Buy-at-Bulk Network Design with Protection

Spyridon Antonakopoulos
Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974
email: spyros@research.bell-labs.com

Chandra Chekuri
Department of Computer Science, University of Illinois, Urbana, IL 61801
email: chekuri@cs.illinois.edu

Bruce Shepherd
Department of Mathematics and Statistics, McGill University, Montreal, Canada
email: bruce.shepherd@mcgill.ca

Lisa Zhang
Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974
email: ylz@research.bell-labs.com

We consider approximation algorithms for buy-at-bulk network design, with the additional constraint that demand pairs be protected against a single edge or node failure in the network. In practice, the most popular model used in high speed telecommunication networks for protection against failures, is the so-called 1+1 model. In this model, two edge or node-disjoint paths are provisioned for each demand pair. We obtain the first non-trivial approximation algorithms for buy-at-bulk network design in the 1+1 model for both edge and node-disjoint protection requirements. Our results are for the single-cable cost model, which is prevalent in optical networks. More specifically, we present a constant-factor approximation for the single-sink case, and an $O(\log^3 n)$ approximation for the multi-commodity case. These results are of interest for practical applications and also suggest several new challenging theoretical problems.

Key words: Buy-at-bulk; network design; 1+1 protection; approximation algorithms

MSC2000 Subject Classification: Primary: 68Q25 Analysis of algorithms and problem complexity; Secondary: 90B18 Communication networks

OR/MS subject classification: Primary: Networks/graphs, theory; Secondary: Analysis of algorithms, suboptimal algorithms

1. Introduction. The telecommunications industry is the inspiration for numerous network optimization problems. In this paper, we consider buy-at-bulk network design problems that arise in the design and operation of modern optical core networks [6]. These networks are characterized by the following two salient features: (i) very high capacity achieved via DWDM (Dense Wavelength Division Multiplexing) based optical transmission technology and (ii) expensive equipment exhibiting economies of scale. In such networks, each link carries enormous amounts of traffic and hence the failure of a link or a node represents an unacceptable degradation of service. Therefore, fault tolerance is an integral part of the design. Although there are a variety of ways to ensure fault tolerance, one of the most commonly used solutions in optical core networks is to set up, for each commodity, so-called *dedicated* or *1+1* protection. This amounts to reserving a pair of disjoint paths between the source and destination nodes of each commodity. The popularity of the 1+1 model comes from its operational simplicity and high restoration speed.

Disjointness may be defined in several ways, according to requirements of the commodity in question. For instance, the commonly used measures include "site-disjointness", where the two paths do not share any common nodes; edge-disjointness, where the two paths do not share any common links; and cable or fiber-disjointness, where the two paths must use distinct fibers/cables if they go through the same link. In this context, a central problem faced by network operators and equipment vendors is to build a cost-effective and bandwidth-efficient network that supports a multitude of traffic at the desired level of protection. The network operators look to utilize their network resources as efficiently as possible, and the equipment vendors seek to find innovative cost advantages to obtain a competitive edge in bidding for contracts from the network providers. We refer the reader to [6, 29, 25, 28] for in-depth descriptions of the various issues in optical network design.

We give a formal description of the optimization problem that abstracts the above problem. The input consists of an undirected edge-weighted graph $G = (V, E)$, and a set $\mathcal{D}$ of h node pairs $s_1t_1, s_2t_2, \ldots, s_ht_h$ that represent different traffic demands. Each pair has a non-negative demand value $\mathsf{dem}(i)$ that needs to be routed between s_i and t_i and also specifies a protection requirement. Herein we restrict our attention to the 1+1 model, with each demand requiring node-disjoint protection. A feasible solution consists of a collection of path pairs $(P_1, R_1), \ldots, (P_h, R_h)$, where P_i and R_i are internally node-disjoint paths between s_i and t_i and each carries a reserved bandwidth of $\mathsf{dem}(i)$. If these paths induce a requirement of b_e units of bandwidth on edge e of the network, then equipment that can support this requirement has to be purchased.

Now, let us discuss the cost model for purchasing bandwidth on the edges. In this paper, we focus on a simple cost model, namely the single-cable cost model: bandwidth can be purchased in integer multiples of a *cable* of capacity μ. The cost of purchasing a cable on edge e is c_e. Thus, the cost of purchasing a bandwidth of b_e units on edge e is $f_e(b_e) = \lceil b_e/\mu \rceil c_e$. The objective is to minimize the total cost $\sum_e f_e(b_e)$ over all possible choices of $(P_1, R_1), \ldots, (P_h, R_h)$. The single-cable cost function closely models DWDM networks, where each optical fiber carries the same number of wavelengths μ, and each edge e has a cost c_e for deploying one copy of a fiber; the cost accounts for equipment along the edge and at the end nodes of the edge (see [6]). We give an overview of more general cost functions, namely the non-uniform and the uniform multi-cable functions, in Section 1.3.

Observe that, even in the single-cable setting, the buy-at-bulk problem captures, as special cases, some well-known NP-hard connectivity problems such as the minimum-cost Steiner tree and the minimum-cost Steiner forest problems. Moreover, Andrews [1] has shown that even the single-cable problem without protection constraints is hard to approximate to within an $\Omega\big(\log^{1/4-\epsilon} n\big)$ factor; this separates the approximability of the buy-at-bulk problem from those of connectivity problems. In the connectivity setting, survivability and protection constraints have long been studied and include classical problems. Jain [20] devised the important iterative rounding method that yields a 2 approximation algorithm for the survivable network design problem (SNDP), in which the goal is to find a minimum-cost subgraph that satisfies given edge connectivity requirements between each pair of nodes in a graph. In [13] this technique was extended to handle node connectivity, when the requirements are restricted to be in the set $\{0, 1, 2\}$.

Buy-at-bulk network design without protection has received substantial attention in the past decade, including some recent work on super-constant lower bounds in the simplest setting [1], and poly-logarithmic upper bounds in the most general non-uniform setting [7]. On the contrary, the variant with protection has not been so far considered in the literature on approximation algorithms. One reason for this is the difficulty of the buy-at-bulk problem, even without protection constraints. Although the first approximation algorithm for the uniform multi-cable setting appeared in 1997 [3], the algorithm is based on embedding into tree metrics [4] and this approach is not applicable (in a direct fashion at least) to more general settings, nor to connectivity requirements larger than one. It is only recently that alternative algorithms [5, 7] were developed that not only handled the non-uniform cost functions, but also provided new algorithmic approaches and insights. Further, for SNDP, both the iterative rounding method [20] and the earlier primal-dual approach [32] strongly rely on the structural properties of the underlying linear program, which do not hold for the buy-at-bulk problem.

Our primary motivation to study this problem arose while developing a sequence of optical network design tools at Bell Laboratories. We realized the ubiquity of the 1+1 model in practice, the lack of theoretical understanding of protected buy-at-bulk network design and a dearth of useful heuristic methods for the problem. Most algorithms used in practice are based on simple ad hoc methods combining greedy algorithms, local improvement and some enumeration. We hope this paper serves as a starting point in addressing the challenges from the theoretical point of view, as well as in providing insights that lead to more sophisticated and effective heuristics.

1.1 Results. We give approximation algorithms for buy-at-bulk network design in the 1+1 protection model for the single-cable setting. Observe that the 1+1 edge-disjoint protection problem can be reduced in a straightforward fashion to the 1+1 node-disjoint protection problem. In fact, for the edge-disjoint case our arguments can be substantially simplified; however, our focus here is on the node version, as it is the version arising more commonly in practice. We note that hardness results for the unprotected problems carry over to their protected counterparts via simple reductions.

Our first result is for the single-sink problem. This is the special case of the problem where all the pairs have one terminal node in common. In other words, the pairs are $st_1, st_2, \ldots, st_h$ and s is a common sink. We present an $O(1)$ approximation algorithm for it and also establish an $O(1)$ integrality gap for a natural linear programming relaxation.

Our second result is an $O(\log^3 h)$ approximation for the multi-commodity problem; recall that h is the number of demand pairs in the problem instance. In particular, we show that that a ρ approximation for the single-sink problem via a natural LP relaxation yields an $O(\rho \log^3 h)$ approximation for the multi-commodity problem; we then combine this with our result for the single-sink problem. We conjecture that a technique developed in the recent work of Kortsarz and Nutov [22] for the unprotected buy-at-bulk problem applies to our setting as well. If indeed applicable, we would expect it to lead to an improved ratio of $O(\log^2 h)$.

1.2 Overview of algorithmic ideas. The high-level framework of our algorithms is reminiscent of familiar approaches that have been applied to buy-at-bulk network design without protection. Nevertheless, the transition to the protected setting requires some new algorithmic ideas and in the following, we give a brief overview of these.

For the single-sink problem, we take advantage of the single-cable model to start with a good lower bound on the optimal solution: we compute a minimum-cost subgraph H of G that has two node-disjoint paths from each terminal t_i to the sink s. The graph H is used in a clustering procedure to find aggregation points, called *centers*. The idea is to route the flow of each terminal t_i to two distinct centers, via node-disjoint paths. Furthermore, the centers need to receive $\Omega(\mu)$ flow, so that they can route to the sink independently. We remark that clustering and re-routing of flow, as above, is a natural algorithmic paradigm that has been applied in algorithms for single-sink unprotected buy-at-bulk [26, 2, 15, 17, 23]. For the unprotected case, a simple tree based clustering procedure suffices, where each cluster contains terminals with $\Theta(\mu)$ amount of demand, and a center can be chosen arbitrarily from the cluster.

In the protected case, in particular the node-disjoint setting, a straightforward clustering procedure as above does not guarantee that each terminal can find disjoint paths to two distinct centers. We give a clustering procedure that enables us to overcome this difficulty; a distinctive feature of this procedure is that it may create clusters that enclose an arbitrarily large (compared to μ) amount of demand, but in that case the cluster is required to satisfy some special property that can be exploited. Some of the methods we employ in sending flow to two centers are inspired by the work in [8], however the node-disjointness calls for several new technical ideas.

The multi-commodity problem is considerably harder to approach directly, and here we build on the recent algorithmic paradigm developed for the unprotected non-uniform problem [7]. At the high level, the algorithm uses an iterative greedy approach. In each iteration, it finds a partial solution of good *density* amongst the remaining demand pairs, where density is the ratio of the solution cost to the number of pairs connected. In [7] the problem of finding a partial solution of good density is effectively reduced to a single-sink problem. A key step in the reduction is to show the existence of a solution with near-optimal density that also has a *junction* structure: demand pairs connect to each other via a common junction node r, and this enables us to employ single-sink techniques by guessing r.

A similar scheme can be applied to the edge-disjoint protection problem; nevertheless, this does not suffice for the node-protected version. Indeed, even if terminals s_i and t_i individually have two node-disjoint paths to r, they may still not be 2-node-connected. To overcome that difficulty, we show that the basic junction scheme can be extended to use a *pair* of nodes (u, v), as a junction through which multiple pairs connect. We believe this new scheme may offer an interesting idea for new heuristics, which should be evaluated against the current methods that are based on greedy approaches.

1.3 Related work. We briefly discuss closely related work, beginning with a review of more general cost models. The most general cost model considered in the buy-at-bulk problem is the *non-uniform* case, where each $e \in E$ has an associated concave or sub-additive function $f_e : \mathbb{R}^+ \to \mathbb{R}^+$ and $f_e(b_e)$ is the cost of purchasing b_e units of bandwidth on e. In the *uniform* cost model, f_e is restricted to be equal to $c_e f(b_e)$, where c_e is a non-negative constant specified for edge e and $f : \mathbb{R}^+ \to \mathbb{R}^+$ is a sub-additive function common to all edges. This is also called the *multi-cable* cost model, because an alternative and approximately equivalent (within a factor $2+\epsilon$ for any fixed $\epsilon > 0$) definition stipulates that there exists a fixed set of τ cable types, with capacities $\mu_1 < \mu_2 < \cdots < \mu_\tau$ and costs per unit length $c_1 < c_2 < \cdots < c_\tau$, such that the cost-to-bandwidth ratio decreases: $c_1/\mu_1 > c_2/\mu_2 > \cdots > c_\tau/\mu_\tau$. Moreover, the cost of installing a cable of type i on edge e is $c_e c_i$. Hence, to support b_e units of flow on edge e, one needs to purchase the cheapest combination of cables of total capacity at least b_e.

As mentioned above, the buy-at-bulk problem has so far been studied only in the unprotected setting. One of the early approximation algorithm formulations of the problem was due to Salman et al. [26]. In subsequent work, a number of variants have been considered. Even the simplest versions of buy-at-bulk network design, including the single-sink single-cable problem, are APX-hard since they generalize the Steiner tree problem; a $(\rho_{\mathrm{ST}}+2)$-approximation is given in [18], however, where ρ_{ST} is the approximation ratio available for Steiner tree.

Regarding the uniform multi-commodity problem, Awerbuch and Azar [3] showed that it is easy to solve on a tree and then reduced the problem on general graphs to one on a tree using embeddings into random tree metrics [4, 12], thus obtaining an $O(\log n)$ approximation. For the uniform single-sink problem, Andrews and Zhang [2] gave an approximation ratio that was independent of the number of nodes (but did depend on the cost function); an $O(1)$ approximation was first achieved by Guha et al. [15], with subsequent refinements and improvements in the ratio [30, 17, 21, 14]. For a special case of the multi-commodity problem called the rent-or-buy problem, an $O(1)$ approximation is known [17].

For the non-uniform single-sink problem, Meyerson et al. [24] presented an $O(\log n)$ approximation, while Charikar and Karagiozova [5] gave an $\exp\big(O(\sqrt{\log n \log\log n})\big)$ approximation for the non-uniform multi-commodity problem. More recently, the first poly-logarithmic approximation was obtained in [7], which also introduced the junction scheme that we now extend. The ratio can be improved to $O\big(\log^3 h\big)$ if demand values are polynomially bounded with respect to h [22].

Andrews [1] showed that there is no $O\big(\log^{1/2-\epsilon} n\big)$ approximation algorithm for the non-uniform multi-commodity problem, unless NP has efficient randomized algorithms. In the uniform case, including the single-cable model, the hardness factor becomes $O\big(\log^{1/4-\epsilon} n\big)$. Moreover, for the single-sink non-uniform problem a hardness factor of $O(\log\log n)$ is known due to Chuzhoy et al. [10].

Connectivity problems have a rich history in classical combinatorial optimization, and there is a vast literature on the subject. We refer to Schrijver [27] for exact algorithms and classical results and [31, 19, 20, 13] for pointers to approximation algorithms. In particular, Jain [20] and Fleischer et al. [13] present 2 approximation algorithms for SNDP and the element connectivity problem (which generalizes SNDP), respectively. In [13] a 2 approximation algorithm is also obtained for the node-connectivity version of SNDP when the requirements are restricted to lie in the set $\{0, 1, 2\}$; we make use of this algorithm.

2. Single-sink buy-at-bulk with protection. An instance of the node-protected single-sink problem consists of a graph $G = (V, E)$, a sink node $s \in V$, a set of terminals $\mathcal{T} = \{t_1, t_2, \ldots, t_h\} \subseteq V \setminus s$, and a demand function $\mathsf{dem} : \{1, 2, \ldots, h\} \to \mathbb{N}^*$, where $\mathbb{N}^*$ is the set of positive integers. We use $\mu \in \mathbb{N}^*$ throughout to denote the capacity of the cable that can be installed in integral copies on any edge $e \in E$, at a cost c_e per cable. Thus, carrying bandwidth b_e on e costs $\lceil b_e/\mu \rceil c_e$.

Our algorithm consists of three high level steps that follow the outline given in Section 1.

- **Connectivity:** Find a subgraph $H = (V_H, E_H)$ of G such that each t_i has two node-disjoint paths to s in H.
- **Clustering:** Partition the node set V_H into disjoint subsets $X_1, X_2, \ldots, X_l$ called *clusters*, such that for $1 \le j \le l$ the induced subgraph $H[X_j]$ is connected and $\mathsf{dem}(X_j) \ge \mu$, where $\mathsf{dem}(X_j) = \sum_{t_i \in X_j} \mathsf{dem}(i)$. Clusters exhibit additional properties that facilitate analysis.
- **Routing:** Use the clusters to identify a node set $\mathcal{S} \subseteq V_H$, whose elements are called *centers*. For each terminal $t_i \in \mathcal{T}$, send $\mathsf{dem}(i)$ flow to each of two distinct centers in $\mathcal{S}$ using node-disjoint paths, such that in total every center receives $\Omega(\mu)$ flow from terminals. Then, independently for each center $x \in \mathcal{S}$, find the cheapest two node-disjoint paths from x to s and route all of the flow accumulated at x to s along these paths.

The correctness of the above scheme is implied by the proposition below.

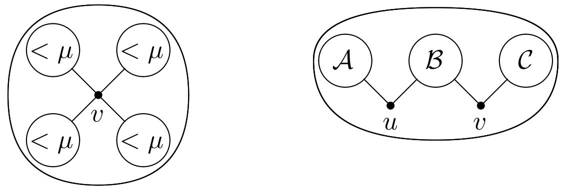

Figure 1: A typical star-like cluster with 4 components (left) and a typical twin cluster with two special nodes u, v (right).

PROPOSITION 2.1 *Let P and R be paths from terminal t to distinct centers x_1 and x_2, respectively, such that P and R are node-disjoint except at t. Let P_{x_1}, R_{x_1} be internally node-disjoint paths from x_1 to s, and P_{x_2}, R_{x_2} be internally node-disjoint paths from x_2 to s. Then, there exist two internally node-disjoint paths from t to s in $P \cup R \cup P_{x_1} \cup R_{x_1} \cup P_{x_2} \cup R_{x_2}$.*

PROOF. It suffices to show that there is no cut vertex $\tilde{v}$ in $G_t = P \cup R \cup P_{x_1} \cup R_{x_1} \cup P_{x_2} \cup R_{x_2}$ separating t and s. To the contrary, suppose there exists such a $\tilde{v}$ and consider the connected components of $G_t \setminus \tilde{v}$. If x_1 belonged to the same component as t, then P_{x_1} and R_{x_1} would both have to pass through $\tilde{v}$ to reach s, which is a contradiction. Therefore, either x_1 coincides with $\tilde{v}$ or is in a different component than t; a similar statement is true for x_2. Since t can reach x_1 and x_2 via the paths P and R, both P and R must then contain $\tilde{v}$, whether as an endpoint or as an internal node, which is impossible since P, R do not share any nodes other than t. □

2.1 Connectivity. We apply the 2 approximation algorithm from [13] for the node-connectivity version of the survivable network design problem on G, with a connectivity requirement of 2 between s and t_i, for each $t_i \in \mathcal{T}$, and 0 for every other pair of nodes. Let $H = (V_H, E_H)$ be the subgraph returned. We install one cable on each edge of H and hence $\mathsf{cost}(H) \leq 2\,\mathsf{cost}(\text{OPT}_{\text{SS}})$, where OPT_{SS} is the optimal solution to the node-protected single-sink problem.

If $\sum_{i=1}^{h} \mathsf{dem}(i) \leq \mu$, then H itself immediately qualifies as a solution to our problem, so we are done. The remainder of the algorithm deals with the case where $\sum_{i=1}^{h} \mathsf{dem}(i) > \mu$. For simplicity, we henceforth assume that H is 2-node-connected, because the clustering and routing procedures can be applied to each 2-node-connected component of H separately.

2.2 Clustering. We describe a method to partition H (in fact, any 2-node-connected node-weighted graph) into clusters, as mentioned earlier. A cluster X is called *small* if $\mathsf{dem}(X) < \mu$; *normal* if $\mu \leq \mathsf{dem}(X) \leq 2\mu$; and *jumbo* if $\mathsf{dem}(X) > 2\mu$. Ideally, we would like to partition V_H so that all clusters are normal. However, this is not always possible. Instead, we allow jumbo clusters in the partition, as long as they possess certain structural properties.

In particular, a jumbo cluster X is *star-like* if and only if there exists a *special node* $v \in X$ such that every connected component of $H[X \setminus v]$ contains $< \mu$ demand. Observe that a jumbo cluster with exactly one terminal (with $> 2\mu$ demand) and any number of non-terminal nodes satisfies this definition vacuously. Additionally, a jumbo cluster X is *twin* if and only if there exists a *set of special nodes* $W \subseteq X$ with $|W| \geq 2$, such that for each special node $v \in W$ the following two properties hold: (a) there is a connected component of $H[X \setminus v]$ that includes all nodes in $W \setminus v$ and contains $< 2\mu$ demand, and (b) the union of all the other components of $H[X \setminus v]$ together with v contain $< \mu$ total demand. Figure 1 provides a visualization of the above definitions.

REMARK 2.1 *A twin cluster may contain at most 3μ demand, as a direct consequence of the definition, whereas there is no upper bound on the amount of demand contained in a star-like cluster.*

The clustering result below is central to the routing arguments used in our algorithm. As the proof of the lemma is somewhat involved and the routing procedure of the third step relies solely on the statement of the lemma, we defer the proof to Section 2.5, after we present the algorithm in its entirety.

Lemma 2.1 *The node set V_H can be partitioned into node-disjoint clusters $X_1, X_2, \ldots, X_l$ in polynomial time, such that for $1 \le j \le l$: (a) the induced subgraph $H[X_j]$ is connected; (b) $\mathsf{dem}(X_j) \ge \mu$; and (c) if $\mathsf{dem}(X_j) > 2\mu$, then X_j is either star-like or twin.*

2.3 Routing. We now describe a scheme to implement the routing step of the algorithm using the clustering of H. That involves several phases, and the analysis goes hand-in-hand with each phase. Given some partition of V_H into clusters $X_1, X_2, \ldots, X_l$, we say that an edge $e = uv$ is an *inter-cluster* edge if its endpoints lie in distinct clusters. Otherwise, the edge is called *intra-cluster*. A node $v \in X_j$ is a *border node* of cluster X_j if and only if it is incident to some inter-cluster edge.

Phase 1: We process each cluster X_j separately. First of all, take a spanning tree T_j of $H[X_j]$ and find a *balanced node separator* v_j of T_j, with respect to the amount of demand. In other words, v_j is such that every connected component of $T_j \setminus v_j$ contains at most $\mathsf{dem}(X_j)/2$ demand. (Any balanced node separator of T_j will do, even if it is not unique.) Moreover, call v_j the *center* of X_j. A few noteworthy special cases:

- If X_j contains exactly one terminal, the latter is obviously the only choice for v_j. On top of that, no further processing is required for X_j in this phase of the routing.
- If X_j is star-like, its special node is a balanced separator of every spanning tree of $H[X_j]$, so we choose it as v_j by default.
- If X_j is twin, observe that none of its special nodes qualify as balanced separators. Indeed, for any such special node v, the connected component of $H[X_j \setminus v]$ that includes all other special nodes must contain $> \mathsf{dem}(X_j) - \mu$ demand, which exceeds $\mathsf{dem}(X_j)/2$ because $\mathsf{dem}(X_j) > 2\mu$. Clearly, then, the center v_j of X_j has to lie somewhere within that component.

Proposition 2.2 *For each terminal $t_i \in X_j \setminus v_j$, there exist two internally node-disjoint paths $P_1(t_i)$ and $P_2(t_i)$ using edges of $H[X_j]$, both starting from t_i, such that $P_1(t_i)$ ends at v_j and $P_2(t_i)$ ends at a border node $b(t_i)$ of X_j.*

Proof. Create a graph $\mathcal{G}$ from H as follows: contract all nodes of $V_H \setminus X_j$ into a node v^*, then add a new node v_0 and new edges $v_j v_0$, $v^* v_0$. Since $H[X_j]$ is connected, neither v^* nor v_0 is a cut vertex of $\mathcal{G}$. Furthermore, if $u \in X_j$ is a cut vertex of $\mathcal{G}$, then it is also a cut vertex of H, contradicting H's biconnectivity. Therefore, $\mathcal{G}$ has no cut vertices either, i.e. it is also 2-connected. Hence, there exist two internally node-disjoint paths from t_i to v_0 in $\mathcal{G}$, of which one goes through v_j and the other through v^*. Denote the former path by $P_1(t_i)$ and the latter by $P_2(t_i)$. After deleting the last edge of $P_1(t_i)$ and the last two edges of $P_2(t_i)$, these two paths satisfy all required properties. □

For each $t_i \in X_j \setminus v_j$, consider the paths $P_1(t_i)$, $P_2(t_i)$ implied by the above proposition. Then, extend $P_2(t_i)$ by adding an inter-cluster edge $\big(b(t_i), b'(t_i)\big)$, where $b'(t_i)$ belongs to some other cluster $X_{j'}$. We refer to $b'(t_i)$ as the *entry point* of t_i to $X_{j'}$. Send flow equal to $\mathsf{dem}(i)$ along each of $P_1(t_i)$, $P_2(t_i)$. For any subset $\hat{\mathcal{T}} \subseteq \mathcal{T}$ of terminals, we refer to the elements of $\{P_1(t_i) \mid t_i \in \hat{\mathcal{T}}\}$ as the P_1 paths of the terminals in $\hat{\mathcal{T}}$. P_2 paths are similarly defined.

Lemma 2.2 *In Phase 1, every intra-cluster edge of H carries a flow of at most 3μ and every inter-cluster edge carries flow of at most 4μ (in particular, at most 2μ flow from each cluster its endpoints belong to).*

Proof. We first bound the total flow on any intra-cluster edge induced by this routing phase. If X_j is normal, then the total flow carried on an edge e is at most 2μ, because $\mathsf{dem}(X_j) \le 2\mu$ and for every terminal $t_i \in X_j$ at most one of $P_1(t_i)$ and $P_2(t_i)$ passes through e. If X_j is star-like, on the other hand, the maximum flow per edge is $< \mu$, since the paths originating in one component of $H[X_j \setminus v_j]$ have no common edges with paths originating in another component, and each component has $< \mu$ demand. Finally, if X_j is twin with special node set W_j, the maximum flow per intra-cluster edge equals $\mathsf{dem}(X_j) < 3\mu$. However, for any $v \in W_j$, note that the P_2 paths of the terminals contained in each small component of $H[X_j \setminus v]$ must pass via a border node within that same component. The reason is that the corresponding P_1 paths need to go through v, to reach the center v_j. As a result, regardless of how other terminals of X_j are routed, no more than 2μ flow from this cluster is routed via any one of its border nodes, and hence via any inter-cluster edge. □

Phase 2: Again, we examine each X_j individually, but now the focus is on *foreign* flow, i.e. flow that arrives at X_j on P_2 paths of terminals in other clusters. Recall from Phase 1 the spanning tree T_j of $H[X_j]$, and root it at v_j. We will extend the P_2 paths along T_j in a greedy manner, as follows. We process the nodes in T_j in a bottom-up fashion starting from the leaves; a node w is processed only after all its descendants in T_j have been processed.

When processing a node $w \neq v_j$, let $S(w)$ be the set of terminals that send foreign flow to w, whose total amount we denote by $F(w)$. If $F(w) < 4\mu$, then this flow is forwarded to w's parent node $p(w)$, which means that the P_2 paths of the terminals in $S(w)$ are extended up to $p(w)$. Naturally, the forwarded flow is later taken into consideration when processing $p(w)$.

Otherwise, consider the P_2 paths of the terminals in $S(w)$, and in particular the path segments between their respective entry points to X_j and w. These segments define a tree $T(w)$, which is a subgraph of T_j. Observe that every edge of $T(w)$ currently carries at most 4μ foreign flow. We find a balanced separator x_w of $T(w)$, assuming that the weight of a node equals the total demand of the terminals in $S(w)$ for which it is an entry point. Then, x_w becomes an *auxiliary center*, and if $x_w \neq w$ we re-route the P_2 paths along edges of $T(w)$, so that they all end up at x_w instead. This re-routing may take place in conjunction with the search for x_w, as described below.

We begin by rooting the tree $T(w)$ at w and considering w as a candidate for balanced node separator. Indeed, if w receives at most $F(w)/2$ foreign flow from each one of its children in $T(w)$, then we are done. Otherwise, let w' be the unique child that sends $> F(w)/2$ flow to w. Among all P_2 paths arriving at w, we truncate at w' those that pass via that node, and extend all others from w to w'. Clearly, the total flow of the extended paths is $< F(w)/2$, which is less than than of the truncated paths. Thus, there is now less flow on edge ww' than before, so every edge of $T(w)$ still carries at most 4μ foreign flow. Moreover, w' qualifies as a balanced node separator, unless one of its own children (say w'') sends $> F(w)/2$ flow to it. In the former case, we just declare w' as x_w; else, we repeat the above path rearrangements, from w' to w'' this time, and so on. After $O(n)$ iterations, this procedure determines x_w and re-routes the P_2 paths from w to x_w, while keeping the flow carried by every edge of $T(w)$ bounded by 4μ. Hereafter, these paths are fixed and the processing of w is complete.

Finally, the cluster center v_j is processed last. If $F(v_j) < 4\mu$, then no further processing is necessary, else the procedure of the previous paragraph is applied. Note that the auxiliary center thus created may coincide with v_j, but we treat them as separate entities to simplify the subsequent analysis. In any case, the following lemma is readily established.

Lemma 2.3 *In Phase 2, each intra-cluster edge of H carries at most 4μ foreign flow.*

Corollary 2.1 (of Lemmata 2.2 & 2.3) *In Phases 1 and 2 together, the aggregate flow on every edge of H does not exceed $\max\{3\mu+4\mu, 4\mu\} = 7\mu$, and therefore the cost of this routing is bounded by $7\,\mathsf{cost}(H)$.*

Phase 3: For a cluster X_j with center v_j, denote by $\alpha(v_j)$ the total flow accumulated in v_j during the first two phases. Note that $\alpha(v_j)$ includes flow coming from terminals of $X_j \setminus v_j$, foreign flow (which cannot exceed 4μ, as per the discussion above), plus $\mathsf{dem}(i_j)$ if v_j is itself a terminal with index i_j. Likewise, for an auxiliary center x_w we define $\alpha(x_w)$ as the total foreign flow accumulated in x_w. Moreover, let

$$\beta(x) = \begin{cases} 6\mu & \text{if } x \text{ is the center of a normal cluster;} \\ 7\mu & \text{if } x \text{ is the center of a twin cluster;} \\ \mu\lceil\alpha(x)/\mu\rceil & \text{if } x \text{ is the center of a star-like cluster or an auxiliary center.} \end{cases}$$

Clearly, $\beta(x) \geq \alpha(x)$ and $\beta(x) \geq 3\mu$. Consider a new instance I' of the node-protected single-sink problem on the graph G, with sink s and a terminal set $\mathcal{T}' = \{t'_1, t'_2, \ldots t'_{h'}\}$ whose elements are all the cluster centers and auxiliary centers. Additionally, the demand of a terminal $t'_i \in \mathcal{T}'$ is given by $\beta(t'_i)$, which equals 3μ or a larger multiple of μ. For this particular instance, an optimal solution OPT'_{SS} can be found in polynomial time: for every terminal t'_i, find the shortest cycle containing t'_i and s, and then route $\beta(t'_i)$ flow along each of the two paths from t'_i to s induced by the cycle; note that this is the same as finding two internally node disjoint paths from t'_i to s with minimum total length. Observe that, as a result of the routing, the amount of flow on any edge is an exact multiple of μ.

To show the optimality of OPT'_{SS}, we formulate a linear programming relaxation LP1 for the protected single-sink buy-at-bulk problem instance I'. In LP1, $f_i^{u\to v}$ (respectively, $f_i^{v\to u}$) is a variable that indicates whether or not edge $uv \in E$ is used in the solution to carry from u to v (respectively, from v to u) flow originating at t'_i. Moreover, constraint (1d) enforces node-disjointness. Owing to the integrality of single-sink min-cost flow, we easily deduce that OPT'_{SS} is an optimal solution for said LP relaxation and, *a fortiori*, for the buy-at-bulk instance I'.

$$\mathsf{LP1}: \quad \min \quad \sum_{i=1}^{h'} \frac{\beta(t'_i)}{\mu} \sum_{uv\in E} c_{uv}(f_i^{u\to v} + f_i^{v\to u}) \tag{1a}$$

$$\text{s.t.} \quad \sum_{v:uv\in E} (f_i^{u\to v} - f_i^{v\to u}) = 0 \qquad \forall 1 \le i \le h',\ u \in V\setminus\{t'_i, s\} \tag{1b}$$

$$\sum_{v:t'_i v\in E} \left(f_i^{t'_i\to v} - f_i^{v\to t'_i}\right) = 2 \qquad \forall 1 \le i \le h' \tag{1c}$$

$$\sum_{v:uv\in E} f_i^{u\to v} \le 1 \qquad \forall 1 \le i \le h',\ u \in V\setminus\{t'_i, s\} \tag{1d}$$

$$f_i^{u\to v}, f_i^{v\to u} \ge 0 \qquad \forall 1 \le i \le h',\ uv \in E \tag{1e}$$

The next lemma relates the total cost of OPT'_{SS} to that of OPT_{SS}.

LEMMA 2.4 *For the optimal solution* OPT'_{SS} *to instance* I', $\mathsf{cost}(\text{OPT}'_{\text{SS}}) < 21\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\text{OPT}_{\text{SS}})$.

PROOF. Since OPT'_{SS} is optimal for LP1 on the instance I', it suffices to construct a – hypothetical – feasible *fractional* solution SOL' for LP1 with cost not exceeding $21\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\text{OPT}_{\text{SS}})$. In SOL', the flow from each terminal $t'_i \in \mathcal{T}'$ may be split among several paths leading to s, provided that no more than half this flow passes via any single node other than t_i and s, as stipulated by (1d). More specifically, every t'_i sends its flow to one or more of $t_1, t_2, \ldots, t_h$ (i.e. the terminals of the original instance I), which is then routed to s. Let us examine cluster centers and auxiliary centers separately; henceforth, any reference to terminals implies those of the original instance.

In a cluster X_j, the center node v_j sends $2\beta(v_j)$ flow to terminals of X_j along edges of the spanning tree T_j of $H[X_j]$, so that each terminal $t_i \in X_j$ receives flow $2\beta(v_j)\cdot \mathsf{dem}(i)/\,\mathsf{dem}(X_j)$. Of course, if v_j is itself a terminal, its own share of flow is not routed anywhere at this point. We claim that each terminal t_i thus receives at most $12\,\mathsf{dem}(i)$ flow. This holds because if X_j is a normal cluster then $2\beta(v_j)/\,\mathsf{dem}(X_j) < 2\cdot 6\mu/\mu = 12$; if X_j is twin then $2\beta(v_j)/\,\mathsf{dem}(X_j) < 2\cdot 7\mu/(2\mu) = 7$; and if X_j is star-like then $2\beta(v_j)/\,\mathsf{dem}(X_j) < 2\mu\big\lceil\big(\mathsf{dem}(X_j)+4\mu\big)/\mu\big\rceil/\,\mathsf{dem}(X_j) < 7$, since $\mathsf{dem}(X_j) > 2\mu$.

We also argue that every edge of $H[X_j]$ carries $\le 7\mu$ flow, which implies that the overall cost of this routing step does not exceed $7\,\mathsf{cost}(H)$. On one hand, since v_j is the balanced separator of T_j, the flow on any edge and through any node (except v_j) does not exceed $\beta(v_j)$. Therefore, if X_j is a normal or twin cluster, the flow on every edge is at most $\beta(v_j) \le 7\mu$. On the other hand, if X_j is a star-like cluster, each component of $H[X_j \setminus v_j]$ originally contained $< \mu$ demand, so now it receives from v_j less than $2\beta(v_j)\cdot\mu/\,\mathsf{dem}(X_j) < 7\mu$ flow in total. This means that the flow on any edge of $H[X_j]$ is $< 7\mu$.

The last step is to route all flow that has accumulated at $t_1, t_2, \ldots, t_h$ to s. Each terminal t_i sends the flow that it receives (at most $12\,\mathsf{dem}(i)$ from the above discussion) to s, following the same two node-disjoint paths used in OPT_{SS}. We don't know OPT_{SS}, of course, however that is no obstacle for the purpose of our proof. It is easy to see that scaling up OPT_{SS} by 12 provides enough capacity to perform this routing, thus we derive that:

CLAIM 1. The partial cost of SOL' due to cluster centers is at most $7\,\mathsf{cost}(H) + 12\,\mathsf{cost}(\text{OPT}_{\text{SS}})$.

Moreover, consider an auxiliary center x_w in some cluster X. Recall from the description of Phase 2 the definitions of $S(w)$ and $T(w)$. Now, x_w sends $2\beta(x_w)$ flow to terminals in $S(w)$, using their P_2 paths in the opposite direction.

Back in Phase 2, the flow due to these paths on any edge and through any node of $T(w)$ (except x_w) did not exceed $\min\{4\mu, \alpha(x_w)/2\}$, since x_w is the balanced separator of $T(w)$. Additionally, as a corollary of the arguments in the proof of Lemma 2.2, in Phase 1 said paths put no more than 2μ flow on any edge and through any node not in $H[X]$. In SOL', by comparison, the amount of flow is increased $2\beta(x_w)/\alpha(x_w)$ times. Nevertheless, it is true that at most $\beta(x_w)$ flow passes via any single node, whether in X or not, because $2\beta(x_w)/\alpha(x_w) \cdot \min\{4\mu, \alpha(x_w)/2\} \leq \beta(x_w)$ and $2\beta(x_w)/\alpha(x_w) \cdot 2\mu \leq \beta(x_w)$, as a consequence of $\alpha(x) \geq 4\mu$.

Next, we bound the total amount of flow from auxiliary centers carried by an arbitrary intra-cluster edge $e \in H$. First of all, let $X \in \{X_1, X_2 \dots, X_l\}$ be the unique cluster such that $H[X] \ni e$. There may be at most one auxiliary center $x_w \in X$ such that $e \in T(w)$, and the flow from x_w does not exceed

$$2\beta(x_w)/\alpha(x_w) \cdot \min\{4\mu, \alpha(x_w)/2\} = \min\{8\mu \cdot \beta(x_w)/\alpha(x_w), \beta(x_w)\} < 9\mu \ .$$

Furthermore, e may carry flow from auxiliary centers in other clusters that is routed along the P_2 paths of terminals in X. As mentioned earlier, the proof of Lemma 2.2 implies that in Phase 1 the amount of this flow is $< 2\mu$. In SOL' it becomes $< \frac{5}{2} \cdot 2\mu = 5\mu$, since for any auxiliary center x we have $\alpha(x) \geq 4\mu$, which implies that $2\beta(x)/\alpha(x) < \frac{5}{2}$. Hence, the cumulative upper bound on the flow is $9\mu + 5\mu = 14\mu$.

On the other hand, every arbitrary inter-cluster edge of H carried at most 4μ flow to auxiliary centers in Phase 1, so in SOL' it carries $< \frac{5}{2} \cdot 4\mu = 10\mu$ flow, by a similar reasoning. As a result, the flow on any edge of H is at most 14μ. Finally, routing the flow from the terminals to s is feasible by scaling up OPT_{SS} by $2\beta(x)/\alpha(x) < \frac{5}{2}$, at a cost less than $3\,\mathsf{cost}(\text{OPT}_{\text{SS}})$.

Claim 2. The partial cost of SOL' due to auxiliary centers is at most $14\,\mathsf{cost}(H) + 3\,\mathsf{cost}(\text{OPT}_{\text{SS}})$.

Claims 1 and 2 complete the proof. □

Corollary 2.1 and Lemma 2.4 together imply that the overall cost of our solution is at most

$$7\,\mathsf{cost}(H) + 21\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\text{OPT}_{\text{SS}}) = 28\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\text{OPT}_{\text{SS}}) \leq$$
$$\leq 71\,\mathsf{cost}(\text{OPT}_{\text{SS}}) \ .$$

It is also relatively straightforward to verify its feasibility, by applying Proposition 2.1. Therefore,

Theorem 2.1 *The node-protected single-sink single-cable buy-at-bulk problem is $O(1)$ approximable.*

2.4 LP relaxation and its integrality gap. So far, we have evaluated buy-at-bulk solutions under a *cable capacity* cost model. In other words, the cost of using edge e is $c_e\lceil b_e/\mu\rceil$, where μ is the cable capacity, b_e is the flow on e and c_e is the cable cost for e. Compare this model with the *fixed + incremental* cost model (FI), otherwise known as the *cost-distance* model [2, 24]. In the FI model, each edge e has a fixed cost c_e and an incremental cost ℓ_e. Additionally, the cost of purchasing bandwidth b_e on e is given by $f_e(b_e) = c_e + \ell_e \cdot b_e$. When restricted to the single-cable case, the FI model specializes to having $\ell_e = c_e/\mu$ for each e, so that $f_e(b_e) = c_e(1 + b_e/\mu)$. Since $c_e\lceil b_e/\mu\rceil \leq c_e(1 + b_e/\mu) \leq 2c_e\lceil b_e/\mu\rceil$, the cost of a solution under the single-cable FI model is at most twice that under the cable capacity model.

Let us formulate a linear programming relaxation $\mathsf{LP2}$ for the protected single-sink buy-at-bulk problem under the single-cable FI model. In the formulation, $x(e)$ is a variable that indicates whether or not edge e is in the solution; $\mathcal{Q}_i$ is the collection of simple cycles containing the sink s and the terminal $t_i \in \mathcal{T}$; $f(Q)$ is a variable indicating whether flow from t_i is carried to s using the node-disjoint paths on the cycle $Q \in \mathcal{Q}_i$; and finally $\ell_Q = \sum_{e \in Q} \ell_e$ is the total length of the edges in Q, where the edge length ℓ_e equals the incremental cost c_e/μ per unit flow. Observe that the first term in the objective function (2a) represents the fixed cost, which depends only on which edges are used in the network, while the second term is the incremental cost, that is proportional to the flow carried by these edges.

Theorem 2.2 *The linear program* $\mathsf{LP2}$ *has an integrality gap of $O(1)$ for the single-cable FI cost model.*

We first consider a special case that is useful in the subsequent analysis.

Proposition 2.3 *The linear program* $\mathsf{LP2}$ *has an integrality gap of at most $1 + 1/\xi$, if* $\mathsf{dem}(i) \geq \xi\mu$ *for all $1 \leq i \leq h$.*

Proof. For each terminal t_i, let $\mathsf{dem}(i) = \xi_i\mu$ for some $\xi_i \geq \xi$. Let $Q_i \in \mathcal{Q}_i$ be the cycle containing t_i and s that minimizes the quantity $c_{Q_i} = \sum_{e \in Q_i} c_e$. Route all of t_i's demand along the two paths induced

$$
\begin{aligned}
\mathsf{LP2}: \quad \min \quad & \sum_{e \in E} c_e x(e) + \sum_{i=1}^{h} \mathsf{dem}(i) \sum_{Q \in \mathcal{Q}_i} \ell_Q f(Q) && && \text{(2a)} \\
\text{s.t.} \quad & \sum_{\substack{Q \in \mathcal{Q}_i \\ Q \ni e}} f(Q) \le x(e) && \forall e \in E,\ 1 \le i \le h && \text{(2b)} \\
& \sum_{Q \in \mathcal{Q}_i} f(Q) \ge 1 && \forall 1 \le i \le h && \text{(2c)} \\
& x(e), f(Q) \ge 0 && \forall e \in E,\ Q \in \textstyle\bigcup_i \mathcal{Q}_i && \text{(2d)}
\end{aligned}
$$

by this cycle. We do this for each terminal independently, and this creates a feasible solution of FI cost not exceeding $\sum_i \left(c_{Q_i} + \xi_i \mu \ell_{Q_i}\right) = \sum_i (1+\xi_i) c_{Q_i}$, because $\ell_e = c_e/\mu$ implies that $\ell_{Q_i} = c_{Q_i}/\mu$. However, any fractional solution has incremental cost at least $\sum_i \xi_i \mu \ell_{Q_i} = \sum_i \xi_i c_{Q_i}$. Since $(1+\xi_i)/\xi_i \le 1 + 1/\xi$, the cost of the integral solution is at most $1 + 1/\xi$ times the optimal fractional cost. □

Hence, we immediately observe that $\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}'_{\mathrm{SS}}) \le \frac{4}{3} \mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}'_{\mathsf{LP2}})$, where $\mathrm{OPT}'_{\mathsf{LP2}}$ is the optimal fractional solution to $\mathsf{LP2}$, adapted for the instance I'. This is because all terminals in I' have demand at least 3μ, and because $\mathrm{OPT}'_{\mathrm{SS}}$ is constructed exactly as described in the above proof.

Previously, we obtained an $O(1)$ approximation by giving an algorithm to route the demands and then comparing the cost of the routing to the cost of an optimal (integral) solution. To prove Theorem 2.2, we use the *same* algorithm; however, we compare its cost to $\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$, the cost of an optimum fractional solution $\mathrm{OPT}_{\mathsf{LP2}}$ to $\mathsf{LP2}$. We follow the same notation as in Sections 2.1-2.3.

Lemma 2.5 $\mathsf{cost}(H) \le 2\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$.

Proof. Recall that H is obtained by iterative rounding of the optimal solution to the following LP formulation of the node-connectivity version of the survivable network design problem [13].

$$
\begin{aligned}
\mathsf{LP3}: \quad \min \quad & \sum_{e \in E} c_e x(e) && && \text{(3a)} \\
\text{s.t.} \quad & \sum_{e \in \delta(S,S')} x(e) \ge 2 - |V \setminus (S \cup S')| && \forall S, S' \subseteq V \text{ such that } S \cap S' = \varnothing, && \text{(3b)} \\
& && s \in S, \text{ and } \mathcal{T} \cap S' \ne \varnothing && \\
& 0 \le x(e) \le 1 && \forall e \in E && \text{(3c)}
\end{aligned}
$$

Since constraints (2b)-(2d) imply (3b) and (3c), the value of the optimal solution to $\mathsf{LP3}$ is clearly a lower bound on $\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$, and $\mathsf{cost}(H)$ is at most twice that. Therefore, $\mathsf{cost}(H) \le 2\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$. □

We also give a lemma similar to Lemma 2.4.

Lemma 2.6 $\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}'_{\mathsf{LP2}}) < 22\,\mathsf{cost}(H) + 15\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$.

Proof. Consider the optimal fractional solution $\mathrm{OPT}'_{\mathsf{LP2}}$ to $\mathsf{LP2}$ for I'. We bound its cost by giving a hypothetical solution SOL' such that $\mathsf{cost}_{\mathrm{FI}}(\mathrm{SOL}') < 22\,\mathsf{cost}(H) + 15\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$. In fact, the construction of SOL' is precisely the same as in the proof of Lemma 2.4, and the only issue is how to derive the desired bound in the FI cost model. We briefly sketch a few details. As discussed earlier, in SOL' we route the flow from terminals in I' to the original terminals $t_1, \ldots, t_h$, such that each edge of H carries less than 21μ flow and each t_i receives at most $15\,\mathsf{dem}(i)$ flow. The FI cost of the flow on H is less than $\sum_{e \in H}(c_e + 21\mu \cdot c_e/\mu) \le 22\,\mathsf{cost}(H)$. Moreover, the cost of fractionally routing $15\,\mathsf{dem}(i)$ from each t_i to the source is easily seen to be at most $15\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$: we simply use the optimal solution $\mathrm{OPT}_{\mathsf{LP2}}$ with the demand for each terminal scaled up by a factor of 15. Thus, we have exhibited a feasible fractional solution SOL' of cost not exceeding $22\,\mathsf{cost}(H) + 15\,\mathsf{cost}_{\mathrm{FI}}(\mathrm{OPT}_{\mathsf{LP2}})$. □

We can now prove Theorem 2.2 as follows. The algorithm for instance I uses edges of H in the first two routing phases, and then uses $\mathrm{OPT}'_{\mathrm{SS}}$ in the third phase. The cost of the first two phases is at most

$8\,\mathsf{cost}(H)$, because each edge of H carries at most 7μ flow. Proposition 2.3 and Lemma 2.6 imply that the cost of OPT'_{SS} is less than $\frac{88}{3}\,\mathsf{cost}(H) + 20\,\mathsf{cost}_{\text{FI}}(\text{OPT}_{\mathsf{LP2}})$. Consequently, the overall cost is at most $\frac{112}{3}\,\mathsf{cost}(H) + 20\,\mathsf{cost}_{\text{FI}}(\text{OPT}_{\mathsf{LP2}})$, which by Lemma 2.5 does not exceed $\frac{284}{3}\,\mathsf{cost}_{\text{FI}}(\text{OPT}_{\mathsf{LP2}})$.

2.5 Proof of Lemma 2.1. We need a few more definitions that pertain to a given partition of V_H into clusters $X_1, X_2, \ldots, X_l$. Recall that $v \in X_j$ is a *border node* if and only if there exists a $u \in X_{j'}$, $j' \neq j$, such that the edge $uv \in E_H$. In that case, X_j and $X_{j'}$ are *neighboring* clusters and v is a *neighbor* of $X_{j'}$. Moreover, a node v is called *critical* if and only if: (a) it belongs to a cluster X with $\mathsf{dem}(X) \geq \mu$; (b) at least one of the connected components of $H[X \setminus v]$ contains $< \mu$ demand; and (c) it is a neighbor of one or more small clusters. We say that these clusters *contend* for the critical node. Furthermore, a component of $H[X \setminus v]$ that has $\geq \mu$ demand is called *self-sufficient.*

Initially, let each node of V_H be in a cluster of its own. In each iteration of our clustering procedure, we check which of the following four transformations are feasible, and apply the one with the highest priority. This is repeated until there are are no small clusters left. Note that a critical node may become non-critical, or vice-versa, from one iteration to the next. The transformations are listed below, in order of decreasing priority.

(i) If the total demand in two neighboring clusters is at most 2μ, merge them.

(ii) If a small cluster X has a neighbor node v in a cluster Y such that v is not critical, move v to X.

(iii) Consider a cluster Y with a critical node v. Separate any self-sufficient components of $H[Y \setminus v]$ into new clusters. Keep v and all other components together. If this remaining cluster is small, merge it immediately with a small cluster contending for v, just as in transformation (i).

(iv) If there exists a small cluster X that contends for a critical node v of cluster Y, then create a jumbo cluster by merging X and Y into a new cluster Z.

It is easy to see that a small cluster always has some neighboring cluster, because H is 2-connected and $\sum_{i=1}^{h} \mathsf{dem}(i) > \mu$ (refer to Section 2.1), hence at least one of these transformations applies. We now prove the correctness and running time of the clustering procedure.

Lemma 2.7 *After each iteration there are only small, normal, star-like, and twin clusters.*

Proof. The proof is by induction on the number of iterations. At the beginning, each cluster contains one node. If that node is also a terminal with at least μ demand, then the cluster is either normal or (degenerately) star-like, otherwise it is small. Additionally, the following property is self-evident:

Claim 3. Let X be a star-like (respectively, twin) cluster. If some $X' \subseteq X$ constitutes a jumbo cluster, i.e. $H[X']$ is connected and $\mathsf{dem}(X') > 2\mu$, then X' is also star-like (respectively, twin).

Indeed, because $H[X']$ is connected and $\mathsf{dem}(X') > 2\mu$, any v that is a special node of X must belong to X' as well, since all connected components of $H[X \setminus v]$ contain $< 2\mu$ demand. Moreover, v is also a special node of X', according to the definition of a star-like or twin cluster, respectively. Therefore, Claim 3 is valid.

Consider the above transformations that may be applied in each iteration. The first transformation merges two clusters into a new cluster that is either small or normal. The second transformation removes a non-critical node v from Y to a small cluster X. Y is not small, otherwise it could be merged with X, so by hypothesis it must be either normal, star-like, or twin. Since v is not critical, Claim 3 guarantees that the cluster (or clusters) that remain in Y's place after removing v will still belong to one of the aforementioned three classes, though possibly not the same as before. Furthermore, X may remain a small cluster or become normal or star-like after the addition of v.

The third transformation separates a self-sufficient component $H[Y \setminus v]$ of cluster Y into a cluster of its own. Let Y_1 be this new cluster, and $Y_2 = Y \setminus Y_1$. Clearly, Y is not small. If Y is normal, then Y_1 is also a normal cluster and Y_2 is either normal or small. If Y is a star-like cluster, then v obviously cannot be Y's special node. Thus, Y_1 is either a normal or star-like cluster, by Claim 3, and Y_2 is small. If Y is a twin cluster, then again by Claim 3 and the fact that $\mathsf{dem}(Y) < 3\mu$ it is easy to deduce that Y_1 is either normal or twin and Y_2 is either small or normal. Of course, if Y_2 is small, then it is promptly merged with another small cluster to produce a small or normal cluster.

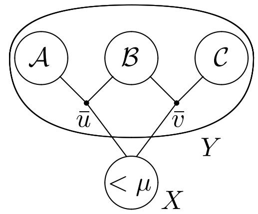

Figure 2: An example for which transformation (iv) (second case) is necessary. Since no other transformation applies, $\mathsf{dem}(\mathcal{A}) + \mathsf{dem}(\bar{u}) + \mathsf{dem}(\mathcal{B}) < \mu$ and $\mathsf{dem}(\mathcal{B}) + \mathsf{dem}(\bar{v}) + \mathsf{dem}(\mathcal{C}) < \mu$. Compare with Figure 1.

For the fourth transformation, we distinguish two cases. First, suppose X contends for a single critical node v of Y. The connected components of $H[Z \setminus v]$ are $H[X]$ and the components of $H[Y \setminus v]$, none of which is self-sufficient, because otherwise transformation (iii) would apply. Therefore, Z is star-like with special node v. Second, suppose X contends for more than one critical nodes of cluster Y (e.g. as in Figure 2). Let $U \subseteq Y$ be the set of critical nodes contended for by X, with $|U| \geq 2$. We say that $w \in U$ is *interesting* if and only if all other nodes in U belong to only one component $H[V_w]$ of $H[Y \setminus w]$, where $V_w \subseteq Y \setminus w$. We claim that that there must exist at least *two* interesting critical nodes $\bar{u}, \bar{v} \in U$, which we prove later. After the merger, then, $H[V_{\bar{u}} \cup X]$ is a connected component of $H[Z \setminus \bar{u}]$ containing $< 2\mu$ demand, since $\mathsf{dem}(V_{\bar{u}}) < \mu$. All other components of $H[Z \setminus \bar{u}]$ are precisely the components of $H[Y \setminus \bar{u}]$ excluding $H[V_{\bar{u}}]$, and hence together with $\bar{u}$ they contain $< \mu$ total demand, otherwise they would form (part of) a self-sufficient component of $H[Y \setminus \bar{v}]$. By symmetry, analogous properties hold for the components of $H[Z \setminus \bar{v}]$, so Z is a twin cluster with at least two special nodes: $\bar{u}$ and $\bar{v}$.

Finally, we establish the claim in the previous paragraph. Take a spanning tree $T(Y)$ of $H[Y]$ and consider its leaves. If a leaf node of $T(Y)$ is not in U, then delete it and repeat this procedure on the remaining tree. This eventually produces a tree $T'(Y) \subseteq T(Y)$ that contains every node in U and whose leaves all belong to U. It is easy to see that every leaf of $T'(Y)$ is an interesting critical node, and that $T'(Y)$ has at least two leaves, being a tree with no less than $|U| \geq 2$ nodes. □

Lemma 2.8 *The clustering procedure terminates in at most* $2|V_H|^2$ *iterations.*

Proof. Denote by n_1 the combined number of normal, star-like, and twin clusters, by n_2 the number of small clusters, and by n_3 the number of nodes in small clusters. We examine how these quantities change during each iteration. Obviously, at all times $0 \leq n_\zeta \leq |V_H|$, $\zeta = 1, 2, 3$, since every cluster contains at least one node. Observe that n_1 never decreases and n_2 never increases. Furthermore, if both n_1 and n_2 are left unchanged during an iteration, then n_3 strictly increases; this latter case pertains only to transformations (ii) and (iii). Hence, there can be no more than $|V_H|$ consecutive iterations in which both n_1 and n_2 remain constant, which implies that the procedure performs at most $2 \cdot |V_H|^2$ iterations in total. □

Each transformation uses basic graph theoretic operations and can be implemented in polynomial time. This completes the proof of Lemma 2.1.

3. From single-sink to multi-commodity. In this section we consider the node-protected multi-commodity buy-at-bulk problem. We establish that a ρ approximation for the single-sink problem implies an $O\big(\rho \log^2 h \log D\big)$ approximation for the multi-commodity problem via a natural LP relaxation, where $D = \sum_i \mathsf{dem}(i)$. Note that our result can be applied to the general FI model, i.e. even without the single-cable restriction $\ell_e = c_e/\mu$ that was introduced in Section 2.4. This is significant because the general FI model is essentially equivalent to the non-uniform model. However, in the single-cable model that is of interest here, the dependence on D can be removed and the ratio becomes $O\big(\rho \log^3 h\big)$. The results in Section 2.4 imply that $\rho = O(1)$ for the single-cable model, and thus we obtain an $O\big(\log^3 h\big)$ approximation for the multi-commodity single-cable problem. To simplify the exposition, throughout this section we assume unit demands ($\mathsf{dem}(i) = 1$ for $1 \leq i \leq h$) and prove the ratio of $O\big(\log^3 h\big)$ in this setting. The extension to the general case of arbitrary demands is demonstrated later.

We use the algorithmic paradigm from [7], as outlined in Section 1. The main technical ingredient is an extension of the *junction tree* concept from [7]. We define a structure which we call a *junction-structure*, more precisely a two-node junction-structure, as shown below.

To begin with, let us formulate the objective function for the multi-commodity problem in the general FI model. Recall that c_e and ℓ_e are the fixed and incremental cost (henceforth also called *length*) of e. Given two nodes a, b and a subgraph H of G, we let $\ell 2_H(a,b)$ be the minimum-length cycle of H containing a and b. Note that this is the same as the minimum length of two node-disjoint paths between a and b in H. Then, the objective is to find $E' \subseteq E$ that minimizes $\sum_{e \in E'} c_e + \sum_{s_i t_i \in \mathcal{D}} \ell 2_{G[E']}(s_i, t_i)$.

A *two-node junction* is an unordered pair of nodes u, v with $u \neq v$, and is denoted by $\widehat{uv}$. We say that a node x is *two-connected to a junction* $\widehat{uv}$ in a graph H if there exist paths P and R in H that connect x to u and x to v, respectively, and are node-disjoint (except at x). Denote by $\ell 2_H(x, \widehat{uv})$ the minimum total length of two such paths.

Proposition 3.1 *Let H be a graph in which s_i and t_i are two-connected to a junction $\widehat{uv}$ and there exists a cycle containing u and v. Then there is a cycle in H containing s_i and t_i of length at most $\ell 2_H(s_i, \widehat{uv}) + \ell 2_H(t_i, \widehat{uv}) + \ell 2_H(u, v)$.*

Proof. Consider the minimum-length cycle Q_{uv} implied by $\ell 2_H(u,v)$ and the paths $P_{s_i u}, P_{s_i v}$ and $P_{t_i u}, P_{t_i v}$ implied by $\ell 2_H(s_i, \widehat{uv})$ and $\ell 2_H(t_i, \widehat{uv})$, respectively. It suffices to show that in the graph H' defined by $Q_{uv} \cup P_{s_i u} \cup P_{s_i v} \cup P_{t_i u} \cup P_{t_i v}$, there are two internally node disjoint paths between s_i and t_i. Suppose not, then there is a cut vertex w in H' such that s_i and t_i are in two different components of $H' \setminus w$. Since u, v have a cycle containing them in H', either both of them are in the same connected component of $H' \setminus w$ or w is one of u, v. In either case, this implies that s_i or t_i cannot reach both u and v via disjoint paths, a contradiction to our assumption that both are two-connected to $\widehat{uv}$. □

Given a subset $\mathcal{D}'$ of the demands, a *junction-structure for* $\mathcal{D}'$ rooted at a two-node junction $\widehat{uv}$ is a subgraph $H(\widehat{uv})$ of G satisfying the requirements of Proposition 3.1 for every s_i and t_i such that $s_i t_i \in \mathcal{D}'$. Hence, we can connect the pairs in $\mathcal{D}'$ using edges of $H(\widehat{uv})$, with cost no more than

$$\sum_{e \in E(H(\widehat{uv}))} c_e + \sum_{s_i t_i \in \mathcal{D}'} \left(\ell 2_{H(\widehat{uv})}(s_i, \widehat{uv}) + \ell 2_{H(\widehat{uv})}(t_i, \widehat{uv}) + \ell 2_{H(\widehat{uv})}(u, v) \right). \quad (4)$$

Quantity (4) is called – somewhat abusively – the *cost* of junction-structure $H(\widehat{uv})$.

Given a multi-commodity instance with unit demands, let OPT_{MC} be the optimal solution. We first show the existence of a junction-structure of density $O\left(\frac{\log h}{h}\right) \mathsf{cost}(\text{OPT}_{\text{MC}})$, where density is defined to be the ratio of the cost of the junction-structure to the number of demand pairs connected by it. Although this existence proof builds upon the ideas in [7], to ensure node-disjointness we need a more sophisticated argument in Lemma 3.2. Using the $O(1)$ integrality gap of the single-sink problem, we further show how to find a junction-structure whose density is at most $O(\log h)$ times the optimal density, namely a structure of density at most $O\left(\frac{\log^2 h}{h}\right) \mathsf{cost}(\text{OPT}_{\text{MC}})$. We now remove the demands whose source-destination nodes are connected and recurse on the remaining ones. This gives us an approximation ratio of $O\left(\log^3 h\right)$ for the protected multi-commodity buy-at-bulk problem.

3.1 Existence of a low-density junction-structure. To show the existence of a junction-structure with low density, we assume knowledge of the edge set $E^* \subseteq E$ of an optimal solution OPT_{MC} to the given multi-commodity instance and find a low-density junction-structure from E^*. Let $G^* = G[E^*]$ be the graph induced on E^*. Let $L = \sum_i \ell 2_{G^*}(s_i, t_i)/h$ be the *average length* of the demand pairs in the optimal solution. A demand $s_i t_i$ is *short* if $\ell 2_{G^*}(s_i, t_i)$ is at most $2L$. By Markov's inequality, more than half of the demands are short:

Proposition 3.2 *At least $h/2$ demands are short.*

We now restrict our attention to these short demands. For each demand pair $s_i t_i$, we fix a shortest cycle Q_i through s_i and t_i in G^*. Subsequently, we present an algorithm that decomposes G^* into connected *edge-disjoint*[1] induced subgraphs $G_1^* = G[E_1^*], G_2^* = G[E_2^*], \ldots, G_a^* = G[E_a^*]$. For a subgraph

[1] This is in contrast to the 1+1 edge-protection case where the subgraphs can be chosen to be node-disjoint. The node-disjoint property is relevant for the buy-at-bulk problem with node costs [7].

H of G^*, we define a ball $B_H(\widehat{uv}, r)$ with center $\widehat{uv}$ and radius r to contain vertices $x \in V(H)$ for which $\ell 2_H(x, \widehat{uv}) \le r$. We abuse notation and use $B_H(\widehat{uv}, r)$ also to denote the induced subgraph. A demand pair $s_i t_i$ is *captured* by a ball $B_H(\widehat{uv}, r)$ if both s_i and t_i are contained in the ball. A pair $s_i t_i$ *intersects* $B_H(\widehat{uv}, r)$ if it is not captured by the ball, but the ball contains some edge in the cycle Q_i.

We choose a short demand pair uv as center $\widehat{uv}$ and define a sequence of radii $r_p = 2Lp$, for $p \in \mathbb{N}^*$. We begin with the ball $B_{G^*}(\widehat{uv}, r_1)$; if the number of captured demands is at least the number of intersected demands, we make the ball the first component G_1^*. Otherwise, the number of captured demands is fewer than the number of intersected demands. In this case, we consider progressively larger balls, of radii r_2, r_3, and so on. Let $\bar{p}$ be the smallest index such that the number of demands captured by $B_{G^*}(\widehat{uv}, r_{\bar{p}})$ is equal to or greater than the number of demands intersected by the same ball. Then, $B_{G^*}(\widehat{uv}, r_{\bar{p}})$ becomes the first component G_1^*. We remove all *edges* in $B_{G^*}(\widehat{uv}, r_{\bar{p}})$ from G^* and all demands that are either captured or intersected by that ball; these intersected demands are henceforth considered *lost*. We recurse on the residual of G^* and the remaining demands to create components $G_2^*, G_3^*, \ldots, G_a^*$, until no demands are left. We let $\mathcal{D}_j$ be the set of demands captured by the component G_j^*, and let (u_j, v_j) denote the center we have arbitrarily chosen for G_j^*. Since lost demands are fewer than captured demands, the following lemma also holds.

Lemma 3.1 *The total number of demands that are captured by $G_1^*, G_2^*, \ldots, G_a^*$ is at least $h/4$.*

We show below that one of the components corresponds to a low-density junction-structure. The construction of the ball-growing algorithm has the following property.

Lemma 3.2 *Any demand intersected by the ball $B_{G^*}(\widehat{uv}, r_p)$ is captured by $B_{G^*}(\widehat{uv}, r_{p+1})$.*

Proof. Assume that $B_{G^*}(\widehat{uv}, r_p)$ intersects some demand pair $s_i t_i$. It suffices to argue that $\ell 2_{G^*}(s_i, \widehat{uv}) \le r_{p+1}$. By symmetry, a similar inequality shall then hold for t_i, too. Start from s_i and move in one direction along Q_i. Denote by x the first node in $B_{G^*}(\widehat{uv}, r_p)$ thus encountered, and by $P_{s_i x}$ the segment of Q_i traversed. Then, go back to s_i and move in the opposite direction along Q_i. Denote by y the first node in $B_{G^*}(\widehat{uv}, r_p)$ encountered, and by $P_{s_i y}$ the segment of Q_i traversed. Note that x and y are distinct; since Q_i and $B_{G^*}(\widehat{uv}, r_p)$ share an edge, at least two nodes on Q_i are in $B_{G^*}(\widehat{uv}, r_p)$. Moreover, $P_{s_i x}$ and $P_{s_i y}$ are node-disjoint, by construction. Let P_{xu} and P_{xv} be the two node-disjoint paths from x to u and v, respectively, whose combined length is at most r_p. Similarly, let P_{yu} and P_{yv} be the two node-disjoint paths from y to u and v, respectively, whose combined length is at most r_p. By the choice of x and y, the paths P_{xu}, P_{xv}, P_{yu} and P_{yv} are node-disjoint from $P_{s_i x}$ and $P_{s_i y}$, other than at x and y. We distinguish the following two cases.

Case 1: $x \notin P_{yu} \cup P_{yv}$ and $y \notin P_{xu} \cup P_{xv}$ (see Figure 3, left). Let F be the subgraph induced on $Q_i \cup P_{xu} \cup P_{xv} \cup P_{yu} \cup P_{yv}$. Add a dummy node v_0 to F, such that v_0 is adjacent to u and v only, via zero-length edges. Then, create a single-sink min-cost flow problem on F, where two units of flow are sent from s_i to v_0. Assume that each node except s_i and v_0 has unit capacity and zero cost, and each edge has cost equal to its length.

Consider the following fractional solution: s_i sends one unit of flow to x along $P_{s_i x}$ and one unit to y along $P_{s_i y}$; x sends 1/2 units of flow to u along P_{xu} and 1/2 units to v along P_{xv}; and y sends 1/2 units of flow to u along P_{yu} and 1/2 units to v along P_{yv}. It is easy to see that node capacities are not exceeded and the routing cost is at most

$$\frac{1}{2}\ell(P_{xu} \cup P_{xv}) + \frac{1}{2}\ell(P_{yu} \cup P_{yv}) + \ell(Q_i) \le \frac{1}{2}r_p + \frac{1}{2}r_p + 2L = r_{p+1} \; . \tag{5}$$

The integrality of single-sink min-cost flow implies the existence of two node-disjoint integral paths between s_i and u_0, of total cost no more than the fractional cost given in (5).

Case 2: Otherwise, without loss of generality, suppose that P_{yu} goes through x (see Figure 3, right); the other sub-cases are symmetrical. P_{yv} and the segment of P_{yu} between x and u yield two node-disjoint paths from x to u and from y to v, whose combined length is less than r_p. Consequently, there exist two node disjoint paths from s_i to u and v, whose total length is at most $\ell(P_{yu}) + \ell(P_{yv}) + \ell(Q_i) \le r_p + 2L = r_{p+1}$. □

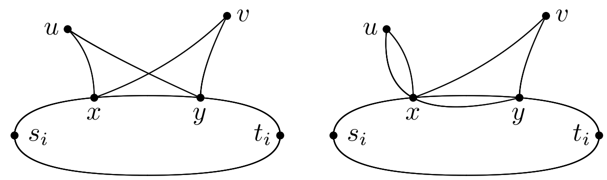

Figure 3: Existence of two node-disjoint paths from s_i (symmetrically t_i) to u and v.

Lemma 3.3 *For $1 \le j \le a$ and every node x in G_j^*, $\ell 2_{G_j^*}(x, \widehat{u_j v_j}) \le 2L \cdot (1 + \log h)$. In particular, for each demand st captured by G_j^*, $\ell 2_{G_j^*}(s, \widehat{u_j v_j}) + \ell 2_{G_j^*}(t, \widehat{u_j v_j}) + \ell 2_{G_j^*}(u_j, v_j) \le 2L \cdot (3 + 2\log h)$.*

Proof. Let H be the residual graph of G^* after the first $j-1$ components $G_1^*, \ldots, G_{j-1}^*$ are created and their edges removed. From Lemma 3.2 and the construction of the algorithm, every time the ball grows from $B_H(\widehat{u_j v_j}, r_p)$ to $B_H(\widehat{u_j v_j}, r_{p+1})$, the number of demands captured by $B_H(\widehat{u_j v_j}, r_{p+1})$ is at least twice the number captured by $B_H(\widehat{u_j v_j}, r_p)$. Since the total number of pairs is h, the number of times the radius is increased is at most $\lceil \log h \rceil$. Thus, G_j^* has radius at most $2L \cdot (1 + \log h)$. Clearly then,

$$\ell 2_{G_j^*}(s, \widehat{u_j v_j}) + \ell 2_{G_j^*}(t, \widehat{u_j v_j}) + \ell 2_{G_j^*}(u_j, v_j) \le 2 \cdot 2L \cdot (1 + \log h) + 2L = 2L \cdot (3 + 2\log h) \ ,$$

as we sought to prove. □

Theorem 3.1 *Given a multi-commodity instance of the protected buy-at-bulk problem, there exists a junction-structure of density $O\big(\frac{\log h}{h}\big)\, \mathsf{cost}(\text{OPT}_{\text{MC}})$.*

Proof. The subgraphs $G_1^*, \ldots, G_a^*$ are edge-disjoint by construction, and each G_j^* constitutes a junction-structure for the corresponding demand set $\mathcal{D}_j \subseteq \mathcal{D}$. Taking Lemma 3.3 into account, we have

$$\sum_{j=1}^{a} \left(\sum_{e \in E(G_j^*)} c_e + \sum_{s_i t_i \in \mathcal{D}_j} \big(\ell 2_{G_j^*}(s_i, \widehat{u_j v_j}) + \ell 2_{G_j^*}(t_i, \widehat{u_j v_j}) + \ell 2_{G_j^*}(u_j, v_j) \big) \right) \le$$

$$\le \sum_{e \in E(G^*)} c_e + \sum_{s_i t_i \in \mathcal{D}} 2L \cdot (3 + 2\log h) \le O(\log h) \cdot \mathsf{cost}(\text{OPT}_{\text{MC}}) \ . \quad (6)$$

Since $\sum_{j=1}^{a} |\mathcal{D}_j| \ge h/4$, by Lemma 3.1, applying a simple averaging argument on (6) establishes the theorem. □

3.2 Finding a low-density junction-structure. Using the single-sink single-cable approximation algorithm as a subroutine, an $O(\log h)$ approximation to the minimum-density junction-structure can be derived. This is a consequence of the theorem below.

Theorem 3.2 *There is an $O(\log h)$ approximation for the min-density protected single-sink single-cable problem.*

Proof. We closely follow the argument used in [7]. First, let us formulate a linear programming relaxation for the density version of the protected single-sink problem. In other words, the objective is to minimize the ratio of the cost of the network to the number of terminals it connects to the sink. For each terminal t_i, we introduce a variable y_i that indicates whether or not t_i is connected to s in the solution. By normalizing the sum $\sum_{i=1}^{h} y_i$ to 1, we ensure that the linear objective function represents the density of the solution.

Claim. The linear program LP7 is a valid relaxation for the min-density protected single-sink problem. It can be solved optimally in polynomial time, either by using the ellipsoid algorithm on its dual, or by first transforming it via standard techniques into a polynomial-size linear program with edge-based flow variables (e.g. similar to LP1).

Let $(\mathbf{x}^*, \mathbf{f}^*, \mathbf{y}^*)$ denote a basic optimal solution to LP7. We then partition the terminals into groups $\mathcal{T}_z \subseteq \mathcal{T}$, where $0 \le z \le \bar{z} = \lceil \log h \rceil$, depending on the corresponding y_i^* values. More specifically,

$$\begin{aligned}
\mathsf{LP7}: \quad \min \quad & \sum_{e\in E} c_e x(e) + \sum_{i=1}^{h}\sum_{Q\in\mathcal{Q}_i} \ell_Q f(Q) && && \text{(7a)}\\
\text{s.t.} \quad & \sum_{\substack{Q\in\mathcal{Q}_i\\ Q\ni e}} f(Q) \le x(e) && \forall e\in E,\ 1\le i\le h && \text{(7b)}\\
& \sum_{Q\in\mathcal{Q}_i} f(Q) \ge y_i && \forall 1\le i\le h && \text{(7c)}\\
& \sum_{i=1}^{h} y_i = 1 && && \text{(7d)}\\
& x(e), f(Q), y_i \ge 0 && \forall e\in E,\ Q\in \textstyle\bigcup_i \mathcal{Q}_i,\ 1\le i\le h && \text{(7e)}
\end{aligned}$$

$\mathcal{T}_z = \{t_i \mid y^*_{\max}/2^{z+1} < y^*_i \le y^*_{\max}/2^z\}$, where $y^*_{\max} = \max_i y^*_i$. Observe that $\sum_{z=0}^{\bar{z}} \sum_{t_i\in\mathcal{T}_z} y^*_i \ge \frac{1}{2}$, hence there exists one group $\mathcal{T}_\theta$ such that $\sum_{t_i\in\mathcal{T}_\theta} y^*_i = \Omega(1/\log h)$. Furthermore, $2^\theta/(y^*_{\max}|\mathcal{T}_\theta|) = O(\log h)$.

Finally, we solve the protected single-sink problem for the terminals in $\mathcal{T}_\theta$ only, by invoking the approximation algorithm from Section 2, and demonstrate that the resulting solution is an $O(\log h)$ approximation to the min-density protected single-sink problem. Indeed, let U be the value of the optimal solution to LP7. We may obtain a feasible solution to LP2 (considering only terminals in $\mathcal{T}_\theta$) by scaling up the optimal solution to LP7 by a factor of $\lambda = 2^{\theta+1}/y^*_{\max}$. The cost of this solution is at most λU. By the proof of Theorem 2.2, the algorithm of Section 2 can yield an integral solution, for terminals in $\mathcal{T}_\theta$, of value $O(\lambda U)$. Its density is $O(\lambda U)/|\mathcal{T}_\theta|$, which is $O(\log h)U$, by the choice of θ. Since U is a lower bound on the density of the optimal solution, the proof is complete. □

We now describe how to approximate the minimum-density junction-structure. Once again, the ideas are similar to those in [7], but the details are more elaborate. The step of guessing the junction $\widehat{uv}$ of a min-density junction-structure is implemented, as is standard, by trying each possible pair of nodes as a candidate junction, and keeping the best result.

Then, we relax this problem to an LP very similar to that for the min-density single-sink problem. Create a new graph G' by adding an artificial sink node σ to the graph G and connecting it to u and v via edges $u\sigma, v\sigma$ such that $\ell_{u\sigma} = \ell 2_G(u,v)$, $c_{u\sigma} = \mu \cdot \ell_{u\sigma}$, and $c_{v\sigma} = \ell_{v\sigma} = 0$. Assume, without loss of generality, that each node in the original graph G is the endpoint of at most one demand pair in $\mathcal{D}$. Consider LP7 on G', with σ as sink and $\mathcal{T}' = \{s_1, t_1, s_2, t_2, \ldots, s_h, t_h\} = \{t'_1, t'_2, \ldots, t'_{2h}\}$ as the set of terminals. Moreover, place an additional set of constraints in LP7:

$$y_i = y_j \qquad \forall i,j \text{ such that there exists index } q \text{ with } t'_i = s_q \text{ and } t'_j = t_q$$

Suppose that the minimum-density junction-structure OPT* has density γ^*. It is straightforward to convert OPT* to a feasible solution of this new linear program, with density between $\frac{1}{2}\gamma^*$ and γ^*; it may not be exactly $\frac{1}{2}\gamma^*$, because the fixed cost of some junction-structure edges may be double-counted in the objective function of the LP.

Apply the algorithm from Theorem 3.2 to the optimal solution $(\mathbf{x}^*, \mathbf{f}^*, \mathbf{y}^*)$ of the modified LP7 above. Observe that the rounding procedure ensures that for any i,j such that $y^*_i = y^*_j$, either both t'_i and t'_j are connected to σ, or neither is. Thus, the resulting solution SOL* to the single-sink density problem can be converted back to a junction-structure. By Theorem 3.2, SOL* has density $O(\log h)\gamma^*$, so the corresponding junction-structure also has density $O(\log h)\gamma^*$. Combined with Theorem 3.1, this yields:

THEOREM 3.3 *Given a multi-commodity instance of the protected single-cable buy-at-bulk problem, there is a polynomial time algorithm that finds a junction-structure with density* $O\left(\frac{\log^2 h}{h}\right) \mathsf{cost}(\text{OPT}_{\text{MC}})$.

As mentioned before, we use an iterative greedy algorithm, similar to the classic one for set cover. In each iteration, we invoke Theorem 3.3 to find an approximate junction-structure in the residual instance and remove the demand pairs that are connected by the structure to obtain the residual instance for the next iteration. Hence, THEOREM 3.4 *The node-protected multi-commodity single-cable buy-at-bulk problem can be approximated by a factor of* $O(\log^3 h)$.

3.3 Approximation for arbitrary demands. Theorem 3.1 can be extended to multi-commodity instances with arbitrary demands, in which case it guarantees the existence of a junction-structure with density $O\big(\frac{\log h}{D}\big)\,\mathsf{cost}(\text{OPT}_{\text{MC}})$, where $D = \sum_i \mathsf{dem}(i)$ and density is defined as the cost of a junction-structure divided by the total demand of the pairs it connects. Thus, for the protected multi-commodity buy-at-bulk problem we would obtain an approximation ratio that depends on D. In the single-cable model, we avoid this dependence as follows.

Suppose $\mathsf{dem}(i) \geq \mu$ for some pair s_it_i. Then we can route this pair independently of other pairs, by finding a shortest cycle Q_i for s_it_i and routing $\mathsf{dem}(i)$ using $\lceil \mathsf{dem}(i)/\mu \rceil$ cables on the cycle. This costs $\lceil \mathsf{dem}(i)/\mu \rceil \cdot \ell(Q_i)$ under the cable capacity cost model, or $\big(\lceil \mathsf{dem}(i)/\mu \rceil + 1\big) \cdot \ell(Q_i)$ under the FI cost model. By contrast, removing s_it_i from the demand set $\mathcal{D}$ reduces the cost of the optimal solution by at least $\lfloor \mathsf{dem}(i)/\mu \rfloor \cdot \ell(Q_i)$, under either model. Consequently, routing all such demand pairs in the aforementioned manner incurs an overall cost not exceeding $3\,\mathsf{cost}(\text{OPT}_{\text{MC}})$.

Henceforth, assume that $\mathsf{dem}(i) < \mu$ for each i. In G, we find a 2-approximation to the minimum-cost subgraph H in which s_i and t_i are two-connected, using the algorithm from [13]. This is similar to the the first step in the single-sink algorithm. We install exactly one cable on each edge of H. Clearly, the cost of this network is at most $2\,\mathsf{cost}(\text{OPT}_{\text{MC}})$. Using the capacity installed on H, we route all pairs s_it_i such that $\mathsf{dem}(i) \leq \mu/h$, using an arbitrary cycle for each such pair. Note that the total flow on any edge is at most $h \cdot \mu/h \leq \mu$, and therefore the capacity is sufficient.

After these steps, any demand s_it_i that remains has the property that $\mu/h < \mathsf{dem}(i) < \mu$. As a result, the ratio between the maximum and minimum demand is at most h. By re-scaling we can ensure that the minimum demand value is 1, in which case the maximum demand value is at most h and the sum D of all demands is at most h^2. We replace a pair s_it_i with $\mathsf{dem}(i)$ distinct pairs of demand 1 each, by artificially duplicating the nodes s_i and t_i. The total number of pairs in the new unit-demand instance is $O(h^2)$, and therefore its solution can be approximated within a factor of $O(\log^3 h)$, as already established. Hence, Theorem 3.4 also applies to arbitrary demands.

3.4 Extension to the general FI cost model. Both Theorem 3.1 and the reduction from the min-density junction-structure problem to the single-sink one are also valid for the general FI model. In this case, though, the construction of the graph G' becomes slightly more complicated: edge $u\sigma$ is replaced by a set of parallel edges, whose fixed and incremental costs correspond to those of polynomially many cycles in G containing u and v.

More specifically, take an arbitrary junction-structure $H(\widehat{uv})$ and note that each edge e of the shortest cycle containing u and v contributes between $c_e + \ell_e \cdot \sum_{s_it_i \in \mathcal{D}'} \mathsf{dem}(i)$ and $c_e + 2\ell_e \cdot \sum_{s_it_i \in \mathcal{D}'} \mathsf{dem}(i)$ to the cost defined in (4), where $\mathcal{D}' \neq \varnothing$ is the set of demands connected by the structure. Let κ be the smallest integer such that $2^\kappa \geq 2 \cdot \sum_{s_it_i \in \mathcal{D}'} \mathsf{dem}(i)$. We may assume that every such edge contributes exactly $c_e + \ell_e \cdot 2^\kappa$ to (4), losing only a factor of 4 in the approximation.

Now, for each possible value of κ between 1 and $\lceil \log D \rceil + 1$, consider a cycle Q_κ in G that minimizes the quantity $\sum_{e \in Q_\kappa} c_e + \ell_e \cdot 2^\kappa$. We add $\lceil \log D \rceil + 1$ parallel edges joining u and σ to G', where the κ^{th} such edge has fixed cost $\sum_{e \in Q_\kappa} c_e$ and incremental cost $\sum_{e \in Q_\kappa} \ell_e$. It is easy to see that the value of the optimal solution to the modified $\mathsf{LP7}$ on G' is no more than a constant factor away from the minimum junction-structure density. Consequently, if a bound on the integrality gap of $\mathsf{LP2}$ for either the uniform or the non-uniform cost model were known, via a constructive result analogous to Theorem 2.2, we could generalize Theorems 3.2 and 3.3 – and, ultimately, Theorem 3.4 – accordingly.

4. Conclusions. Our main contributions are the formal introduction of the protected buy-at-bulk network design problem and the first approximation algorithms for it in the single-cable setting. One important question is the approximability of the protected buy-at-bulk problem in the uniform multiple-cable and non-uniform cost models. Our results in Section 3 pertaining to the general FI cost model indicate that it is sufficient to focus on the single-sink version of the problem. There has been some progress on this question [9], although the ratios obtained are far from satisfactory. On a different note, recent work of Chuzhoy and Khanna [11] obtained the first non-trivial approximation algorithms for vertex-connectivity SNDP when the connectivity requirements are larger than 2; this and related ideas [9, 16] may be fruitful in obtaining algorithms for buy-at-bulk network design with higher connectivity requirements. Even though there may not be immediate practical applications for buy-at-bulk with connectivity requirements larger than two, we believe that it is a problem of theoretical interest. From a practical perspective, we hope the concepts of clustering and junction structures may inspire the design of more effective heuristics for the design of real-world DWDM networks.

Acknowledgments. This work was partially supported by a basic research grant N00014-06-1-0396 from ONR to Bell Laboratories. The work of the first author was performed at Columbia University, supported by NSF grant CCF-0430946 and an Alexander S. Onassis Foundation Scholarship. The second author acknowledges support from NSF grant CCF-072872. A preliminary version of this paper appeared in Proceedings of IEEE FOCS, 2007.

References

[1] Matthew Andrews, *Hardness of buy-at-bulk network design*, Proceedings of IEEE FOCS, 2004, pp. 115–124.

[2] Matthew Andrews and Lisa Zhang, *Approximation algorithms for access network design*, Algorithmica **34** (2002), no. 2, 197–215.

[3] Baruch Awerbuch and Yossi Azar, *Buy-at-bulk network design*, Proceedings of IEEE FOCS, 1997, pp. 542–547.

[4] Yair Bartal, *On approximating arbitrary metrics by tree metrics*, Proceedings of ACM STOC, 1998, pp. 161–168.

[5] Moses Charikar and Adriana Karagiozova, *On non-uniform multicommodity buy-at-bulk network design*, Proceedings of ACM STOC, 2005, pp. 176–182.

[6] Chandra Chekuri, Paul Claisse, René-Jean Essiambre, Steven Fortune, Daniel C. Kilper, Wonsuck Lee, Nachi K. Nithi, Iraj Saniee, Bruce Shepherd, Christopher A. White, Gordon Wilfong, and Lisa Zhang, *Design tools for transparent optical networks*, Bell Labs Technical Journal **11** (2006), no. 2, 129–143.

[7] Chandra Chekuri, Mohammad T. Hajiaghayi, Guy Kortsarz, and Mohammad R. Salavatipour, *Approximation algorithms for nonuniform buy-at-bulk network design*, SIAM Journal on Computing **39** (2010), no. 5, 1772–1798.

[8] Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd, *The all-or-nothing multicommodity flow problem*, Proceedings of ACM STOC, 2004, pp. 156–165.

[9] Chandra Chekuri and Nitish Korula, *Single-sink network design with vertex connectivity requirements*, Proceedings of FSTTCS, 2008.

[10] Julia Chuzhoy, Anupam Gupta, Joseph Naor, and Amitabh Sinha, *On the approximability of some network design problems*, ACM Transactions on Algorithms **4** (2008), no. 2, Article 23.

[11] Julia Chuzhoy and Sanjeev Khanna, *An $O(k^3 \log n)$ approximation algorithm for vertex-connectivity survivable network design*, Proceedings of IEEE FOCS, 2009, pp. 437–441.

[12] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, Journal of Computer and System Sciences **69** (2004), 485–497.

[13] Lisa Fleischer, Kamal Jain, and David P. Williamson, *An iterative rounding 2-approximation algorithm for the element connectivity problem*, Journal of Computer and System Sciences **72** (2006), no. 5, 838–867.

[14] Fabrizio Grandoni and Giuseppe F. Italiano, *Improved approximation for single-sink buy-at-bulk*, Proceedings of ISAAC, 2006, pp. 111–120.

[15] Sudipto Guha, Adam Meyerson, and Kamesh Munagala, *A constant factor approximation for the single sink edge installation problem*, SIAM Journal on Computing **38** (2009), no. 6, 2426–2442.

[16] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi, *Online and stochastic survivable network design*, Proceedings of ACM STOC, 2009, pp. 685–694.

[17] Anupam Gupta, Amit Kumar, Martin Pál, and Tim Roughgarden, *Approximation via cost-sharing: Simpler and better approximation algorithms for network design*, Journal of the ACM **54** (2007), no. 3, Article 11.

[18] Refael Hassin, R. Ravi, and F. Sibel Salman, *Approximation algorithms for a capacitated network design*, Algorithmica **38** (2003), no. 3, 417–431.

[19] Dorit S. Hochbaum, *Approximation algorithms for NP-hard problems*, PWS Publishing Company, 1997.

[20] Kamal Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica **21** (2001), no. 1, 39–60.

[21] Raja Jothi and Balaji Raghavachari, *Improved approximation algorithms for the single-sink buy-at-bulk network design problem*, Proceedings of SWAT, 2004, pp. 336–348.

[22] Guy Kortsarz and Zeev Nutov, *Approximating some network design problems with node costs*, Proceedings of APPROX-RANDOM, 2009, pp. 231–243.

[23] Andrew McGregor and Bruce Shepherd, *Island hopping and path colouring, with applications to WDM network design*, Proceedings of ACM-SIAM SODA, 2007, pp. 864–873.

[24] Adam Meyerson, Kamesh Munagala, and Serge Plotkin, *Cost-distance: Two metric network design*, SIAM Journal on Computing **38** (2008), no. 4, 1648–1659.

[25] Rajiv Ramaswami and Kumar N. Sivarajan, *Optical networks: A practical perspective*, 2nd ed., Morgan Kaufmann, 2001.

[26] F. Sibel Salman, Joseph Cheriyan, R. Ravi, and Sairam Subramanian, *Buy at bulk network design: Approximating the single-sink edge installation problem*, SIAM Journal on Optimization **11** (2000), no. 3, 595–610.

[27] Alexander Schrijver, *Combinatorial optimization: Polyhedra and efficiency*, Springer-Verlag, 2003.

[28] Arun K. Somani, *Survivability and traffic grooming in WDM optical networks*, Cambridge University Press, 2006.

[29] Thomas E. Stern and Krishna Bala, *Multiwavelength optical networks: A layered approach*, Addison-Wesley, 1999.

[30] Kunal Talwar, *Single-sink buy-at-bulk LP has constant integrality gap*, Proceedings of IPCO, 2002, pp. 475–486.

[31] Vijay V. Vazirani, *Approximation algorithms*, Springer, 2001.

[32] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani, *A primal-dual approximation algorithm for generalized Steiner network problems*, Combinatorica **15** (1995), 534–454.