Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 3069–3089
Computer Networks
www.elsevier.com/locate/comnet

Tackling group-to-tree matching in large scale group communications ☆

Li Lao [a], Jun-Hong Cui [b,*], Mario Gerla [a]

[a] Computer Science Department, University of California, Los Angeles, United States
[b] Computer Science and Engineering Department, University of Connecticut, Storrs, United States

Received 12 September 2005; received in revised form 25 August 2006; accepted 10 January 2007
Available online 19 January 2007

Responsible Editor: A. Kshemkalyani

Abstract

As a mechanism to efficiently support group communications, multicasting, faces a serious state scalability problem when there are large numbers of groups in the network. Recently, a novel solution called *Aggregated Multicast* has been proposed, in which multiple groups can share one delivery tree. A key problem in Aggregated Multicast is group-to-tree matching (i.e., assigning groups to proper trees). In this paper, we formally define this problem, and formulate two versions of the problem: static and dynamic. We analyze the static version and prove that it is NP-complete. To tackle this hard problem, we propose three algorithms: one optimal (using Linear Integer Programming, or ILP), one near-optimal (using Greedy method), and one Pseudo-Dynamic algorithm. For the dynamic version, we present a generic dynamic on-line algorithm. Simulation study has been conducted to evaluate the performance of the algorithms. Our results show that: (1) for the static problem, the Greedy algorithm is a feasible solution and its performance is very close to the optimal ILP solution, while the Pseudo-Dynamic algorithm is a good heuristic for many cases where Greedy does not work well; (2) for the dynamic problem, the generic dynamic on-line algorithm is a very practical solution with promising performance and reasonable computation requirement.

Keywords: Multicast; Group communication; Aggregated multicast; Group-to-tree matching

1. Introduction

With the rapid development of the Internet, there are many emerging large scale multi-user applications, such as distributed interactive simulation (DIS), distributed network games, teleconferencing and telemedicine. All these applications involve multi-point group communications (that is, delivering data from one or more sources to multiple receivers). To support these applications efficiently, multicast is usually employed, in which a concept of group is introduced: sources send data to an advertised group; receivers who are interested in the data need to subscribe to the group in order to receive the data.

☆ This material is based upon work supported by the National Science Foundation under Grant No. 0435515 and Grant No. 0435230.
* Corresponding author. Tel.: +1 (860) 4868951.
E-mail addresses: llao@cs.ucla.edu (L. Lao), jcui@cse.uconn.edu (J.-H. Cui), gerla@cs.ucla.edu (M. Gerla).

1389-1286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2007.01.003

In multicast protocols, a tree delivery structure is widely used due to its resource efficiency. Each in-tree node maintains forwarding state, and data packets are duplicated at fork nodes and are forwarded only once over each link. Traditionally, each multicast group uses one delivery tree. To manage multicast groups, resources (e.g., memory to maintain group forwarding state) and control overhead (e.g., setup and maintenance of multicast trees) are required. When there are large numbers of multicast groups in the network, a lot of resources and management overhead will be involved. Hence, the network performance will be tremendously degraded. This issue is referred to as multicast state scalability problem. It will be exacerbated with the increasing demand of multi-user applications.

The recognition of the state scalability problem has prompted a significant amount of interesting work. Some approaches resort to different multicast architectures, such as application-layer multicast [18,9] and Xcast [6], aiming to completely eliminate multicast state at routers. These solutions either sacrifice multicast efficiency or increase packet processing overhead at routers. Some other schemes attempt to reduce forwarding state at non-branching tree nodes [29,27,11,30] or aggregate forwarding state at individual nodes [24,28,8]. However, all of them only consider reducing forwarding state at routers without solving the control overhead issue.

On the other hand, a recently proposed approach called Aggregated Multicast [16] exploits both the resource and control overhead issues. In this scheme, multiple multicast groups are aggregated to share a single delivery tree (which is called an *aggregated tree*). This way, the total number of trees in the network may be significantly reduced and thus the forwarding state would be decreased accordingly. Aggregated multicast involves group-to-tree matching (i.e., assigning groups to trees) since proper trees should be found to deliver data for groups. To solve the state scalability problem, the objective of group-to-tree matching algorithms would be to minimize the resources and control overhead. In previous study [26,14,13], several Aggregated Multicast protocols using heuristic on-line group-to-tree matching algorithms have been proposed; however, there is no formal analysis of the group-to-tree matching problem, and there is no formal evaluation of how good the on-line heuristics are in comparison with optimal solutions.

In this paper, we formally define the group-to-tree matching problem and formulate two versions of the problem: static and dynamic. In the static version, we assume all the groups are known beforehand, i.e., we have the knowledge of global group information. This case is useful for multicast tree pre-dimensioning based on long-term traffic measurement. We analyze the complexity of this problem and show that it is NP-complete. We propose three algorithms: one optimal (using Linear Integer Programming, or ILP), one near-optimal (using Greedy method), and one Pseudo-Dynamic algorithm. By simulation study, we show that the Greedy method is much faster and less time consuming than ILP while the performance is not significantly sacrificed. The Pseudo-Dynamic algorithm is even faster and more resource efficient, but it trades off the performance for efficiency. For the dynamic version of the problem, groups dynamically join and leave, and there is no global information about all the groups. This is more meaningful for managing on-line systems. We present a generic dynamic on-line group-to-tree matching algorithm, and evaluate its performance by comparing it with its upper bound (obtained using the static algorithms) and existing on-line heuristics. We find that the generic dynamic on-line algorithm is a practical solution to the dynamic group-to-tree matching problem with limited performance penalty and reasonable computation requirement.

The rest of this paper is organized as follows. In Section 2, we first describe and formulate the group-to-tree matching problem. Then we present several algorithms for both the static and dynamic versions of the problem, and give formal time complexity analysis of these algorithms (Sections 3–5). After that, we conduct simulation study and compare different algorithms quantitatively in Section 6. Finally, we discuss some related work in Section 7, and conclude the paper in Section 8.

2. The group-to-tree matching problem

2.1. Problem description

In traditional multicast, each group uses one delivery tree, while in Aggregated Multicast [16], multiple groups are forced to share one aggregated tree. Thus, to implement Aggregated Multicast, we need to conduct group-to-tree matching to assign groups to proper trees. One thing to note is that Aggregated Multicast is targeted for a single domain, especially a transit domain, where member dynamics are usually not as significant as in stub