On the number of aggregated multicast trees in a domain

Joanna Moulierac

Advisor: Miklós Molnár
IRISA/University of Rennes I
Campus de Beaulieu, 35 042 Rennes Cedex, France
Phone number: +33 2 99 84 71 94
Email: {joanna.moulierac}@irisa.fr

***Keywords*— Multicast, aggregation, aggregated tree.**

I. Introduction

In order to allow the deployment of multicast over Internet, tree aggregation has been proposed in [1]. Multicast tree aggregation reduces the number of multicast forwarding states stored in the routers by allowing multiple groups to share the same delivery tree. In this way, this proposition can solve the problem of multicast scalability.

Consider the figure 1 where multicast tree aggregation is depicted:

- If multicast is deployed in this domain, three trees are needed for the three multicast groups g_1, g_2 and g_3. In order to configure these trees, and if we consider bidirectional trees, 15 forwarding states are required. Recall that it is necessary to store as many forwarding states as there are routers covered by trees.
- Now, if tree aggregation is deployed in this domain, only two trees can be utilized as the groups g_1 and g_3 can use the same tree. In order to configure these two trees, only 10 forwarding states are required. The packets in the domain are routed with the labels (corresponding to the trees) and not with the multicast address.

Consequently, in the domain, the number of forwarding states increases with the number of trees configured in the domain. As multiple groups share the same tree, there are less trees than groups and the number of forwarding states is strongly reduced in the domain.

Several protocols exist in order to achieve tree aggregation. Usually, for these algorithms, an entity is responsible of the aggregation and is requested for each new group. The protocol STA [2] propose to speed up the aggregation algorithm by an efficient sort of the trees and by an improved selective function. The protocol Q-STA [3] and AQoSM [4] deal with QoS and especially with bandwidth constraints. The protocol DMTA [5] is a distributed protocol that proposes to add significance to labels.

In this paper, we discuss about the number of aggregated trees needed to be configured initially in a given domain. Indeed, it is usually agreed that for a domain with b routers (routers that can be attached to multicast members), then there are 2^b different multicast groups. We consider in this paper, that a multicast group is a set of routers (the set of the multicast members of groups). Consequently, the number of different multicast groups is the enumeration of all the possible configuration of routers. Obviously, it is not possible to configure a tree for each group (as in multicast) and then to configure the 2^b trees. However, we will show that the number of possible different trees needed to be configured can be strongly reduced and is significantly smaller than the number of different groups.

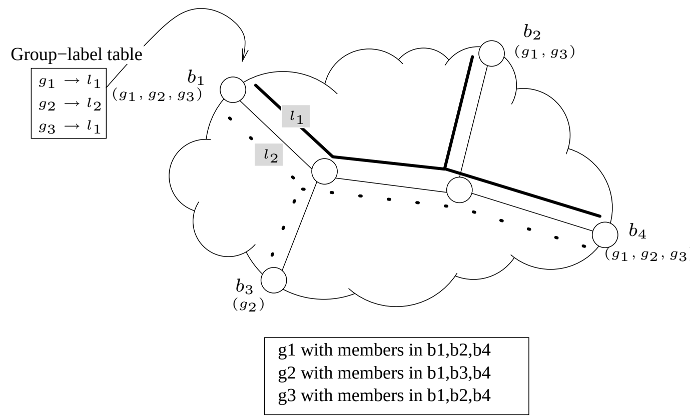

Fig. 1. The groups g_1 and g_3 can use the same delivery tree.

II. How to configure the set of trees?

The question is, given a set of k multicast groups, how to configure a set of trees that will be used by these k groups. This question is even more important if the network administrator wants to configure the set of trees that can cover any of the 2^b multicast groups without wasting too much bandwidth.

A. Main principles and proposed algorithm

Figure 2 shows that a tree can cover several different multicast groups. For example the tree represented on the figure is the native tree, considering an algorithm of tree construction A for the group (b_1, b_2, b_3, b_4), but also for the group (b_1, b_2, b_4) and for the group (b_1, b_3, b_4). That means that only one tree can be configured for these three groups without any wastage of bandwidth.

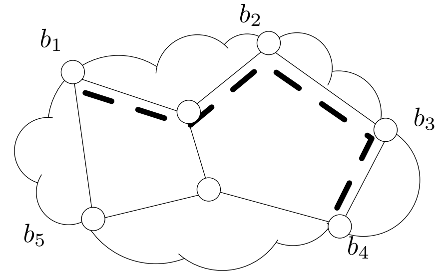

Fig. 2. The tree represented on the figure covers different multicast groups.

Network	Number of different Groups	Number of different Trees (MST)	Number of different Trees (SPT)
Nsfnet Network [6]	2 036	370	131
Abilene Network [7]	16 369	4 785	958
Geant Network [8]	262 125	48 942	8 222

TABLE I

NUMBER OF DIFFERENT TREES FOR ALL THE POSSIBLE MULTICAST GROUPS

Obtaining the number of different trees analytically is quite difficult, because the analysis would greatly depend on the topology and on the algorithm of tree construction. That is why we propose Algorithm 1 to compute this set of different trees for k given multicast groups. A given tree can be built for two different groups and in this case it is not added in $\mathcal{T}$. At the end of the algorithm, each of the k groups can be covered by a tree in $\mathcal{T}$ and all the trees in $\mathcal{T}$ are native trees for groups, *i.e.* no bandwidth is wasted if the groups utilized these trees.

Input: A domain with B routers, an algorithm A for the construction of trees, a set $\mathcal{G}$ of k multicast groups
Output: A set of trees $\mathcal{T}$ built with algorithm A covering all the k groups
$\mathcal{T} \leftarrow \emptyset$
for $i = 1$ to k **do**
The group g_i is the i-th of the k groups
Compute a tree t_i covering g_i using algorithm A
if There is no tree in $\mathcal{T}$ that covers exactly the same routers as t_i **then**
add t_i to $\mathcal{T}$
return $\mathcal{T}$

Algorithm 1: Configuring the set of multicast trees

B. Simulations and Results

We simulate the Algorithm 1 on different topologies, Nsfnet [6] (with 11 routers), Abilene [7] (with 14 routers) and Geant [8] (with 18 routers). The results of these simulations are represented on table I. During the simulations, the set $\mathcal{G}$ contained all the possible multicast groups of the domain. Note that the number of different groups for a domain with b routers is equal to $(2^b - b - 1)$ as we consider only the groups with at least 2 members. Two algorithms of tree construction are considered: firstly, minimum spanning trees are built (in the metric closure graph in order to get approximate Steiner trees) and secondly, shortest-path trees are built from a source randomly chosen among the members. Note that we consider bidirectional trees and that the shortest-path tree can be utilized in both ways, our only concern for the moment is that the members are covered by the tree.

The results presented in the table show that the number of different trees is very low compared to the number of different groups. Note that only a subset of these trees can be configured, only a subset covering the k given multicast groups with $k << 2^b$. The table shows that with shortest path tree algorithm, less trees are built than with minimum spanning trees. This can be explained as a shortest path trees covers more routers than a minimum spanning tree and then it can cover more groups. Consequently, more bandwidth is wasted with shortest-path trees than with minimimum spanning trees.

Moreover, the mean number of forwarding states in routers to be kept is rather small. Indeed, if we consider shortest-path trees, for Nsfnet, the routers keep around 62 forwarding states while for Abilene, around 522 states are kept and for Geant, around $4\,534$ states.

III. TOWARDS A DISTRIBUTED PROTOCOL

Considering that the number of possible trees is small allows to configure all of them during initialization process. In this way, a tree is configured for any new multicast group and no more multicast forwarding entry has to be added when new groups arrive.

If the trees are configured off-line, a distributed multicast tree aggregation protocol can be proposed. Indeed, current tree aggregation protocols are either centralized either ineffectively distributed. Centralized protocols are not failure tolerant and in distributed protocols, there are permanently control messages between the entities responsible of the aggregations.

If the set of trees is configured off-line, each entity responsible of the aggregation is aware of the trees configured in the domain and does not need to request the other entities if no tree is found for a given group (see BEAM [9] for more details). Indeed, with our proposition, the aggregation can be done directly by the entity requested which searches an aggregated tree in the set of initially configured trees. In this way, no more control messages are needed between the entities and for the configuration of the new aggregated trees.

IV. CONCLUSION AND PERSPECTIVES

This short analysis gives us several perspectives of research. Indeed, we can think of configuring a set of multicast trees for a given domain. The number of multicast forwarding states needed to be stored is rather small as shown during the simulations. This set of trees can be rather stable and the routers do not need to configure others forwarding states for new multicast groups except in case of failures where a reconfiguration may be done. This allows to think of a distributed protocol where the entities responsible of the aggregation will not need to exchange messages in order to aggregate new groups.

REFERENCES

[1] A. Fei, J.-H. Cui, M. Gerla, and M. Faloutsos, “Aggregated Multicast: an Approach to Reduce Multicast State,” in *Global Internet Symposium*, November 2001.
[2] A. Guitton and J. Moulierac, “Scalable Tree Aggregation for Multicast,” in *8th International Conference on Telecommunications (ConTEL)*, June 2005, Best student paper award.
[3] J. Moulierac and A. Guitton, “QoS Scalable Tree Aggregation,” in *IFIP Networking*, ser. LNCS, no. 3462, May 2005.
[4] J.-H. Cui, L. Lao, M. Faloutsos, and M. Gerla, “AQoSM: Scalable QoS multicast provisioning in Diff-Serv networks,” *Computer Networks*, May 2005.
[5] J. Moulierac and A. Guitton, “Distributed Multicast Tree Aggregation,” Inria, Tech. Rep. 5636, July 2005.
[6] “Nsfnet network,” http://www.cybergeography.org/atlas/nsfnet_t1.gif, 11 nodes, 14 edges.
[7] “Abilene network,” http://abilene.internet2.edu, 18 nodes, 30 edges.
[8] “Geant network,” http://www.cybergeography.org/atlas/geant_large.gif, 18 nodes, 30 edges.
[9] J.-H. Cui, L. Lao, D. Maggiorini, and M. Gerla, “BEAM: A Distributed Aggregated Multicast Protocol Using Bi-directional Trees,” in *IEEE International Conference on Communications (ICC)*, May 2003.