNS topology only success ratios are measured. The vBNS topology is so small that it is meaningless to measure the bandwidth metric.

#### 4.3 Simulation results

In each run we vary one or more of the following parameters: group members, groups number, and the delay requirement of each receiver. For each given set of parameters, 100 experiments are performed, and the final results are obtained by averaging the results from them.

Figures 6 and 7 give the success ratios of the QDMR-LD, YAM-LD, QMRP-LD and SPT-LD schemes under different cases in the vBNS topology and Mbone topology. In Fig.6(a), the number of receivers is set to 10 and the delay bound required by each receiver is set to 0.2s. As the number of groups increases, the traffic load increases. As a result, the success ratio of each algorithm decreases. But the percentage of receivers that fail to attain their requested QoS is significantly lower under QDMR-LD than those under the other three algorithms. Figure 7(a) shows the same result as Fig.6(a). Figure 6(b) depicts the success ratios of the four algorithms under the case that the groups number and group members are set to 250 and 10 respectively as the delay bound increases from 0 to 0.3s. We can see that the success ratios increase under all the four algorithms as the delay bound is relaxed, and the success ratio of QDMR-LD outperforms those of the other three algorithms. The result is consistent with that in Ref.[15]. Figure 7(b) depicts the success ratios of the four algorithms under the case that the delay bound and

groups number are set to 0.3s and 10 respectively as the number of group members increases from 10 to 50. As the group members increase, the success ratios of the four algorithms decrease. But the success ratio of QDMR-LD also surpasses those of the other three algorithms.

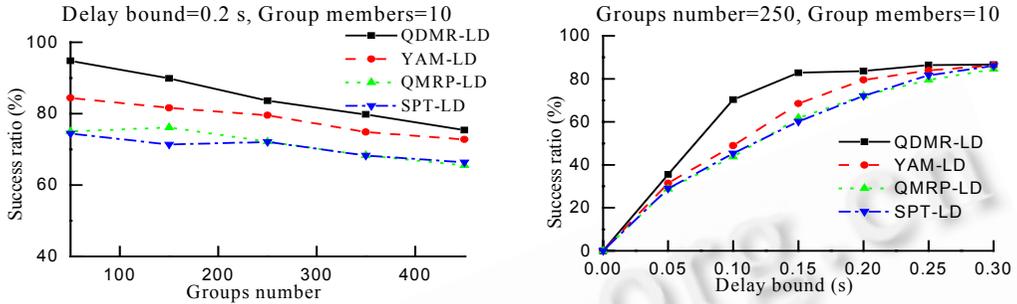


Fig.6 The success ratio results of the four schemes under different given parameters in the vBNS topology

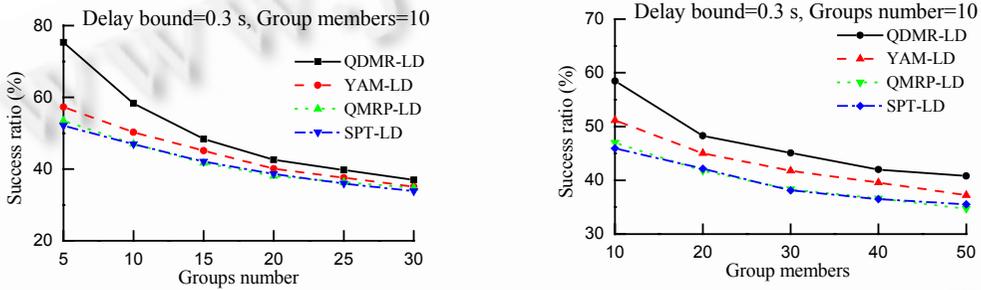


Fig.7 The success ratios of the four schemes under different given parameters in the Mbone topology

Figure 8 shows the average bandwidth used per member under the four algorithms in the Mbone topology. As the group members increase, the average bandwidth used per member decreases. At the same time, we can see that the average bandwidth used per member under YAM-LD and QDMR-LD is smaller than that of SPT-LD and QMRP-LD, which means that the multicast tree constructed under YAM-LD and QDMR-LD is more efficient as compared to that under SPT-LD and QMRP-LD. Because both YAM-LD and QDMR-LD are heuristic algorithms, a new receiver selects the feasible minimum cost path between the currently existing tree and the receiver. Therefore, the bandwidth used per member under YAM-LD and QDMR-LD is lower than that under QMRP-LD and SPT-LD. However, compared with YAM-LD, QDMR-LD employs RBMF<sup>[15]</sup> rather than RPF to select a feasible branch. Thus, the bandwidth used per member under QDMR-LD is prone to be a little higher than that under YAM-LD. For example, at the point that the number of group members is set to 30, the

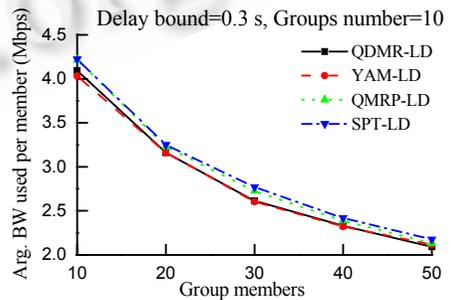


Fig.8 The bandwidth used per member under four schemes in the Mbone topology

amounts of average bandwidth used per member under QDMR-LD, YAM-LD, QMRP-LD and SPT-LD are 2.61381 Mbps, 2.60378 Mbps, 2.72537 Mbps and 2.76993 Mbps respectively.

The control message overhead of QDMR is analyzed theoretically in Ref.[15]. Figures 9(a) and (b) present the simulation results of the average message overhead under the four algorithms in the vBNS and Mbone topologies respectively. Similar results are observed in Ref.[15]. The message overheads of QDMR-LD and YAM-LD increase faster than those of QMRP-LD and SPT-LD as the search hops increase. The relatively high message overhead of QDMR-LD is a reasonable cost justified by the high success ratio and the low multicast tree cost. When applying QDMR-LD in a small-scale network environment, the TTL of the search can be limited to a small number according to its topology and diameter, hence, the subsequent overhead of control messages is restrained in a considerably acceptable range. For example, in Fig.9(a), when the TTL of the search is set to 6 in the vBNS topology, the average message count is 9.99, and the average message overhead is 399.6 bytes if the length of a search message is taken as 40 bytes. This overhead is rather negligible when compared to the link capacities, though it costs a little more than other schemes, in the vBNS topology.

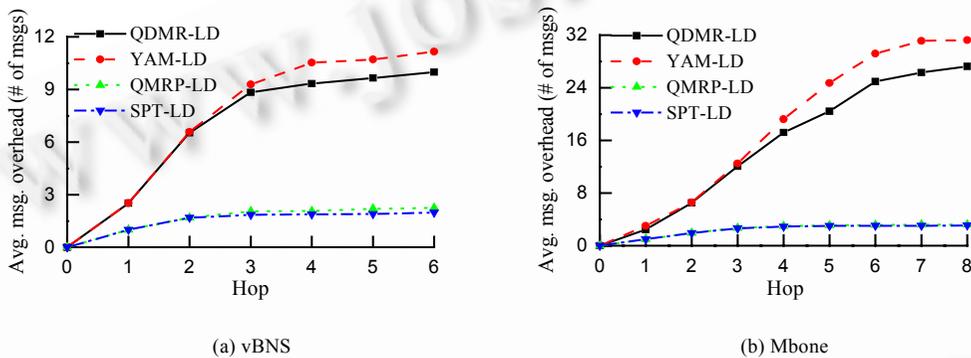


Fig. 9 The average message overhead under four schemes in the vBNS and Mbone topologies

## 5 Conclusions

This paper proposes a QoS-based dynamic multicast routing algorithm for streaming layered data, which can support QoS requirements in terms of bandwidth and delay bound imposed by heterogeneous and dynamic joining receivers. Each node in the network maintains only local state. The construction of a multicast tree is initiated by the receivers. To find a feasible path to the multicast tree, QDMR-LD adopts RBMF<sup>[15]</sup> mode in which satisfying a new receiver's QoS requirements is firstly considered. Therefore, QDMR-LD can increase the joining success ratio. At the same time, it builds a resource-efficient multicast tree by means of (1) adding a new receiver to the multicast tree via a feasible path with the minimum cost between the tree and the receiver; (2) pruning the needless part of the multicast tree when a receiver leaves. Simulation results show that QDMR-LD increases the success ratio and lowers the multicast tree cost compared with the other three schemes.

The message overhead makes QDMR-LD not scalable well for large networks. In order to achieve scalability, we are intending to introduce hierarchical structure to describe the network topology in our future research.

## References:

- [1] Li B, Liu J. Multirate video multicast over the Internet: An overview. *IEEE Network*, 2003,17(1):2~6.

- [2] Kwon G, Byers JW. Smooth multirate multicast congestion control. In: Matta I, ed. Proc. of the IEEE INFOCOM. San Francisco: IEEE Communications Society, 2003. 1022~1032.
- [3] McCanne S, Jacobson V, Vetterli M. Receiver-Driven layered multicast. In: Deering S, ed. Proc. of the ACM SIGCOMM. Stanford: ACM Press, 1996. 117~130.
- [4] Byers JW, Horn G, Luby M, Mitzenmacher M, Shaver W. FLID-DL: Congestion control for layered multicast. IEEE JSAC Special Issue on Network Support for Multicast Communication, 2002,20(8):1558~1570.
- [5] Byers JW, Luby M, Mitzenmacher M. Fine-Grained layered multicast. In: Ammar M, ed. Proc. of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1143~1151.
- [6] Wen WS, Mukherjee B, Chan SH, Ghosal D. LVMSR: An efficient algorithm to multicast layered video. Elsevier Computer Networks, 2003,41(4):363~383.
- [7] Maxemchuk NF. Video distribution on multicast networks. IEEE Journal on Selected Areas in Communications, 1997, 15(3):357~372.
- [8] Shacham N. Multipoint communication by hierarchically encoded data. In: Bovopoulos AD, ed. Proc. of the IEEE INFOCOM. Florence: IEEE Communications Society, 1992. 2107~2114.
- [9] Lin HC, Liu MY. Dynamic multicast routing for distributing hierarchically encoded video data. In: Bradlow H, ed. Proc. of the IEEE GLOBECOM. Sydney: IEEE Communications Society, 1998. 3035~3040.
- [10] Wang B, Hou JC. QoS-Based multicast routing for distributing layered video to heterogeneous receivers in rate-based networks. In: Katzela I, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 480~489.
- [11] Nakauchi K, Morikawa H, Aoyama T. A network-supported approach to layered multicast. In: Miltra U, ed. Proc. of the IEEE ICC. Helsinki: IEEE Communications Society, 2001. 1227~1231.
- [12] Nguyen TH, Nakauchi K, Kawada M, Morikawa H, Aoyama T. Rendezvous points based layered multicast. IEICE Trans. on Communications, 2001,E84-B(12):3133~3140.
- [13] Fei A, Mario G. Receiver-Initiated multicasting with multiple QoS constraints. In: Katzela I, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 62~70.
- [14] Zappala D, Zhou D. Performance evaluation of path searching heuristics for multicast QoS routing. In: Luijten R, ed. Proc. of the IEEE Int'l Conf. on Computer Communications and Networks. Miami: IEEE Communications Society, 2002. 248~254.
- [15] Lu HM, Xiang Y, Shi ML, Yang M. A new bandwidth and delay-constrained distributed multicast routing. Acta Electronica Sinica, 2002,30(12A):1978~1981 (in Chinese with English abstract).
- [16] Carlberg K, Crowcroft J. Building shared trees using a one-to-many joining mechanism. ACM Computer Communication Review, 1997,27(1):5~11.
- [17] Faloutsos M, Banerjee A, Pankaj R. QoSMIC: Quality of service sensitive multicast Internet protocol. In: Oran D, ed. Proc. of the ACM SIGCOMM. Vancouver: ACM Press, 1998. 144~153.
- [18] Yan S, Faloutsos M, Banerjee A. QoS-Aware multicast routing for the Internet: The design and evaluation of QoSMIC. IEEE/ACM Trans. on Networking, 2002,10(1):54~66.
- [19] Chen S, Nahrstedt K, Shavitt Y. A QoS-aware multicast routing protocol. In: Katzela I, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 1594~1603.
- [20] Chen S, Nahrstedt K, Shavitt Y. A QoS-aware multicast routing protocol. IEEE Journal on Selected Areas in Communications, 2000,18(12):2580~2592.
- [21] Fenner B, Handley M, Holbrook H, Kouvelas I. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification (Revised). INTERNET DRAFT, Work in Progress, 2003. <http://www.ietf.org/internet-drafts/draft-ietf-pim-sm-v2-new-07.ps>
- [22] Network Simulator ns-2. <http://www.isi.edu/nsnam/>
- [23] Casner S. Major MBONE routers and links. 1994. <ftp.isi.edu/mbone/mbone-topology.ps>

#### 附中文参考文献:

- [15] 陆慧梅,向勇,史美林,杨敏.一种基于带宽和时延约束的分布式组播路由算法.电子学报,2002,30(12A):1978~1981.