Designing Overlapping Networks for Publish-Subscribe Systems

Jennifer Iglesias[1], Rajmohan Rajaraman[2], R. Ravi[1], and Ravi Sundaram[2]

[1]Carnegie Mellon University
[2]Northeastern University

17 April 2015

Abstract

From the publish-subscribe systems of the early days of the Internet to the recent emergence of Web 3.0 and IoT (Internet of Things), new problems arise in the design of networks centered at producers and consumers of constantly evolving information. In a typical problem, each terminal is a source or sink of information and builds a physical network in the form of a tree or an overlay network in the form of a star rooted at itself. Every pair of pub-sub terminals that need to be coordinated (e.g. the source and sink of an important piece of control information) define an edge in a bipartite demand graph; the solution must ensure that the corresponding networks rooted at the endpoints of each demand edge overlap at some node. This simple overlap constraint, and the requirement that each network is a tree or a star, leads to a variety of new questions on the design of overlapping networks.

In this paper, for the general demand case of the problem, we show that a natural LP formulation has a non-constant integrality gap; on the positive side, we present a logarithmic approximation for the general demand case. When the demand graph is complete, however, we design approximation algorithms with small constant performance ratios, irrespective of whether the pub networks and sub networks are required to be trees or stars.
Keywords: Approximation Algorithms, Steiner Trees, Overlay Networks, Publish-Subscribe Systems, Integrality Gap, VPN.

1 Introduction

In large Internet publishing systems, a variety of sources of information constantly refresh their content, while a set of subscribers continuously pull this updated information. The recent widespread adoption of the "Internet of Things" and "Web 3.0" tools similarly involves the constant real-time sharing of information between producers of relevant content and their corresponding consumers. Common examples include syndication systems as well as distributed databases that contain information originating at sources with sinks interested in the most up to date copies.

A natural approach to enable efficient information transfer in such systems is to build a *cost-effective* collection of networks, one for each publisher and supplier: the publishers push their updates to a set of locations via their respective networks, while the subscribers pull the information, refreshed by multiple publishers, from these intermediate nodes using their own networks. Note that each subscriber network needs only to *overlap* those publishers' networks that are of interest. Such interests are represented by an auxiliary bipartite *demand* graph with publishers on one side, subscribers on the other, and edges (of interest) between the two. Since the individual networks are being used for scatter/gather or push/pull operations (by publishers/subscribers respectively) the two natural structures are: trees and overlay stars. Trees correspond to situations where the entity (e.g. a pusher, such as Facebook, or a puller, such as the IRS) has sufficient network presence to employ multicast/reverse-multicast while overlay stars correspond to point-to-point communication.

This basic framework gives rise to a class of problems we have christened *DON* or Design of Overlapping Networks. Given their relevance to developments in the Internet ecosystem, these theoretical problems are significant from a practical perspective. Our central goal is to settle the polynomial-time approximability for the most general DON problem, in which we have an arbitrary demand graph, and arbitrary choice of tree or overlay star by each publisher/subscriber. In this paper, we obtain a constant approximation for the special case when all subscribers are interested in all publishers, and a logarithmic approximation for the general case. The latter approximation is, in fact, with respect to the value of a natural linear programming relaxation of the problem. In a contrasting result, we establish a non-constant integrality gap for this linear program. However, the exact approximability status of the general DON problem remains tantalizingly open.

1.1 Problem Definition. In the general DON problem we are given an undirected graph $G = (V(G), E(G))$ with non-negative costs on the edges $c : E \to \mathbb{Z}^+$, subsets of nodes $P, S \subseteq V$ (publishers and subscribers respectively), the type of network to be installed for each publisher and subscriber, $Type : P \cup S \to \{\text{tree}, \text{star}\}$, and an auxiliary demand graph $D = (V(D), E(D)$ where $V(D) = P \cup S$ and $E(D) \subseteq P \times S$ specifying (publisher, subscriber) pairs whose networks are required to overlap (intersect); the goal is to build a collection of networks satisfying the input requirements. We assume that the edge costs form a metric: they are symmetric and satisfy the triangle inequality. Any instance of the general DON problem can be split into four sub-instances. When the type requirement $Type$ is tree (resp., star) for all publishers and subscribers we refer to the problem as *tree-tree DON* (resp. *star-star DON*). We also use *tree-star DON* to refer to the two problem variants where on one side, say the publishers, we have rooted trees while on the other, we have rooted stars. We use the prefixes *general* and *complete* to denote arbitrary and complete demand graphs, respectively, as in general tree-tree DON or complete tree-star DON, etc. Thus, the term general (complete) DON refers to the problem where the demand graph is arbitrary (complete) and the type requirement may vary across terminals.

We denote the installed network by $N = (V, E_N)$; P_i denotes the network installed at publisher p_i and S_j the network rooted at subscriber s_j. Then, the multi-graph $N = (\cup_{p_i \in P} P_i) \cup (\cup_{s_j \in S} S_j)$ is the (multi-set) union of all the installed networks. The cost of N is the sum of the costs of all the constituent networks, with each edge counted as many times as the number of individual networks they are present in. Recall that the installed networks are operated autonomously by each publisher and subscriber, and thus the cost of an edge needs to be multiplied by the number of such independent networks that build and utilize it in their updates.

1.2 Results and Techniques. We present new algorithms and results for several DON problems.

1. We conjecture that a polynomial-time constant-factor approximation for general DON is

not achievable. We present in Section 2.1 an $\Omega(\log\log n)$ integrality gap for a natural LP relaxation of the general tree-tree DON; note that this result also extends to the general DON problem. This integrality gap proof, which is our strongest technical contribution, is based on a novel reduction from a well-studied LP relaxation for the group Steiner problem, applied to a hypercube demand graph instance of DON.

On the positive side, we present an $O(\log n)$-approximation algorithm for the general DON problem in Section 2.2. The main ingredient of our result is a constant-factor approximation algorithm for tree-star DON on tree metrics, by a careful deterministic rounding of an LP relaxation of the problem. The logarithmic approximation for general DON follows by extending to general metrics and combining with results for the star-star and tree-tree variants.

2. We next study the complete DON variants where the demand graph is complete. We give constant-factor approximation algorithms for all three variants— tree-tree, star-star and tree-star— which together yield a constant-factor approximation for complete DON. Unlike our algorithm for general DON, all of our algorithms for the complete demand case are combinatorial; they combine structural characterizations of near-optimal solutions with interesting connections to access network design and facility location problems.

   (a) Our approximation factor for complete tree-tree DON in Section 3.1 is $4\rho_{TS}$ where ρ_{TS} is the best known approximation for the tree-star Access Network Design problem (which is generalized by the Connected Facility Location or CFL problem). Thus $\rho_{TS} \leq \rho_{CFL} \leq 4$ [3].

   (b) Our approximation factor for complete star-star DON in Section 3.2 is 4α, where α is the best approximation factor achieved for uncapacitated facility location, improves over the result in [1].

   (c) For the complete tree-star-DON problem, we get a $4\rho_{TS}$-approximation in Section 3.3.

1.3 Related Work. Data Dissemination Networks. Our formulation of DON generalizes network data dissemination problems first studied in [1]. Using our terminology, the relevant results of [1] are $O(\log n)$-approximation algorithms for general tree-tree DON and general star-star DON, and a 14.57-approximation for the complete star-star DON. Our work improves the approximation factor for complete star-star DON to under 6 (since the current best approximation for uncapacitated facility location is 1.488 [11]), and presents new results for many other DON problems. The star-star DON problem is also closely related to the minimum-cost 2-spanner problem studied in [2, 10]. In particular, a greedy algorithm essentially along the same lines as an algorithm of [2] yields an $O(\log n)$-approximation for the star-star-DON problem even when the underlying distances do not form a metric.

Network Design. There has been considerable work in network design, which is concerned with the design of network structures that satisfy some connectivity properties and optimize some underlying cost structure [20]. Well-known problems in this area include the minimum Steiner tree [5], group Steiner tree [6], and general survivable networks [19]. One key distinction between many of these network design problems and DON is that the desired solution in DON is a collection of networks (as opposed to a single network), and each edge contributes to the total cost of the solution as many times as it occurs in the network collection. On the other hand, the goal in many classical network design problems is to build a single network. Note that the problem of building a single minimum-cost network such that every pair of nodes in a given demand graph is connected in the network is exactly the generalized Steiner network problem, for which polynomial-time constant-factor approximations exist.

Multicommodity facility location. Another stream of work has addressed the extension of facility location problems to reach clients with additional restrictions on the facility opening costs, to reach facilities more robustly [18], or with the addition of services that facilities open to satisfy the clients with various cost functions governing the installation of services and facilities [14, 16]. The work in [12] arising from publisher-subscriber mechanisms is most closely related to our work, and rather than use a network from each publisher, models the publisher as a commodity that can be supplied at various nodes in the network by installing "facilities" of appropriate costs; the subscribers build minimum-cost networks to reach these facility installations of the appropriate publishers.

Access Networks and Connected Facility Location. Our algorithms for the complete DON problem are connected to to the access network design and facility location problems. In a version of the Access Network Design problem [13], we are given an undirected graph, a root node and nonnegative metric costs on the edges, along with a subset of terminal nodes. The goal is to design a backbone network in the form of a tree or tour which is built with higher speed and higher quality cables, while the terminals access the backbone using direct access edges. Thus the overall network is a backbone rooted Steiner tree (or tour), with access networks that are stars arising from the terminals and ending at the nodes of the backbone. We are given a cost multiplier μ that denotes the cost overhead factor for the backbone compared to the access network and the objective to be minimized is the total cost $\mu \cdot c(\text{backbone network } T) + \sum_{stars\ s} c(s)$. A $\rho_{AN} = 3 + 2\sqrt{2}$-approximation is presented for this problem [13] when the backbone is a ring and the access network is a star. Subsequent work [8, 17, 3] present constant-factor approximations for other generalizations and variants of this problem as well using LP rounding and primal-dual methods. The current best approximation factor for the CFL generalization is $\rho_{CFL} \leq 4$ [3], which also extends to tree-star access network design, i.e. $\rho_{AN} \leq \rho_{CFL} \leq 4$.

Virtual Private Network. The DON problems are also closely related to the VPN and asymmetric VPN problems. The VPN problem can be solved exactly [7], while the asymmetric VPN problem is NP-Hard but has a constant approximation [15]. The VPN problems differ from DON as each VPN problem instance seeks only one network, while a DON instance builds multiple networks. Nevertheless, we are able to decompose an approximate solution for asymmetric VPN into multiple networks, and obtain a useful approximation for the complete tree-tree DON problem (Appendix B). As we establish in Section 3.1, however, a more direct approach yields a much better approximation factor.

2 DON with General Demands

In this section, we consider approximation algorithms for the general DON problem. We present in Section 2.1 an $\Omega(\log \log n)$ integrality gap for a natural LP relaxation of the tree-DON problem (with general demands). In Section 2.2, we present an $O(\log n)$-approximation algorithm for general DON.

2.1 Integrality Gap for Tree-Tree DON. In this section, we show that a natural LP formulation for the tree-tree DON problem has super-constant integrality gap. We note that the same lower bound on integrality gap extends to the appropriate LP for general DON. A natural integer program, $\mathbf{IP_{BT}}$ for the tree-tree DON problem is as follows (all variables are $0-1$): we let $r \in P \cup S$ denote a publisher or subscriber node that serves as the root of its tree N_r; z_e^r is an indicator variable that is 1 iff edge $e \in E(G)$ is in tree N_r; y_h^r is an indicator variable that is 1 iff vertex h is in tree N_r; $x_h^{r,s}$ is an indicator variable that is 1 iff vertex h is in both trees, N_r and N_s; $\mathcal{C}$ will refer to a cut which is a subset of vertices of $V(G)$ and $E(\mathcal{C})$ will denote the edges between $\mathcal{C}$ and its complement $V(G) \setminus \mathcal{C}$. The integer program $\mathbf{IP_{BT}}$ for the tree-tree DON has the following nontrivial constraints.

$$
\begin{aligned}
\min \sum_{r \in V(D), e \in E(G)} c_e z_e^r & & \\
\sum_{e \in E(\mathcal{C})} z_e^r &\geq y_h^r & \forall \mathcal{C}, r \in \mathcal{C}, h \in V(G) \setminus \mathcal{C} \\
x_h^{r,s} &\leq y_h^r & \forall r, s \in V(D), h \in V(G) \\
\sum_{h \in V(G)} x_h^{r,s} &\geq 1 & \forall (r,s) \in E(D)
\end{aligned}
$$

$$\mathbf{IP_{BT}}$$

The first set of cut covering constraints enforce that the tree rooted at r is connected with all nodes h for which y_h^r is set to one. The third set enforces all pairs of terminals r, s in the demand graph must meet in some hub vertex h, while the second set enforces that if a node h is used as a hub for a pair, it is required to occur in both these trees. Relaxing the above integer program by allowing the variables to take values in $[0,1]$ gives us the natural linear program $\mathbf{LP_{BT}}$. Observe that the feasible integral points of the linear program are exactly the solutions to the integer program.

Theorem 1 *For every sufficiently large n, there exist instances of tree-tree DON with $n = |V(G)|$ for which $\mathbf{LP_{BT}}$ has an $\Omega(\log\log n)$ integrality ratio.*

Proof. Recall that the integrality ratio of a (minimizing) linear program is the minimum ratio between any feasible integral point and the optimum fractional solution. Our proof will proceed by a reduction from a linear program for the Group Steiner Tree (GST) problem.

Given a tree T with edge costs and a collection of groups of leaves, the Group Steiner Tree problem is to find a minimum cost subtree such that at least one vertex from every group is connected to the root. In [9] it was shown that a natural linear program for the GST problem has an $\Omega(\log^2 n)$ integrality ratio even when the input metric costs c arise from an underlying tree. Similar to the linear program for the tree-tree DON problem we present the linear program $\mathbf{LP_{GST}}$ as the relaxation of an integer program $\mathbf{IP_{GST}}$ with $0-1$ variables.

$$
\begin{aligned}
\min \sum_{e\in E(T)} x_e c_e & \\
g_i \cap \mathcal{C} &= \phi \quad \forall \mathcal{C}, r \in \mathcal{C} \\
\sum_{e\in E(\mathcal{C})} x_e &\geq 1
\end{aligned}
$$

$\mathbf{IP_{GST}}$

Here is an intuitive explanation of what the variables in $\mathbf{IP_{GST}}$ represent: x_e is an indicator variable that is 1 iff edge $e \in E(T)$ is in the solution subtree; $g_i, 1 \leq i \leq k$ are the k groups and $\mathcal{C}$ is a subset of vertices of $V(G)$ referring to a cut and $E(\mathcal{C})$ will denote the edges between $\mathcal{C}$ and its complement $V(G) \setminus \mathcal{C}$. The main cut covering constraints enforce that each group is connected to the root node r.

As stated before [9] show that $\mathbf{LP_{GST}}$ has an integrality ratio of $\Omega(\log^2 n)$ even on tree metrics when k, the number of groups, is $\Omega(n)$ where $n = |V(T)|$.

Given an instance, T_{GST} of $\mathbf{LP_{GST}}$ with $n = V(T)$ vertices and k groups, we transform it into an instance, GD_{BT} of $\mathbf{LP_{BT}}$ with $N = 2^k n = |V(G)|$ vertices in the host graph such that

- corresponding to every fractional solution, of value f_{GST}, of $\mathbf{LP_{GST}}$ there is a fractional solution of value $f_{BT} = 2^k f_{GST}$ to $\mathbf{LP_{BT}}$, and
- corresponding to every feasible integral point, of value I_{BT}, of $\mathbf{LP_{BT}}$ there is a feasible integral point of value $I_{GST} = \frac{I_{BT}(1+\log k)}{2^k}$ to $\mathbf{LP_B T}$.

The transformation is intuitive: we take a "graph product" of the Group Steiner Tree instance with a hypercube of dimension k, where k is the number of groups. In more detail, given the tree T for the Group Steiner Tree instance, the demand graph D is a hypercube of dimension k, with the ith dimension of the hypercube being associated with group g_i. The host graph G has 2^k copies of T, one for each vertex of the demand hypercube (where the demand edges go between the roots of the corresponding trees). For every edge (r, s) in D in the ith dimension we connect pairwise with zero-cost edges the leaves in group g_i of the copy of T corresponding to r, with the leaves in group g_i of the copy of T corresponding to s.

It is easy to see that $f_{BT} = 2^k f_{GST}$ for the above transformation - observe that replicating the fractional solution to $\mathbf{LP_{GST}}$ in each of the 2^k copies of T is a valid fractional solution to $\mathbf{LP_{BT}}$.

For the other direction, i.e., to see $I_{GST} = \frac{I_{BT}(1+\log k)}{2^k}$ first observe that for edge (r, s) in dimension i of the demand hypercube, at least one of the trees corresponding to r or s must cross dimension i and the only way to cross dimension i is along a 0-cost edge connecting two corresponding group g_i leaves. Now note that any tree T' in an integral solution to $\mathbf{LP_{BT}}$ can be transformed into a subtree of T by keeping an edge in T if T' contains the corresponding edge in any copy of T in G. Let the subtree of T so obtained be called the *retract* of T'. It is easy to see that if T' ever crosses dimension j then a leaf in group g_j is connected to the root of T in its retract and that the cost of a retract is never more than the cost of the original T'. By our earlier observation for any edge (r, s) in dimension i at least one of the two retracts, that of the tree corresponding to r or corresponding to s must connect a node in group g_i to the root. Hence if we select a node in D at random and take its retract then any given group is connected with probability at least $1/2$ and it has expected cost $\frac{I_{BT}}{2^k}$. Thus if we take the union of $1+\log k$ retracts chosen uniformly at random then the resulting subgraph of T has expected cost $\frac{I_{BT}\log k}{2^k}$ and the probability any given group is not connected to the root is less than $\frac{1}{k}$. Since there are k groups this means there exists a subgraph of T, connecting the root to every group, of cost at most $\frac{I_{BT}(1+\log k)}{2^k}$, i.e., $I_{GST} = \frac{I_{BT}(1+\log k)}{2^k}$.

From $f_{BT} \le 2^k f_{GST}$ and $I_{GST} \le \frac{I_{BT}(1+\log k)}{2^k}$ it follows that $\frac{I_{BT}}{f_{BT}} \ge \frac{I_{GST}}{f_{GST}}(1+\log k)$. By [9], when $k = \theta(n)$ we have that $\frac{I_{GST}}{f_{GST}} = \Omega(\log^2 n)$ from which it follows that $\frac{I_{BT}}{f_{BT}} = \Omega(\frac{\log^2 n}{\log k}) = \Omega(\log n)$ but the size of the transformed instance is $N = 2^k n$, i.e., $n = \Omega(\frac{\log N}{\log\log N})$. In other words, the integrality gap $\frac{I_{BT}}{f_{TT}} = \Omega(\log n) = \Omega(\log(\frac{\log N}{\log\log N})) = \Omega(\log\log N)$ $\square$

2.2 $O(\log n)$ approximation. We next show that the general DON problem can be approximated to within an $O(\log n)$ factor in polynomial time. As discussed in Section 1, the general DON problem can be split into three problems: tree-tree DON, star-star DON, and tree-star DON. In previous work, $O(\log n)$-approximation algorithms have been developed for tree-tree DON and star-star DON [1]. We now present an $O(\log n)$-approximation for tree-star DON, implying an $O(\log n)$-approximation for general DON.

Our $O(\log n)$-approximation for tree-star DON is obtained by deriving a constant-factor approximation for the special case of tree metrics, and invoking the standard reduction from general metrics to tree metrics [4]. Our constant-factor approximation algorithm, which rounds an LP relaxation, essentially generalizes a result of [12] on multicommodity facility location from a uniform facility cost case to the case where the facility costs form a tree metric.

Theorem 2 *The tree-star DON problem with general demands on tree metrics can be approximated to within a constant factor in polynomial time. This implies an $O(\log n)$-approximation algorithm for general DON on general metrics.*

Due to space constraints, we defer the algorithm and the proof of Theorem 2 to Appendix A.

3 DON with Complete Demands

In this section, we present constant factor approximation algorithms for the DON problem when the demand graph is complete. We obtain this result by deriving constant-factor approximations for the three variants—tree-tree, star-star and tree-star— in the following subsections.

3.1 Complete tree-tree DON. The complete tree-tree DON problem has an interesting connection to the asymmetric VPN problem [15], which we can exploit to obtain a constant-factor approximation. As we discuss this reduction and explain in Appendix B, however, the approximation we obtain using the best known asymmetric VPN algorithm is 49.84. Here, we present a more direct approximation algorithm for which we are able to establish a much better approximation factor.

Theorem 3 *There is a $4\rho_{TS}$-approximation algorithm for complete tree-tree DON, where ρ_{TS} is the best factor for the tree-star access network design problem.*

In the rest of this subsection, we give a proof of Theorem 3. Given N^*, an optimal solution, let us denote the publisher and subscriber networks by $P_1, P_2, \ldots, P_k$ and $S_1, S_2, \ldots, S_l$ where we index the nodes so that we have $c(P_1) \leq c(P_2) \leq \cdots \leq c(P_k)$ and $c(S_1) \leq c(S_2) \leq \cdots \leq c(S_l)$. Let c_P^* and c_S^* denote the total cost of the publisher and subscriber trees. Let s_j be the subscriber whose network is S_j and let p_i be the publisher whose network is P_i. Note that feasibility requires that $P_i \cap S_j \neq \emptyset$ for all i, j. Let us also assume without loss of generality that $c(P_1) \leq c(S_1)$

The key transformation of the optimal solution is a reconfiguration of the subscriber networks where we replace each tree S_j for $j \neq 1$ by the direct edge from subscriber node j to subscriber node 1 concatenated with the subscriber tree S_1. In other words, we set $S'_j = \{(s_j, s_1)\} \cup S_1$ for every subscriber $s_j \neq s_1$. Let us assign $S'_j = S_j$.

Note that the modified subscriber trees are still feasible since the original subscriber tree S_1 intersects every publisher tree. We now bound the cost of the additional edge from subscriber j to the subscriber 1, the root of S_1.

Lemma 1 *For every subscriber $j \neq 1$, we have $c_{i1} \leq 3c(S_j)$.*

Proof. To see this, note that by taking the path from j in S_j to its intersection with P_1 and following it to the intersection of P_1 and S_1 and continuing along S_1 to the subscriber node 1, we have found a path from j to 1 of cost no more than the sum of the costs of S_j, P_1 and S_1. However, since $c(S_j) \geq c(S_1) \geq c(P_1)$, the length of this path is at most $3c(S_j)$. □

Given that every subscriber contains the tree S_1, it is particularly simple to design the publisher network P'_i (for publisher p_i) that needs to reach this tree: it will simply be a direct edge that represents the shortest path from the publisher p_i to the tree S_1. The union of all such direct edges gives a collection of stars that end at the subscriber tree S_1. Furthermore since the subscriber tree S_1 is going to be used by every subscriber node, its cost must be counted $|S| = l$ times in the objective.

The resulting problem of finding the best tree for S_1 is exactly the tree-star access network design problem [13] with the root being subscriber 1, the multiplier $M = |S|$ and the terminals being $R = P$, the publisher nodes. Given an optimal solution N^* for the complete tree-tree DON problem, we thus have a solution to the tree-star access network instance of cost at most $c(P^*) + |S| \cdot c(S_1)$. We thus have the following lemma.

Lemma 2 *For the correct choice of the subscriber node 1 as the root with multiplier $|S|$ and terminals P, there is a solution to the tree-star access network design problem of cost at most $c_P^* + |S| \cdot c(S_1)$.* □

Proof of Theorem 3: The approximation algorithm tries every subscriber node as the root of the tree-star access network problem formulated above. By adding the direct edge from each other subscriber to this root, and extending the backbone tree with each such edge, we get a solution to the complete tree-tree DON problem. The algorithm keeps the solution of smallest total cost among all choices of the root subscriber node. The total cost of the solution is the sum of the cost of the tree-star access network design problem and the sum of the costs of the direct edges from the subscribers to the root. By Lemma 1, the latter cost is no more than three times the cost of the tree (with the multiplier of —S—) in the solution to the tree-star access network design problem. By 2 and the ρ_{TS}-approximation factor for the tree-star access network design problem, we thus obtain a total cost of at most $4\rho_{TS}(c_P^* + |S| \cdot c(S_1))$ which is at most $4\rho_{TS}$ times the cost of N^*. □

It is not hard to extend the above methods to the case when the input terminals are partitioned into more than two subsets, say $R = P_1 \cup P_2 \cup \ldots \cup P_k$ and the demand graph is the complete k-partite graph between these k subsets. By considering the partition that has the cheapest tree network in the optimal solution to be in P_1, the above argument can be adapted to give a constant-factor approximation. We omit the details in this extended abstract.

3.2 Complete star-star DON. In this section, we present a constant-factor approximation for complete star-star DON.

Theorem 4 *There is a 4α-algorithm for complete star-star DON, where α is the best approximation achievable for metric uncapacitated facility location.*

Our algorithm and the proof of Theorem 4 are based on an argument that there exists a constant-factor approximate solution that has a special structure; our algorithm then computes a constant-factor approximate solution with this special structure.

Given a solution where the publisher network is $P_1, P_2, \ldots, P_k$ and subscriber network is $S_1, S_2, \ldots S_l$, let P_1 be the publisher network of smallest cost and S_1 be the subscriber network of smallest cost without loss of generality. Also, let $\sigma(P_i)$ denote the set of nodes (which we refer to as hubs for P_i) in the star for the ith publisher. Likewise, let $\sigma(S_j)$ be the set of nodes in the star for the jth subscriber. Thus, we can refer to solutions using the maps defined by σ and denote the optimal one by σ^*. The next lemma shows a near-optimal solution with a very simple structure.

Lemma 3 *There exists a solution σ such that $c(\sigma) \leq 4c(\sigma^*))$, and either $\sigma(S_i) = \sigma(S_j)$ for all pairs of subscriber networks S_i, S_j and $|\sigma(P_i)| = 1$ for each publisher network P_i or $\sigma(P_i) = \sigma(P_j)$ for all pairs of publisher networks P_i, P_j and $|\sigma(S_i)| = 1$ for each subscriber network S_i.*

Proof. Without loss of generality, let $c(P_1) \leq c(S_1)$. Since each subscriber star intersects all publisher stars, we have $d(s_i, s_1) \leq c(S_i) + c(P_1) + c(S_1) \leq 3c(S_i)$. Let C_1 denote the set of publishers that share any hub with p_1. Let p_2 denote the least-cost publisher not in C_1. Let C_2 be the set of all publishers not in C_1 that share any hub with p_2. In general, let p_{j+1} be the least-cost publisher not in $\bigcup_{1 \leq i \leq j} C_i$. Let C_{j+1} denote the set of all publishers not in $\bigcup_{1 \leq i \leq j} C_i$ that share any hub with p_{j+1}. Let h_j denote any hub in $\sigma^*(s_1) \cap \sigma^*(p_j)$.

Let p'_j be an arbitrary publisher in C_j. We first obtain the following equation $d(p'_j, p_j) \leq 2c(P_{j'})$ (owing to a shared hub and the fact that $c(P_j) \leq c(P_{j'})$). By construction, for any two distinct p_i and p_j, we have $\sigma^*(p_i) \cap \sigma^*(p_j) = \emptyset$; i.e., p_i and p_j do not share any hubs. Note that this may not be true of all pairs of publishers in $C_i \times C_j$.

We now consider two cases. In the first case when $c(P_j) \leq d(s_1, h_j)$, we have all subscribers meet all the publishers in cluster C_j at p_j. Consider any subscriber s_i. It meets p_j at some hub, say h^i_j. Its increase in cost for meeting p_j now is at most $\mathrm{cost}_{\sigma^*}(p_j) \leq d(s_1, h_j)$, which equals one leg of s_1's star. Since two different p_j's do not share any hubs, the h^i_j's (for a given i) are all different. Hence, the total increase in cost for s_i is at most $\sum_j d(s_1, h_j)$, which is at most $c(S_1)$.

If $c(P_j) > d(s_1, h_j)$, then we will have all publishers in C_j, go to s_1. Fix a publisher p'_j in C_j. Its total cost is at most $d(p'_j, p_j) + c(P_j) + d(s_1, h_j) \leq d(p'_j, p_j) + 2c(P_j) \leq 4c(P'_j)$. All the subscribers also go to s_1 to handle this case. We have $d(s_i, s_1) \leq c(S_i) + c(P_1) + c(S_1) \leq 3c(S_i)$.

So overall, we obtain a blowup of at most 4 in the cost for each publisher and each subscriber. We have proved that there exists a solution of cost at most $4 \cdot OPT$ in which every subscriber's star connects to exactly the same set of hubs and every publisher's star is just a line to one of the hubs. □

Proof of Theorem 4: Using Lemma 3, we now give a polynomial-time 4α-algorithm where α is the best approximation achievable for the uncapacitated facility location problem.

Our algorithm considers all possible choices for S_1, a linear number (where by symmetry S_1 could be on either side). For a given choice of S_1, we formulate an uncapacitated facility location problem, with the set of publishers as the clients, and the potential facility locations being the publishers and S_1. The cost of opening a facility at any of these nodes is the sum of the distances of all the subscribers to that node. Given a solution to this facility location problem, we obtain a solution to the complete star-star DON problem as follows: each publisher's star is a singleton edge to the facility it is assigned to; each subscriber's star consists of edges to all the open facilities.

We solve all the linear number of facility location problems, and then the corresponding problems with the roles of subscribers and publishers reversed, and take the best solution. This yields the desired approximation. □

3.3 Complete tree-star DON. We now present a constant factor approximation for complete tree-star DON. Without loss of generality, let us suppose that the publishers will build trees, and the subscribers will build stars. The main idea is to show that either the appropriately defined complete star-star DON solution or complete tree-tree DON solution is within a constant factor of optimal.

Let N^* be an optimal solution. Let the trees be indexed $P_1, P_2, \ldots P_k$ and the stars $S_1, \ldots S_\ell$ such that $c(P_1) \leq \cdots \leq c(P_k)$ and $c(S_1) \leq \cdots \leq c(S_k)$.

First consider the case where $c(P_1) \geq c(S_1)$. Note that every P_i and S_j must have a non-empty intersection. Now for every tree P_j we can redirect it to P_1 and then make a copy of P_1. So we will let: $P'_j = \{(p_j, p_1)\} \cup P_j$.

This solution is feasible because P_1 must intersect all the stars. These additions to the solution cost at most $3c(N^*)$, as seen in lemma 1. Now all the stars can simply take an edge which is the shortest edge to the tree.

The approximation algorithm from this point follows the tree-tree case exactly. In this case, we get that the final solution has cost at most $4\rho_{TS}c(N^*)$. Where ρ_{TS} is the best constant approximation for the tree-star access problem.

Next consider the case that $c(S_1) \geq c(P_1)$. We will now choose p_i's in a similar fashion to the complete star-star DON problem. Let p_1 be the publisher with the smallest cost tree. Let C_1 be all the publishers whose trees meet p_1's tree. Now let p_2 be the smallest tree which does not intersect p_1's tree. Let C_2 be all the publishers not in C_1 who meet p_2's tree. Likewise p_{j+1} will be the smallest tree not in $\cup_{1 \leq i \leq j} C_i$. Let C_{j+1} be all the publishers which intersect p_{j+1}'s tree not in $\cup_{1 \leq i \leq j} C_i$.

Now from hereon, the proof follows that for the complete star-star DON case. Hence we have a solution within a constant factor of optimal where all the stars go to s_1 (the subscriber with star S_1), and some of the publishers; P_{open}. Each tree goes to the nearest node in $S_1 \cup P_{open}$. This establishes the following lemma.

Lemma 4 *The complete tree-star DON has an $O(1)$-approximate solution in which either all the subscribers go to some hubs and each tree goes to the nearest hub among a set of one subscriber* *and some publishers, or where all the publisher trees are identical and all the subscribers go to the closest node in that tree.* □

For solving the complete tree-star DON problem, we apply our constant-factor approximation algorithm for the complete tree-tree DON instance, together with our constant-factor algorithm for the complete star-star DON instance, and take the better of the two. This completes our argument showing that complete tree-star DON can be approximated to within a constant factor.

References

[1] R. C. Chakinala, A. Kumarasubramanian, K. A. Laing, R. Manokaran, C. Pandu Rangan, and R. Rajaraman. Playing push vs pull: models and algorithms for disseminating dynamic data in networks. In *SPAA*, pages 244–253, 2006.

[2] Yevgeniy Dodis and Sanjeev Khanna. Designing networks with bounded pairwise distance. In *STOC*, pages 750–759, 1999.

[3] Friedrich Eisenbrand, Fabrizio Grandoni, Thomas Rothvoss, and Guido Schafer. Connected facility location via random facility sampling and core detouring. *JCSS*, 76(8):709 – 726, 2010.

[4] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *STOC*, pages 448–455, 2003.

[5] M. R. Garey and D. S. Johnson. *Computers and intractability.* Freeman Press, 1979.

[6] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000.

[7] Navin Goyal, Neil Olver, and F B. Shepherd. The vpn conjecture is true. In *STOC*, pages 443–450, 2008.

[8] Anupam Gupta, Jon M. Kleinberg, Amit Kumar, Rajeev Rastogi, and Bülent Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. *STOC*, pages 389–398, 2001.

[9] Eran Halperin, Guy Kortsarz, Robert Krauthgamer, Aravind Srinivasan, and Nan Wang. Integrality ratio for group Steiner trees and directed Steiner trees. *SODA*, pages 275–284, 2003.

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

[11] Shi Li. Approximation algorithms for network routing and facility location problems. *Doctoral Dissertation, Princeton University*, Januray 2014.

[12] Laura J. Poplawski and Rajmohan Rajaraman. Multicommodity facility location under group steiner access cost. *SODA*, pages 996–1013, 2011.

[13] R. Ravi and F. Sibel Salman. Approximation algorithms for the traveling purchaser problem and its variants in network design. *ESA*, pages 29–40, 1999.

[14] R. Ravi and A. Sinha. Multicommodity facility location. *SODA*, pages 342–349, 2004.

[15] Thomas Rothvoss and Laura Sanita. On the complexity of the asymmetric vpn problem. In *APPROX/RANDOM*, pages 326–338, 2009.

[16] David B. Shmoys, Chaitanya Swamy, and Retsef Levi. Facility location with service installation costs. In *SODA*, pages 1088–1097, 2004.

[17] Chaitanya Swamy and Amit Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[18] Chaitanya Swamy and David B. Shmoys. Fault-tolerant facility location. In *SODA*, pages 735–736, 2003.

[19] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. In *STOC*, pages 708–717, 1993.

[20] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

A An $O(\log n)$-approximation for DON with general demands

In this section, we will present a constant approximation algorithm for tree-star DON on tree metrics and establish Theorem 2.

We first present a linear programming relaxation for the problem. Let T denote the given tree which is our metric. For a publisher j and an edge e of T, let z_e^j represent the extent to which j's tree uses e. For a subscriber i and leaf node v, let y_v^i denote the extent to which i's star visits v. For leaf node u, subscriber i and publisher j such that (i, j) is in the demand graph, let $x_u^{i,j}$ denote the extent to which j meets i at u. Let $d(u, v)$ denote the distance between u and v under the tree metric; abusing notation somewhat, let $d(e)$ denote the distance between the two endpoints of the edge e. Then, we have the following LP.

$$\begin{array}{lll}
\text{minimize} & \sum_{j,e} z_e^j d(e) + \sum_{i,u} y_u^i d(i,u) & \\
\text{subject to} & & \\
& z_e^j \geq \sum_{u: e \in P_{ju}} x_u^{i,j} & \text{for all } i,\ j,\ e \\
& \sum_u x_u^{i,j} \geq 1 & \text{for all } (i,j) \text{ in demand graph} \\
& y_u^i \geq x_u^{i,j} & \text{for all } i,\ j,\ u \\
& x_u^{i,j}, y_u^i, z_e^j \geq 0 & \text{for all } i,\ j,\ u,\ e
\end{array}$$

We now present our algorithm. We introduce some useful notation first. Let Y_v^i denote the sum of y_w^i, over all leaves w in the subtree rooted at v. Similarly, let $X_v^{i,j}$ denote the sum of $x_w^{i,j}$, over all leaves w in the subtree rooted at v.

1. Solve the above LP to obtain a fractional solution (x, y, z).

2. For every subscriber i:

- For every node v such that $Y_v^i \geq 1/3$ and there is no child c of v such that $Y_c^i \geq 2/3$: we mark node v.
- For each marked node v such that no ancestor of v is marked, we add v to $\sigma(v)$; we refer to v as a type-C hub for i.
- For every node v such that (a) there is no ancestor of v that is a type-C hub for i, and (b) there are two children c_1 and c_2 of v such that $Y_{c_1}^i \geq 1/3$ and $Y_{c_2}^i \geq 1/3$, we add v to $\sigma(v)$; we refer to v as a type-A hub for i.
- For every node v that is an ancestor of a type-C hub, we define W_v^i to be sum, over every child c of v that is not an ancestor of a type-C hub, of Y_c^i.
- For every path p from the root or a type-A hub node to a descendant type-A hub or type-C hub node: we divide p into minimal contiguous segments such that the sum of W_v^i, over all v in the segment, is at least 1/3; for each such segment, we create a type-B hub for i at the lowest node in the segment.
- The star network for i connects i to each type-A, -B, and -C hub.

3. For every publisher j, the tree network consists of all edges e such that $z_e^j \geq 1/3$.

We prove Theorem 2 by establishing the following two lemmas.

Lemma 5 *For any edge (i, j) in the demand graph, the tree of publisher j overlaps with the star of subscriber i at least one node.*

Proof. Fix publisher j and subscriber i such that (i, j) is an edge in the demand graph. Consider some subtree rooted at a node r_0 such $X_{r_0}^{i,j}$ is at least 1/3 in the LP solution, while for any child c of r_0, $X_c^{i,j} < 1/3$. Suppose $(r_0, r_1, \ldots, r_f)$ denote the path from r_0 to the root of the tree.

We first show that if there is a type-C hub at a node r_k, then the tree of publisher j includes node r_k. By our algorithm's choice of locating type-C hubs, it follows that $Y_{r_{k-1}}^i < 2/3$. Therefore, publisher j meets subscriber i less than 2/3 in the subtree rooted at r_{k-1}. We consider two cases. If j is in the subtree rooted at r_{k-1}, then for the edge $e = (r_{k-1}, r_k)$, $z_e^j \geq 1/3$. Otherwise, since $X_{r_{k-1}}^{i,j} \geq X_{r_0}^{i,j} \geq 1/3$, again we have $z_e^j \geq 1/3$. Thus, in both cases, we ensure that the tree for publisher j contains r_k.

We next show that if there is a type-A or type-B hub for i at a node r_k and there are no hubs for i at any r_g, $0 \leq g < k$, then the tree for j must include r_k. Since there is no type-A hub at any r_g, $0 \leq g < k$, each r_g has at most one child that has a descendant with a type-C hub; if there were two such children, then r_g would have a type-A hub. Furthermore, there must be a type-C hub in the subtree rooted at r_0; if not, then the first ancestor of r_0 to have a hub would have a type-C hub, which would contradict our assumption. So suppose there is a type-C hub in the subtree rooted at r_0, say under the child r_{-1} of r_0. Then, it must be the case that the sum of $W_{r_g}^i$, over $0 \leq g < k$, is at most 1/3, since otherwise we would have a type-B hub at r_g. Furthermore, by the definition of r_0, $X_{r_{-1}}^{i,j} < 1/3$. This implies that j meets i to an extent of 1/3 outside the subtree rooted at r_{k-1} and at least 1/3 inside the subtree rooted at r_{k-1}. Thus, regardless of where j is located for edge $e = (r_{k-1}, r_k)$, we will have $z_e^j \geq 1/3$, ensuring that the tree for publisher j contains r_k. □

Lemma 6 *The total cost of the tree and star networks is at most a constant factor times the LP optimal.*

Proof. An edge e is added to the tree of publisher j exactly when $z_e^j \geq 1/3$. Therefore, the cost of the tree network of j is within the cost for j in the LP.

We next consider the costs of the subscriber stars. There are three parts to it. First is the distance to the type-A hubs. If a type-A hub for i is created at a node r, then there exist two children c_1 and c_2 of r such that $Y_{c_1}^i$ and $Y_{c_2}^i$ are both at least 1/3. Clearly, i is either not in the subtree rooted at c_1 or not in the subtree rooted at c_2. In either case, the cost for i in the LP solution for reaching the fractional hubs in one of c_1 or c_2 is at least $d(i, r)/3$. Adding this over all the type-A hubs yields a cost that is at most 3 times the LP cost for i.

If a type-C hub is created at a node r, then we consider two cases: the LP cost associated with the fractional hubs under the subtree at r is at least $d(i, r)/3$.

If a type-B hub is created at a node r, then consider the sequence of ancestors a of r, whose W_a^i add up to 1/3. The cost of i reaching the fractional hubs in the LP that contribute to these W_a^i is at least $d(i, r)/3$.

The fractional hubs against which we have charged the type-C and type-B hubs are different, so the cost for the type-B and type-C hubs is at most 3 times the LP cost for i, yielding an $O(1)$-approximation for the overall total cost. □

B Alternate Algorithm for Complete Tree-Tree DON using VPNs

We use results from the asymmetric VPN problem [7, 15] and show how it gives an approximation for complete tree-tree DON. First we introduce the VPN problem. Given a graph G, with edge costs c, and marginals for each vertex, the VPN problem is to build a network of minimum cost such that for any set of pairwise demands which obey the marginals, the flow can be routed on our network. A set of pairwise demands obeys the marginals if the demands a vertex is involved in does not exceed its marginal. One crucial distinction between VPN and the DON problems is that while the VPN problem seeks the design of a single network, DON problems seek networks for every node involved.

Now we define asymmetric VPN. Here the flows are directed, and each vertex has two marginals, one for how much can flow out of the node, and one for how much can flow into the node. We restrict to the case where the terminals allow 1 flow out of the node and no flow in, *sources*, or they allow 1 flow into the node, and no flow out, *sinks*.

It turns out that for asymmetric VPN, there is always a tree solution which is within a constant factor of an optimal solution [15]. We now use this tree solution to get a solution for complete tree-tree DON.

Lemma 7 *Given a complete tree-tree DON problem, consider an asymmetric VPN problem with the same input as the DON problem, with the subscribers as sources, and the publishers as sinks. Then, any tree solution for the asymmetric VPN problem can be transformed into a solution of the same cost for the complete tree-tree DON problem.*

Proof. Let T be the tree which is a solution to asymmetric VPN. Since our solution to asymmetric VPN is a tree which is adjacent to all the publishers and subscribers, then every edge in the tree induces a partition of the terminal nodes.

Consider any edge $e \in T$; we decide which trees use e. Let S_1, P_1 be the subscribers and publishers respectively on one side of the partition; likewise let S_2, P_2 be the remaining subscribers and publishers respectively. Now let $a = \min(|S_1|, |P_2|)$ and $b = \min(|P_1|, |S_2|)$. Now a valid demand matrix would be to require a unit flow from a elements of S_1 to a elements of P_2 and to require a unit flow from b elements of S_2 to b elements of P_1. These $a + b$ flows must all cross e since T is a tree, therefore e has multiplicity at least $a + b$.

Now if $|S_1| \leq |P_2|$, then we assign e to be in the trees for the elements of S_1, otherwise e is in the trees for the elements of P_2. Likewise if $|S_2| \leq |P_1|$ we assign e to be in the trees for the elements of S_2, otherwise e is in the trees for the elements of P_1. The number of times we use e is

$$\min(|S_1|, |P_2|) + \min(|S_2|, |P_1|) = a + b$$

So, we don't overuse e.

We next need to show that the edges assigned to a node form a tree. Since the original structure was a tree, we only need to show that the edges assigned to a terminal t are connected. Without loss of generality, suppose that a copy of e was assigned to be in T_s for $s \in S_1$. Let Q be the path in the tree T from e to s. Let $e' \in Q$. Let V' be the vertices on the same side as s of the partition formed by removing e' from T. We know that $S \cap V' \subseteq S_1$. Likewise we know that $P_2 \subseteq P \cap V'^C$. Since $|S_1| \leq |P_2|$ (because $e \in T_s$), then we know $|S \cap V'| \leq |P \cap V'^C|$. So, e' is assigned to be in the tree for nodes in $S \cap V'$. Therefore we have that $Q \subseteq T_s$. Therefore e is connected to s. Hence the graphs formed by our assignment scheme are connected.

Lastly, we must show that for every $s \in S$ and $p \in P$ then T_s and T_p intersect. Consider $s \in S$ and $p \in P$. Let Q be the path in T from p to s and e be an edge in Q. When we look at the S_1, S_2, P_1, P_2 formed by removing e, then either $s \in S_1$ and $p \in P_2$, or $s \in S_2$ or $p \in P_1$. Without loss of generality, assume $s \in S_1$ and $p \in P_2$. Then e is assigned to be in either the tree for all elements of S_1 or for all elements in P_2. So e is in either T_s or T_p. Since T_s and T_p are connected subtrees of the same tree, and $Q \subseteq T_S \cup T_p$ then T_s and T_p meet at some vertex in Q. Therefore, all demands are satisfied and this is a valid solution to the complete tree-tree DON problem. $\square$

This provides an 49.84 approximation algorithm for the complete tree-tree DON problem.