# Playing Push vs Pull: Models and Algorithms for Disseminating Dynamic Data in Networks*

R. C. Chakinala
Dept. of Comp. Sci. & Engr.
IIT Madras
ravich@cse.iitm.ernet.in

A. Kumarasubramanian
Dept. of Elec. Eng.
IIT Madras
abishek@iitm.ac.in

K. A. Laing
Dept. of Comp. Sci.
Tufts University
laing@cs.tufts.edu

R. Manokaran
Dept. of Comp. Sci. & Engr.
IIT Madras
rajsekar@cse.iitm.ernet.in

C. Pandu Rangan
Dept. of Comp. Sci. & Engr.
IIT Madras
rangan@iitm.ernet.in

R. Rajaraman
College of Comp. & Info. Sci.
Northeastern University
rraj@ccs.neu.edu

## ABSTRACT

Consider a network in which a collection of source nodes maintain and periodically update data objects for a collection of sink nodes, each of which periodically accesses the data originating from some specified subset of the source nodes. We consider the task of efficiently relaying the dynamically changing data objects to the sinks from their sources of interest. Our focus is on the following "push-pull" approach for this data dissemination problem. Whenever a data object is updated, its source relays the update to a designated subset of nodes, its push set; similarly, whenever a sink requires an update, it propagates its query to a designated subset of nodes, its pull set. The push and pull sets need to be chosen such that every pull set of a sink intersects the push sets of all its sources of interest. We study the problem of choosing push sets and pull sets to minimize total global communication while satisfying all communication requirements.

We formulate and study several variants of the above data dissemination problem, that take into account different paradigms for routing between sources (resp., sinks) and their push sets (resp., pull sets) – multicast, unicast, and controlled broadcast – as well as the aggregability of the data objects. Under the multicast model, we present an optimal polynomial time algorithm for tree networks, which yields a randomized $O(\log n)$-approximation algorithm for $n$-node general networks, for which the problem is hard to approximate within a constant factor. Under the unicast model, we present a randomized $O(\log n)$-approximation algorithm for non-metric costs and a matching hardness result. For metric costs, we present an $O(1)$-approximation and matching hardness result for the case where the interests of any two sinks are either disjoint or identical. Finally, under the controlled broadcast model, we present optimal polynomial-time algorithms.

While our optimization problems have been formulated in the context of data communication in networks, our problems also have applications to network design and multicommodity facility location and are of independent interest.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network Communications, Network Topology*; G.2.2 [**Discrete Mathematics**]: Graph Theory— *Graph Algorithms, Network Problems, Trees.*

## General Terms

Algorithms, Performance, Theory.

## Keywords

Network Design, Push & Pull, Data Dissemination, Multicast Tree, Approximation Algorithms, NP-Completeness

## 1. INTRODUCTION

Consider a network in which a collection of source nodes maintain and update data objects that are periodically accessed by a set of sink nodes, each of which is interested in the data originating from some specified subset of the source nodes. Moreover, suppose that they always require the latest data from all their sources of interest at the same time in order to construct a current and complete view of all the interesting data within the network. Such scenarios arise in diverse network applications including event notification in sensor networks and publish-subscribe systems [17, 20]. For instance, in an environmental monitoring system, several geographically distributed sensors (sources) continually measure local conditions and a distributed collection of network monitors (sinks) periodically require a picture of the environment sensed by the network. Similarly, in a

*Chakinala, Kumarasubramanian, and Manokaran were partially supported by a generous gift from Northeastern University alumnus Madhav Anand. Rajaraman was partially supported by NSF grant IIS-0330201. Laing was partially by the Mellon Foundation and the Faculty Research Awards Committee of Tufts University, while visiting Northeastern University.


*SPAA'06,* July 30–August 2, 2006, Cambridge, Massachusetts, USA.

publish-subscribe system, each subscriber (sink) may specify a rate for receiving updates from an arbitrary subset of the publishers (sources), each of which may be generating new information at an arbitrary rate.

A natural problem in the above scenario is to design an efficient mechanism for relaying the dynamically changing data from the sources to the sinks while satisfying all communication requirements. One approach is for the source nodes to always update all their interested sinks as soon as a data object is updated. This is clearly very inefficient if the source updates are much more frequent than the sink queries. Similarly, an approach based purely on the sinks querying the sources whenever they need an update is wasteful when the sink queries are highly frequent. In fact, even a middle-ground approach in which sources with low update frequencies update their interested sinks while the remaining sources receive queries can be seen to be inefficient since it does not exploit any network locality characteristics of the accesses.

In this paper, we study a *push-pull* approach toward disseminating dynamic data in networks. Whenever a data object is updated, its source informs a designated subset of nodes, referred to as its *push set*. Similarly, when an updated view of the network is required at a sink, it queries a designated subset of nodes, its *pull set*. The push and pull sets need to be chosen such that every pull set of a sink must intersect the push sets of all its sources of interest. Three kinds of communication costs are incurred: the data propagation cost from the sources to their push sets, the query propagation cost from the sinks to their pull sets, and the response cost carrying the updates from the nodes at the intersection of the push and pull sets to the sinks. We consider the question of how to choose push sets and pull sets to minimize total global communication while satisfying all communication requirements.

## 1.1 Our contributions

We formulate optimization problems that consider three different paradigms for routing data between sources and their push sets, and sinks and their pull sets: *multicast*, *unicast*, and *controlled broadcast*. In the multicast model, the network is an undirected weighted graph and a node $u$ may communicate with a set $S$ of nodes through a subtree $T$ of the graph that connects $u$ and $S$ while an incurring a cost proportional to the sum of the weights of the edges of $T$. In the unicast model, the network is a complete weighted graph over all nodes; all communication is through point-to-point unicast, with each edge weight indicating the cost of communicating between the two endpoints. Motivated by applications in ad hoc networks, we also consider a third model, the controlled broadcast, in which a source or sink communicates with its push or pull set by broadcasting to nodes within a certain distance range in the network.

We also distinguish between the case where data may be aggregated within the network without changing its size (such as computing a sum or a maximum), and the case where data simply has to be collected at the sinks without being processed within the network. This dimension affects the contribution of the response costs to the total communication cost. Section 2 presents the formal model definitions and all of the problem formulations.

- Under the multicast model, we first present an optimal polynomial-time algorithm for tree networks. An interesting characteristic of the algorithm for trees is that a globally optimal solution is obtained by computing suitable "locally optimal solutions" for each edge of the tree. Using embedding of general metrics into tree metrics, we also obtain polynomial-time randomized $O(\log n)$-approximate solutions for general graphs. Our optimization problem under the multicast model is a generalization of the minimum Steiner tree problem, and hence, MAXSNP-hard for general graphs. These results are described in Section 3.

- Under the unicast model, we first present a polynomial-time randomized $O(\log n)$-approximation for arbitrary cost functions and an asymptotically-matching hardness of approximation result. We then present an $O(1)$-approximation for metric cost functions in the special case where the interest sets of any two sinks are either disjoint or identical, and a matching hardness result, via a reduction from the metric uncapacitated facility location problem. Both of the approximation algorithms are based on rounding linear programming relaxations of the problem. These results are described in Section 4.

- We show that the data dissemination problem under the controlled broadcast model can be solved optimally in polynomial time by proving that all the vertices of the polytope of a particular linear programming formulation are integral. By studying the dual of this linear program, we develop an optimal polynomial-time combinatorial algorithm using maximum flows. These results are described in Section 5.

## 1.2 Related Work

To the best of our knowledge, the optimization problems solved in this paper have not been studied earlier. The most closely related work is the recent study on event notification in wireless sensor networks by Liu et al. [17] and the FeedTree project [20]. Liu et al. considered regular two-dimensional grids and planar Euclidean networks, in which sensors generate events periodically at a frequency that is identical for all sensors, while information sinks issue queries at a frequency identical for all sinks. The focus of their study is a particular class of communication mechanisms, whereby the source sensors propagate their updates and the sinks propagate their queries along specified geometric patterns on the Euclidean plane. From a systems perspective, the *FeedTree* project considers the problem of efficient distribution of RSS feeds using peer-to-peer multicast [20]. Compared to [17], our problem formulations are much more general with regard to the underlying networks, the dynamics of the data, as well as the permitted communication structures. Our work begins to build a valuable theoretical foundation for projects like *Feedtree* [20].

While our optimization problems have been formulated in the context of data communication in networks, our problems also have applications to network design and multicommodity facility location. In particular, the minimum-cost 2-spanner problem studied by Dodis and Khanna [6] and Kortsarz and Peleg [14, 15], is a special case of our data dissemination problem under the unicast model for non-metric cost functions. Our algorithm of Section 4 is an alternative $O(\log n)$-approximation algorithm for the problem, matching the bound of [6]. The optimization problem in the multicast model is a generalization of the classic minimum Steiner tree problem [8] and is related to the generalized Steiner network [24] and group Steiner problems [19]. For instance, if the push sets of the sources are fixed, then the problem reduces to solving an instance of the group Steiner problem for each sink.

Under the unicast model, the data dissemination problem is a generalization of the well-studied uncapacitated facility location problem [21, 12, 25, 22], many variants of which have also been studied [9, 23, 3, 16]. In particular, when we have a single source and many sinks, our problem reduces to uncapacitated facility location. Thus, our data dissemination problem in the unicast model can be viewed as a form of multi-commodity facility location (with sources as commodities and sinks as clients), a class of problems recently studied by Ravi and Sinha [18]. From a technical standpoint, however, there are crucial differences between the two studies: the model of [18] includes an initial installation cost at each node for placing the first facility, a cost for which our model does not have any provision. On the other hand, the service cost for a client in [18], is additive over different commodities, while our aggregation model allows for the sharing of client service costs across commodities.

## 2. PROBLEM FORMULATIONS

We consider a network in which a subset of sink nodes periodically access data objects that are stored and periodically updated at another subset of source nodes. We let $\mathcal{P}$ and $\mathcal{Q}$ denote the sets of sources and sinks, respectively. We will assume that each source node has only one data object, because multiple objects can be modelled by multiple colocated source nodes. Each source $i$ updates its data object at an average frequency of $p_i$ while each sink $j$ accesses data by issuing queries at an average frequency of $q_j$. Each sink $j$ has an *interest set* $I_j$ that is the set of all sources whose data objects $j$ would like to obtain in each query. For convenience, we assume throughout this paper that all data objects are of a uniform size. Many of our results extend to the case of non-uniform data lengths. We defer a discussion on this issue to Section 6.

We adopt the following conceptual framework for the dissemination of information between the sources and sinks. Whenever a source $i$ updates its data object, $i$ propagates this update to a *push set* $P_i$ of nodes in the network. Similarly, any query issued by a sink $j$ is propagated to a *pull set* $Q_j$ of nodes in the network. To ensure that each query issued by $j$ obtains the updates generated by all sources in $I_j$, we require that $i \in I_j \Rightarrow P_i \bigcap Q_j \neq \emptyset$. A node in the nonempty set $P_i \bigcap Q_j$ then propagates the desired information generated by $i$ to $j$.

The total communication cost for a given network, source and sink sets and their frequencies, depends on both the choice of the push and pull sets, as well as the underlying routing mechanism. In all of our problems, our goal is to minimize the following objective function which reflects the total communication cost:

$$\sum_{i \in \mathcal{P}} p_i \cdot \mathrm{SetC}(i, P_i) + \sum_{j \in \mathcal{Q}} q_j \cdot \mathrm{SetC}(j, Q_j) + \sum_{j \in \mathcal{Q}} q_j \cdot \mathrm{RespC}(j), \quad (1)$$

where $\mathrm{SetC}(i, S)$ is the communication cost between node $i$ and the set $S$ of nodes, and $\mathrm{RespC}(j)$ is the sum over all $i$ in $I_j$ of the cost of propagating query responses from a node in the intersection of $Q_j$ and $P_i$ to node $j$. We assume that $\mathrm{SetC}(i, \{i\}) = 0$, so we can assume without loss of generality that $i \in P_i$. Similarly $j \in Q_j$ for all $j \in \mathcal{Q}$ . We now elaborate on the SetC and RespC terms, beginning with the latter.

**Response cost.** We consider two response cost models. In the *aggregation* model, we assume that multiple distinct data objects can be aggregated to yield a single data object of the size of an individual object. In this model, the responses are propagated along paths that are reverse of query propagation, and aggregated as the paths meet and merge; so we set $\mathrm{RespC}(j) = \mathrm{SetC}(j, Q_j)$. In the *non-aggregation* model, we assume that data objects from two different sources cannot be aggregated, and hence contribute separately to the communication cost; so we set $\mathrm{RespC}(j) = \sum_{i \in I_j} \mathrm{MinC}(P_i \cap Q_j, j)$, where $\mathrm{MinC}(S, j)$ is the cost of the least-cost path from a node in $S$ to $j$, denoted $P(S, j)$.

**Routing mechanism.** The communication cost between a node $i$ and a set $S$ of nodes, namely $\mathrm{SetC}(i, S)$, depends on the underlying routing model. We consider three models, each based on a standard communication paradigm.

In the *multicast* model, data is routed from a node $i$ to a set $S$ on a multicast tree $T$ connecting $i$ to $S$. The network is modeled by a graph $G = (V, E)$ with a nonnegative cost $c_{uv}$ for each edge $(u, v)$. For a given multicast tree $T$, we define the function $\mathrm{cost}(T)$ as the sum of the costs of the edges of the tree. Thus, in the multicast model, we seek a collection of push- and pull-sets $(\{P_i\}, \{Q_j\})$, together with their respective multicast trees $(\{T_i\}, \{T'_j\})$, that minimize Equation 1, with $\mathrm{SetC}(i, P_i) = \mathrm{cost}(T_i)$ and $\mathrm{SetC}(j, Q_j) = \mathrm{cost}(T'_j)$; we set $\mathrm{RespC}(j)$ to be $\mathrm{cost}(T'_j)$ in the aggregation model, and $\sum_{i \in I_j} \mathrm{MinC}(P_i \cap Q_j, j)$, otherwise.

In the *unicast* model we assume that all communication is through point-to-point unicast. For each pair of nodes $u$ and $v$, we associate a nonnegative cost $d_{uv}$ of communicating between $u$ and $v$. For the unicast model, we set $\mathrm{SetC}(u, S) = \sum_{k \in S} d_{uk}$ and optimize Equation 1. Again we set $\mathrm{RespC}(j)$ equals $\mathrm{SetC}(j, Q_j)$ in the aggregation model and $\sum_{i \in I_j} \mathrm{MinC}(P_i \cap Q_j, j)$, otherwise. We separately consider the *metric* case (in which we assume that $d$ satisfies reflexivity, symmetry and the triangle inequality), and the *non-metric case* in which $d$ is reflexive and symmetric, but may not satisfy the triangle inequality.

In the *controlled broadcast* model, we assume that communication to a set of nodes is by means of controlled broadcast, whereby a node broadcasts data through the network up to a specified distance, reaching all nodes within this distance. The network is modeled by a metric distance function $d$ and a cost function $c$. The distance between two nodes $u$ and $v$ is given by $d_{uv}$, and the cost of communicating from $i$ within a distance $d$ is given by $c(i, d)$. For any $i$, we assume that $c(i, d)$ is non-decreasing with increasing $d$. For the controlled broadcast model, we seek a radius $r_i$ (resp., $r_j$) for each source $i$ (resp., sink $j$) such that setting $P_i = \{v \mid d_{iv} \leq r_i\}$ and $\mathrm{SetC}(i, P_i) = c(i, r_i)$ (resp., $Q_j = \{v \mid d_{jv} \leq r_j\}$ and $\mathrm{SetC}(j, Q_j) = c(j, r_j)$) minimizes Equation 1. Here $\mathrm{RespC}(j)$ equals $c(j, r_j)$ in the aggregation model, and $\sum_{i \in I_j} \mathrm{MinC}(P_i \cap Q_j, j)$ in the non-aggregation model.

## 3. THE MULTICAST MODEL

In this section we consider a general model which specializes to both the aggregation and the non-aggregation cases of the multicast problem. Let $x_{uvi}$ be a 0-1 variable that indicates that edge $uv$ is in $T_i$. Similarly $y_{uvj}$ indicates that $uv$ is in $T'_j$ and $z_{uvij}$ indicates that $i \in I_j$ and $uv$ is in $P(T_i \cap T'_j, j)$. Finally, we introduce a new parameter $m_{ij}$, which represents the average frequency of the responses. Our algorithms in this section work for arbitrary $m_{ij}$. For the non-aggregation case, we set $m_{ij} = q_j$, and for the aggregation case, we set $m_{ij} = 0$ and double the $q_j$ values. By choosing other values for $m_{ij}$, we can model other scenarios as well; for instance, setting $m_{ij} = min(p_i, q_j)$ models the non-aggregation scenario in which we omit responses when there is no new data. The objective function may be rewritten as follows:

$$\sum_{i \in \mathcal{P}} p_i \sum_{uv \in E} c_{uv} x_{uvi} + \sum_{j \in \mathcal{Q}} q_j \sum_{uv \in E} c_{uv} y_{uvj} + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{Q}} m_{ij} \sum_{uv \in E} c_{uv} z_{uvij}$$

In section 3.1 we present a polynomial-time exact combinatorial algorithm for trees. Then in section 3.2 we present an $O(\log n)$-approximation algorithm for general graphs, based on the technique of embedding arbitrary metrics into tree metrics.

## 3.1 An optimal polynomial algorithm for trees

When $G$ is a tree $T = (V, E)$, the distance $\mathrm{MinC}(T_i \cap T'_j, j)$ is simply the sum of edge weights on a shortest path $P(T_i, j)$ from any node in $T_i$ to $j$. First we rearrange the sums so that we sum by edges last, and then split the coefficient of the $c_{uv}$ term into two components, one for each direction of overall information flow. For any edge $uv$, let $S_{uv}$ be the largest subtree that contains $u$ and excludes $v$. Note that $S_{vu} = V \setminus S_{uv}$. We now substitute $V = S_{uv} \cup S_{vu}$ into the equation and note that when $i$ and $j$ are both in $S_{uv}$ the edge $uv$ can not possibly be on the path $P(T_i, j)$. Therefore the sum $\sum_{i \in S_{uv}} \sum_{j \in S_{uv}} m_{ij} z_{uvij}$ is zero, and similarly $\sum_{i \in S_{vu}} \sum_{j \in S_{vu}} m_{ij} z_{uvij}$ is also zero. Thus we obtain:

$$\sum_{uv \in E} c_{uv} \left[ \sum_{i \in S_{uv}} p_i x_{uvi} + \sum_{j \in S_{vu}} q_j y_{uvj} + \sum_{i \in S_{uv}} \sum_{j \in S_{vu}} m_{ij} z_{uvij} \right]$$
$$+ \sum_{uv \in E} c_{uv} \left[ \sum_{i \in S_{vu}} p_i x_{uvi} + \sum_{j \in S_{uv}} q_j y_{uvj} + \sum_{i \in S_{vu}} \sum_{j \in S_{uv}} m_{ij} z_{uvij} \right]$$

This grouping into two components for each $uv$ is quite natural because tradeoffs in the sizes of $T_i$, $T'_j$ and $P(T_i, j)$ involve $c_{uv}$ when $i \in S_{uv}$ and $j \in S_{vu}$ (or else when $i \in S_{vu}$ and $j \in S_{uv}$). We minimize the objective function by independently minimizing the coefficients of $c_{uv}$ in each of the two components above. The argument for both components is symmetric, so we present only one of them in detail. For any edge $ab$, let $P_{ab} = \{i \in S_{ab} \mid p_i > 0\}$ and similarly let $Q_{ab} = \{j \in S_{ab} \mid q_j > 0\}$. Since $p_i = 0$ for all $i \in S_{uv} \setminus P_{uv}$ and $q_j = 0$ for all $j \in S_{vu} \setminus Q_{vu}$, we can replace $S_{uv}$ and $S_{vu}$ with $P_{uv}$ and $Q_{vu}$ respectively in the coefficient of the first component of the above equation, and we obtain the coefficient:

$$\alpha_{uv} = \sum_{i \in P_{uv}} p_i x_{uvi} + \sum_{j \in Q_{vu}} q_j y_{uvj} + \sum_{i \in P_{uv}} \sum_{j \in Q_{vu}} m_{ij} z_{uvij}$$

We can informally interpret this to be the cost (per unit edge weight) of using edge $uv$ to communicate events from $P_{uv}$ to queries in $Q_{vu}$ at the given frequencies. This includes sources in $P_{uv}$ pushing data across from $u$ to $v$, queries sent by sinks in $Q_{vu}$ going across from $v$ to $u$ and responses from $S_{uv}$ into $S_{vu}$ going across from $u$ to $v$. We will show how to optimize $\alpha_{uv}$ locally for each edge and obtain a globally consistent solution. Define $X_{uv} = \{(i, j) \mid i \in P_{uv}, j \in Q_{vu} \text{ and } i \in I_j\}$. We will define a set of tokens $\{x_{ij} \mid (i, j) \in \mathcal{P} \times \mathcal{Q}\}$ for constructing graphs. For $(i, j) \in X_{uv}$, we can represent the constraints for membership of edge $uv$ in subtrees of the form $T_i$, $T'_j$ and $P(T_i, j)$ using the following bipartite graph. Define $G_{uv}$ with $P_{uv}$ as one part and $Q_{vu} \cup \{x_{ij} \mid (i, j) \in X_{uv}\}$ as the other part. The set of edges of $G_{uv}$ is defined by $E(G_{uv}) = X_{uv} \cup \{(i, x_{ij}) \mid (i, j) \in X_{uv}\}$. Finally we associate weights $p_i$, $q_j$ and $m_{ij}$ with the nodes in $i \in P_{uv}$, $j \in Q_{vu}$ and $x_{ij}$ such that $(i, j) \in X_{uv}$ respectively.

LEMMA 3.1. *For each directed edge $e = uv$, the weight of a minimum weight vertex cover of $G_{uv}$ is precisely the minimum value of $\alpha_{uv}$ in any feasible solution.*

PROOF. We show that there exists a vertex cover of $G_{uv}$ with weight $W$ if and only if there exists a feasible solution in which $\alpha_{uv} = W$. First, suppose $C$ is a vertex cover of weight $W$. For $i \in P_{uv} \cap C$, define $T_i = T[S_{uv} \cup \{v\}]$, and for $i \in P_{uv} \setminus C$, define $T_i = T[S_{uv}]$, and for all $i \in P_{uv}$, define the pull tree $T'_i = (\{i\}, \emptyset)$. For all $j \in Q_{vu} \cap C$, define $T'_j = T[S_{vu} \cup \{u\}]$ and for all $j \in Q_{vu} \setminus C$, define $T'_j = T[S_{vu}]$ and finally define source trees $T_j = T$ for all $j \in Q_{vu}$. For all $x_{ij} \in C \cap \{x_{ij} \mid (i, j) \in X_{uv}\}$, define $P(T_i, j) = P(u, j)$, and finally for all $x_{ij} \in \{x_{ij} \mid (i, j) \in X_{uv}\} \setminus C$, define $P(T_i, j) = P(v, j)$. It is easy to verify that this is a feasible solution, and that $\alpha_{uv} = W$.

Conversely suppose we have a feasible solution $T_i$, $T'_j$ and $P(T_i, j)$ for all $i$ and $j$, in which $\alpha_{uv} = W$. We can construct a corresponding vertex cover $C$ of weight $W$ as follows. If there exists $(i, j) \in X_{uv}$ such that $e \in T_i$ then we include $i$ into $C$, at a cost of $p_i$. Similarly if there exists $(i, j) \in X_{uv}$ such that $e \in T'_j$ then we include $j$ into $C$, at a cost of $q_j$, and finally, for all $(i, j) \in X_{uv}$, if $e \in P(T_i, j)$ then we include $x_{ij}$ into $C$, at a cost of $m_{ij}$.

We claim that $C$ is a vertex cover of $G_{uv}$. Consider edges in $G_{uv}$ of the form $(i, j)$ where $(i, j) \in X_{uv}$. Since the solution is feasible, $(i, j) \in T_i$, or $(i, j) \in T'_j$ (because if neither subtree contains $(i, j)$, then there exists an $(i, j) \in X_{uv}$ for which $T_i \cap T'_j = \emptyset$. So either $i \in C$ or $j \in C$. On the other hand for edges in $G_{uv}$ of the form $(i, x_{ij})$ for some $(i, j) \in X_{uv}$, then since the solution is feasible, either $(i, x_{ij}) \in T_i$, in which case $i \in C$, meaning $(i, x_{ij})$ is covered, or $(i, x_{ij}) \notin T_i$, and therefore $(i, x_{ij}) \in P(T_i, j)$ and $x_{ij} \in C$. Therefore $C$ is a cover of $G_{uv}$. □

It is well known that for bipartite graphs like $G_{uv}$, the minimum weight vertex cover can be computed in polynomial time, using maximum flow algorithms. The standard procedure is to orient all the edges in $G_{uv}$ from $A = P_{uv}$ to $B = Q_{vu} \cup \{x_{ij} \mid (i, j) \in X_{uv}\}$ and assign them infinite capacities. We then create a new supersource $s$ and edges $(s, i)$ with capacity $p_i$ for all $i \in P_{uv}$. We also create a new supersink $t$ and edges $(j, t)$ for all $j \in Q_{vu}$, each with capacity $q_j$, and finally we create edges $(x_{ij}, t)$ for all $(i, j) \in X_{uv}$, each with capacity $m_{ij}$. Call this graph $G'_{uv}$. A max flow from $s$ to $t$ can be used to determine a minimum cut $R$. From $R$, we obtain a minimum weight vertex cover $C_{uv} = (A \setminus R) \cup (B \cap R)$ for $G_{uv}$.

For each edge $uv$ there may be many different minimum-weight vertex covers. A consistent tie-breaking scheme is needed to ensure that the resulting structures $T_i$, $T'_j$ and

$P(T_i, j)$ are connected. To resolve this problem, we will first prove the following lemma, which will enable us to define a canonical minimum weight vertex cover. Let $A$ be one part of a bipartite graph $G$. We say a minimum weight vertex cover of $G$ is $A$-maximal if it maximizes its weight in $A$. The following lemma shows that $A$-maximal covers are unique and we will use this to show how to achieve a globally consistent solution by using covers for $G_{uv}$ that are $P_{uv}$-maximal. Thus every edge $uv$ will prefer to do as much pushing as possible among all the possible cheapest configurations. Let $w(A)$ be the sum of node weights of nodes in a set $A$.

Lemma 3.2. *Let $G = (A \cup B, E)$ be a node-weighted bipartite graph, let $A_1, A_2 \subseteq A$ and let $B_1, B_2 \subseteq B$. If $A_1 \cup B_1$ and $A_2 \cup B_2$ are both $A$-maximal minimum weight vertex covers, then $A_1 = A_2$ and $B_1 = B_2$.*

Proof. Let $A_1 \cup B_1$ and $A_2 \cup B_2$ (where $A_1, A_2 \subseteq A$ and $B_1, B_2 \subseteq B$) both be minimum weight vertex covers of a bipartite graph $G = (A \cup B, E)$, which are $A$-maximal. Suppose $A_1 \cup B_1 \neq A_2 \cup B_2$. Then $A_1 \neq A_2$ or $B_1 \neq B_2$, that is, $A_1 \setminus A_2 \neq \emptyset$ or $A_2 \setminus A_1 \neq \emptyset$ or $B_1 \setminus B_2 \neq \emptyset$ or $B_2 \setminus B_1 \neq \emptyset$.

If $u \in A_1 \setminus A_2$ and $N(u) \subseteq B_1$, then $(A_1 \cup B_1) \setminus \{u\}$ is a lighter cover. So $u \in A_1 \setminus A_2$ implies that $N(u) \cap (B \setminus B_1) \neq \emptyset$. Now if $v \in N(u)$ then $v \notin B \setminus (B_1 \cup B_2)$, otherwise the edge $uv$ is not covered by $A_2 \cup B_2$. So every node in $A_1 \setminus A_2$ has at least one neighbor in $B_2 \setminus B_1$ and none in $B \setminus (B_1 \cup B_2)$. By symmetry, every node in $B_1 \setminus B_2$ has at least one neighbor in $A_2 \setminus A_1$ and none in $A \setminus (A_1 \cup A_2)$.

Now $w(A_1 \cup B_1) = w(A_2 \cup B_2)$ since both vertex covers have minimum weight. Subtracting out the weights $w(A_1 \cap A_2)$ and $w(B_1 \cap B_2)$, we get that $w(A_1 \setminus A_2) + w(B_1 \setminus B_2) = w(A_2 \setminus A_1) + w(B_2 \setminus B_1)$. So $w(A_1 \setminus A_2) - w(A_2 \setminus A_1) = w(B_2 \setminus B_1) - w(B_1 \setminus B_2)$.

Now $w(A_1) = w(A_2)$ since both minimum weight vertex covers are $A$-maximal. So $w(A_1 \setminus A_2) - w(A_2 \setminus A_1) = 0$, and therefore we also get that $w(B_2 \setminus B_1) - w(B_1 \setminus B_2) = 0$. That is, $w(A_1 \setminus A_2) = w(A_2 \setminus A_1)$ and $w(B_1 \setminus B_2) = w(B_2 \setminus B_1)$.

First suppose $w(B_2 \setminus B_1) < w(A_2 \setminus A_1)$. Then, starting with $A_1 \cup B_1$, we can exclude $A_1 \setminus A_2$, thus uncovering some edges with endpoints in $A_1 \setminus A_2$ and $B_2 \setminus B_1$. This can be corrected by including $B_2 \setminus B_1$, so the set $(A_1 \cap A_2) \cup (B_1 \cup B_2)$ is also a vertex cover. Moreover, since $w(B_2 \setminus B_1) < w(A_2 \setminus A_1)$, the new vertex cover is strictly lighter, contradicting the choice of $A_1 \cup B_1$ as a minimum weight cover.

Next, suppose $w(B_2 \setminus B_1) > w(A_2 \setminus A_1)$. This implies $w(B_1 \setminus B_2) > w(A_2 \setminus A_1)$. Starting with $A_1 \cup B_1$, we can exclude $B_1 \setminus B_2$ and include $A_2 \setminus A_1$, and obtain another strictly lighter vertex cover $(A_1 \cup A_2) \cup (B_1 \cap B_2)$, a contradiction.

So it must be the case that $w(B_2 \setminus B_1) = w(A_2 \setminus A_1)$. If both of these quantities are nonzero then we can start with $A_1 \cup B_1$, exclude $B_1 \setminus B_2$ and include $A_2 \setminus A_1$, and obtain another vertex cover $(A_1 \cup A_2) \cup (B_1 \cap B_2)$, which is the same weight, but which has strictly more weight in $A$ than before, another contradiction. So it can only be the case that $w(A_1 \setminus A_2) = w(A_2 \setminus A_1) = w(B_1 \setminus B_2) = w(B_2 \setminus B_1) = 0$. Thus $A_1 = A_2$ and $B_1 = B_2$. $\square$

Next, we generalize the notion that if an edge is a push edge, then it must be a push edge all the way back to the source, and if it is a pull edge it must be a pull edge all the way to the sink, and finally if it is a response edge, it must be a response edge all the way to the sink. For that we require the following lemma:

Lemma 3.3. *Let $A$, $A'$, $B$, $B'$ and $B''$ be five sets of nodes with assigned nonzero node weights[1], for which the two families $\{A, A', B, B'\}$ and $\{A, B, B''\}$ are each pairwise disjoint. Let $A_1, A_2 \subseteq A$, $B_1, B_2 \subseteq B$, $A_3 \subseteq A'$, $B_3 \subseteq B'$ and $B_4 \subseteq B''$. Let $E_0$ be a set of edges between $A$ and $B$, $E_1$ be a set of edges between $A'$ and $B$, $E_2$ be a set of edges between $A$ and $B''$, and $E_3$ be a set of edges between $A'$ and $B'$. Let $G_1$ be a bipartite graph with parts $A \cup A'$ and $B \cup B'$, and edges $E_0 \cup E_1 \cup E_3$. Let $G_2$ be a bipartite graph with parts $A$ and $B \cup B''$, and edges $E_0 \cup E_2$. If $A_1 \cup B_1 \cup A_3 \cup B_3$ is the $(A \cup A')$-maximal minimum weight vertex cover of $G_1$, and $A_2 \cup B_2 \cup B_4$, is the $A$-maximal minimum weight vertex cover of $G_2$, then $A_1 \subseteq A_2$ and $B_1 \supseteq B_2$.*

Proof. Let $A_1 \cup B_1 \cup A_3 \cup B_3$ be $(A \cup A')$-maximal in $G_1$, and let $A_2 \cup B_2 \cup B_4$ be $A$-maximal for $G_2$. Every node $u$ in $A_1 \setminus A_2$ has to have at least one neighbor $v$ outside $B_1$. If not, then $N(u) \subseteq B_1$ in $G_1$, so we can drop $u$ from the cover $A_1 \cup B_1 \cup A_3 \cup B_3$ and obtain a lighter cover for $G_1$. Moreover, $v \in N(u)$ in $G_1$ implies that $v \notin B \setminus (B_1 \cup B_2)$, otherwise $uv$ is not covered by $A_2 \cup B_2 \cup B_4$ in $G_2$. Thus, every node in $A_1 \setminus A_2$ has at least one neighbor in $B_2 \setminus B_1$ and none in $B \setminus (B_1 \cup B_2)$. Similarly we can show that every node in $B_2 \setminus B_1$ has at least one neighbor in $A_1 \setminus A_2$ and none in $A \setminus (A_1 \cup A_2)$.

If $w(A_1 \setminus A_2) < w(B_2 \setminus B_1)$, then we can obtain a lighter cover for $G_2$ by starting with $A_2 \cup B_2 \cup B_4$, excluding $B_2 \setminus B_1$, and including $A_1 \setminus A_2$. The new cover is $(A_1 \cup A_2) \cup (B_1 \cap B_2) \cup B_4$. Note that it covers $G_2$ and is lighter, a contradiction.

Similarly, if $w(A_1 \setminus A_2) > w(B_2 \setminus B_1)$, we can obtain a lighter cover, but for $G_1$ this time, by starting with $A_1 \cup B_1 \cup A_3 \cup B_3$, excluding $A_1 \setminus A_2$, and including $B_2 \setminus B_1$. The new cover is $(A_1 \cap A_2) \cup (B_1 \cup B_2) \cup A_3 \cup B_3$. Note that it covers $G_1$ and is lighter, a contradiction.

Now suppose $w(A_1 \setminus A_2) = w(B_2 \setminus B_1) \neq 0$. Then there exists a cover for $G_2$ with strictly more weight in $A$. Starting with $A_2 \cup B_2 \cup B_4$, exclude $B_2 \setminus B_1$, and include $A_1 \setminus A_2$. The newly formed cover is $(A_1 \cup A_2) \cup (B_1 \cap B_2) \cup B_4$. Note that it covers $G_2$ and while it is the same weight, it has a higher amount of weight in $A$. This is again a contradiction.

Therefore we are left with the case $w(A_1 \setminus A_2) = w(B_2 \setminus B_1) = 0$, from which it follows that $A_1 \subseteq A_2$ and $B_1 \supseteq B_2$ as required, since all node weights are nonzero. $\square$

Algorithm 3.4. *For each directed edge $uv$ (ie. consider each undirected edge in both directions separately), construct the graph $G_{uv}$, and find its canonical minimum cut $C_{uv}$. For all $i \in P_{uv}$, if $i \in C_{uv}$ then include $uv$ in $T_i$. For all $j \in Q_{vu}$ if $j \in C_{uv}$ then include $uv$ in $T'_j$. Finally, for all $(i, j) \in X_{uv}$ if $x_{ij} \in C_{uv}$ then include $uv$ in $P(T_i, j)$.*

The correctness of this algorithm is proven in Lemma 3.6 below, by applying Lemma 3.3 with the following definition:

Definition 3.5. *Let $Y \otimes Z = \{x_{ij} \mid i \in Y, j \in Z, i \in I_j\}$, and let $\otimes$ have lower precedence than $\setminus$. Let $u, v, w$ be three consecutive nodes on any path. We define $G_1 = G_{vw}$, $G_2 = G_{uv}$ and the following:*

[1]Nodes with zero weights for either push or pull frequency are assumed to not generate or consume any data, and therefore no connections are required from or to these nodes.

$$\begin{array}{ll} A = P_{uv} & B = Q_{wv} \cup (P_{uv} \otimes Q_{wv}) \\ A' = P_{vw} \setminus P_{uv} & B' = P_{vw} \setminus P_{uv} \otimes Q_{wv} \\ A_1 = C_{vw} \cap P_{uv} & B'' = Q_{vu} \setminus Q_{wv} \cup (P_{uv} \otimes Q_{vu} \setminus Q_{wv}) \\ A_2 = C_{uv} \cap P_{uv} & B_1 = C_{vw} \cap (Q_{wv} \cup (P_{uv} \otimes Q_{wv})) \\ A_3 = C_{vw} \cap P_{vw} \setminus P_{uv} & B_2 = C_{uv} \cap (Q_{wv} \cup (P_{uv} \otimes Q_{wv})) \\ & B_3 = C_{vw} \cap (P_{vw} \setminus P_{uv} \otimes Q_{wv}) \\ & B_4 = C_{uv} \cap (Q_{vu} \setminus Q_{wv} \cup (P_{uv} \otimes Q_{vu} \setminus Q_{wv})) \end{array}$$

LEMMA 3.6. *The edge sets* $\{T_i \mid i \in V\}$, $\{T'_j \mid j \in V\}$ *and* $\{P(T_{i,j}) \mid i, j \in V\}$ *resulting from Algorithm 3.4 are all connected, and can correctly be called subtrees and paths.*

PROOF. To show that the edge set $T_i$ is connected for each $i \in \mathcal{P}$, it suffices to show that if $i \in P_{vw} \cap C_{vw}$ then for all $rs \in P(i, v)$, $i \in C_{rs}$. That in turn follows by induction if we can show that if $i \in P_{vw} \cap C_{vw}$ and $i \neq v$ and $uv$ is the next edge on a path from $v$ back to $i$, then $i \in C_{uv}$. Suppose $i \in P_{vw} \cap C_{vw}$. Since $i \in P_{vw}$ and $uv \in P(i, v)$, it follows that $i \in P_{uv}$. Also it follows from $i \in C_{vw}$ that $i \in A_1$, by definition 3.5. Lemma 3.3 gives us that $A_1 \subseteq A_2$, and since $A_2 \subseteq C_{uv}$, it follows that $i \in C_{uv}$.

Similarly we show that the edge set $T'_j$ is also connected for each $j \in \mathcal{Q}$, by showing that if $j \in Q_{vu} \cap C_{uv}$ then for all $rs \in P(v, j)$, $j \in C_{rs}$. The inductive step in this case is to show that if $j \in Q_{vu} \cap C_{uv}$ and $j \neq v$ and $vw$ is the next edge on a path from $v$ to $j$, then $j \in C_{vw}$. Suppose $j \in Q_{vu} \cap C_{uv}$. Since $j \in Q_{vu}$ and $vw \in P(v, j)$, it follows that $j \in Q_{wv}$. Also, $j \in C_{uv}$ implies $j \in B_2$. So by Lemma 3.3, $j \in B_2 \subseteq B_1 \subseteq C_{vw}$.

Finally we show that for every pair $(i, j) \in \mathcal{P} \times \mathcal{Q}$, the edge set $P(T_i, j)$ is connected, by showing that if $x_{ij} \in X_{uv} \cap C_{uv}$ for some edge $uv$, then for all $rs \in P(v, j)$, $x_{ij} \in C_{rs}$. Again the main step is to show that if $x_{ij} \in X_{uv} \cap C_{uv}$ and $j \neq v$ and $vw$ is the next edge on the path from $v$ to $j$, then $x_{ij} \in C_{vw}$. Suppose $x_{ij} \in X_{uv} \cap C_{uv}$. First, $x_{ij} \in X_{uv}$ implies $i \in P_{uv}$ and $j \in Q_{vu}$ and $i \in I_j$. Now since $vw \in P(v, j)$, $j \in Q_{wv}$. Moreover $P_{uv} \subseteq P_{vw}$. So $x_{ij} \in X_{vw}$. Next, notice that $x_{ij} \in C_{uv}$ implies $x_{ij} \in B_2$, by definition of $B_2$. So we use Lemma 3.3 again to obtain $x_{ij} \in B_2 \subseteq B_1 \subseteq C_{vw}$. □

Having established correctness, it remains to show that Algorithm 3.4 runs in polynomial time. To establish this, we need to verify that the push-biased (that is, $P_{uv}$-maximal) minimum weight vertex cover of $G_{uv}$ can be computed in polynomial time. To this end, consider the following two lemmas (the first one is well known in the folklore so we omit the proof, but though the second lemma is simple, we are not aware of situations in which it has been used).

LEMMA 3.7. *If* $R_1$ *and* $R_2$ *are two minimum cuts of* $G'_{uv}$ *that are reachable from* $s$ *in the residual networks of maximum flows* $f_1$ *and* $f_2$ *respectively, then* $R_1 = R_2$. □

LEMMA 3.8. *If* $R$ *is the minimum cut for* $G'_{uv}$ *that is reachable from* $s$ *in the residual network of a maximum flow* $f$ *and* $C = (A \setminus R) \cup (B \cap R)$, *then* $C$ *is* $P_{uv}$*-maximal.*

PROOF. It is well-known that $C$ is a minimum weight vertex cover. It only remains to show that it is $A$-maximal. From the definition of $C$, it must be the case that $w(A \setminus R)$ is as large as possible for any minimum cut, and equivalently, that $w(A \cap R)$ is as small as possible. Suppose not. Then there exists a minimum cut $R'$ such that $w(A \cap R') < w(A \cap R)$. Now consider $R \cap R'$ which, being the intersection of two minimum cuts, is also a minimum cut. Let $f'$ be a maximum flow that saturates $R \cap R'$, and let $v \in R \setminus R'$. Clearly $v \in R$, but $v$ is not reachable in the residual network of $f'$, which contradicts Lemma 3.7. So $w(A \cap R)$ is in fact as small as possible. □

**Distributed implementation.** Our algorithm in the multicast model for trees achieves global optimality by independently solving local optimization problems for each edge of the network. This enables a simple three-phase distributed implementation of the algorithm. The first phase consists of a global exchange in which each network node learns of the interest sets and frequencies of all of the sources and sinks. The second phase is entirely local, in which each node computes the minimum vertex cover for the problem defined for each of its adjacent edges. The resulting local solutions are directly used in the final long-running phase in which the information published by the sources is continually pushed to the respective push sets and pulled by the sinks from their respective pull sets in a cost-optimal manner.

The second and third phases are self-explanatory. The first parameter exchange phase can be carried out by a communication step in which every source and sink node inform other nodes of their frequency and interest sets. The number of bits communicated in this phase is proportional to the total number of sources and sinks and the sizes of their interest sets. Since this is a one-time cost, this communication can be amortized against the cost incurred during the long-running phase. Furthermore, if the number of different frequency values and the number of different interest sets are small, the total amount of communication in the first phase can be significantly reduced. We finally note that any change in the set of sources, set of sinks, frequencies, or interest sets, can be broadcast within the network, following which each node can locally compute an updated solution.

## 3.2 Results for general graphs

We now present a randomized $O(\log n)$-approximation for general graphs in the multicast model, based on the technique of embedding arbitrary metrics into tree metrics [1, 7, 13]. In particular, we first use the 2-HST (hierarchically well-separated tree) construction of Fakcharoenphol et al [7], which $O(\log n)$-probabilistically approximates the metric $d$ over the given graph. We then apply the result of Konjevod et al.[13], who showed that any $k$-HST resulting from weak or strong probabilistic partitions can be replaced by a tree whose vertex set is that of the original graph at the cost of a constant factor in stretch. By a standard argument, we obtain a randomized $O(\log n)$-approximation for general graphs. One can also obtain a deterministic $O(\log n)$ approximation via the deterministic rounding techniques of [2, 7].

On the hardness side, the data dissemination problem is easily seen to be a generalization of the classic minimum Steiner tree problem, which is NP-hard to approximate to within a factor 96/95 [5].

**Approximation algorithm for general graphs.**

THEOREM 3.9 (KONJEVOD ET AL.[13]). *Any* $k$*-HST* $T'$ *resulting from weak or strong probabilistic partitions can be replaced by a tree* $T$ *whose vertex set is* $V(G)$, *such that* $d_G(u, v) \leq d_T(u, v) \leq 2d_{T'}(u, v)k/(k-1)$ *for any* $u, v \in V(G)$.

We apply this result to the 2-HST construction of Fakcharoenphol et al [7].

THEOREM 3.10 (FAKCHAROENPHOL ET AL.[7]). *The distribution over tree metrics resulting from (their) algorithm $O(\log n)$-probabilistically approximates the metric $d$.*

Let $(\mathcal{S}_G, \mathcal{S'}_G)$ be an optimal solution in a given graph $G$ with cost $OPT(G)$. Let $T$ be a tree (defined on the nodes of $G$) selected at random from the distribution of metric-spanning trees that $O(\log n)$-probabilistically approximates the metric $d$ of shortest distances in a graph $G$. Convert every edge $e$ in a structure of $(\mathcal{S}_G, \mathcal{S'}_G)$ into the corresponding path $P(e)$ within $T$. (By a structure we mean a push tree, a pull tree or a response path). Since two structures $T_i$ and $T'_j$ in the optimal solution of $G$ intersect in $G$, they will also intersect in $T$. Thus we obtain a family of structures in $T$ which is feasible for $T$, with cost $O(\log n) \cdot OPT(G)$. Therefore the optimal solution $OPT(T)$ of $T$ satisfies $OPT(T) \leq O(\log n) \cdot OPT(G)$.

After running the Algorithm of section 3.1 on $T$, we obtain the value $OPT(T)$ of an optimal solution $(\mathcal{S}_T, \mathcal{S'}_T)$ for $T$. We project it back into the graph $G$ as follows: every edge $e$ in a structure ($T_i$ say) of $(\mathcal{S}_T, \mathcal{S'}_T)$ is replaced by the corresponding path $P^{-1}(e)$ in $G$. Call the images of the endpoints of $e$ *essential nodes* if $e$ is in a subtree like $T_i$ or $T'_j$. Otherwise $e$ is in a response path from $u$ to $v$, say, and then let the images of *only* $u$ and $v$ be essential in $G$ for this structure. The resulting structures in $G$ are connected but may not necessarily be subtrees, so we replace each such structure with the Steiner tree of $G$ (or a 2-approximate MST approximation of the Steiner tree) or else the shortest path (in the case of a response path) that contains its essential nodes. Since structure intersections in $T$ only occur at essential points, the required intersection properties hold in $G$, and the cost of each such structure in $G$ is at most twice the cost of its precursor in $T$. Therefore we have obtained an approximate solution $ALG(G)$ which is at most twice $OPT(T)$. Therefore $ALG(G) \leq 2 \cdot OPT(T) \leq O(\log n) \cdot OPT(G)$. Thus we have shown that:

THEOREM 3.11. *There is an expected $O(\log n)$-approximation for the Multicast problem in general graphs.* □

# 4. APPROXIMATION ALGORITHMS FOR THE UNICAST MODEL

In this section, we present approximation algorithms and hardness results for the unicast model. Recall that in the unicast model, we are given distances $d_{uv}$ for every pair of nodes $u$ and $v$ in the network, and the goal is to determine the push-sets $P_i$ and pull-sets $Q_j$ that minimize the total communication cost

$$\sum_{i\in\mathcal{P}} p_i \sum_{k\in P_i} d_{ik} + \sum_{j\in\mathcal{Q}} q_j \sum_{k\in Q_j} d_{kj} + \sum_{j\in\mathcal{Q}} q_j \cdot \mathrm{RespC}(j),$$

subject to $P_i \cap Q_j \neq \emptyset$ for all $i \in I_j$. We first consider the aggregation model. In Section 4.1, we present an $O(\log n)$-approximation algorithm for the arbitrary distance functions and interest sets, using a variant of the standard randomized rounding scheme, and an asympototically matching hardness result. In Section 4.2, we present a constant-factor approximation for metric distances in the special case when the interest set for every sink is identical, and the MAXSNP-hardness for this special case. Finally, in Section 4.3, we consider the non-aggregation model where we show that the problem reduces to solving multiple instances of the uncapacitated facility location problem.

## 4.1 The general unicast model with aggregation

In this section, we present an $O(\log n)$ approximation for the information dissemination problem in the unicast model for general (non-metric) cost functions, when responses are aggregated. In the aggregation model, $\sum_{j\in\mathcal{Q}} \mathrm{RespC}(j)$ is simply $\sum_{j\in\mathcal{Q}} q_j \sum_{k\in Q_j} d_{kj}$; so we can replace the response cost by doubling the sink frequencies. Thus, we can ignore response costs without loss of generality. Our algorithm is based on rounding a linear programming relaxation for the problem. We begin by presenting an integer program for the problem.

$$\min \quad \sum_{i\in\mathcal{P}} p_i \sum_{k\in V} d_{ik}x_{ik} + \sum_{j\in\mathcal{Q}} q_j \sum_{k\in V} d_{kj}y_{kj},$$

$$\text{subject to} \quad \begin{cases} r_{ijk} \leq x_{ik} \\ r_{ijk} \leq y_{kj} \\ \sum_k r_{ijk} \geq 1 \end{cases}, \text{ where } x_{ik}, y_{kj}, r_{ijk} \in \{0,1\}.$$

Consider the linear programming (LP) relaxation obtained by replacing the $\in \{0,1\}$ constraint in the integer program by the $\geq 0$ constraint. Let $(x^*, y^*, r^*)$ denote an optimal solution to the LP. Our rounding procedure is as follows.

1. We zero out any variable whose value is at most $1/n^2$. As a result of this, the solution may no longer be feasible since the third condition of the LP may not be satisfied. However, for any pair $(i,j)$, the sum of the $r^*_{ijk}$'s that are at most $1/n^2$ is at most $1/n$; so we can scale up all of the values by a factor of $1/(1-1/n)$ and obtain a feasible solution.

2. We next round each value up to the nearest power of 1/2. Let $(\tilde{x}, \tilde{y}, \tilde{r})$ denote the resulting solution.

3. For any node $k$ and integer $p$, $0 \leq p < 2\log n$, let $X_{pk}$ (resp,. $Y_{pk}$) denote the set of nodes $i$ (resp., $j$) such that $\tilde{x}_{ik} \geq 1/2^p$ (resp., $\tilde{y}_{kj} \geq 1/2^p$).

4. We execute the following randomized rounding step independently for each node $k$ and integer $p$: with probability $\min\{(\log n)/2^p, 1\}$, add $k$ to $P_i$ for all $i$ in $X_{pk}$ and to $Q_j$ for all $j$ in $Y_{pk}$.

LEMMA 4.1. *For any pair $(i,j)$ and node $k$, the probability that $k$ lies in $P_i \cap Q_j$ is at least $\min\{1, \tilde{r}_{ijk}\log(n)\}$.*

PROOF. Follows directly from the fact that $i \in X_{\log(\tilde{r}_{ijk})k}$ and $j \in Y_{\log(\tilde{r}_{ijk})k}$. □

LEMMA 4.2. *For all $i$, $j$, and $k$, the probability that $k$ is in $P_i$ (resp., $k$ is in $Q_j$) is at most $\min\{1, 2\tilde{x}_{ik}\log n\}$ (resp., $\min\{1, 2\tilde{y}_{jk}\log n\}$).*

PROOF. The probability that $k$ is in $P_i$ is at most $\sum_{p:i\in X_{pk}} (\log n)/2^p = 2\tilde{x}_{ik}\log n$. The proof of $Q_j$ is similar. □

THEOREM 4.3. *With high probability, the solution $(\{P_i\}, \{Q_j\})$ is feasible and the cost is $O(\log n)$ times that of the optimal LP solution.*

PROOF. By Lemma 4.1, the probability that $P_i$ and $Q_j$ do not intersect for a given pair is at most $\prod_k (1 - \tilde{r}_{ijk}\log n) \leq e^{-\sum_k \tilde{r}_{ijk}\log n} \leq 1/n^2$. Let $C_i$ be the random variable denoting the cost per unit frequency for $i$. By Lemma 4.2,

$C_i$ is dominated by $\sum_k C_{ik}$, where $C_{ik}$ is the random variable that independently takes value $d_{ik}$ with probability $\min\{1, 2\tilde{x}_{ik}\log n\}$. By standard Chernoff-Hoeffding bounds [4, 11] with high probability, $\sum_k C_{ik}$ is $O(\log n) \cdot \sum_k d_{ik}\tilde{x}_{ik}$, which is $O(\log n)$ times the cost per unit frequency for $i$ in the optimal LP cost. Adding over all of the at most $n$ sources and sinks yields the desired bound on the cost with high probability. □

**Hardness of approximation.** We establish an asymptotically matching bound on the hardness of approximation by reduction from the minimum-cost 2-spanner problem, which is NP-hard to approximate to within an $O(\log n)$-factor on $n$-node graphs in polynomial time [14]. In the minimum-cost 2-spanner problem, we are given an undirected graph $G$ with edge costs and seek a spanning subgraph of $G$ of minimum total cost such that the distance between any two vertices in the subgraph is at most twice that in $G$. This problem reduces to the data dissemination problem by placing a source and sink at each node of $G$, setting the interest set of each sink to be the sources at its neighbors in $G$, the distance between two adjacent nodes to be 1 and non-adjacent nodes to be $\infty$, and all frequencies to be unit. Given any solution to the 2-spanner problem, we set the push set (resp., pull set) to be the set of neighbors of the node in the solution subgraph; the resulting cost is twice the cost of the subgraph. On the other hand, any solution of push and pull sets for the data dissemination problem yields a 2-spanner of at most half the cost. We defer the proof details to the full paper.

## 4.2 The metric unicast model with uniform interest sets and aggregation

In this section, we assume that the interest sets of all the sinks are identical. It is easy to see that any result for this special case extends to the case where the interest sets of any two sinks are either disjoint or identical. Our approximation algorithm is based on a different, deterministic, rounding of the linear program relaxation presented in Section 4.1. Our rounding procedure is based on the filtering technique of [16], as has also been used for other facility location problems [21, 18]. As in Section 4.1, let $(x^*, y^*, r^*)$ denote an optimal solution to the LP. For any node $u$ and real $r$, we define the *ball of radius $r$ around $u$*, $B_u(r)$, as the set $\{v : d_{uv} \le r\}$. Let $C_i$ denote the term $\sum_k d_{ik}x^*_{ik}$, that is, the push cost of node $i$ in the LP solution. Similarly we define the pull cost $C'_j = \sum_k d_{kj}y^*_{kj}$ for each node $j$.

LEMMA 4.4. *For any $i \in \mathcal{P}$, $j \in \mathcal{Q}$, and $\alpha > 1$, we have* $\sum_{k \notin B_i(\alpha C_i)} x^*_{ik} \le 1/\alpha$, *and* $\sum_{k \notin B_j(\alpha C'_j)} y^*_{kj} \le 1/\alpha$. □

We compute two sets $S$ and $S'$ from $\mathcal{P}$ and $\mathcal{Q}$ and the linear program solution. Set $S$ is obtained by going through the sources in non-decreasing order of their $C_i$ values. We initially set $S$ to $\emptyset$. Let $\beta > 1$. When we consider source $i$, we add $i$ to $S$, iff there does not exist $\ell \in S$ such that $d_{i\ell} \le \beta(C_i + C_\ell)$. We similarly compute $S'$.

We now compute an $O(1)$-approximate feasible solution as follows. For each $i$ in $\mathcal{P}$, let $\ell_i$ be an arbitrary point in $S$ such that $C_{\ell_i} \le C_i$ and $d_{i\ell} \le \beta(C_i + C_\ell)$; note that if $i \in S$, then $\ell_i$ is simply $i$, while it is well-defined for $i \notin S$ by our construction of $S$ above. We similarly define $\ell'_j$ for each $j$ in $\mathcal{Q}$. For each $i \in \mathcal{P}$, we set $P_i = \{i\} \cup \{\ell_i\} \cup \{j : j \in S' \text{ and } C'_j \le C_i\}$, and for each $j$ in $\mathcal{Q}$, we set $Q_j = \{j\} \cup \{\ell'_j\} \cup \{i : i \in S \text{ and } C_i < C'_j\}$.

LEMMA 4.5. *For each $i \in \mathcal{P}$ and $j \in \mathcal{Q}$, $P_i \cap Q_j \ne \emptyset$.* □

THEOREM 4.6. *For each $i \in \mathcal{P}$ and $j \in \mathcal{Q}$, we have* $\sum_{k \in P_i} d_{ik} \le O(C_i)$, *and* $\sum_{k \in Q_j} d_{kj} \le O(C'_j)$.

PROOF. Consider the elements of the set $P_i$. These include $i$, $\ell_i$, and all $j \in S'$ such that $C'_j \le C_i$. We have $d_{ii} = 0$ and $d_{i\ell_i} \le \beta(C_i + C_{\ell_i}) \le 2\beta C_i$. Consider the balls of radius $\beta C'_j$ around $j$, for all $j$ in $S'$. By our construction, these balls $B_j(\beta C'_j)$ are all disjoint. Therefore, at most one of these balls contains $i$. Let $S_i$ denote the set of $j \in S'$ such that $C'_j \le C_i$ and $i \notin B_j(\beta C'_j)$. For every $j \in S_i$, consider the ball $B_j(\alpha C'_j)$ contained within $B_j(\beta C'_j)$, where $\alpha < \beta$. By Lemma 4.4, we obtain that $\sum_{k \in B_j(\alpha C'_j)} r^*_{ijk} \ge 1 - 1/\alpha$. Since these balls are all disjoint and none of them contains $i$, we obtain that

$$C_i \ \ge \ \sum_{j \in S_i} (d_{ij} - \alpha C'_j) \sum_{k \in B_j(\alpha C'_j)} r^*_{ijk} \ \ge \ \sum_{j \in S_i} d_{ij}\left[1 - \frac{\alpha}{\beta}\right]\left[1 - \frac{1}{\alpha}\right]$$
$$= \ \frac{(\beta - \alpha)(\alpha - 1)}{\alpha\beta} \sum_{j \in S_i} d_{ij}.$$

In cases where $i \in B_j(\beta C'_j)$ for some $j$, $d_{ij} \le \beta C'_j \le \beta C_i$. Thus, recalling that $d_{i\ell_i} \le 2\beta C_i$, we can write:

$$\sum_{k \in P_i} d_{ik} \ \le \ (3\beta + \frac{\alpha\beta}{(\beta - \alpha)(\alpha - 1)})C_i.$$

This completes the proof of the first inequality. The proof of the second is analogous. If we set $\alpha = 1.69$ and $\beta = 2.86$, we obtain a 14.57-approximation. □

We next show that the data dissemination problem in the metric unicast model is NP-hard to approximate to within a constant factor, even in the special case of uniform interest sets. Our hardness result is via an approximation-preserving reduction from the facility location problem.

Consider an instance of the uncapacitated facility location problem over a set $D$ of $n$ points with metric distances given by function $d$, with unit facility cost and unit demand at each node. The class of such instances is known to be NP-hard and cannot be approximated in polynomial-time to within a factor of 1.278 unless $NP \subset DTIME(n^{\log\log n})$ [10]. We reduce this problem to the metric unicast problem with uniform interest sets as follows. The set of nodes is $V = D \cup \{t\}$, where $t$ is a new node at distance $1/\epsilon$ from every node in $D$, for some $\epsilon < 1/\max_{u,v \in D} d_{uv}$. Clearly this defines a metric space. We let $D$ be the set of sources, each with frequency equal to its demand, and $t$ be the lone sink with frequency $\epsilon$.

Consider any optimal solution to the data dissemination problem. Since there is only one sink, the push sets are all singleton sets. Let $S$ be the union of all push sets. If $S$ contains $t$ and has size at least two, then we can replace $t$ in any push set by any node in $S$ and reduce the total cost. Furthermore, if $S$ equals $\{t\}$, then the total cost is greater than the case when $S$ is the singleton set containing any source node. Thus, we can assume that $S$ does not contain $t$. In this case, the total communication cost is exactly equal to the cost incurred when opening facilities at nodes in $S$ in the facility location problem, thus completing the reduction. To make the reduction approximation preserving, if we set $\epsilon < 1/(\alpha \sum_{u,v \in D} d_{uv})$, we obtain that an $\alpha$-approximate solution to the data dissemination problem yields an $\alpha$-approximate solution to the facility location problem, thus yielding the desired hardness result.

### 4.3 Response cost without aggregation

In the non-aggregation model for response costs, RespC$(j)$ equals $\sum_{i \in I_j} d(P_i \bigcap Q_j, j)$, where $d(S, j)$ is the distance to j from the node in $S$ that is nearest to $j$. Since we can assume without loss of generality that $Q_j \subseteq \bigcup_{i \in I_j} P_i$, it follows that the total response cost is thus at least the query propagation cost. It is easy to see that the problem of minimizing the sum of the push costs and response costs can be reduced to $|\mathcal{P}|$ instances of the uncapacitated facility location problem, one for every source node. Thus, constant-factor approximations for the facility location problem yield constant-factor approximations in the non-aggregation model. We defer a discussion of the best approximation factors obtained to the full paper.

## 5. CONTROLLED BROADCAST MODEL

In this section, we provide a polynomial-time optimal algorithm for the controlled broadcast model, for each of the two response cost models. Throughout this section, we assume for simplicity that the metric distances $d_{ij}$ are induced by an undirected unweighted graph. Our results can be extended to the case of weighted graphs; we defer this discussion to the full paper.

In the controlled broadcast problems, we seek $r_i$ and $r_j$ for each $i \in \mathcal{P}$, $j \in \mathcal{Q}$, such that that $r_i + r_j \geq d_{ij}$ for all $i \in I_j$. When the responses are aggregated, then the objective is to minimize $\sum_{i \in X} p_i c(i, r_i) + 2 \sum_{j \in Y} q_j c(j, r_j)$. When there is no aggregation, then the objective is to minimize the total cost $\sum_{i \in X} p_i c(i, r_i) + \sum_{j \in Y} q_j c(j, r_j) + \sum_{i \in I_j} \mathrm{MinC}(P_i \cap Q_j, j)$.

We formulate the problems as integer linear programs as follows. Let $M = \max_{i,j}\{d_{ij}\}$. Note that $M < n$. For each $i \in \mathcal{P}$ define 0-1 variables $x_{i1}, x_{i2}, \ldots x_{iM}$ and similarly for each $j \in \mathcal{Q}$, define variables $y_{j1}, y_{j2}, \ldots y_{jM}$. The relation between these variables and the variables $r_i, r_j$ is given by the constraints $r_i \geq k \Leftrightarrow x_{ik} = 1$ and $r_j \geq l \Leftrightarrow y_{jl} = 1$. In the aggregation model, we have the following linear constraints:

$$\begin{aligned} x_{ik} + y_{jl} \geq 1 & \quad \forall e = (i, j) \in E;\ \forall k, l \geq 1 : k + l = d_{ij} + 1 \\ x_{ik} \leq x_{i(k-1)} & \quad \forall i \in \mathcal{P};\ 1 < k \leq M \\ y_{jl} \leq y_{j(l-1)} & \quad \forall j \in \mathcal{Q};\ 1 < l \leq M. \end{aligned} \tag{2}$$

In order to capture the response cost in the non-aggregation model, we introduce a third set of variables $z_{ijl}$, for every $j$, $i \in I_j$, and $k$, which indicates whether $i$'s data item is propagated to $j$ by a node that is at least distance $\ell$ away. We have the following additional constraints:

$$\begin{aligned} x_{ik} + z_{ijl} \geq 1 & \quad \forall e = (i, j) \in E;\ \forall k, l \geq 1 : k + l = d_{ij} + 1 \\ z_{ijl} \leq y_{jl} & \quad \forall j \in \mathcal{Q}, i \in I_j; 1 \leq l \leq M \\ z_{ijl} \leq z_{ij(l-1)} & \quad \forall j \in \mathcal{Q}, i \in I_j;\ 1 < l \leq M. \end{aligned} \tag{3}$$

In the aggregation model, we seek to minimize the linear objective $\sum_{ik} \alpha_{ik} x_{ik} + 2 \sum_{jl} \beta_{jl} y_{jl}$, where $\alpha_{ik} = p_i(c(i,k) - c(i, k-1))$ and $\beta_{jl} = q_j(c(j,l) - c(j, l-1))$. In the non-aggregation model, we seek to minimize $\sum_{ik} \alpha_{ik} x_{ik} + \sum_{jl} \beta_{jl} y_{jl} + \sum_{ijl} q_j z_{ijl}$.

It is easy to see that both of the above linear programs are, in fact, special cases of the following linear program over variable sets $U = \{u_1, \ldots, u_p\}$ and $V = \{v_1, \ldots, v_q\}$, with $E_1$, $E_2$, and $E_3$ as arbitrary subsets of $U \times V$, $U^2$, and $V^2$, respectively

$$\begin{aligned} \min & \quad \sum_{u \in U} f_u \cdot u + \sum_{v \in V} g_v \cdot v, \\ \text{subject to} & \quad \begin{cases} u + v \geq 1 & \forall (u, v) \in E_1 \\ u_1 \geq u_2 & \forall (u_1, u_2) \in E_2 \\ v_1 \geq v_2 & \forall (v_1, v_2) \in E_3. \end{cases} \end{aligned} \tag{4}$$

THEOREM 5.1. *All the vertices of the polytope of the linear program 4 are integral; hence, an optimal integral solution can be found in polynomial time.*

PROOF. Let $S = (u_1, \ldots, u_p, v_1, \ldots, v_q)$ be a feasible solution with at least one fractional variable, and let $\theta = min_{0 < u_i, v_j < 1}\{u_i, 1 - u_i, v_j, 1 - v_j\}$. Let $S_1$ denote a solution vector derived from $S$ in which all fractional $u_i$ are incremented by $\theta$ and all fractional $v_j$ and are decremented by $\theta$. Similarily, define $S_2$ as the solution vector in which all fractional $u_i$ are decremented by $\theta$ and all fractional $v_j$ are incremented by $\theta$. The solution vectors $S_1$ and $S_2$ satisfy the constraints of the linear program and hence are feasible solutions. Further, $S = \frac{1}{2} S_1 + \frac{1}{2} S_2$ and hence $S$ is a linear combination of $S_1$ and $S_2$. Hence $S$ is not a vertex. Thus, any solution vector containing at least one fractional variable can not be a vertex. Hence, all vertices of the above linear program are integral. □

We now present a more efficient combinatorial algorithm for solving the linear program of Equation 4 by considering its dual, which maximizes $\sum_{(u,v) \in E_1} a_{u,v}$ subject to

$$\sum_{(u,v) \in E_1} a_{u,v} + \sum_{(u,u') \in E_2} b_{u,u'} - \sum_{(u',u) \in E_2} b_{u',u} \leq f_u \qquad u \in U \tag{5}$$

$$\sum_{(u,v) \in E_1} a_{u,v} + \sum_{(v,v') \in E_3} c_{v,v'} - \sum_{(v',v) \in E_3} c_{v',v} \leq g_v \qquad v \in V. \tag{6}$$

Consider the directed graph with the vertex set $\{s, t\} \cup U \cup V$, where $s$ and $t$ are the source and sink, respectively. Let $E_4 = \{(s, u) \mid u \in U\}$, $E_5 = \{(v, t) \mid v \in V\}$, $E_3' = \{(v, v') \mid (v', v) \in E_3\}$. The edge set in the graph is $E_1 \cup E_2 \cup E_3' \cup E_4 \cup E_5$. We set the capacity of an edge $(s, u)$ to be $f_u$ and that of $(v, t)$ to be $g_v$. All other capacities are infinity. Let $a_{u,v}, b_{u,u'}, c_{v,v'}$ denote the flow through edges through $E_1$, $E_2$, and $E_3'$ respectively. Then, the flow constraint at each vertex in $U$ gives constraints (5) and at vertices in $V$ give constraints (6), respectively. Further, since $E_1$ is a cut, the flow through $E_1 = \sum_{u,v} a_{u,v}$ is the total flow from $s$ to $t$. Thus, the maximum flow solves the dual LP, from which we can obtain a solution to the primal LP.

The LP (4) is general enough to capture the data dissemination problem under the multicast model for tree networks, studied in Section 3.1. It thus yields an alternative polynomial-time algorithm for that problem. This algorithm, however, has higher complexity than the algorithm of Section 3.1 since the size of the maximum flow graph in the above algorithm is at least the sum of the sizes of the bipartite graphs whose vertex covers are computed in the algorithm of Section 3.1.

## 6. CONCLUDING REMARKS

Our model described in Section 2 assumes uniform lengths for all data items. Non-uniform data lengths affect both the push costs and the response costs. The effect of non-uniform data lengths on the push costs can be easily modeled by scaling the source frequencies appropriately. The impact on response costs depends on whether the responses are aggregated. If the responses are not aggregated, then non-uniform data lengths can be easily modeled by multiplying the length of the data item of a source $i$ to the response cost for the data item to every sink $j$ such that $i$ is in $I_j$. Thus all of our results hold when the responses are not aggregated. If the responses are aggregated, then we need an additional model that specifies the size of the aggregation of two data items or aggregates, an issue that we do not consider in this paper. Our results do apply in the special case when all of the aggregates have uniform size.

Two significant open problems left by our work are to resolve the gap between the $O(\log n)$ upper bound and the $O(1)$ lower bound on the approximation ratios for both the multicast model on general graphs and the metric unicast model with general interest sets. It would also be interesting to consider scenarios where the data update and query frequencies are unknown and may change with time, thus requiring solutions that estimate these frequencies and adapt to these changes. Heuristics for such adaptive schemes under special cases have been studied in [17].

## 7. REFERENCES


[1] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[2] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group Steiner trees and $k$-median. In *Proc. 30th Annual ACM Symposium on Theory of Computing*, pages 114–123, 1998.

[3] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k -median problem (extended abstract). In *31st Annual ACM Symposium on Theory of Computing*, pages 1–10, 1999.

[4] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[5] M. Chlebìk and J. Chlebìkovà. Approximation hardness of the steiner tree problem on graphs. In *SWAT '02: Proceedings of the 8th Scandinavian Workshop on Algorithm Theory*, pages 170–179, London, UK, 2002. Springer-Verlag.

[6] Y. Dodis and S. Khanna. Designing networks with bounded pairwise distance. In *31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 750–759, 1999.

[7] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC '03: Proc. of the 35th annual ACM Symposium on Theory of computing*, pages 448–455, New York, NY, USA, 2003. ACM Press.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[9] M. X. Goemans and M. Skutella. Cooperative facility location games. In *Symposium on Discrete Algorithms*, pages 76–85, 2000.

[10] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31:228–248, 1999.

[11] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[12] K. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1999.

[13] G. Konjevod, R. Ravi, and F. S. Salman. On approximating planar metrics by tree metrics. *Information Processing Letters*, 80(4):213–219, 2001.

[14] G. Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[15] G. Kortsarz and D. Peleg. Generating low-degree 2-spanners. *SIAM Journal on Computing*, 27(5):1438–1456, 1998.

[16] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44:245–249, 1992. Tech Report CS-92-37.

[17] X. Liu, Q. Huang, and Y. Zhang. Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 122–133, New York, NY, USA, 2004. ACM Press.

[18] R. Ravi and A. Sinha. Multicommodity facility location. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 342–349, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[19] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science WG-89, Lecture Notes in Computer Science, volume 411*, pages 196–210. Springer-Verlag, 1990.

[20] D. Sandler, A. Mislove, A. Post, and P. Druschel. Feedtree: Sharing web micronews with peer-to-peer event notification. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, Ithaca, New York, Feb. 2005.

[21] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *29th ACM Symposium on Theory of Computing*, pages 265–274, 1997.

[22] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *9th IPCO, LNCS 2337*, pages 240–257, 2002.

[23] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. In *5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX '02) LNCS 2462*, pages 256–270, 2002.

[24] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. In *Proc. of the 25th annual ACM symposium on Theory of computing*, pages 708–717, 1993.

[25] N. E. Young. K-medians, facility location, and the Chernoff-Wald bound. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 86–95, 2000.