Computer Networks 57 (2013) 2916–2930

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

On multi-stream multi-source multicast routing ☆

Yuh-Rong Chen [a], Sridhar Radhakrishnan [a,*], Sudarshan Dhall [a], Suleyman Karabuk [b]

[a] School of Computer Science, The University of Oklahoma, 110 W. Boyd St., Room 150, Norman, OK 73019, USA
[b] School of Industrial Engineering, The University of Oklahoma, 202 W. Boyd St., Room 124, Norman, OK 73019, USA

ARTICLE INFO

Article history:
Received 31 August 2012
Received in revised form 28 March 2013
Accepted 11 May 2013
Available online 6 July 2013

Keywords:
Multicast
Multi-source
Multi-tree
Tree packing
Optimization

ABSTRACT

Multicasting has been used to conserve bandwidth and reduce network traffic for delivering a single data stream to a set of destinations. The problem of conserving bandwidth is a challenging one when one considers multiple data streams with multiple sources for each and a set of destinations that subscribe to one or more of these streams. The Multi-stream Multi-source Multicast Routing Problem (MMMRP) is to determine multiple multicasting trees on a given network, rooted at sources (nodes in the network) that are responsible for delivering one or more data streams to a set of destinations. Since several multicast trees co-exist on the same network, our goal is to construct these trees in such a way that the minimum residual bandwidth on the links that are shared among the trees is maximized. We prove that MMMRP is $\mathcal{NP}$-hard and apart from providing an IP formulation, we have also provided a heuristic algorithm *MMForests*, which runs in polynomial-time. We compared and contrasted MMMRP with known algorithms for the *multicast tree packing problem* and our exhaustive empirical evaluations show that our heuristic has a very low execution-time while achieving near-optimal residual bandwidth. In addition, our heuristic is very scalable as it is able to produce results for networks with thousands of nodes, unlike the other ones that are based on Steiner tree heuristics.

1. Introduction

Multicasting is an efficient way to deliver multimedia contents or large files from a single source to multiple destinations. A multicasting tree rooted at the source is usually used for multicasting. The internal nodes of a multicasting tree duplicate every packet they receive and send it to all their children in the tree. Some of the internal nodes could be destination nodes and all leaf nodes are certainly destination nodes. Consider the Internet with its router topology. A (native) multicast data distribution tree is formed with routers as its internal nodes and end-hosts (computers) as the source and destinations, which are connected to the routers. The Internet Group Management Protocol (IGMP) [1] is used by an end-host (in IPv4) to signal its interest in a particular multicast group to its closest router. The Multicast Listener Discovery (MLD) [2] protocol is used for the same purpose in IPv6. Construction of the multicast data distribution tree is achieved by a multicast routing protocol, such as Protocol Independent Multicast [3]. However, explicit multicast routing is usually forbidden by Internet Service Providers on their publicly-visible routers, thus denying (native-mode) multicast data transmission to their customers.

There has been a significant amount of research work that involves construction of multicasting trees to satisfy various Quality-of-Service (QoS) requirements of multimedia applications. While special trees can be constructed, these trees cannot be readily used with a (native-mode) multicast routing protocol as they require changes to router software. To overcome this issue, application layer multicasting has been proposed and widely implemented

☆ A preliminary version of this paper was presented in GLOBECOM 2012.

* Corresponding author. Tel.: +1 405 325 3038.
E-mail addresses: yrchern@ou.edu (Y.-R. Chen), sridhar@ou.edu (S. Radhakrishnan), sdhall@ou.edu (S. Dhall), karabuk@ou.edu (S. Karabuk).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.05.012

[4–7]. In this approach, end-hosts form a connected topology called an *overlay network*. In an overlay network, each link is the Internet path formed by Internet routers connecting two end-hosts. This architecture is very flexible in the sense that newer protocols can be easily incorporated at the end-hosts, but is less efficient because the multicasting paths may sometimes involve overlapping Internet paths [8]. In this paper, we do not make any special assumptions on the network architecture under consideration.

There has also been a growing interest in building multiple multicast trees. Castro et al. [5] developed SplitStream, where they split the source stream into k stripes and multicast them using disjoint multicast trees, i.e., the trees do not share common interior nodes. The destinations (or subscribers) then obtain each stripe from a different tree. Birrer et al. [9] address the issue of bandwidth, especially it being the bottleneck as we move closer to the root (or source). They do this by building *fat-trees* for multicasting, wherein the outgoing links near the root have higher bandwidth compared to links that are further away from the root.

One approach to solving the multiple multicast problem is to build multicasting trees for each stream and combine the multicast trees. This approach may not always produce a result (or one that is desirable). For example, say we have a source s and a destination t and there are two video streams that need to be sent from s to t and each consumes 1 unit of bandwidth. If we solve the problem for each of the streams individually, we may get two edge-joint paths that consume 2 units of the bandwidth on the common edges. A better solution with less congestion could be two edge-disjoint paths from s to t, which results in 1 unit of bandwidth usage. Several papers have addressed these issues for different scenarios such as minimum interference routing between source–destination pairs in multi-protocol label switched (MPLS) networks [10,11] or multicast tree packing [12–14].

In this paper, we consider the problem of delivering multiple data streams to their destinations taking into consideration that each stream can originate from one or more sources. Our goal is to develop algorithms to reduce the congestion on the communication links and increase their residual bandwidths. The rest of this paper is organized as follows. Several relevant research works are reviewed in Section 2. In Section 3, we introduce the notations, define Multi-stream Multi-source Multicast Routing Problem MMMRP, and prove the $\mathcal{NP}$-hardness of this problem. Integer Programming (IP) formulations are then provided in Section 4 and the heuristic algorithm based on widest path algorithm is presented in Section 5. Performance evaluation and results are presented in Section 6 with conclusions drawn in Section 7.

2. Related work

There have been a number of techniques for creating multiple multicasting trees that optimize various resources. For example, there have been several works have tried to reduce the number of nodes that participate in the multicasting trees [7]. A number of researchers have developed techniques to minimize the total resource consumed by all multicast trees [14], and others that try to reduce the number of shared links among the multiple trees [12,13]. There are also approaches that combine many constraints such as the number of nodes, total bandwidth, and bandwidth constraints on links [12–14].

The minimum interference routing problem is discussed in [10,11]. Kar et al. [10] considered the problem of routing data between source–destination pairs in MPLS networks. Data from the source is routed to destinations using one more edge-disjoint paths. Figueiredo et al. [11] later developed an algorithm that improved its computation time.

Chen et al. [12] considered the multicast tree packing problem wherein groups of participants communicate with other participants within the same group. Each group uses a multicast tree for many-to-many multicasting. The goal of the multicast packing problem is to minimize the maximum congestion (the number of times a link is shared) among the communication links while keeping the size of each multicast tree within a bound. Chen et al. [12] developed IP models together with a heuristic algorithm called *TreePacking*. Their solution methodology involves solving multiple Steiner tree problems individually and then improving the solution by rebuilding (refining) the trees that use the most congested link(s).

The problems considered in [12] assume that each multicast tree requires the same amount of bandwidth, in other words all data streams served require the same bandwidth. Lee and Cho [13] considered the same problem in which the bandwidth consumptions are all different and provided an algorithm called *MMTA*. An additional constraint in [12,13] is to keep the cost of the trees within a bound for Quality-of-Service. Wang et al. [14] address a similar problem but with a different objective wherein they aim to reduce the total cost of the multicast trees (cost on the communication links) while satisfying the bandwidth constraints of the communication links.

The research work mentioned above assumes that the multicast sessions consume constant bandwidth during their lifetime. Ravindran et al. [15] considers the problem of changes to the bandwidth that can occur at various points in a streaming environment and provide a technique to find routing paths.

The main difference between the problem addressed in this paper and the work in [12,14,13] is that our work considers the case in which each multicast session has *one or more sources* that can provide the data stream. The multicast trees that are constructed in [12,14,13] are used for group communication and there is no requirement to take into consideration source nodes. That is, each member in the group performs peer-to-peer communication with others in the group. Existing solutions [12,14,13] are not suitable for the problem under consideration based on the following reasons. First, the existing solutions use Steiner tree heuristics to reduce the number of participating nodes. If we relax the number of nodes constraint and focus on just the bandwidth related constraints, it may be possible to find better solutions that maximize minimal residual bandwidth. Second, we cannot remove the Steiner tree construction parts from the existing heuristics as they