Multicasting in Directed Graphs

Michalis Faloutsos
Dept. of Computer Science
mfalou@cs.toronto.edu

Rajesh Pankaj
University of Toronto
Dept. of Electrical Eng.
pankaj@comm.toronto.edu

Kenneth C. Sevcik
Dept. of Computer Science
kcs@cs.toronto.edu

Abstract

The multicast problem is a significant problem for the efficient use of networks. Although it has been studied extensively, most previous work assumes undirected graphs. In this paper, we study the multicast problem on directed graphs. We show that even the best heuristics of the undirected case do not perform well on directed graphs. Namely, they can perform as bad as if we would route every participant separately in oblivion of the others. Finally, we extend the algorithms for directed graphs, and study their performance.

1 Introduction

Multicasting can be defined as the distribution of the same information stream from one to many nodes concurrently. A lot of emerging applications are of multicast nature, and thus multicasting has become a very active research area. The multicast problem on undirected graphs has been studied quite extensively. It can be stated as follows. Given an undirected weighted graph $G(V, E)$ and a subset of **participant** nodes $S \subseteq V$, we want to find the minimum weight tree that spans the participants. The problem is NP-complete and algorithms for the accurate solution have at least $O(N3^M)$ computational complexity [DW72], which raises the need for good polynomial complexity approximations. A common measure of how good such an approximation is the **worst-case inefficiency (WCI)** of a multicast algorithm that is defined as the maximum ratio of the cost of the tree of the algorithm over the optimal one. A number of algorithms exist with a WCI bounded by 2 [TM80] [KMB81] [RSC86] [Meh88]. A comprehensive survey of the algorithms can be found by Winter [Win87]. The union of the shortest paths between source and destination forms also a solution with a WCI of $O(M)$. We call this algorithm **Shortest Paths** Note that Shortest Paths routes every destination separately in oblivion of the other destinations.

A greedy and intuitive algorithm for the multicast problem was suggested by Takahashi and Matsuyama [TM80]. We start from the root and construct the multicast tree, that we denote as MT, by adding one participant at a time. At each step we connect the participant that is nearest to the partial multicast tree. We call this algorithm **Nearest Participant First** (NPF). Another greedy heuristic was proposed by Kou et al. in [KMB81]. This algorithm finds the minimum spanning tree on graph $G'(S, E')$, which is the complete graph that has only the participants of the original graph and the weights of the edges are equal to the weight of the shortest path between the participants.

Most previous work assumes undirected graphs. In practice, a communications network is usually a directed graph where every bidirectional link corresponds to two opposite directed edges. Furthermore, communication links are not necessarily bidirectional. Without loss of generality, we can assume that for every pair of nodes we have at most one directed edge for each direction. Recently, Ramanathan [Ram96] studied the directed multicast problem (**DMP**) and proved that a constant-factor approximation algorithm is highly unlikely. The proof is an elegant and short reduction of the DMP from the Minimum Set Cover.

In this paper, we study the performance of the algorithms by offering lower and upper bounds. We then show that for highly asymmetric graphs the WCI of the greedy algorithms is $\Theta(M)$, where $\Theta()$ denotes the tight bound.

2 The Directed Multicast Problem

The directed problem can be defined as follows. Given weighted directed graph $G(V, E)$, a set of participants $S \subseteq V$ and $s \in S$ a source node, find the minimum cost tree that has a directed path from s to all the other participants (which we call destinations). The weights of the edges can be represented by a function $V \times V \rightarrow Z_o^+$, which we denote as $e(u, w)$. We define function $d(u, w)$ to represent the cost of the shortest directed path from u to w.

The greedy algorithms we examined so far have to be extended. **Directed Shortest Paths** algorithms: the multicast tree is the union of the shortest directed

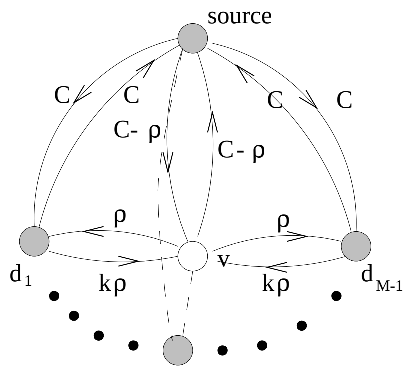

Figure 1: A bad example of a directed graph.

paths from the source to every destination.

Directed NPF algorithm: we start from the root, and, at each step, add the node whose directed path from the tree to itself is of minimum cost.

Directed KMB algorithm: We can construct G' as the graph of shortest paths between the participants, and perform NPF on G' starting from the root. Note that G' is not necessarily complete, and the resulting tree is not necessarily the minimum spanning tree.

In directed graphs the WCI of NPF and KMB is not bounded by a constant, and as we already said the existence of a constant approximation algorithm is not probable.

We want to quantify the asymmetry in the weights of the graph, and so we introduce the following metric.

Definition 1 *We call maximum asymmetry $k(R)$ of a set of edges R the ratio.*

$$\max_{(v,w)\in R}\left(\frac{e(v,w)}{e(w,v)},\frac{e(w,v)}{e(v,w)}\right)$$

It is interesting to study how WCI increases as the difference between the weights of the edges between each pair of nodes become greater. We will call these edges **opposite**. If one of the opposite edges does not exist, then we assume that its weight is infinite.

We prove the following theorems for the upper and lower bound of the NPF and KMB.

Theorem 1 (Upper bound) *The worst-case inefficiency, WCI, of KMB and NPF algorithms in directed graphs is bounded by*

$$min\left(\frac{\sum_{v\in S} d(s,v)}{d_{max}}, 2\ k(T_{OPT})\right)$$

where $k(T_{OPT})$ is the maximum asymmetry of the edges of the optimal tree, d_{max} is the maximum distance of a destination from the source, and T_{OPT} the optimal tree.

PROOF. Let T be the cost of the tree that results from NPF or KMB, and OPT the cost of the optimal solution. Note that T is bounded by the sum of the shortest paths between the root and the destinations, since at every step that joining participant v will never join with a cost higher than $d(s,v)$

$$T \leq \sum_{v_i} d(s,v_i) \leq M\ d_{max}$$

Clearly $d_{max} \leq OPT$. Thus,

$$\frac{T}{OPT} \leq \frac{\sum_{v\in S} d(s,v)}{d_{max}} \leq M$$

The proof that $\frac{T}{OPT} \leq 2\ k(T_{OPT})$ can be found in an article by Ramanathan [Ram96]. ■

It is worth noting that $\frac{\sum_{v\in S} d(s,v)}{d_{max}}$ is bounded by M.

We show how bad the WCI of the algorithms can be by constructing a family of graphs. For these graphs, the WCI can be linear to the maximum asymmetry of the graph, for a sufficiently large set of participants.

We say a node v **reaches** node w if there is a directed edge or path from v to w.

Theorem 2 Lower bound. *The WCI of KMB and NPF in a directed graph of maximum asymmetry k can be as bad as*

$$T \leq (k+1)(1-\frac{k}{k+M-1})OPT$$

where OPT the cost of the optimal tree.

PROOF. It is sufficient to prove the existence of a graph and a session that satisfies the inequality we want to prove. Assume the graph depicted in Fig. 1. The source reaches all the destinations with an edge of cost c, and there is only one non-participants node that we will call the **middle node**. This node reaches all the participants with an edge of cost ρ, while the opposite edges are of cost $k\ \rho$. The source reaches the middle node with a cost of $c-\rho$.

Clearly the optimal tree includes the middle node and its cost OPT is

$$OPT = c-\rho+(M-1)\rho = c+(M-2)\rho \qquad (1)$$

The worst case for NPF/KMB is to miss the middle node. This can actually happen if we assume that

$$c \leq \rho + k\rho = (k+1)\rho \qquad (2)$$

The cost T of the tree of NPF/KMB is the following

$$T = (M-1)c \qquad (3)$$

For the inefficiency of the algorithms we get the following

$$\frac{T}{OPT} = \frac{(M-1)c}{c+(M-2)\rho}$$

We can see that the inefficiency of NPF/KMB is maximized for a large c and a small ρ. However, using (Ineq.2) the maximum value of c for a given ρ is

$$c = (k+1)\rho$$

Substituting c in the inefficiency of the algorithms (Eq.4), we get

$$\frac{T}{OPT} = \frac{(M-1)(k+1)}{k+M-1} = (1 - \frac{k}{k+M-1})$$

This completes the proof. ■

We can notice the following special cases:

1. $k = 1$. It is the unidirectional case for which we get: $T \leq 2(1 - \frac{1}{M})OPT$. Note that this is the tight theoretical bound for the undirected problem as proven by Kou et al. [KMB81]. In this case M is the number of leaves in the optimal tree, as required in the bound for the undirected case.

2. $k \rightarrow \infty$. It is the case where we have one way links, for which we get $T \leq M \cdot OPT$. Note that this is the bound of the Shortest Paths algorithm.

3. $M \rightarrow \infty$ and $M \gg k$. We get : $T \leq (k+1)OPT$. In this case we see clearly the linear relation between WCI and k.

Given the two previous theorems we have the following corollary.

Corollary 1 *For directed graphs with maximum asymmetry at least* $\frac{M}{2}$, *the worst-case inefficiency of NPF and KMB is tightly bounded by* $\Theta(M)$.

Therefore, we observe that in (highly) asymmetric graphs the WCI of NPF, and KMB is comparable to that of the Shortest Paths. However, when the asymmetry is not that large the two greedy algorithms perform better than the Shortest Paths in the worst-case.

3 Experimental Results

We wanted to study the effect of the asymmetry on the performance of the algorithms in practice, and for this, we conducted a series of experiments. We chose to experiment on the family of graphs that we call **Euclidean** graphs, first because they are widely used in network simulations, and second, because, these graphs seem to have some characteristics that we intuitively expect from real networks. We can create a Euclidean

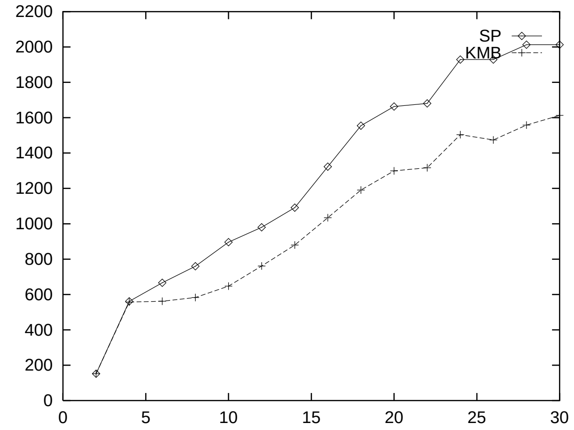

Figure 2: Cost versus number of participants, undirected case.

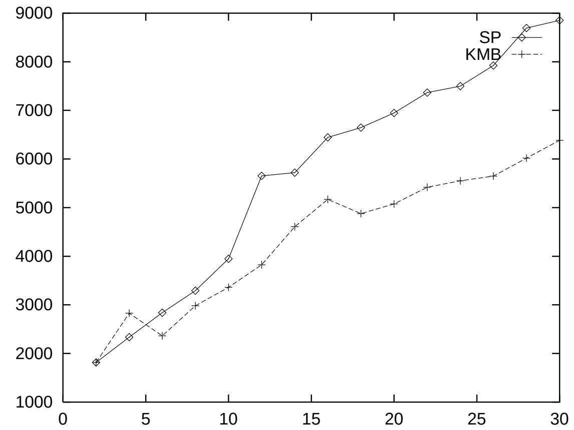

Figure 3: Cost versus number of participants, directed case $K = 30$.

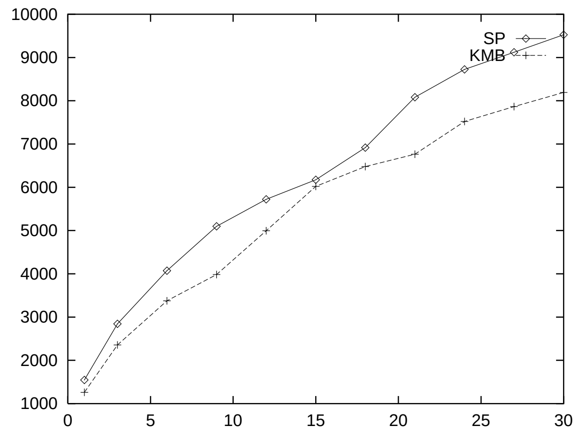

Figure 4: Cost versus asymmetry K for 27 participants.

graph by distributing points in a cartesian grid. Every point corresponds to a node, and the edges are introduced by considering all possible node pairs and favoring probabilistically the pairs whose euclidean distance is small. A detailed description of the creation of such graphs can be found in a work by Doar and Leslie [DL93]. We choose the weights of an edge to be a uniformly distributed number. Finally, in order to make the graph directed, we transform every edges to two opposite (directed) edges. We vary the weights so that the ratio of every pair of opposite edges is equal to the asymmetry constant K. The participants are randomly selected. For the following experiments a graph of 60 nodes was used.

Since the accurate solution is hard to computate, we thought of comparing KMB with Shortest Paths which represents a naive solution. We can see that as the multicast size increases the superiority of the performance of KMB over that of Shortest Paths in both the undirected (Fig.2) and the directed case (Fig.3). However, in the directed case the increase is more dramatic. Finally, Figure 4 displays the tree cost versus the asymmetry factor for 15 participants. Here again, the asymmetry seems to increase the difference in the performance of the two algorithms.

We can conclude that our experiments suggest that, although the worst-case performance of the algorithms seems to converge with the increase of the asymmetry, in practice, their performances diverge in favor of the greedy algorithm.

4 Conclusions

In this paper, we study the problem of multicasting on directed graphs. We define the worst-case inefficiency ratio (WCI) for the comparison of approximation algorithms. We also introduce a metric for the asymmetry of the edges between nodes that we call $K(G)$, which is defined as the maximum ratio between the directed-edge weights between a pair of nodes. Recall that it is highly improbable [Ram96] that there exists a constant-factor approximation algorithm with polynomial runtime complexity. We extend two greedy approximation algorithms of the undirected case to the directed one. Although, in the former, the WCI of the algorithms is bounded by constant (2), in the latter the WCI depends heavily on the asymmetry of the graph. Namely, WCI can be linear to the asymmetry factor, and we show that this bound is tight. Furthermore, we prove that the WCI of the greedy algorithms can never be worse than the Shortest Paths algorithm. With this observation, we can improve the upper bound of WCI of the greedy algorithms. Finally we conclude that, for highly asymmetric graphs, the greedy and the Shortest Paths algorithms have similar worst-case performance. On the other hand, our experimental results indicate that the increase in the asymmetry seems to favour the greedy algorithms when compared with the Shortest Paths. Finally, we would like to stress that the experimental results shown here are an indication of the effect of the asymmetry on the performance of the algorithms. As future work, we plan to do an extended study of the directed problem on various graph families, traffic patterns, and multicast scenarios.

ACKNOWLEDGMENTS. We would like to thank Mr. Tat Man Leo Lau for conducting the experiments.

References

[DL93] Matthew Doar and Ian Leslie. How bad is naive multicast routing? *Proc. IEEE INFOCOM*, pages 82–89, 1993.

[DW72] S.E. Dreyfus and R.A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.

[KMB81] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica, Springer-Verlag*, 15:141–145, 1981.

[Meh88] Kurt Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, pages 125–128, 1988.

[Ram96] S. Ramanathan. An algorithms for multicast tree generation in networks with asummetric links. *Proc. IEEE INFOCOM '96*, pages 337–344, 1996.

[RSC86] V. J. Rayward-Smith and A. Clare. On finding steiner vertices. *Networks*, 16:283–294, 1986.

[TM80] H. Takahashi and A. Matsuyama. An approxiamte solution for the Steiner problem in graphs. *Math. Japonica*, 24(6):573–577, 1980.

[Win87] P. Winter. Steiner problem in networks: a survey. *Networks*, 17:129–167, 1987.