PROCEDURE FOR MULTICAST TREE MAINTENANCE BASED ON MEMBER TIME DURATION

Kais Mnif and *Michel Kadoch*
Lagrit, Ecole de Technologie Supérieure
Montréal (Qc) Canada
H4A 3R5
Email: *kais.mnif.1@ens.etsmtl.ca*

Abstract
In this paper we propose a new maintenance procedure for dynamic multicasting. With our approach, we can guarantee that we can add/delete a node to the existing multicast tree without the need to rebuild all the tree and satisfying QoS constraints, also, our approach tries to find the best route for a new member according to the dynamic of members in group. Simulations results show that tree cost is very close to other algorithm who does not consider multicast group dynamic.

Introduction

Multicast is an efficient mechanism for delivering data from a source to a set of receivers. It has advantageous to minimize the consumption of bandwidth through the network. The multicast group (source, receivers) has access to a shared tree formed by routers. Source sends only one copy of data and routers duplicate data at the output interface to send them to receivers. In the last decade multicast routing has been extensively studied for static group. The destination group is said dynamic when a receiver may join or leave the existing session at any instant of time [1]. The problem for dynamic multicast under QoS constraint can be defined as how to find a route for a newly joining node, such that route can satisfy QoS requirements. In a dynamic system, an existing member can leave the session before his end. The procedure to add a new member to the existing group should be guaranteeing the three following features: firstly, the topology should be maintained stable for the whole system face to the group dynamic. Secondly, the tree cost should stay minimal without the need to reconstruct the tree from the beginning. Finally, the procedure should maintain the QoS requested and avoid its degradation. In the literature, some propositions have been discussed and can be classified into two categories: *naif* algorithms and *greedy* algorithms. With the *naif* technique, new number joins the multicast tree using the shortest path from the source; this technique is simple but doesn't guarantee the cost of the whole tree is minimal. *Greedy* technique tries to resolve this drawback by using the shortest path from any node in the existing multicast tree; nevertheless, this technique has a weakness, if the new member uses a route used by a member who will leave later. In fact, this technique doesn't take into consideration members who went to leave before the end of the session. This route will not be necessarily the shortest one after the departure of this member and we have to look for shortest route for the new member. Which cannot be useful in practice specially for real time application. As we can see

these propositions don't take into consideration of the period that a member will stay in the session. In the following section we will present a new procedure which guarantees the stability of the tree topology with respect to the dynamics of the group.

Maintenance procedure

In the following section we will present the maintenance procedure for a multicast group based on member time duration; the time that the member went to stay in the session. Our proposition takes into consideration of this parameter to find the best route for futures members who went to join the session. The maintenance procedure for a multicast group contains two parts; one part describes how a new member can join a group for a session in progress, and the second part describes how an existing member can leave the session.

Join procedure – the session will begin when the global time (t =0) with the source and receivers who will take part in the session from the beginning. The multicast tree is then constructed for the set of nodes (*source*, *receivers*). Each member should declare the time duration of staying in the session. In other words, the departure time of all receivers is known. While progressing in time and at the time, t_i, and when a new member, M_i, send a request to join the session, our algorithm has to connect the new member using the existing tree; it has to find the best route to this new member, this route should not be affected when one (or more) existing member left the session in the future. With the new route we guarantee that the tree has minimal cost even when topology changes.

Leave procedure – When the leaving time for a node equal to current time, this node must leave the session. It starts by sending a leave request. If the node is a leaf node it can be removed instantaneously. Links and nodes in the same route can be removed recursively, until we reach a node with departure time bigger then the current time. If the node is an internal node or a Steiner node, it can't be removed because it acting for others members. After the node is deleted from the multicast tree,

Simulation results

To evaluate our approach, we went to compare it with the two other methods (*Naif* and *Greedy*). Random graphs are generated using Waxman algorithm [2], parameters are chosen in order to get an average node degree connection in the range]3, 4[, which is close to the average node degree of current network. In each network, we select the first node as the source node. The number of destination is varied from one session to another. A random arrival and departure time are generated for each participating nodes in the considering session. For a fixed number of nodes in a network a different set of destination nodes has been used. In the first simulation we choose two arbitrary nodes not from the multicast tree. The additional cost represents the ration of the cost added for these two members by the original tree cost. We repeat simulation ten times, with same parameters, to get the average value (Figure 1-a). Simulations show that with our algorithm the added cost for two new members is almost equal to the *Greedy* algorithm. This means that our approach don't produce a very high tree cost. In the

second simulation, we went to examine how much time we need to find again a new route for a member who has just connected. With the greedy technique, new member join the multicast tree using short path can contain existing members which went to leave the session before its end, so for this new member we should find a new route. We define the parameter α as a ratio of the number of members who need to find again a new route, by the numbers of new members for one session. Figure 1-b shows the variation of α versus the size group. Using our approach this variation is nil, with the greedy approach this variation is significant ($\alpha \geq 40$ %). simulations results can confirm the efficiency of our approach which takes into consideration the time duration of each member.

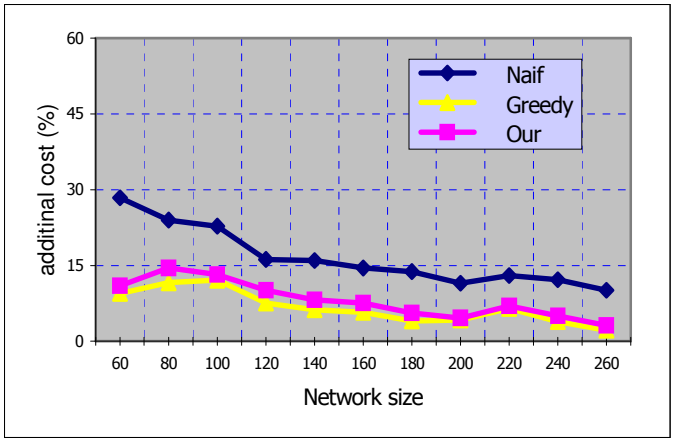

Figure 1-a: Added Cost in percent for two new numbers versus network size

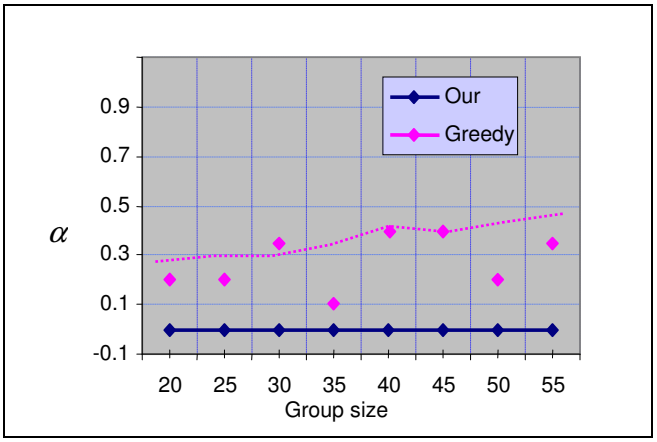

Figure 1-b: α versus group size

Conclusion

Dynamic multicast routing is required for multimedia application and it is an important issues. Satisfying various QoS constraints required an efficient procedure to connect a member who went to join/leave a current session. In this paper, we presented a new approach for the maintenance of multicast tree when a member has to join/leave the session. Compared to existing approaches, our approach gives a good simulation results. Our approach tries to find the best route without any risk caused by the dynamic of members group. More investigation and simulation will be done in the future.

References

[1] C. Huitema (1995). Routing in the Internet. Englewood Cliffs, NJ, Prentice Hall.

[2] B.M. Waxman (1986). "Routing of multiple connections." IEEE Selected Areas in Communications 6(9).