QoS Scalable Tree Aggregation

Joanna Moulierac and Alexandre Guitton

IRISA/University of Rennes I, 35 042 Rennes Cedex, France
{joanna.moulierac, alexandre.guitton}@irisa.fr

Abstract. Some of the main reasons which prevents the deployment of IP multicast are forwarding state scalability and control explosion problems. In this paper, we propose an algorithm called Q-STA (QoS Scalable Tree Aggregation) which reduces the number of forwarding states by allowing several groups to share the same tree. Q-STA accepts groups only if there is enough available bandwidth. Q-STA accepts much more groups and performs faster aggregations than previous algorithms.

1 Introduction

The deployment of multicast is limited mainly due to multicast forwarding state scalability and control explosion problems [1,2]. Tree aggregation is a recent approach to deal with these problems: several groups share the same delivery tree within a domain. Consequently, less trees are maintained in the network and the number of forwarding states in routers is decreased. Moreover, the overhead of control messages required for tree maintenance is reduced. Several algorithms have been proposed to perform tree aggregation [3,4]. Algorithm Aggregated QoS Multicast (AQoSM), described in [5], is a framework to support QoS multicast. The goal of AQoSM is to aggregate groups to trees while respecting bandwidth requirements of groups. Some groups may be refused because of the limited capacity of links.

In this paper, we propose a new algorithm called QoS Scalable Tree Aggregation (Q-STA) based on the framework of AQoSM. Q-STA accepts much more groups than AQoSM by building efficient trees and performing better aggregations. Additionally, the aggregation of Q-STA is faster than with AQoSM. We describe Q-STA in Section 2. Then, in Section 3, we compare AQoSM and Q-STA by extensive simulations.

2 Q-STA algorithm

In order to deal with bandwidth constraints, each group is assigned a bandwidth requirement and each link is assigned a limited bandwidth capacity. To accept a group g, the bandwidth available on the tree has to be sufficient, otherwise g is rejected. Our protocol Q-STA is an extension of our proposition STA [6]. Q-STA has better performance than AQoSM because of the following aspects: (i) Q-STA builds efficient trees by utilizing links with lowest load in order to aggregate further groups to this tree and to balance the load, (ii) Q-STA aggregates groups to minimum cost trees in order to spare network resources and (iii) Q-STA performs fast aggregations by evaluating few trees each time a group arrives whereas AQoSM evaluates all the trees in the multicast tree set.

Each time a new group g arrives, Q-STA considers only a subset of the multicast tree set (iii). This subset corresponds to the trees of cost between $(|g|-1)$ and $c(t_g)$, where $|g|$ is the number of members of g and $c(t_g)$ is the cost of the native tree of g. As soon as a candidate with enough bandwidth available is found, it is chosen for aggregation of g. As trees are evaluated in increasing order of their costs, the chosen tree minimizes the number of links used for the group g (ii). If no tree of adequate cost or with enough bandwidth is found for g, a new tree is builded and added in T. This new tree maximizes the bandwidth available on links (i). The loaded links are avoided and Q-STA builds trees with links that are little utilized by AQoSM.

3 Simulation analysis

We conducted several simulation experiments on the graph Abilene which contains 11 nodes and 14 edges. We choose to assign 1 Gb/s of the 10 Gb/s available as the capacity dedicated to multicast for each link of Abilene. In this way, we take into account the utilization of the links by unicast communications.

We compared three algorithms: PIM-SM, AQoSM (with its bandwidth threshold equals to 0) and Q-STA (the program of Q-STA can be found at [7]). PIM-SM neither performs tree aggregation nor avoid congested links. The groups and their bandwidth requirements were the same for the three algorithms: 50% of the group requests were low-bandwidth (10 Kb/s), 30% of the group requests were medium-bandwidth (100 Kb/s) and the remaining 20% of the group requests were high-bandwidth (1 Mb/s). We implemented the node weighted model with 80% of the nodes having a weight of 0.2 and 20% of the nodes having a weight of 0.6 (see [8]). In this model, nodes with a large weight have an high probability of being members of groups. We varied the number of concurrent groups from $1,000$ to $10,000$. Each plot is an average of 100 simulation scenarios.

3.1 Number of accepted groups

Figure 1 shows the number of groups accepted by PIM-SM, AQoSM and Q-STA. These three algorithms accept the first $5,000$ groups because the network capacity is sufficient. After the first $5,000$ groups, PIM-SM refuses a large number of groups: it accepts only 800 of the next $5,000$ groups. Since PIM-SM does not take into account the available bandwidth of links, a group is refused as soon as its native tree uses saturated links. AQoSM accepts 200 more groups than PIM-SM. By building native trees maximizing the bandwidth available on links, Q-STA accepts $1,200$ groups more than AQoSM.

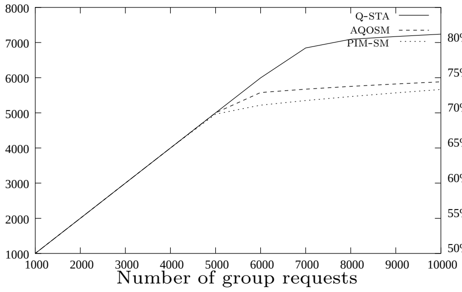

Fig. 1. Number of accepted groups.

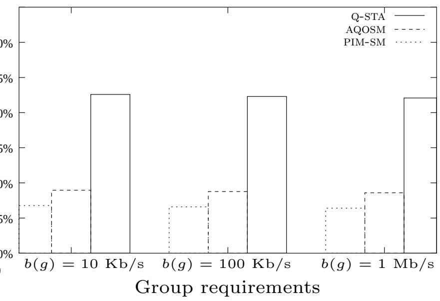

Fig. 2. Percent of accepted groups.

3.2 Percent of accepted groups per bandwidth requirement

The number of accepted groups is not sufficient to evaluate an algorithm if groups have different requirements. Figure 2 plots the percent of accepted groups considering their bandwidth requirements. The three algorithms are fair: they accept groups independently of their bandwidth requirements. Unfair algorithms would have refused high-bandwidth groups in order to accept more low-bandwidth groups. Q-STA accepts 73% of high-bandwidth groups whereas AQoSM accepts only 59% and PIM-SM accepts 57% of these groups.

3.3 Aggregation ratio

The aggregation ratio is defined as the number of trees over the number of concurrent groups. It determines the performance of a tree aggregation algorithm: a low aggregation ratio means that many groups have been aggregated. Figure 3 displays the aggregation ratio of AQoSM and Q-STA. The aggregation ratio of PIM-SM is not plotted since it is equal to 1. Before $6,000$ groups, AQoSM has a lower aggregation ratio than Q-STA: AQoSM builds less trees for the same number of groups. Instead of aggregating at all cost as AQoSM does, Q-STA balances the load by building more trees. After $6,000$ groups, Q-STA is slightly better than AQoSM, because Q-STA accepts more groups than AQoSM without building new trees. Q-STA builds around 50 more trees than AQoSM but handles around $1,200$ more groups. Finally, AQoSM and Q-STA build significantly less trees than PIM-SM (around 300 trees for AQoSM and Q-STA instead of $5,700$ trees for PIM-SM).

3.4 Number of evaluated trees

In AQoSM, all the trees are evaluated each time a new group arrives. In Q-STA, only trees that are likely to be candidates are evaluated. Additionally, Q-STA chooses as a tree for g the first tree with minimum cost that can cover g and that has enough bandwidth. Figure 4 shows the number of evaluated trees by AQoSM and Q-STA. Q-STA evaluates 45% less trees than AQoSM. To deal with

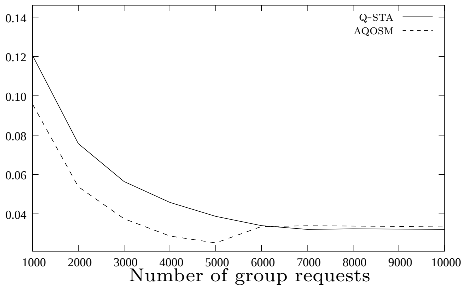

Fig. 3. Aggregation ratio.

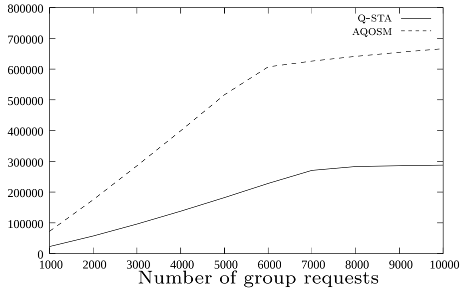

Fig. 4. Number of evaluated trees.

10, 000 group requests, AQoSM evaluates 380, 000 trees more than Q-STA. This reduction of the number of evaluated trees enables faster aggregations.

4 Conclusion

In this paper, we proposed a new algorithm Q-STA for tree aggregation with bandwidth constraints. We compare the performance of our algorithm with the previous algorithm AQoSM by extensive simulations. While AQoSM accepts 200 groups more than PIM-SM, Q-STA accepts 1, 400 more groups than PIM-SM. Indeed, Q-STA spares network resources by building efficient trees and by aggregating groups to low cost trees. Moreover, Q-STA builds as few trees as AQoSM and performs faster aggregations than AQoSM by evaluating 45% less trees.

References

1. Almeroth, K.: The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment. IEEE Network **14** (2000) 10–20
2. Wong, T., Katz, R.: An Analysis of Multicast Forwarding State Scalability. In: IEEE International Conference on Network Protocols (ICNP). (2000)
3. Cui, J.H., Kim, J., Maggiorini, D., Boussetta, K., Gerla, M.: Aggregated Multicast — A Comparative Study. In: IFIP Networking. (2002)
4. Liu, Z.F., Dou, W.H., Liu, Y.J.: AMBTS: A Scheme of Aggregated Multicast Based on Tree Splitting. In: IFIP Networking. (2004) 829–840
5. Cui, J.H., Kim, J., Fei, A., Faloutsos, M., Gerla, M.: Scalable QoS Multicast Provisioning in Diff-Serv-Supported MPLS Networks. In: IEEE Globecom. (2002)
6. Guitton, A., Moulierac, J.: Scalable Tree Aggregation with a Large Number of Multicast Groups. Internal publication 1663, IRISA (2004)
7. Q-STA: (2004) http://www.irisa.fr/armor/lesmembres/Guitton/recherche/qsta/qsta-en.html.
8. Fei, A., Cui, J.H., Gerla, M., Faloutsos, M.: Aggregated Multicast: an Approach to Reduce Multicast State. In: Global Internet Symposium. (2001)