A Preprocessing Algorithm to Implement Greedy-MRS for Selecting a Hierarchical Reliable Multicast Proxy

R. Sasikala and Dr.K.Duraiswamy

IT Department, K.S. Rangasamy College of Technology, Tiruchengode – 637 215, Namakkal(Dt.), Tamil Nadu, India
sasikarthi@gmail.com

Abstract

Many hierarchical reliable multicast (HRM) protocols deploy repair proxies that perform local recovery and feedback consolidation. Placement of proxies is a key design issue in HRM network. The objective of Greedy-MRS algorithm is to place the HRM proxy in a single iteration, but it requires communication with routers frequently to find the efficient proxy. The objective of preprocessing step is to communicate with routers to implement Greedy–MRS algorithm with minimum number of communications between routers. Space needed in the router to store data to find efficient placement of proxies for HRM network is constant and it is independent of number of children. Simulation results indicate that this preprocessing step providing good levels of space and communication complexity and scalability of this algorithm depends on height of subtree.

Keywords: *Proxy, Greedy-MRS, Transmission Reduction, Space Complexity, Time Complexity*

1. Introduction

As computer systems have been widely used, *Internet*, which is a network-of-network, has been greatly developed and rapidly spread all over the world. In addition to unicast transmissions of point-to-point, multicast transmissions of one-to-many and many-to-many have been recently used [1], [2]. A ubiquitous and efficient multicast data delivery service is essential to the success of large-scale group communication applications. Various improvements for their performances from the view points of both hardware and software have been suggested and practiced. In particular, to improve reliability and scalability of multicast, *reliable multicast* was proposed with a number of recovery techniques. Most multicast transport protocols emphasize scalability as a fundamental design goal.

For instance, protocols that need to scale should use receiver-based reliability [3] through negative acknowledgement (NAK) or acknowledgement (ACK) and employ some form of feedback control to avoid feedback implosion [4].

Furthermore, protocols targeted at very large groups usually do not assume knowledge of group membership. Sources simply send to a multicast group address and receivers will join to that group to receive data. Scalability also requires protocol functionality to be distributed to avoid bottlenecks (and, of course, single points of failure).

An efficient approach to address scalability problems is the tree-based hierarchical proxy approach, dividing a multicast distribution tree (multicast group) into several subtrees (subgroups) to form a hierarchy rooted at the source. A representative node, called *proxy or designated local replier*, is designed to detect packet loss and arrange retransmission inside each subgroup. Thus the incidence of *implosion* and *exposure* are suppressed significantly. Meanwhile, bandwidth is saved, enabling large-scale reliable multicast. Moreover, recovery latency is reduced because retransmission comes from a proxy located much more close than the source. Many existing reliable multicast protocol follow this approach [3,10-15,18], e.g. SRM, RMP, LBRM, RMTP, TMTP, PGM and LMS.

Greedy-MRS[5] is one such approach, to design an efficient algorithm for the calculation of an approximate solution to the placement of proxies[17,19,20] for optimal performance in HRM, but this algorithm is not discussing about communication between routers to implement it. This paper designs an approach to coordinate with routers to implement Greedy-MRS algorithm.

2. HRM model

This paper assumes the HRM model [5,16] of the following characteristics.

(1) Consider a single-sender multicast tree where the root is the unique sender, all nodes are receivers and some of them can be act as a proxy node.

(2) The multicast distribution tree (multicast group) is divided into several subtrees (subgroups) to form a hierarchy rooted at the source.

(3) The root of each subtree acts as the proxy inside the subgroup, but as an ordinary member inside the upstream subgroup. The sender itself is a proxy by default.

(4) The topology of the tree is static (IP multicast tree), and loss probabilities at the links and number of children of all intermediate nodes are given.

(5) Proxy multicasts the original data to its own subgroup. Each receiver sends feedback (NAK) to its own proxy when a packet loss is detected, and the proxy retransmits the lost packet to the whole subgroup. We assume that all feedback packets are delivered via an out-of-band channel, so all feedback packets are delivered safely to repair servers. This assumption is also used in [5].

2.1 Greedy-MRS algorithm

The algorithm [5] tries to minimize the number of total link transmissions by maximizing the reduction caused by the placement of proxies. An optimal group partition algorithm is to partition the subtree rooted at a potential node to incur the maximum reduction. The node that can achieve the maximum reduction is placed as proxy in a single iteration to achieve local optimal solution, thus, a global suboptimal solution is achieved finally. The calculation of transmission reduction and selection of new proxy which incurs maximum reduction is the significant step of this algorithm. The transmission reduction in total link transmissions caused by placing node v as the proxy of the entire subtree T_V as follows:

$$\mathrm{Re}\,duction(v) = \underbrace{\sum_{i \in SG(V)} (E[M(AP_V, v) - E'[M(v)]) * p(v,i) * ch_i(v)}_{reduction\ inside\ the\ sub\,group\,SG(v)}$$

$$+ \underbrace{\sum_{j \in (SG(AP_v) - SG(v))U(v)} (E[M(AP_v)] - E'[M(AP_v)]) * p(AP_v, j) * ch_j(AP_v)}_{reduction\ outside\ the\ sub\,group\,SG(v)} \quad (1)$$

where *E[M(i)]* and *E'[M(i)]* are the values of the *E[M]* measure seen on node *i* before and after the placement of new proxy, respectively. The calculation of *E[M]* from bottom-up way was proposed in [4] and *Reduction(v)* in [5] elaborately.

Greedy MRS algorithm can be introduced into network routers (or layer-3 switches) that support HRM protocols. Once proxies are needed, these routers would cooperate with each other to select the appropriate ones to be proxies. However, acting as proxies and supporting *Greedy-MRS* require more storage and processing power, and therefore more expensive to operate than normal routers. The more routers support Greedy-*MRS* in HRM networks, the better solution *Greedy-MRS* provides. The solution can perform best if all routers support *Greedy-MRS.* The technique that how these routers coordinate with each other to implement this algorithm is not discussed elaborately in Liansheng Tan et.al.

2.2 Preprocessing algorithm description

The approach, the proposed preprocessing step of Greedy-MRS algorithm requires minimum number of communication between routers and size of buffer space required in routers is independent of number of children of the node. It does the following operation to simplify the router work to implement Greedy-MRS efficiently.

- Greedy-MRS algorithm calculates transmission reduction to select an efficient proxy. To calculate transmission reduction of each intermediate node *v (reduction(v)),* Eqn. 1 requires the following information :

Probability of node *v* successfully receives a packet from assigned proxy node AP_v *p(APv, v).*

Expected number of transmission for each intermediate node *v* *E[M(v)]*

Expected number of transmission for assigned proxy of node *v* (AP_v) $E[M(AP_v)]$

Expected number of transmissions for node *v* and AP_v, after placement of proxies $E'[M(v)]$, $E'[M(AP_v)]$.

Partial multiplications $p(v, i)*ch_i(v)$ and $p(AP_v,j)*ch_j(AP_v)$

Degree of congestion on node *i* $E[M(i)]/(1-p_i)$

- Initially, a Source Path Message (SPM)[6] starts at the source called as setup packet helps to store parent node address and calculate probability of node receives a packet from proxy node *p(APv, v).*

- After receiving a SPM packet, children nodes prepare refresh packet to calculate expected number of transmissions from bottom up level such as $E[M(AP_v)]$, $E'[M(AP_v)]$, $E[M(v)]$, $E'[M(v)]$, $p(AP_v,v)$ and partial multiplications to compute transmission reduction.

- Intermediate routers calculate transmission reduction value with assistance of SPM and refresh packets and send to proxy node to select a new proxy which has highest transmission reduction value.

2.2.1 Source path message (SPM): Source periodically emits SPM packets downstream side to establish a source path state. Each SPM contains the address of the parent node that it came from; Routers replace this address with their own when they forward an SPM, so that their children will know their parent. Routers support Greedy-MRS, use this information to determine the unicast path back to the source for forwarding E[M] and partial multiplication results.

In addition to parent node address, each node calculates the Eqn.(2) recursively and forward with SPM packet downwards. SPM called as setup packet calculates the probability of each node v receives a packet correctly from an upstream node u and forward towards downstream. The probability that node v receives a packet correctly from an upstream node u is:

$$p(u,v) = \begin{cases} 1, & \text{if } v = u; \\ \prod_{i \in (u,v)} (1 - p_i), & \text{if } v \neq u; \end{cases} \tag{2}$$

where (u, v) is the route between node u and node v, i is an intermediate node on (u, v), with link loss probability p_i.

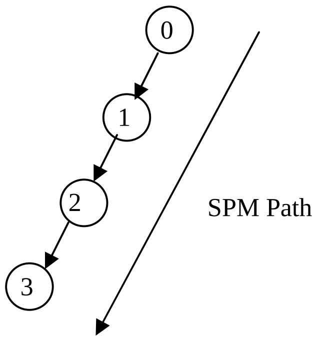

Figure 1. SPM path to find parent node address and p(u,v)

For example, In Figure 1 when node 1 receives a setup packet from node 0, it assigns parent node address as 0 and p(0,1) in table and forwards updated information to node 2 as shown in Table 1.

2.2.2 Refresh packet : A unicast packet send towards upstream direction to calculate expected number of transmissions from bottom up level such as E[M(APv)], E'[M(APv)], E[M(v)], E'[M(v)], p(APv,v) and partial multiplications to compute transmission reduction.

Table 1. Information maintained by the node after receiving a SPM path

Node i	Parent address	$\prod$ (1-p_i)
0	-	1
1	0	1-p1
2	1	(1-p1)*(1-p2)
3	2	(1-p1)(1-p2)(1-p3)

2.2.2.1 Expected number of transmissions (E [M]) : A commonly used measure in the performance analysis of reliable multicast is *E[M]* that first appeared in [7]. It is defined as the expected number of transmissions per packet from a node required for reliable delivery to all receivers below. We calculate the *E[M]* [4] measure by using an approach to find the equivalent link of an arbitrary topology composed of several links in a bottom up way. To calculate *Reduction (v)* for all node, *E[M]* should be calculated at leaves and its value forward towards the proxy. p_i is the loss probability on the link leading to node i,

Each node stores *E[M]* value and same is transferred to parent nodes to calculate *E[M]*. For example, consider the Figure 2 and Figure 3, E[M(1)] will be stored in the intermediate node and transferred to node 0, to calculate E[M(0)]. This process is repeated till packet reaches the proxy node.

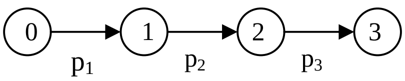

Figure 2. Calculation of *E[M]* for three links in a row

$$E[M(2)] = \frac{1}{(1-p_3)}$$
$$E[M(1)] = \frac{1}{(1-p_2)(1-p_3)} = E[M(2)]\frac{1}{1-p_2}$$
$$E[M(0)] = \frac{1}{(1-p_1)(1-p_2)(1-p_3)} = E[M(1)]\frac{1}{1-p_1}$$
(3),(4),(5)

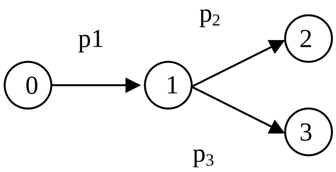

Figure 3. Calculation of *E[M]* for tree type network

$$E[M(1)] = \frac{1}{1-p_{2,3}} = \frac{1}{1-p_2} + \frac{1}{1-p_3} - \frac{1}{1-p_2 p_3}$$

$$E[M(0)] = \frac{1}{1-p_{1,2,3}} = \frac{1}{(1-p_1)(1-p_{2,3})} = E[M(1)]\frac{1}{1-p_1}$$

(6), (7)

2.2.2.2 Partial multiplication: To calculate the transmission reduction in total link transmissions caused by placing node *v* as the proxy of the entire subtree T_v, Greedy-MRS requires Eqn. 1.

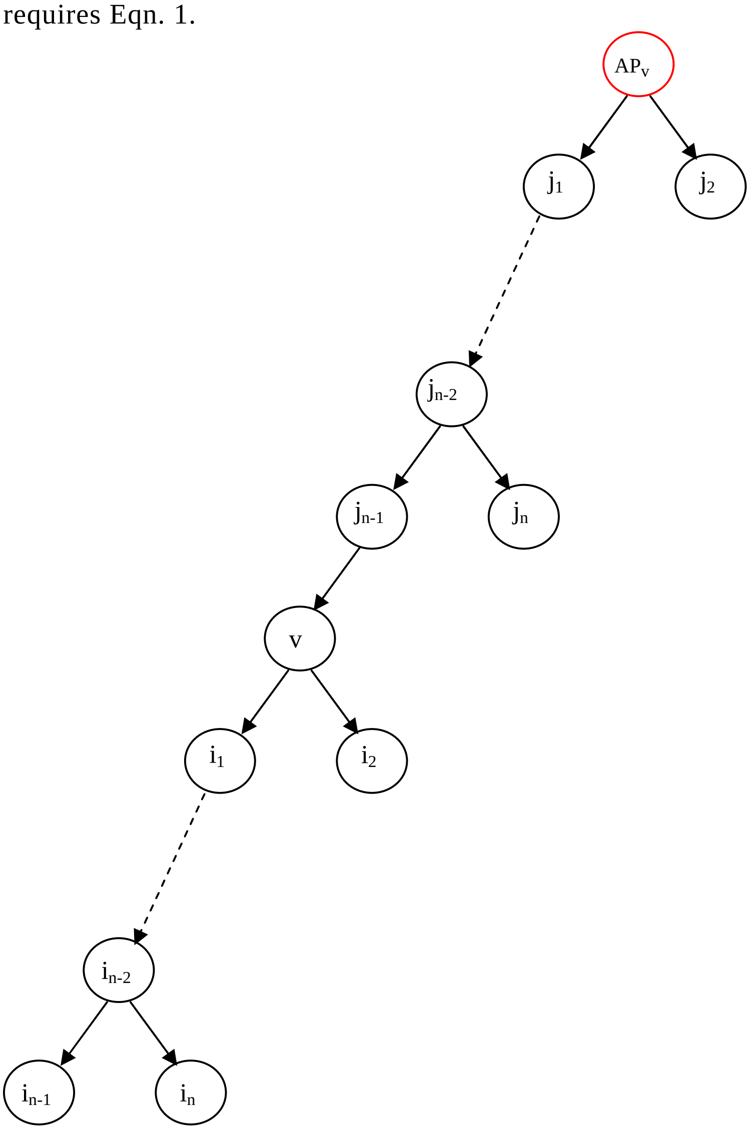

Figure 4. A HRM model with group (AP_v) and subgroup v.

It can be expanded as follows:

$$\begin{aligned}
\mathrm{Re}\,duction(v) = {} & (E[M(AP_v)*p(AP_v,v) - E^{'}[M(v)])*p(v,v)*ch_v(v) \\
& \underbrace{\begin{aligned}
& +(E[M(AP_v)*p(AP_v,v) - E^{'}[M(v)])*p(v,i_1)*ch_{i_1}(v) \\
& +(E[M(AP_v)*p(AP_v,v) - E^{'}[M(v)])*p(v,i_2)*ch_{i_2}(v) \\
& +E(M(AP_v)*p(AP_v,v) - E^{'}[M(v)])*p(v,i_3)*ch_{i_3}(v) \\
& \ldots \\
& +E[M(AP_v)*p(AP_v,v) - E^{'}[M(v)])*p(v,i_n)*ch_{i_n}(v)
\end{aligned}}_{\mathrm{Re}\,duction\ inside\ the\ subgroup\ SG(v)} \\
& \underbrace{\begin{aligned}
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,AP_v)*ch_{AP_v}(AP_v) \\
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,v)*ch_v(AP_v) \\
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,j_1)*ch_{j_1}(AP_v) \\
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,j_2)*ch_{j_2}(AP_v) \\
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,j_3)*ch_{j_3}(AP_v) \\
& \ldots \\
& +(E(M(AP_v) - E^{'}[M(AP_v)])*p(AP_v,j_n)*ch_{j_n}(AP_v)
\end{aligned}}_{\mathrm{Re}\,duction\ outside\ the\ subgroup\ SG(v)}
\end{aligned} \tag{8}$$

Where $v, i_1, i_2, i_3 \ldots i_n$ are members of $SG(v)$, AP_v, v, $j_1, j_2, j_3 \ldots j_n$ are members of $SG(AP_v)$ and $ch_v(AP_v)$ is the number of the children of node v, whose assigned proxy is node AP_v as shown in Figure 4. Eqn. 8 can be written as

$$\begin{aligned}
\mathrm{Re}\,duction(v) = {} & (E[M(AP_v)*p(AP_v,v) - E^{'}[M(v)]) \\
& \underbrace{\begin{aligned}
& *(\mathrm{p(v,v)*ch_v(v) + p(v,i_1)*ch_{i1}(v) + p(v,i_2)*ch_{i2}(v) + p(v,i_3)*ch_{i3}(v)} \ldots \\
& + \mathrm{p(v,i_n)*ch_{in}(v)})
\end{aligned}}_{reduction\ inside\ the\ subgroup\ SG(v)} \\
& \underbrace{\begin{aligned}
& +(E[M(AP_v)] - E^{'}[M(AP_v)])*(p(AP_v,AP_v)*ch_{APv}(AP_v) + p(AP_v,v)*ch_v(AP_v) \\
& \mathrm{p(AP_v,j_1)*ch_{j1}(AP_v) + (AP_v,j_2)*ch_{j2}(AP_v) + p(AP_v,j_3)*ch_{j3}(AP_v)} \ldots \\
& + \mathrm{p(AP_v,j_n)*ch_{jn}(AP_v)}
\end{aligned}}_{reduction\ outside\ the\ subgroup\ SG(v)}
\end{aligned} \tag{9}$$

In general $p(v,i)*ch_i(v)$ is denoted as $pmult(i)$ and it is termed as partial multiplication. The partial multiplication for **reduction inside the group** started at leaves and updated in parent nodes. It can be written as

$$
\begin{aligned}
p(v,i_n)*ch_{in}(v) &= pmult(i_n) \\
&\dots \\
p(v,i_3)*chi_3(v) &= pmult(i_3) \\
p(v,i_2)*ch_{i2}(v) &= pmult(i_2) \\
p(v,i_1)*ch_{i1}(v) &= pmult(i_1) \\
p(v,v)*ch_v(v) &= pmult(v) = ch_v(v)
\end{aligned}
$$

where p(v,v)=1

For example, partial multiplication stored in the node i_{n-2} to calculate *reduction*(i_{n-2}) is *pmult*(i_n)+ *pmult*(i_{n-1})+ *pmult*(i_{n-2}) and partial multiplication to carry over the result to parent node is $(pmult(i_n)+ pmult(i_{n-1}))* p(v, i_{n-2})+(p(v,i_{n-2})*ch_{n-2}(v))$. This recursive procedure is repeated till packet reaches the proxy node. Partial multiplication for **reduction outside the group** started at leaves and updated in parent nodes. It can be written as

$$
\begin{aligned}
&p(AP_v, j_n)*ch_{jn}(AP_v) = pmult(j_n) \\
&\dots \\
&p(AP_v, j_3)*ch_{j3}(AP_v) = pmult(j_3) \\
&p(AP_v, j_2)*ch_{j2}(AP_v) = pmult(j_2) \\
&p(AP_v, j_1)*ch_{j1}(AP_v) = pmult(j_1) \\
&p(AP_v, v)*ch_v(AP_v) = pmult(j_v) \\
&p(AP_v, AP_v)*ch_{AP_V}(AP_v) = pmult(AP_v) = ch_{AP_V}(AP_v)
\end{aligned}
$$

where $p(AP_v,AP_v)=1$

For example, we obtain a recursive form to calculate partial multiplication to carry over the result to parent node j_{n-2} as follows:

$$pmult(j_n) + pmult(j_{n-1}) + pmult(j_{n-2})$$

This recursive procedure is repeated till packet reaches the proxy node.

3. Modified greedy-MRS algorithm

The modified Greedy-MRS algorithm coordinates with routers to calculate transmission reduction with the help of setup and refresh procedure. A setup function uses multicast message(SPM), which starts at source node to find parent node address and $p(AP_v,i)$. A refresh function uses refresh packet starts at leaf nodes to calculate *E[M]*,*E'[M]*, d_i and *pmult(v,i)* at each intermediate node. The refresh packet update above values in each intermediate node and carries this from leaf node to proxy node. Intermediate node stores above values in a table and it is independent of number of children.

Algorithm Greedy-MRS (k, N)
Inputs
k: the number of proxies
N: the set of nodes in the multicast tree T_s
Output
P: the set of proxies

Assumtions
$N=\{s,j_1,j_2,\ldots j_n,v,i_1,i_2,\ldots i_n\}$ — *Set of nodes in the multicast group.*
Initially $AP_v=s$ — *AP – Assigned Proxy. Initially source is the Assigned proxy node for the whole group.*

begin
1. $P=\{s\}$,n=0
2. setup(s)
3. for all $l \in N$ — *where l is a leaf node*
4. refresh (l)
5. for all $u \in N$ $\{AP_u=s\}$
6. while n<k do
7. for all $u \in N$ calculate reduction(u) with the maximum reduction algorithm
8. Find $u \in N$ such that reduction(u) is the greatest
9. for all $v \in B_u$
10. for all $I \in T_v$
11. {if APi is not a descendant of v then APi=v}
12 P=P+{u}, n=n+1
13. end while;
end

Function setup(s)
Assumptions :
Each setup packet has following information
Parent_addr
Pr_loss — *pr_loss = p(s,i)*

1. *Source node s multicast setup packet to all the members in the group*
   Parent_addr=s
   Pr_loss=P(0,0)=1
2. *A node i receiving a setup packet, update the following detail in its table*
   $Parent\_addr_i$ = Parent_addr
   Pr_loss=Pr_loss(1-p(s,i))*
3. *Each node update the following detail in Packet*
   Parent_addr=$addr_i$
   Pr_loss=Pr_loss
4. *Step 2 & 3 will be repeated till setup packet reaches the leaf node.*

Function refesh(l)
Assumtions
Each refresh packet starts from leaf node has the following information
E[M] — *Expected no. of transmissions*

pmult(i) — *Partial multiplication to calculate the reduction inside the subgroup*

pmult(j) — *Partial multiplication to calculate the reduction outside the subgroup*

1. *Leaf node unicast the following information to its parent node*

E[M]
pmult(i)
pmult(j)

2. *An intermediate node i receiving a refresh packet update the following detail in the table and forward to upstream.*

a. *An intermediate node of case-1(as shown in Fig. 2)calculates E[M] as follows:*

$$E[M] = \frac{1}{1-p_1} * \frac{1}{1-p_2} \ldots$$

b. *An intermediate node of case-2(as shown in Fig. 3)calculates E[M] as follows:*

$$E[M] = \frac{1}{1-p_1} + \frac{1}{1-p_2} - \frac{1}{1-p_{1,2}} \ldots$$

c. *Partial multiplication to calculate reduction inside the subgroup is*

$$pmult(i)=ch_i(v)*p(v,i)$$
$$pmult(j)=ch_j(AP_v)*p(AP_v,j)$$

3. *Step 2 will be repeated till refresh packet reaches the source node.*

Greedy-MRS function executes setup and refreshes functions repeatedly, until ***k*** numbers of proxies are placed.

4. Computation procedure on example topology

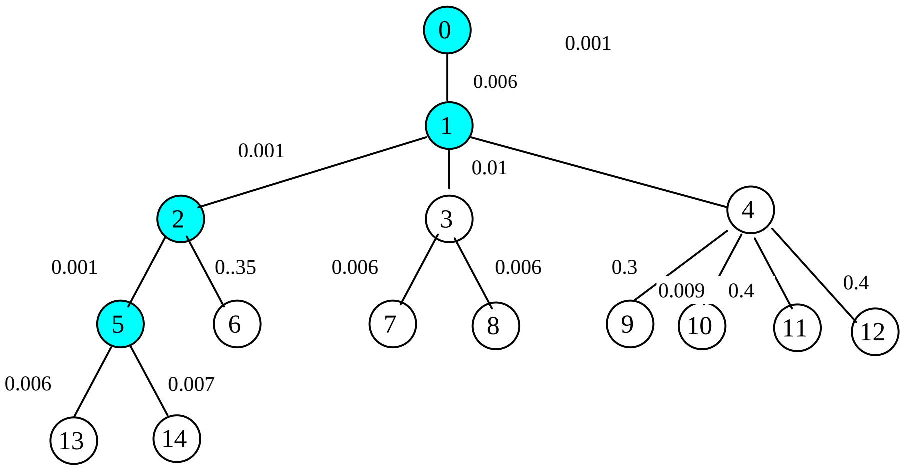

Figure 5. An example topology

Now we describe the computation procedure of reduction (5) on a simple example topology in Figure 5. The tree topology consists of 15 nodes, node 0 is the sender, colored nodes are proxies. Proxies are selected based on transmission reduction value. Initially node 0 executes Greedy-MRS to select a new proxy. Each intermediate node calculates transmission reduction (reduction (i)) value to select a new proxy. For example reduction (5) can be expressed as

$$reduction(5) = \left\{ \begin{array}{l} (E[M(0)] * p(0,5) - E'[M(5)]\ ) * p(5,5)*ch_5(5)\ + \\ (E[M(0)] * p(0,5) - E'[M(5)]\ ) * p(5,13)*ch13(5) + \\ (E[M(0)] * p(0,5) - E'[M(5)]\ ) * p(5,14)*ch14(5) + \end{array} \right\} \text{reduction inside the SG(5}$$

$$\left\{ \begin{array}{l} (E[M(0)]-E'[M(0))*p(0,0)*ch_0(0) + \\ (E[M(0)]-E'[M(0))*p(0,1)*ch_1(0) + \\ (E[M(0)]-E'[M(0))*p(0,2)*ch_2(0) + \\ (E[M(0)]-E'[M(0))*p(0,5)*ch_5(0) + \\ (E[M(0)]-E'[M(0))*p(0,6)*ch_6(0) + \\ (E[M(0)]-E'[M(0))*p(0,3)*ch_3(0) + \\ (E[M(0)]-E'[M(0))*p(0,7)*ch_7(0) + \\ (E[M(0)]-E'[M(0))*p(0,8)*ch_8(0) + \\ (E[M(0)]-E'[M(0))*p(0,4)*ch_4(0) + \\ (E[M(0)]-E'[M(0))*p(0,9)*ch_9(0) + \\ (E[M(0)]-E'[M(0))*p(0,10)*ch_{10}(0) + \\ (E[M(0)]-E'[M(0))*p(0,11)*ch_{11}(0) + \\ (E[M(0)]-E'[M(0))*p(0,12)*ch_{12}(0) + \\ (E[M(0)]-E'[M(0))*p(0,13)*ch_{13}(0) \end{array} \right\} \text{reduction outside the SG(5)}$$

A setup packet starts from source computes the following using Eqn. 2 recursively till it reaches leaf node.

$$p(0,0)=1$$
$$p(0,1)=p(0,0)*p(0,1)$$
$$p(0,2)=p(0,1)*p(1,2) \ldots.$$

Since refresh function follows bottom-up approach, the computations must be performed on the leaf nodes first. Routers inside the SG(5) calculate partial multiplication and $E[M]$ **.**

$$p(5,13)\ *\ ch_{13}(5)\ \ - pmult(13)$$
$$p(5,14)\ *\ ch_{14}(5)\ \ - pmult(14)$$
$$p(5,5)\ *\ ch_5(5)\ \ - pmult(5) = ch_5(5) \quad where\ p(5,5)=1$$

$$E'[M(5)] = 1/1\text{-}p_{13}+1/1\text{-}p_{14}\text{-}1/1\text{-}p_{13},p_{14}$$

Node 5 stores E'[M(5)] and partial multiplication to compute **reduction inside the subgroup SG(5)**.

$$= pmult(13)+pmult(14)+(1*ch_5(5))$$

Partial multiplication values to calculate **reduction outside the subgroup SG(5)** are as follows

$$p(0,9) * ch_9(0) = pmult(9)$$
$$p(0,10) * ch_{10}(0) = pmult(10)$$
$$p(0,11) * ch_{11}(0) = pmult(11)$$
$$p(0,12) * ch_{12}(0) = pmult(12)$$
$$p(0,7) * ch_7(0) = pmult(7)$$
$$p(0,8) * ch_8(0) = pmult(8)$$
$$p(0,5) * ch_5(0) = pmult(5)$$
$$p(0,6) * ch_6(0) = pmult(6)$$
$$p(0,2) * ch_2(0) = pmult(2)$$
$$p(0,3) * ch_3(0) = pmult(3)$$
$$p(0,4) * ch_4(0) = pmult(4)$$
$$p(0,1) * ch_1(0) = pmult(1)$$
$$p(0,0) * ch_0(0) = pmult(0) = ch_0(0) \qquad \text{where } p(0,0) = 1$$

Partial multiplication calculated at step 4 is stored in the node 0, and expected number of transmission *E[M(0)]* will be transferred to all the nodes downstream using multicast. Now Node 5 has partial multiplication for inside and outside the group, *E[M(5)], E[M(0)], E'[M(0)]*. It computes *reduction(5)* and send it to proxy. This is the way all intermediate nodes calculate *reduction(v)*. Proxy node compares all the reduction values and selects a new node as a proxy, which has highest reduction value.

Each Router, i have to maintain the following detail to implement Greedy-MRS algorithm. This is shown in Table 2 for router 5.

Table 2. Data stored in node 5 to calculate reduction(5)

Node i	p_i	No.of children	p(APi,i)	E[M(i)]	E[M(APi)]	pmult (inside the subgroup i)	pmult (outside the subgroup i)
5	p_5	2	p(0,5)	E[M(5)]	E[M(0)]	pmult(13)+pmult(14) +(1*ch_5(5))	pmult(9)+pmult(10) … +1*ch_0(0)

5. Simulation

In order to evaluate the behavior and performance of the existing and proposed scheme, we built a simulation environment using the NS-2.30[8] simulator from U.C. Berkeley/LNBL. NS is a discrete event simulator targeted at networking research. NS allows the user to define arbitrary network topologies, composed of routers, links and shared media. A rich set of built-in and contributed protocol agents are available for selection. The user may instantiate these agents and attaches certain protocols to nodes on the topology. The topologies for simulation are generated by the GT-ITM tool [9]. The simulation is viewed using Network Animator (NAM). NAM is a Tcl/Tk based animation tool for viewing network simulation traces and network link to be a bidirectional with a bandwidth of 15Mb and delay is 10ms. Height of the tree is 4 with probability of loss is fixed randomly. We fixed 4 numbers of proxies including sender node.

5.1. Space complexity

To find reduction (v), algorithm retrieves 7 number of information from the table and it receives 2 number of information from the proxy node. This preprocessing step updates the table for each new proxy selection as shown in Table 2. The number of information processed by proxy node to calculate transmission reduction is same in all nodes, which is independent of number of children of intermediate node. Therefore space complexity of reduction (v) is independent number of children.

5.2. Communication complexity

In these experiments, we use the mean time to measure the effectiveness of this preprocessing step. The mean time is calculated by averaging time spent between sending a setup packet and calculation of *reduction(v)*.

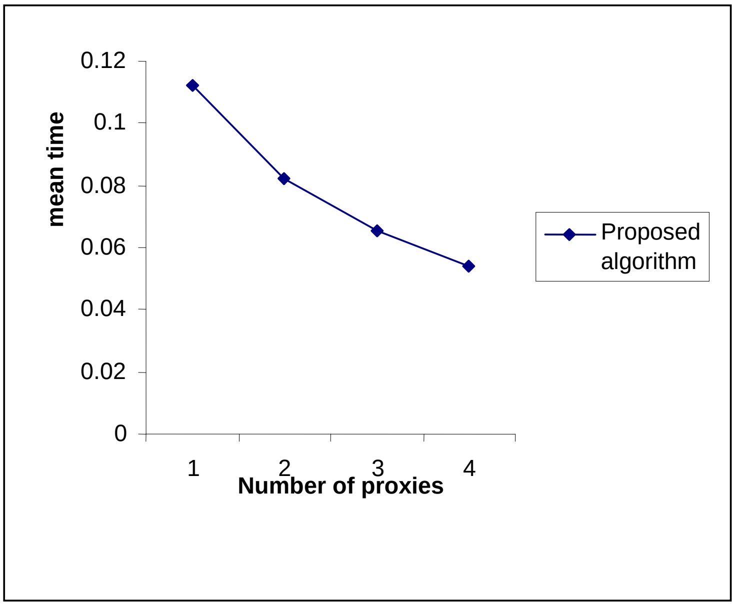

Figure 6. Mean time when number of proxies is varied.

$$Mean\ time = \frac{\sum_{children \in proxy} Time_{setup} - Time_{reduction}}{no.\ of\ children\ in\ the\ proxy}$$

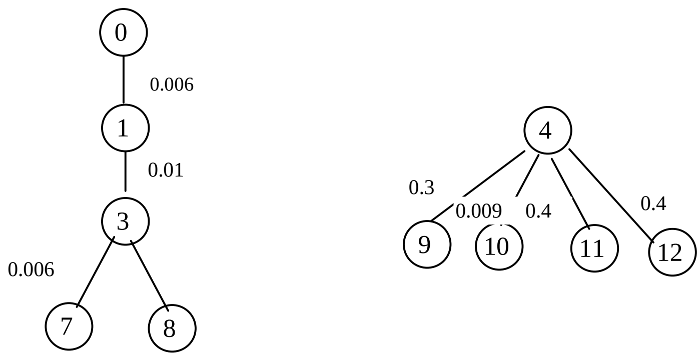

Figure 7. (a) Subtree T_0 (b) Subtree T_4

Table 3. Comparison of mean time for tree height 3 and 1.

Subtree T_v	Height of T_v	Node i	Mean time to calculate reduction(i)	Average
T_0	3	0	0.065216	0.0843296
		1	0.075835	
		3	0.086453	
		7	0.097072	
		8	0.097072	
T_4	1	4	0.021739	0.0302334
		9	0.032357	
		10	0.032357	
		11	0.032357	
		12	0.032357	

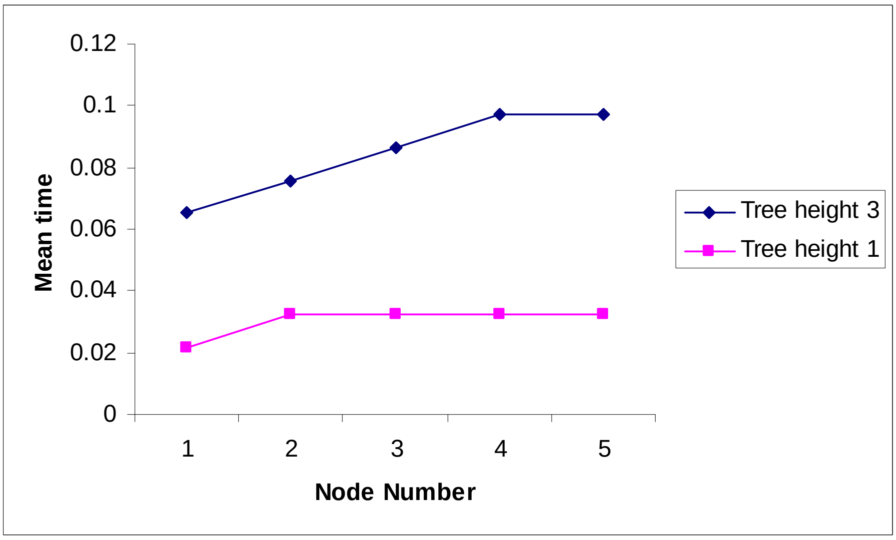

Figure 8. Comparison of average mean time for tree height 3 and 1.

Simulation results are shown in Fig. 6 for number of proxies varies from 1 to 4 with tree height 4. From these results we found that, if we increase the number of proxies, the mean time to calculate reduction (v) is decreased. From the shape of curve, we found that the difference between mean times is reduced when we have more number of proxies and it is practically difficult to choose the best value of k(number of proxies in the tree). Therefore fixing k value by the source is not the best solution as mentioned in the Greedy-MRS algorithm. Also workload of proxy is depends on the reliability level of subgroup not on the number of members in the subgroup. Therefore the new proxy selection process distributed to subgroup instead of source node is the alternative solution. This solution significantly reduces the number of communication between routers. The communication complexity of algorithm is depends on the height of sub tree not on the tree height. Finally, the communication complexity of our proposed algorithm is O (d) where d is the depth of subtree.

5.3. Scalability

Scalability is a critical issue for multicast applications. Greedy-MRS algorithm collecting information from children nodes to calculate transmission reduction (reduction(i)). For example, Figure 7a. and 7b. are subtrees T0, T4 of height 3 and 1 respectively. To select a new proxy, each node calculate reduction(i). Its experimental results are shown in Table 3 and Figure 8. From these results, we found that mean time to calculate reduction is depends on height of the subtree. The following results once again proves that scalability of this algorithm is mainly depends height of subtree Tv, where v is a proxy node.

6. Conclusion

The proposed preprocessing extension helps to implement Greedy-MRS algorithm with less number of communication between routers and maintains less state information at the routers. This algorithm maintains a table in each router of size 7 to implement Greedy-MRS. Initially, it requires a multicast communication to transfer a setup packet from sender to all its children and children have to send unicast packet to sender to update the table. After this initialization, each proxy has to send partial multiplication and E[M(i)] values to its children to find maximum reduction values. An existing proxy node selects a new proxy node which has maximum reduction value. For every new proxy selection, a proxy has to send a message to its children to compute a new reduction values.

This paper found a step to communicate between routers to implement Greedy-MRS algorithm. Numerical example and simulation results shows that, it requires same size of table in all nodes to store information to calculate reduction and it reduces average mean time while increasing number of proxies. The preprocessing step reduces the workload of router significantly.

References

[1] A. Tanenbaum, *Computer Networks*. Englewood Cliffs, NJ: PrenticeHall, 1988.

[2] D. Comer, *Internetworking with TCP/IP*. Englewood Cliffs, NJ: PrenticeHall, 1991.

[3] S. Pingali, D. Towsley, and J. Kurose, “A comparison of sender-initiated and receiver-initiated reliable multicast protocols,” in *J. Select. Areas Commun.*, vol. 15, Apr. 1997, pp. 398–406.

[4] A.M. Costello, S. McCanne, Search party: using randomcast for reliable multicast with local recovery, in: Proc. IEEE INFOCOM’99, vol. 3, New York, NY, USA,1999, pp. 1256–1264.

[5] Liansheng Tan, Li Jin and Yi Pan “Efficient Placement of Proxies for Hierarchical Reliable Multicast” Accepted Manuscript, Journal of Computer Communications 2007

[6] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano, “PGM Reliable Transport Protocol Specification,” IETF, RFC 3208, Dec. 2001.

[7] P. Bhagwat, P. Mishra, S. Tripathi, Effect of topology on performance of reliable multicast communication in IEEE INFOCOM’94, Toronto, Ontario, Canada, 1994, pp. 602–609.

[8] UCB/LBNL/VINT Network Simulator-ns(version 2.30) online http://www.mash-cs.berkeley.edu/ns

[9] GT-ITM, http://www.cc.gatech.edu/projects/gtitm/.

[10] D. Chiu, S. Hurst, M. Kadansky, and J.Wesley, "TRAM: A Tree-Based Reliable Multicast Protocol," Sun Microsystems, Santa Clara, CA, Sun Tech. Rep. SML TR-98-66, July 1998.

[11] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 784 – 803, Dec. 1997.

[12] C. Papadopoulous, G. Parulkar, and G. Verghese, "An Error Scheme for Large-Scale Multicast Applications," in *Proceedings of IEEE INFOCOM'98*, Mar. 1998.

[13] C.Papadopoulos, Guru Parulkar and George Varghese, "Light-Weight Multicast Services(LMS):A Router-Assisted Scheme for Reliable Multicast," in IEEE/CAM Transactions on Networking, Vol.12 No.3, June 2004. pp 456-468.

[14] J. Lin and S. Paul. RMTP: A Reliable Multicast Transport Protocol. In *Proceedings of the IEEE Infocom'96*, pages 1414–1424, San Francisco, USA, March 1996.

[15] H. Holbrook, S. Singhal, and D. Cheriton, "Log-based receiver-reliable multicast for distributed interactive simulation," in *Proc. ACMSIGCOMM*, vol. 25, Oct. 1995, pp. 328–341.

[16] S. Guha, A. Markopoulou, and F. Tobagi, "Hierarchical Reliable Multicast: Performance analysis and Optimal Placement of Proxies," *Computer Communications*, vol. 26, pp. 2070-2081, Apr. 2003.

[17] M. Charikar, S. Guha, E. Tardos, and D. Shmoys, "A constant-factor approximation algorithm for the *k*-median problem," *Proc. 31st Annual Symposium on the Theory of Computing*, Atlanta, Georgia, USA, pp. 1-10, 1999.

[18] S. K. Kasera, J. Kurose, and D. Towsley, "A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast," *Proc. IEEE INFOCOM '98*, San Francisco, CA, USA, vol. 3, pp. 988-995, 1998.

[19] L. Lao, J. H. Cui and M. Gerla, "A Scalable Overlay Multicast Architecture for Large-Scale Applications," *Technical Report*, Department of Computer Science, University of California, Jul. 2004. Available at: http://www.cs.ucla.edu/NRL/hpi/AggMC/papers

[20] R. Yavatkar, J. Griffioen, and M. Sudan, "Reliable Dissemination Protocol for Interactive Collaborative Applications," *Proc. of ACM MULTIMEDIA'95*, Boston, MA, USA, 1995, pp. 333–344.