Discrete Applied Mathematics 216 (2017) 598–608

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Efficient broadcast trees for weighted vertices

Hovhannes A. Harutyunyan [a], Shahin Kamali [b,*]

[a] *Department of Computer Science and Software Engineering, Concordia University, 1515 St. Catherine St. West, Montreal, QC, Canada H3G 1M8*

[b] *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA, 02139, USA*

ARTICLE INFO

Article history:
Received 13 February 2015
Received in revised form 15 September 2016
Accepted 22 September 2016
Available online 15 October 2016

Keywords:
Graph algorithms
Information dissemination
Broadcasting
Network design

ABSTRACT

In this paper, we consider a weighted-vertex model for information dissemination in communication networks. Each node of the network (e.g., a processing machine) is assigned a weight, which represents the delay of the node after receiving data and before sending it to its neighbors. We are interested in the broadcasting problem in which the goal is to disseminate information from one node to all other nodes as quickly as possible. We introduce a simple algorithm for optimal broadcasting from a given node in weighted-vertex trees. We also study the problem of constructing efficient broadcast trees that minimize the required time for broadcasting. We investigate two variants of this problem. First, we show that, given a set of vertices with specified weights, one can construct a tree that connects all vertices and enables broadcasting from *any* vertex in the optimal time of $\Theta(\log n)$. Second, given a set of weighted vertices among which one vertex is specified as the originator, we introduce a polynomial algorithm that connects vertices with a tree in which broadcasting from the originator completes in minimum time.

1. Introduction

In the classic model of communication, a network is modeled by a simple, undirected, unweighted graph. The communication in the network is synchronous and occurs in discrete *rounds*. In each round, a processor (i.e., a vertex) is allowed to send data to at most one of its neighbors. In the broadcasting process, a single vertex, called *the originator*, needs to share its data, called *the message*, with all other vertices. For that, all vertices in the network collaborate to spread data. The broadcasting is said to be *complete* when all vertices receive the message. The goal is to minimize the number of rounds required to complete broadcasting.

Within the above general theme, two flavors of the problem have been studied in the literature. In the *broadcast problem*, a network is presented and the goal is to minimize the broadcast time from a given originator. This problem was formalized by Hajnal et al. [8] and subsequently studied by many others. Another line of research considers the *construction* of efficient topologies that facilitate broadcasting. Here, the goal is to design networks in which broadcasting from *any* vertex can be performed in minimal time (see, e.g., [7]). In practice, adding edges between vertices is costly, and it is desirable to construct sparse graphs with a small number of edges. In particular, the problem of constructing efficient broadcast trees is important in practical scenarios. This problem was first studied by Khachatrian and Haroutunian [16] and independently by Labahn [18] who presented an optimal solution.

* Corresponding author.
E-mail addresses: haruty@cs.concordia.ca (H.A. Harutyunyan), skamali@mit.edu (S. Kamali).

http://dx.doi.org/10.1016/j.dam.2016.09.029

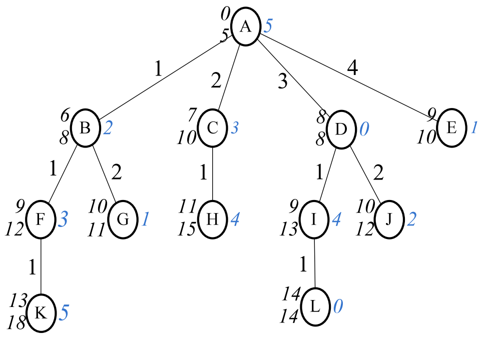

Fig. 1. A broadcast tree rooted at the originator A. The numbers on the right side of the nodes indicate their weights, while the numbers on the left side indicate the time steps at which nodes receive the message (top) and complete their delay (bottom).

The weighted-vertex model of broadcasting is a generalized model in which vertices have non-negative weights. The weight of each vertex represents the number of rounds that it should wait, after receiving the message, before it can send it to any of its neighbors. Our interest in this new model is mainly theoretical; however, the weights might carry practical meanings. In a parallel machine, for example, the weights represent the delays caused by the internal processes that machines (vertices) need to perform before informing their neighbors. The weights can also represent the delay for receiving the message due to the input device limitations. In general, assuming weights for vertices is useful for modeling heterogeneous networks in which nodes have different characteristics. One example of such networks, modeled by weighted vertex graphs, is Satellite–Terrestrial networks in which satellite and terrestrial nodes have different input–output delay and consequently different weights on vertices [1]. We assume the weights of vertices are non-negative integers that are independent of the number of vertices (i.e., they can be thought as being constant values when compared to the size of the graph). When all weights are 0, the model becomes equivalent to the classic model of broadcasting. The model that we consider is inherently different from a related model introduced by Bar-Noy et al. [3]. In that model, each vertex has a switch time delay and a receive time delay, and each edge also has a delay which is larger than the mentioned delays for its endpoints. Since weights are assigned to edges, this model is different from our model. For example, finding an optimal broadcast scheme in a complete graph with weighted edges is NP-hard, while in this paper (Section 3.2) we show that it is polynomially solvable in weighted-vertex graphs. We note that even if the weights on vertices are modeled by receive time delays, the weights on edges are still present, and hence the model of [3] is not a generalization of the one we consider.[1]

In this paper, we study the broadcast problem in the weighted-vertex trees as well as the problem of constructing efficient broadcast trees. In the former problem, similarly to the unweighted case, we are given a weighted-vertex tree and an originator, and the goal is to perform broadcasting as quickly as possible. In the latter problem, we are given a set of vertices with different weights (e.g., heterogeneous processors), and the goal is to construct a tree that spans all vertices. We study two variants of this problem. In the first variant, the goal is to perform broadcasting from *any* vertex as quickly as possible. In the second variant, a vertex is specified as the originator, and the goal is to minimize broadcast time stating from that vertex. The latter variant of the problem is equivalent to broadcasting in complete graphs. While this problem can be easily solved for unweighted graphs, it becomes non-trivial when weights are introduced.

In the weighted-vertex graphs, similar to unweighted ones, any broadcast scheme can be represented by a spanning tree rooted at the originator in which there is a link from u to v if v is informed through node u. For example, consider Fig. 1 in which a broadcast tree is depicted when node A is the originator. In this scheme, A receives the message at $t = 0$. It is busy during next 5 rounds until it completes its delay at $t = 5$. Then A informs B at time $t = 6$, C at time $t = 7$, D at time $t = 8$ and E at time $t = 9$. Similarly, B is busy in the time period [6, 8]. Afterwards, F receives the message through B at $t = 9$ and completes after 3 rounds when it sends the message to K. The broadcasting completes when K completes at $t = 18$.

1.1. Previous work and contribution

The classic model of broadcasting has been extensively studied in the literature (see [14,6,15,13] for surveys of the existing results). Recall that in the broadcast problem, the goal is to minimize the broadcast time in a given graph from a given originator. Johnson proved that the problem is NP-hard for general graphs (reported in [19]). The best existing upper bound is given by Elkin and Kortsarz, who introduced a polynomial-time algorithm with an approximation ratio of $O(\log n/\log\log n)$ for a graph of size n [4]. On the other hand, it is not known whether a polynomial algorithm can achieve a constant approximation ratio.

[1] For example, consider two neighboring vertices x_1 and x_2 with respective weights w_1 and w_2 where $w_1 < w_2$. If the weights are modeled with receive time delays of [3], the edge between the two vertices has a delay of at least w_2, i.e., when x_2 sends the message to x_1, the delay of x_1 to receive would be w_2 units (in contrast to w_1 in the model considered in this paper).