A New Perspective on Vertex Connectivity

Keren Censor-Hillel
Technion
Haifa, Israel
ckeren@cs.technion.ac.il

Mohsen Ghaffari
MIT
Cambridge, MA, USA
ghaffari@mit.edu

Fabian Kuhn
University of Freiburg
Freiburg, Germany
kuhn@cs.uni-freiburg.de

Abstract

Edge connectivity and vertex connectivity are two fundamental concepts in graph theory. Although by now there is a good understanding of the structure of graphs based on their edge connectivity, our knowledge in the case of vertex connectivity is much more limited. An essential tool in capturing edge connectivity are the classical results of Tutte and Nash-Williams from 1961 which show that a λ-edge-connected graph contains $\lceil(\lambda-1)/2\rceil$ edge-disjoint spanning trees.

We argue that connected dominating set partitions and packings are the natural analogues of edge-disjoint spanning trees in the context of vertex connectivity and we use them to obtain structural results about vertex connectivity in the spirit of those for edge connectivity.

More specifically, connected dominating set (CDS) partitions and packings are counterparts of edge-disjoint spanning trees, focusing on vertex-disjointness rather than edge-disjointness, and their sizes are always upper bounded by the vertex connectivity k. We constructively show that every k-vertex-connected graph with n nodes has CDS packings and partitions with sizes, respectively, $\Omega(k/\log n)$ and $\Omega(k/\log^5 n)$, and we prove that the former bound is existentially optimal.

Beautiful results by Karger show that when edges of a λ-edge-connected graph are independently sampled with probability p, the sampled graph has edge connectivity $\tilde{\Omega}(\lambda p)$. Obtaining such a result for vertex sampling remained open. We illustrate the strength of our approach by proving that when vertices of a k-vertex-connected graph are independently sampled with probability p, the graph induced by the sampled vertices has vertex connectivity $\tilde{\Omega}(kp^2)$. This bound is optimal up to poly-log factors and is proven by building an $\tilde{\Omega}(kp^2)$ size CDS packing on the sampled vertices while sampling happens.

As an additional important application, we show CDS packings to be tightly related to the throughput of routing-based algorithms and use our new toolbox to yield a routing-based broadcast algorithm with optimal throughput $\Omega(k/\log n + 1)$, improving the (previously best-known) trivial throughput of $\Theta(1)$.

1 Introduction and Related Work

Vertex and edge connectivity are two core graph-theoretic concepts as they are basic measures for the robustness and throughput capacity of a graph. While by now a lot is known about edge connectivity and its connections to related graph-theoretic properties and problems, our knowledge about vertex connectivity is much scarcer and many related problems remain open.

As an example, given a graph G, assume that each edge or vertex is independently sampled with probability p. How large should p be such that the subgraph given by the sampled edges or the one induced by the sampled vertices is connected (this problem is sometimes studied under the title of *network reliability*). Intuitively, the larger the connectivity of G is, the smaller we should be able to choose p such that the sampled subgraph remains connected. For edge connectivity and sampling edges, Lomonosov and Poleskii [32] verified this intuition already four decades ago: if $p = \Omega(\frac{\log n}{\lambda})$, where λ is the edge-connectivity of G, then the edge-sampled graph is connected with high probability (w.h.p.)[1] and this threshold is optimal. In the special case of sampling edges of a complete graph, this corresponds to the $\frac{\ln n}{n}$ probability threshold for connectivity in the Erdős-Rényi random graph model. Karger [21] showed that assuming $p = \Omega(\frac{\log n}{\lambda})$, the edge-connectivity of the edge-sampled graph will be around λp, w.h.p., and in fact, for such p, the size of each edge cut remains around its expectation. In the following years, these results, and extensions thereof, have emerged as powerful tools, having implications for many problems such as graph sparsifiers [5], finding or approximating minimum edge cuts [5, 21, 22], max-flow [21, 25], network design problems [21], and getting an FPTAS for all-terminal network reliability [23].

In contrast, prior to our work, for vertex sampling, even the most basic of these questions remained open and it was not even known how large p should be for the vertex sampled graph to remain (just simply) connected. We prove results of the same flavor as the ones discussed above, but in the context of vertex sampling rather than edge sampling. In particular, we show that if each vertex of a k-vertex connected graph G is independently sampled with probability p, then w.h.p., the sampled vertices induce an $\tilde{\Omega}(kp^2)$-vertex-connected graph. We also show that this is existentially tight up to $\log$-factors.[2]

The main hurdle on the way to proving these results is that there can be an exponential number of *"small"* vertex cuts. When arguing that the subgraph induced by randomly sampled vertices is connected, one essentially needs to show that for each vertex cut of the graph, at least one vertex is sampled. However, it has been shown that even the number of the minimum vertex cuts of a k-vertex connected graph can be as large as $\Theta(2^k(n/k)^2)$ [19]. Note that this is in stark contrast to the case of edge cuts, where the number of minimum edge cuts is known to be bounded by $O(n^2)$ and the number of edge cuts of size $\alpha \cdot \lambda$ in a graph with edge-connectivity λ is at most $O(n^{2\alpha})$ [20, 24]. This $O(n^{2\alpha})$ bound is the main tool in studying edge cuts after random sampling, with which a simple application of Chernoff and union bounds solves the problem.

An essential tool in grasping edge connectivity is the famous results of Tutte and Nash-Williams [36, 38] from 1961 (presented independently), which shows that a λ-edge-connected graph contains at least $\lceil\frac{\lambda-1}{2}\rceil$ edge-disjoint spanning trees (see [29]). One way to interpret this result is as follows: Recall that Menger's theorem tells us that in a λ-edge-connected graph, each pair of vertices are connected via at least λ edge-disjoint paths. Tutte-Nash-Williams gives us asymptotically the same number of paths but with much stronger structure compared to Menger's. We get $\lceil\frac{\lambda-1}{2}\rceil$ edge-disjoint paths between each pair of vertices, one through each spanning tree, such that the paths of each pair are colored via colors exactly 1 to $\lceil\frac{\lambda-1}{2}\rceil$, and the paths of different colors are edge disjoint (for any set of pairs of vertices). This powerful result leads to numerous applications for different problems concerning edge-connectivity, e.g., the best known min-cut algorithm and the tightest proof of the aforementioned $O(n^{2\alpha})$ upper bound on the number of α-minimum edge-cuts (see e.g., [21]), which in turn leads to the edge-sampling results.

[1] We use the phrase "with high probability" to indicate that an event happens with probability at least $1 - \frac{1}{n^c}$ for a constant $c \geq 1$.
[2] For exact statements and a more extensive discussion of our results, we refer to Section 1.1.

Our main technical contribution is presenting similar structural results for vertex connectivity instead of edge connectivity. Thus, the focus is also on vertex-disjointness rather than edge-disjointness. When requiring vertex disjointness, obviously we can not ask for *spanning* trees. The closest option which allows vertex-disjointness is using *dominating trees*. For a graph $G = (V, E)$, a subgraph $T = (V_T, E_T)$ of G is a dominating tree if it is a tree and each vertex $v \in V \setminus V_T$ has at least one neighbor in V_T. As we explain next, dominating trees are indeed the right substitutes for spanning trees: Recall that Menger's theorem tells us that in a k-vertex-connected graph, each pair of vertices are connected via k internally vertex-disjoint paths. Given k' vertex-disjoint dominating trees, we get a system of colored paths analogous to that of Tutte and Nash-Williams: each pair of vertices is connected via k' paths, exactly one from each color 1 to k', where paths of each color go through one dominating tree and paths of different colors are internally vertex-disjoint.

It is cleaner to work with connected dominating sets (CDSs) rather than dominating trees as then, we only need to specify the vertices of each set and the edges are the induced ones. It is straight-forward to see that this transition is without loss of generality (by removing cycle-creating edges). We note that CDSs are structures that have been studied extensively in graph theory and theory of computing (see e.g. [7, 8, 14, 35]).

As our structural results about vertex connectivity, we present methods to decompose each k-vertex-connected graph into $\tilde{\Theta}(k)$ disjoint (or almost disjoint) connected dominating sets (CDSs). These results are formalized as solutions to CDS partition and packing problems[3]. A *CDS partition* of size K is defined as a partition of the vertices of a graph into K (vertex-disjoint) connected dominating sets [16, 17, 39]. *CDS packing* problem is the natural generalization of CDS partitions, where each vertex can be broken into smaller pieces and each piece is given to one CDS. More formally, a CDS packing is a collection of CDSs with positive weights such that for each vertex v, the sum of the weights of all CDSs containing v is at most 1. The size of a CDS packing is the total weight of all CDSs in the collection. In the context of a system of colored paths, a CDS packing provides a relaxed version of such a path system, where paths of different colors are allowed to have vertex-overlaps but the weighted overlap in each vertex is bounded by 1.

It is easy to see that any CDS partition or packing in a graph with vertex connectivity k has size at most k [16, 39], simply because each CDS must contain at least one vertex from every vertex cut. However, showing any converse to this relation has remained wide open and even for graphs with vertex connectivity as large as, e.g., $\Theta(\sqrt{n})$, even finding an $\omega(1)$ CDS packing seems non-trivial. We constructively show that every k-vertex connected graph G has a CDS packing of size at least $\Omega(k/\log n)$. We also show that this is optimal in the sense that for all n and $k \leq n/4$, there are k-vertex connected n-vertex graphs for which the largest CDS packing has size at most $O(k/\log n)$. A generalized version of this construction leads to the aforementioned vertex sampling result, by building a CDS packing on the sampled vertices while the sampling is happening. Also, with a similar construction, we prove that every k-vertex connected graph has a CDS partition of size at least $\Omega(k/\log^5 n)$. Surprisingly, this proof itself requires the sampling result.

CDS packings also lead to an interesting and important result in networking algorithms. Consider the communication model where in each time unit, each node of a network can send one B-bits message to all its neighbors. We show that the achievable total throughput when globally broadcasting messages using routing-based algorithms (that is, without *network coding*) can exactly be characterized by the size of the largest CDS packing of the network graph. This exactly corresponds to the aforementioned systems of colored paths where now each message relates to one color. As a consequence, we get that in k-vertex-connected networks, the (existentially) optimal routing-based broadcast throughput is $\Theta(k/\log n)$ messages per round.

---

[3] We remark that CDS packing and partition are significantly different problems from the standard study of CDSs in theory of computing where usually the goal is to find just one CDS with certain properties (see e.g. [7, 8, 14, 35]), whereas CDS packing and partition aim at finding many CDSs that are almost or completely vertex-disjoint. Also, these problems are very different from dominating sets problems, which have also received a vast amount of attention (see e.g. [10, 18]).

The CDS partition problem was introduced in [17] where the size of a maximum CDS partition is called the *connected domatic number* (CDN). [39] shows a number of results about CDN; e.g., that it is upper bounded by the vertex connectivity. [16] shows that the CDN of planar graphs is at most 4. [34] argues that CDS partition can help to balance energy-usage in wireless sensor networks.

1.1 Results

1.1.1 Vertex Connectivity vs. CDS Packing and CDS Partition:

As described, our core technical contribution is presenting structural results for vertex connectivity by providing efficient algorithms to construct CDS packings and partitions of size close to vertex connectivity k. Here, we present the exact statements and some related discussions:

Theorem 1.1. *Every k-vertex-connected n-vertex graph has a CDS packing of size $\Omega\big(\frac{k}{\log n}+1\big)$*[4].

Specifically, we show how to construct a collection of k CDSs, each consisting of $O\big(\frac{n\log n}{k}\big)$ vertices, such that each vertex is in at most $O(\log n)$ of the CDSs. We remark that consequent to our work, Ene et al. [9] presented a nice alternative proof for Theorem 1.1, which uses the metarounding result of Carr and Vempala [6] and the Min-Cost-CDS approximation result of Guha and Khuller [14]. They also show that in planar graphs and minor-closed families of graphs, the bound can be improved to $\Omega(k)$.

Using a similar construction style to that of Theorem 1.1, we also obtain an efficient way to get a large CDS partition, which leads to Theorem 1.2. We find it interesting and somewhat surprising that for this CDS partition construction, we need to use the vertex sampling result (which is presented later in Theorem 1.4).

Theorem 1.2. *Every k-vertex-connecgted graph G has a CDS partition of size $\Omega\big(\frac{k}{\log^5 n}+1\big)$, and if $k=\Omega(\sqrt{n})$, then G has a CDS partition of size $\Omega\big(\frac{k}{\log^2 n}+1\big)$.*

We complement these results by showing that the $\Omega\big(\frac{k}{\log n}\big)$ CDS packing bound is existentially optimal.

Theorem 1.3. *For any sufficiently large n, and any $k\in[1,n/4]$, there exist n-vertex graphs with vertex connectivity k where the maximum CDS packing (or partition) size is $O\big(\frac{k}{\log n}+1\big)$.*

Proof of Theorem 1.3, presented in Appendix A, builds on a base graph H, which has vertex connectivity $O(\log n)$ and maximum CDS packing size $O(1)$, and then uses the probabilistic method [4] to show that for any $k\in[1,n/4]$, there exists a subgraph G of H which shows the claimed logarithmic gap.

1.1.2 Vertex Connectivity and Random Sampling:

Using our new perspective on vertex connectivity, we analyze the vertex connectivity of the graph obtained when randomly sampling a subset of the vertices of a graph. Note that in the following, for a graph $G=(V,E)$ and set of vertices $S\subseteq V$, $G[S]$ denotes the subgraph of G induced by S.

Theorem 1.4. *Consider a k-vertex-connected, n-vertex graph $G=(V,E)$ and let S be a subset of V where each vertex $v\in V$ is included in S (i.e.,* sampled*) independently with probability p. W.h.p., the graph $G[S]$ has vertex-connectivity $\Omega\big(\frac{kp^2}{\log^3 n}\big)$.*[5]

We prove this result by constructing an $\Omega\big(\frac{kp^2}{\log^3 n}\big)$ CDS packing of graph G that only uses vertices in S. We do this construction in parallel with the process of vertices being sampled. Since the size of each CDS packing is upper bounded by vertex connectivity, we get the theorem. Note that this proof shows something even slightly stronger than Theorem 1.4 as the created CDSs are all in $G[S]$ but they even dominate G.

[4]It is straightforward to extend Theorem 1.1 to vertex-capacitated graphs, where for each vertex $v\in V$, the total overlap of CDSs in v is bound to a capacity $C(v)\geq 0$, instead of just 1. This is by normalizing capacities to integers, then substitiuting each node v with a clique of $\min\{C(v),k\}$ vertices. If capacitated vertex connectivity is, e.g., $k\geq 2n^2$, then with scaling down and rounding, we can keep the time complexity and the number of vertices polynomial, without loosing more than a constant factor in the bounds.

[5]Note that the theorem requires $k=\Omega(\log^3 n)$ to be meaningful. Such a polylogarithmic lower bound on k is necessary for all our statements to become non-trivial. This is in all cases necessary as, e.g., shown by Observation C.1.

The bound of Theorem 1.4 is the best possible up to logarithmic factors: simply consider two cliques of size k, connected via a matching of k edges. When sampling vertices with probability p, the new vertex connectivity is given by the number of surviving matching edges, which has expected value of kp^2 as each matching edge survives with probability p^2.

1.1.3 Vertex Connectivity and Broadcast Throughput:

Our results have an interesting application in networking since they lead to routing-based broadcast algorithms with optimal throughput. A *routing-based* (a.k.a. *store-and-forward*) algorithm corresponds to the classical paradigm of routing where messages are viewed as atomic tokens and each node only stores and forwards messages and can not combine them (or parts of them). This is in contrast to the newer (more general and complex) paradigm of *network coding* (see [2] and citations thereof) where each node can send any B-bits function of the received messages. Consider the synchronous network model where in each communication round (time unit), each node can send one message of size at most B bits to all of its neighbors. This model is motivated, e.g., by wireless networks when working above the MAC layer (see e.g. [27, 28]).

For instance, consider the widely-studied *gossiping problem* (a.k.a. *all-to-all broadcast*) where each node has one B-bits size message and each node should receive all the messages. A trivial routing-based solution is to gather the messages in one node and then broadcast them in a pipe-lined fashion, leading to an $O(n)$ rounds solution. This only uses the fact that the graph is connected. A similar $O(n)$ bound follows from [28, Lemma 6.1]. Now, as a toy example, suppose that the graph has vertex connectivity say $n/100$—almost like a clique. It follows from [15] that in this case, network coding solves the problem in just $O(1)$ rounds. However, when restricted to routing, the above trivial $O(n)$ solution remains the best we could do, prior to this work. With CDS packing, we get an optimal $O(\log n)$ rounds routing-based algorithm for this toy example[6].

In general, CDS packing is tightly related to the throughput of routing-based algorithms for concurrent global broadcasts. The brief intuition is that we can route messages along different CDSs (almost) simultaneously and in fact, looking at the transcript of any high-throughput routing-based broadcast algorithm, we can generate a large CDS-packing by following the routes that messages take.

Theorem 1.5. *A CDS packing with size K provides a store-and-forward backbone with broadcast throughput $\Omega(K)$ messages per round. Inversely, a store-and-forward broadcast algorithm with throughput K messages per round induces a CDS packing of size at least K.*

The proof is deferred to Appendix B. From Theorem 1.5, we get that our CDS packing result (Theorem 1.1) also gives a routing-based broadcast algorithm with optimal throughput $\Theta(\frac{k}{\log n})$. Techniques of [15] show that network coding can achieve a throughput of $\Theta(k)$. Thus, our $\Theta(\frac{k}{\log n})$ CDS-packing implies that the *network coding advantage* (i.e., the throughput gain provided by network coding compared to routing) for simultaneous broadcasts is a tight $\Theta(\log n)$.

We note that *network coding* is studied extensively (see [2] and its over 5000 citations) and since its gains are usually accompanied by new complications and costs (see e.g. [11]), determining the *network coding advantage* for different networking models is one of the important related questions, which is of interest both in theory (see, e.g., [1, 3, 13, 30, 31]) and in practice (specially for wireless networks; see [26] and its over 1000 citations). In particular, in a seminal paper, Li et al. [31] use the Tutte-Nash-Williams edge-disjoint spanning trees result to show that in undirected wired networks—the model where in each round each node can send one distinct message to each of its neighbors—the network coding advantage is $\Theta(1)$. Agarwal and Charikar [1] prove an $\Omega(\log^2 n/\log^2 \log n)$ advantage in directed wired networks, with related upper bounds remaining wide open. Very recently, Alon et al. [3] show an $\Omega(\log\log n)$ advantage for wireless network model below the MAC layer (i.e., with collisions).

[6] A simpler $\tilde{O}(\sqrt{n})$ round solution is achievable using Theorem 1.7 or using the proof style of the first part of Lemma 4.2.

1.2 Brief Overview of Approach

For all our results, we need to find or prove the existence of a collection of vertex-disjoint CDSs of some graph H. Whereas in all cases, domination will be straightforward, obtaining connectivity of all CDSs simultaneously is much more challenging as it exactly corresponds to treating all vertex cuts. To cope with the problem of facing exponentially many small vertex cuts, we use a simple *layering* idea, roughly explained as follows. We partition the vertices of H into $\Theta(\log n)$ layers (using different methods). We go through the layers one-by-one and establish connectivity by growing components as we proceed. Growing components turns out easier than proving connectivity directly and we can show that using this step-wise approach, it suffices to consider only polynomially many vertex cuts of H, where cuts of each layer are selected depending on the outcome of the cuts of the previous layers.

Although this *layering* trick seems simple and innocent, it is in fact quite powerful as it provides a new almost-trivial proof for connectivity after edge-sampling (see [21,32]). Actually, this approach proves a novel and interesting generalization of the edge sampling results, for which the older proofs break down:

Theorem 1.6. *Let G be a λ-hyperedge connected hypergraph with n vertices and suppose we sample hyperedges of G, each independently with probability $p = \Omega(\frac{\log n}{\lambda})$. Then, the hypergraph with the sampled hyperedges is connected w.h.p.*

Proof. We view the sampling in $\Theta(\log n)$ layers, where in each layer, each hyperedge that is not sampled in the previous layers gets sampled with probability $\Theta(1/\lambda)$. Consider layer $\ell + 1$ of sampling and suppose that the factor with the sampled hyperedges of layers 1 to ℓ is not connected. For each connected component $\mathcal{C}$ of this factor, $\mathcal{C}$ has at least λ hyperedges that connect it to other components. Thus, with the arrival of the sampled hyperedges of layer $\ell + 1$, with constant probability, at least one of these hyperedges of $\mathcal{C}$ gets sampled and so $\mathcal{C}$ gets connected to at least one other component. Therefore, using Markov's inequality, we get that with the addition of sampled hyperedges of layer $\ell + 1$, with constant probability, the number of connected components goes down by a constant factor. After $\Theta(\log n)$ layers, the number of connected components becomes 1 w.h.p. and thus, connectivity is achieved. □

A similarly simple argument proves a very special case of Theorem 1.4:

Theorem 1.7. *Let G be a k-vertex-connected graph with n vertices suppose we sample each vertex independently with probability $p \geq \Omega(\frac{\log n}{\sqrt{k}})$. W.h.p., the sampled vertices induce a connected graph.*

Proof Sketch. Again, we view the sampling in $L = \Theta(\log n)$ layers, where now in each layer, each vertex that is not sampled in the previous layers gets sampled with probability $\Theta(1/\sqrt{k})$. It is easy to see that the sampled vertices of the first $L/2$ layers are a dominating set of the graph, w.h.p. Having that, consider a layer $\ell > L/2$ and consider the subgraph induced by the sampled vertices of layers 1 to ℓ. From Menger's theorem, we know that each connected component of this subgraph is connected to other components via at least k vertex-disjoint paths (in total), that are made of non-sampled internal nodes. Thanks to the domination provided by the first $L/2$ layers, we get a stronger statement: each such component is connected to other components via at least k vertex-disjoint paths, each *with at most* 2 *non-sampled internal nodes* (thus the length of each of these paths is at most 3). We call these *connector paths*. With the arrival of the sampled vertices of layer $\ell + 1$—which are sampled with probability $\Theta(1/\sqrt{k})$—each component gets connected to another component with at least a constant probability. Thus, again after $L = \Theta(\log n)$ layers, sampled vertices induce a connected graph, w.h.p. □

Theorem 1.7 also shows another interesting thing: we can get a CDS partition of size $\Theta(\sqrt{k}/\log n)$ simply by putting each vertex in a random one of $\Theta(\sqrt{k}/\log n)$ CDSs. In Observation C.1 (proven in

Appendix C), we show that for $k \leq n^{1-\varepsilon}$ for any constant $\varepsilon > 0$, the sampling probability threshold of Theorem 1.7 is indeed tight, up to an $O(\sqrt{\log n})$ factor. For larger k, Theorem 1.7 is tight up to $O(\log n)$.

Unfortunately, the proof of our main results are not as simple as Theorems 1.6 and 1.7, because for them we need to create $\tilde{\Theta}(k)$ (or $\tilde{\Theta}(kp^2)$ in the case of sampling) disjoint or almost-disjoint CDSs. In particular, in each layer, while growing towards connectivity, connected components of different CDSs compete to acquire the new vertices. Resolving this competition while satisfying growth of all (or most) CDSs requires careful assignment of nodes of the new layer to the CDSs. This also introduces probabilistic dependencies that can be fatal and lead to incorrect proofs, if not handled carefully. In fact, we are loosing a $\Theta(\log^2 n)$ factor in the bound of Theorem 1.4 and a $\Theta(\log^3 n)$ factor in that of Theorem 1.2 to cure these dependencies. We present a closer explanation of these issues and also the formal construction and proofs of our results in Section 3 and Section 4.

2 Preliminaries

Notations We usually work with an undirected graph $G = (V, E)$ as our main graph and define $n = |V|$. For a subset $S \subseteq V$ of the vertices, we use $G[S]$ to denote the subgraph of G induced by S.

Definition 2.1 (Dominating Set and Connected Dominating Set)**.** *Given a graph $G = (V, E)$, a set $S \subseteq V$ is called a dominating set iff each vertex $u \in V \setminus S$ has a neighbor in S. The set S is called a connected dominating set (CDS) iff S is a dominating set and $G[S]$ is connected. If S is a dominating set of G, we also say that S dominates V.*

Definition 2.2 (CDS Partition)**.** *A* CDS partition *of a graph $G = (V, E)$ is a partition $V_1 \cup \cdots \cup V_t = V$ of the vertices V such that each set V_i is a CDS. The* size *of a CDS partition is the number of CDSs of the partition. The maximum size of a CDS partition of G is denoted by $K_{CDS}(G)$.*

Figure 1 presents a graph with a CDS partition of size 2, where vertices of each color form a CDS.

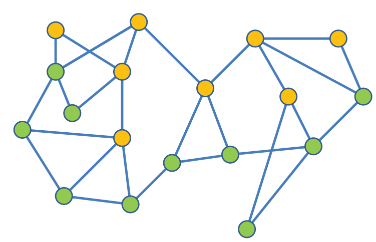

Figure 1: A CDS partition of size 2

Definition 2.3 (CDS Packing)**.** *Let $CDS(G)$ be the set of all CDSs of a graph G. A* CDS packing *of G assigns a non-negative weight x_τ to each $\tau \in CDS(G)$ such that for each vertex $v \in V$, $\sum_{\tau \ni v} x_\tau \leq 1$. The* size *of this CDS packing is $\sum_{\tau \in CDS(G)} x_\tau$. The maximum size of a CDS packing of G is denoted by $K'_{CDS}(G)$.*

Note that a CDS partition is a special case of a CDS packing where each $x_\tau \in \{0, 1\}$. In other words, CDS packing is the LP relaxation of CDS partition when formulating CDS partition as an integer programming problem in the natural way. Consequently, we have $K_{CDS}(G) \leq K'_{CDS}(G)$ for every graph G.

We remark that the maximum CDS partition size of graph G is sometimes also called the *connected domatic number* of G [16, 17, 39]. Analogously, the maximum CDS packing size can be referred to as the *fractional connected domatic number* of G.

As each CDS must contain at least one vertex of each vertex cut, we get $K_{CDS}(G) \leq k$ [16, 39]. Based on the same basic argument, the same upper bound applies to CDS packings (details in Appendix A):

Proposition 2.4. *For each graph with vertex-connectivity k, we have $K_{CDS}(G) \leq K'_{CDS}(G) \overset{(*)}{\leq} k$.*

Note that Theorem 1.3 (proven in Appendix A) shows that the gap in the inequality $(*)$ can be $\Omega(\log n)$.

3 Construction of CDS Packing, and Vertex Connectivity After Sampling

In this section, we prove our main CDS packing result, Theorem 1.1, and our main sampling result, Theorem 1.4. Refer to Section 1.1 for the statements. In particular, we show an efficient construction of a CDS packing of size $O(k/\log n)$. Recall that Theorem 1.3 (proven in Appendix A) shows that this bound is existentially optimal. Moreover, when vertices are independently sampled with probability p, the same construction-style creates a CDS packing of G with size $\Omega(kp^2/\log^3 n)$, using only the sampled vertices. Thus, following Proposition 2.4, this CDS packing acts as a witness to the vertex connectivity of the sampled graph and it proves Theorem 1.4.

To prove Theorems 1.1 and 1.4 together, we explain the construction in the case of vertex sampling with probability p and give a CDS packing with size $\Omega(kq^2/\log n)$, where $q = 1-(1-p)^{1/(3L)}$ and $L = \lambda \log n = \Theta(\log n)$ is the number of layers (as described in Section 1.2). Since $q \geq \frac{p}{6L} = \Theta(\frac{p}{\log n})$, the CDS packing size is $\Omega(kq^2/\log n) = \Omega(kp^2/\log^3 n)$, thus proving Theorem 1.4. When we set $p = 1$ (i.e., no sampling), we get $q = 1$ and thus, the CDS packing size becomes $\Omega(k/\log n)$ as claimed by Theorem 1.1.

3.1 Construction of the CDS Packing

As outlined in Section 1.2, to construct the claimed CDS packing, we partition the vertices of the graph into L layers and we construct CDSs as we go through the layers one-by-one. For our arguments to work, we need the subgraph induced by each layer to have large vertex connectivity. This is hard to achieve when partitioning the vertices of the original graph G into layers. Instead of arguing directly about G, we therefore transform G into a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that we call the *virtual graph* and which is defined next[7].

Virtual Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: For each vertex $v \in V$ (and a sufficiently large constant λ), create $3L = 3\lambda \log n$ *virtual vertices* that are copies of v, one for each layer ℓ in $[1, L]$. Connect two virtual vertices if and only if they are copies of the same real vertex or copies of two adjacent real vertices. Note that for each layer ℓ, the virtual vertices of layer ℓ induce a copy of G. For each set of virtual vertices $\mathcal{W} \subseteq \mathcal{V}$, define the projection $\Psi(\mathcal{W})$ of $\mathcal{W}$ onto G as the set $W \subseteq V$ of real vertices w, for which at least one virtual copy of w is in $\mathcal{W}$.

Proposition 3.1. *Two vertices in $\mathcal{G}$ are connected if and only if they project to the same vertex or to neighboring vertices in G. Thus, $\mathcal{G}[\mathcal{W}]$ is connected (resp. dominating) iff $G[\Psi(\mathcal{W})]$ is connected (resp. dominating).*

To translate the sampling to $\mathcal{G}$, consider the following process: sample each virtual vertex with probability $q = 1-(1-p)^{1/(3L)}$ and then sample each real vertex $v \in V$ if and only if at least one of its virtual copies is sampled (i.e., the sampled real vertices are obtained by projecting the sampled virtual vertices onto G). The probability of each real vertex being sampled is exactly $1-(1-q)^{3L} = p$. Henceforth, we work on $\mathcal{G}$ assuming that each virtual vertex is sampled independently with probability $q = 1-(1-p)^{1/(3L)} \geq \frac{p}{6L}$.

To construct the promised CDS packing on the sampled real vertices, we create a CDS *partition* of size $\Omega(kq^2) = \Omega(kp^2/\log^2 n)$ on the sampled virtual vertices. Since each real vertex has $\Theta(\log n)$ virtual copies, by giving a weight of $1/\Theta(\log n)$ to each CDS, we directly get the claimed CDS packing.

In the rest of this section, we work on $\mathcal{G}$ and show how to construct $t = \delta \cdot kq^2$ vertex-disjoint connected dominating sets on the sampled virtual vertices, for a sufficiently small constant $\delta > 0$. We have t *classes* and we assign each sampled virtual vertex to one class, such that, eventually each class is a CDS, w.h.p. To organize the construction, we group the virtual vertices in L layers, putting three copies of graph G in each layer. Inside each layer, the three copies are distinguished by a *type number* in $\{1, 2, 3\}$.

We now present the notations that we use throughout the construction and the analysis. Let $\mathcal{S}_\ell^i$ be the set of sampled virtual vertices of layers 1 to ℓ that are assigned to class i. Let N_ℓ^i be the number of connected

[7] We note that the virtual graph is a basic construct used in graph theory and it appears under various names. See, e.g., [12, 33].

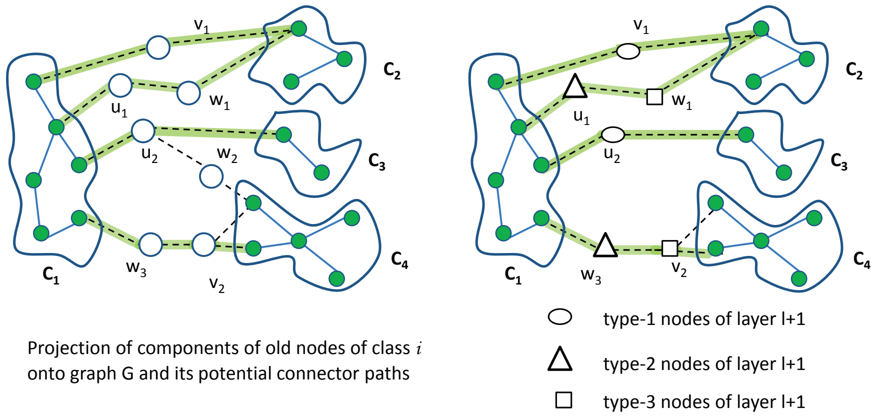

Figure 2: Potential Connector Paths for component $\mathcal{C}_1$ in layer $\ell+1$ copies of G

components of $\mathcal{G}[\mathcal{S}_\ell^i]$. Finally, define $M_\ell := \sum_{i=1}^{t}(N_\ell^i - 1)$ to be the total number of excess components after considering layers $1, \ldots, \ell$, compared to the ideal case where each class is connected.

Initially $M_1 \leq n$. The idea is to do the assignments of vertices to classes such that M_ℓ decreases essentially exponentially with ℓ, until it becomes zero, meaning each class induces a connected sub-graph.

The class assignments are performed in a recursive manner based on the layer numbers. We begin the assignment with a jump-start, assigning sampled virtual vertices of layers 1 to $\frac{L}{2}$ to random classes. We show in Lemma 3.2 that this already gives domination. The proof is simple and deferred to Appendix D.

Lemma 3.2 (Domination Lemma). *W.h.p., for each class i, $\mathcal{S}_{L/2}^i$ dominates $\mathcal{V}$.*

Note that the domination of each class follows directly from this lemma. For the rest of this section, we assume that for each class i, $\mathcal{S}_{L/2}^i$ dominates $\mathcal{V}$, and we use this property later to get short connector paths (in the flavor of those in the proof of Theorem 1.7 in Section 1.2).

After the first $\frac{L}{2}$ layers, we go over the layers one by one and for each layer $\ell \in [L/2, L-1]$, we assign vertices of layer $\ell+1$ to classes based on the assignments of vertices of layers 1 to ℓ. In the rest of this section, we explain this assignment for layer $\ell+1$. We refer to vertices of layers 1 to ℓ as *old vertices* whereas vertices of layer $\ell+1$ are called *new vertices*. The goal is to perform the class assignment of the new sampled vertices such that (in expectation) the number of connected components decreases by a constant factor in each layer (formalized by the *Fast Merger Lemma* presented as Lemma 3.5. During the recursive assignments, our main construction tool will be the concept of *connector paths* (see the proof of Theorem 1.7), defined next.

3.2 Connector Paths

Consider a class i, suppose $N_\ell^i \geq 2$, and consider a component $\mathcal{C}$ of $\mathcal{G}[\mathcal{S}_\ell^i]$. We use the projection $\Psi(\mathcal{S}_\ell^i)$ onto graph G as defined above. A path P in G is called a *potential connector* for $\Psi(\mathcal{C})$ if it satisfies the following three conditions: (A) P has one endpoint in $\Psi(\mathcal{C})$ and the other endpoint in $\Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$, (B) P has at most two internal vertices, (C) if P has exactly two internal vertices and has the form s, u, w, t where $s \in \Psi(\mathcal{C})$ and $t \in \Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$, then w does not have a neighbor in $\Psi(\mathcal{C})$ and u does not have a neighbor in $\Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$. Condition (C) is an important condition, requiring *minimality* of each potential connector path. That is, there is no potential connector path connecting $\Psi(\mathcal{C})$ to another component of $\Psi(\mathcal{S}_\ell^i)$ via only u or only w.

From a potential connector path P on graph G, we derive a potential connector path $\mathcal{P}$ on virtual graph $\mathcal{G}$ by determining the types of related internal vertices as follows: (D) If P has one internal real vertex w, then for $\mathcal{P}$ we choose the virtual vertex of w in layer $\ell+1$ in $\mathcal{G}$ with type 1. (E) If P has two internal real vertices w_1 and w_2, where w_1 is adjacent to $\Psi(\mathcal{C})$ and w_2 is adjacent to $\Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$, then for $\mathcal{P}$ we choose the virtual vertices of w_1 and w_2 in layer $\ell+1$ with types 2 and 3, respectively. Finally, for each endpoint w of P we add the copy of w in $\mathcal{S}_\ell^i$ to $\mathcal{P}$. A given potential connector path $\mathcal{P}$ on the virtual vertices of layer $\ell+1$

is called a *connector path* if and only if the internal vertices of $\mathcal{P}$ are sampled. We call a connector path that has one internal vertex a *short connector path*, whereas a connector path with two internal vertices is called a *long connector path*. Because of condition (C), and rules (D) and (E) above, we get the following fact:

Proposition 3.3. *For each class i, each type-2 virtual vertex u of layer $\ell+1$ is on connector paths of at most one connected component of $\mathcal{G}[\mathcal{S}_\ell^i]$.*

Figure 2 demonstrates an example of potential connector paths for a component $\mathcal{C}_1 \in \mathcal{G}[\mathcal{S}_\ell^i]$. The figure on the left shows graph G, where the projection $\Psi(\mathcal{S}_\ell^i)$ is indicated via green vertices, and the green paths are potential connector paths of $\Psi(\mathcal{C}_1)$. On the right side, the same potential connector paths are shown, where the type of the related internal vertices are determined according to rules (D) and (E) above, and vertices of different types are distinguished via different shapes (for clarity, virtual vertices of other types are omitted).

The following lemma shows that each component that is not alone in its class is likely to have many connector paths. The proof is deferred to Appendix D.

Lemma 3.4 (Connector Abundance Lemma). *Consider a layer $\ell \geq L/2$ and a class i such that $\mathcal{S}_{L/2}^i \subseteq \mathcal{S}_\ell^i$ is a dominating set of $\mathcal{G}$ and $N_\ell^i \geq 2$. Further consider an arbitrary connected component $\mathcal{C}$ of $\mathcal{G}[\mathcal{S}_\ell^i]$. Then, with probability at least $1/4$, $\mathcal{C}$ has at least t [8] internally vertex-disjoint connector paths.*

3.3 Recursive Step of Class Assignment

From Lemma 3.4 we know that for each connected component of each class i with $N_\ell^i \geq 1$, with probability at least $1/2$, this component has at least t connector paths. For each such component, pick exactly t of its connector paths. Using Markov's inequality, we get that with probability at least $1/4$, we have at least $M_\ell \cdot t$ connector paths in total, over all the classes and components. We use these connector paths to assign the class numbers of vertices of layer $\ell+1$. This part is done in a greedy fashion, in three stages as follows:

(I) For each type-1 new vertex v: For each class i, define the *class-i-degree of v* to be the number of connected components of class i that have a *short* connector path through v. Let Δ be the maximum *class-i-degree* of v as i ranges over all classes, and let i^* be a class that attains this maximum. If $\Delta \geq 1$: Assign v to class i^*. Also, remove all connector paths of all classes that go through v and remove all connector paths of all the connected components of class i^* that have v on their short connector paths.

(II) For each type-3 new vertex u: For each class i, define the *class-i-degree of u* to be the number of connected components of class i which have a *long* connector path through u. Let Δ be the maximum *class-i-degree* of u as i ranges over all classes and let i^* be a class that attains this maximum. If $\Delta \geq 1$: Assign u to class i^*. Moreover, each of the Δ long connector paths of class i^* that goes through u also has a type-2 internal vertex. Let these type-2 vertices be $v_1, \ldots, v_\Delta$. Assign $v_1, \ldots, v_\Delta$ to class i^*. Then, remove all the connector paths that go through u or any of the vertices $v_1, \ldots, v_\Delta$. Also remove all connector paths of each component of class i^* that has a connector path going through u.

(III) Assign each remaining new vertex to a random class.

Lemma 3.5 (Fast Merger Lemma). *For each $\ell \geq \frac{L}{2}$ and every assignment of the sampled vertices of layers $1, \ldots, \ell$ to classes such that for all classes i, $\mathcal{S}_{L/2}^i$ is a dominating set of $\mathcal{G}$, we have (a) $M_{\ell+1} \leq M_\ell$, and (b) with probability at least $1/2$, $M_{\ell+1} \leq \frac{5}{6} \cdot M_\ell$.*

The proof is based on an accounting argument that uses the total number of remaining connector paths over all classes and components as the budget, and shows that number of components that are merged, each with at least one other component, is at least $\frac{M_\ell}{3}$. We defer the details to Appendix D.

[8] Recall that $t = \delta \cdot kq^2$, for a small enough constant δ, is the number of classes.

4 Construction of CDS Partition

We now prove our CDS partition result, Theorem 1.2. See Section 1.1 for the statement. To achieve this CDS partition, we use the general construction style of the CDS packing of Theorem 1.4. Here, we explain the key changes: since in a CDS partition, each node can only join one CDS, we cannot use the layering style of Theorem 1.4, which uses $\Theta(\log n)$ copies of G and where each node can join $O(\log n)$ CDSs. Instead, we use *random* layering: each node chooses a random *layer number* in $\{1, \dots, L\}$ and a random *type number* in $\{1, 2, 3\}$. The construction is again recursive, with first assigning nodes of layers 1 to $L/2$ randomly to one of t random classes. This suffices to give domination. Here, the number of classes $t = \delta \frac{k}{\log^2 n}$ if $k = \Omega(\sqrt{n})$ and $t = \delta \frac{k}{\log^5 n}$, otherwise. After that, for each $\ell \geq 2$, we assign class numbers of nodes of layer $\ell + 1$ based on the configuration of classes in layers 1 to ℓ, using the same greedy algorithm as in Section 3.3. Next, we re-define the connector paths, incorporating the random layering.

Connector Paths for CDS Partition: Let V_ℓ^i be the set of all nodes of layers 1 to ℓ in class i. Consider a component $\mathcal{C}$ of $G[V_\ell^i]$. Define potential connector paths on G as in Section 3.2 (conditions (A) to (C)). Then, for each potential connector path on G, this path is called a *connector path* if its internal nodes are in layer $\ell + 1$ and the types of its internal nodes satisfy rules (D) and (E) in Section 3.2.

The key technical change compared to the CDS packing of Section 3, appears in obtaining a Connector Abundance Lemma, which we present in two versions, depending on the magnitude of vertex-connectivity.

Lemma 4.1 (**Connector Abundance Lemma**)**.** *For each class i and layer $\ell \leq L/2$ such that $N_\ell^i \geq 2$, for each connected component $\mathcal{C}$ of $G[V_\ell^i]$, with probability at least $1/2$, $\mathcal{C}$ has at least $\Omega(\frac{k}{\log^5 n})$ internally vertex-disjoint connector paths, with independence between different layers $\ell \leq L/2$.*

Lemma 4.2 (**Stronger Connector Abundance Lemma for Large Vertex Connectivity**)**.** *Assume $k = \Omega(\sqrt{n})$. W.h.p., for each class i and layer $\ell \leq L/2$ such that $N_\ell^i \geq 2$, each connected component $\mathcal{C}$ of $G[V_\ell^i]$ has at least $\Omega(\frac{k}{\log^2 n})$ internally vertex-disjoint connector paths.*

The proof of Lemma 4.1 uses the sampling result of Theorem 1.4. Roughly speaking, for each layer $\ell \leq L/2$, from Lemma 4.1 we get that the graph induced by layers $\ell + 1$ to L has vertex connectivity at least $\Omega(\frac{k}{\log^3 n})$ and then, we get that each component $\mathcal{C}$ has at least $\Omega(\frac{k}{\log^5 n})$ internally vertex-disjoint connector paths (with internal nodes of right type and layer $\ell+1$). The proof of Lemma 4.2 is more involved. Intuitively, for each class i, it first contracts components of $G[V_1^i]$ and then argues about all $2^{O(n/k)}$ cuts of the resulting graph, using the fact that the large vertex connectivity $k = \Omega(\sqrt{n})$ gives a good enough concentration to compensate for this large number of cuts. The proofs of Lemmas 4.1 and 4.2 are deferred to Appendix E.

5 Open Questions

As the most important follow-up research direction, it is certainly interesting to try to use the new perspective on vertex connectivity provided in this paper to approach other problems related to vertex connectivity. Besides that, this paper leaves a number of interesting and important open questions. Most of our results are only tight up to logarithmic factors and it would be interesting to close these gaps. In particular, when sampling each node with probability p, assuming that kp^2 is large enough, we suspect that the remaining vertex connectivity should be $\Theta(kp^2)$. We also believe that construction from Section 4 to compute a CDS partition should actually allow to obtain a CDS partition of size $\Omega(k/\log^2 n)$ for all values of k and not just for $k = \Omega(\sqrt{n})$. Note that our approach would immediately give this if we could prove that when sampling with probability $1/2$, the remaining vertex connectivity is still $\Omega(k)$ rather than $\Omega(k/\log^3 n)$ as we get from Theorem 1.4. Note that even proving an $\Omega(k/\log^2 n)$ lower bound on the size of a maximum CDS partition would still leave a logarithmic gap with respect to the upper bound of Theorem 1.3.

Acknowledgment: We thank David Karger and Bernhard Haeupler for helpful discussions and comments.

References

[1] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proc. IEEE Inf. Theory Workshop*, pages 247–249, 2004. 4

[2] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. Inf. Theory*, 46(4):1204–1216, 2000. 4

[3] N. Alon, M. Ghaffari, B. Haeupler, and M. Khabbazian. Broadcast throughput in radio networks: routing vs. network coding. *Manuscript*, 2013. 4

[4] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley & Sons, New York, 1992. 3, 13

[5] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proc. 28th ACM Symp. on Theory of Computing (STOC)*, pages 47–55, 1996. 1

[6] R. Carr and S. Vempala. Randomized metarounding (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, STOC '00, pages 58–62, New York, NY, USA, 2000. ACM. 3

[7] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on*, volume 1, pages 376–380 vol.1, 1997. 2

[8] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad hoc networks using a spine. In *Computer Communications and Networks, 1997. Proceedings., Sixth International Conference on*, pages 34–39, 1997. 2

[9] A. Ene, N. Korula, and A. Vakilian. Improved approximation algorithms for connected domatic partitions and related problems. *http://arxiv.org/abs/1305.4308*, 2013. 3

[10] U. Feige, M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM J. Computing*, 32(1):172–195, 2002. 2

[11] C. Fragouli and E. Soljanin. Network coding fundamentals. *Foundations and Trends in Networking*, 2(1), 2007. 4

[12] D. Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming*, 1(1):168–194, 1971. 7

[13] A. Goel and S. Khanna. On the network coding advantage for wireless multicast in euclidean space. In *Proc. 7th Conf. on Inf. Processing in Sensor Netw. (IPSN)*, pages 64–69, 2008. 4

[14] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, Apr. 1998. 2, 3

[15] B. Haeupler. Analyzing network coding gossip made easy. In *Proc. 43rd ACM Symp. on Theory of Computing (STOC)*, pages 293–302, 2011. 4

[16] B. L. Hartnell and D. F. Rall. Connected domatic number in planar graphs. *Czech. Math. Journal*, 51:173–179, 2001. 2, 6

[17] S. T. Hedetniemi and R. Laskar. Connected domination in graphs. Technical Report 414, Clemson University, SC, USA, 1983. 2, 6

[18] S. T. Hedetniemi and R. Laskar. Topics on domination. *Annals of Discrete Mathematics*, 48, 1991. 2

[19] A. Kanevsky. On the number of minimum size separating vertex sets in a graph and how to find all of them. In *Proc. 1st ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 411–421, 1990. 1

[20] D. R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-out algorithm. In *Proc. 4th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 21–30, 1993. 1

[21] D. R. Karger. Random sampling in cut, flow, and network design problems. In *Proc. 26th ACM Symp. on Theory of Computing (STOC)*, pages 648–657, 1994. 1, 5

[22] D. R. Karger. Using randomized sparsification to approximate minimum cuts. In *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 424–432, 1994. 1

[23] D. R. Karger. A randomized fully polynomial time approximation scheme for the all terminal network reliability problem. In *Proc. 27th ACM Symp. on Theory of Computing (STOC)*, pages 11–17, 1995. 1

[24] D. R. Karger. Minimum cuts in near-linear time. In *Proc. 28th ACM Symp. on Theory of Computing (STOC)*, pages 56–63, 1996. 1

[25] D. R. Karger and M. S. Levine. Finding maximum flows in undirected graphs seems easier than bipartite matching. In *Proc. 30th ACM Symp. on Theory of Computing (STOC)*, pages 69–78, 1998. 1

[26] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 243–254, 2006. 4

[27] F. Kuhn, N. Lynch, and C. Newport. The abstract mac layer. *Distributed Computing*, 24(3):187–206, 2011. 4

[28] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 513–522, 2010. 4

[29] S. Kundu. Bounds on the number of disjoint spanning trees. *J. of Comb. Theory, Series B*, 17(2):199 – 203, 1974. 1

[30] M. Langberg and M. Médard. On the multiple unicast network coding conjecture. In *Proc. 47th Allerton Conf. on Comm., Control, and Computing*, pages 222–227, 2009. 4

[31] Z. Li, B. Li, and L. C. Lau. A constant bound on throughput improvement of multicast network coding in undirected networks. *IEEE Trans. Inf. Theory*, 55(3):1016–1026, 2009. 4

[32] M. V. Lomonosov and V. P. Polesskii. Lower bound of network reliability. *Problems of Information Transmission*, 7:118–123, 1971. 1, 5

[33] L. Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95 – 98, 1972. 7

[34] R. Misra and C. Mandal. Rotation of CDS via connected domatic partition in ad hoc sensor networks. *IEEE Trans. on Mobile Computing*, 8(4):488–499, 2009. 2

[35] T. Moscibroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pages 8 pp.–, 2005. 2

[36] C. S. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. of the London Math. Society*, 36:445–450, 1961. 1

[37] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *Proc. 15th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 286–294, 2003. 13

[38] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *J. of the London Math. Society*, 36:221–230, 1961. 1

[39] B. Zelinka. Connected domatic number of a graph. *Math. Slovaca*, 36:387–392, 1986. 2, 6

A Upper Bounding the Size of CDS-Packing

In this section, we first present the missing proof of Proposition 2.4, and then, in the more interesting part, prove Theorem 1.3, which shows that in some graphs, the maximum CDS packing size is a $\Theta(\log n)$ factor smaller than the vertex connectivity.

Proof of Proposition 2.4. Consider a vertex cut $\mathcal{C} \subseteq V$ of G that has size exactly k. Each CDS τ must include at least one node in $\mathcal{C}$. For each CDS $\tau \in CDS(G)$, pick one node $v \in C$ as a representative of τ in the cut and let us denote it by $Rep(\tau)$. Thus, for any CDS-Packing of G, we have

$$\sum_{\tau \in CDS(G)} x_\tau = \sum_{v \in \mathcal{C}} \sum_{\substack{\tau \in CDS(G) \\ s.t.\ v = Rep(\tau)}} x_\tau \leq \sum_{v \in \mathcal{C}} 1 = |\mathcal{C}| = k.$$

Since the above holds for any CDS-Packing of G, we get that $K'_{CDS}(G) \leq k$. □

Let us recall the statement of Theorem 1.3

Theorem 1.3. *(restated) For any large enough n and any $k \in [1, n/4]$, there exists an n-node graph G with vertex connectivity k such that $K'_{CDS}(G) = O\big(\frac{k}{\log n} + 1\big)$.*

To prove this theorem, in Lemma A.1, we present a graph H with vertex connectivity k, size between 2^k and 4^k, and $K'_{CDS}(\mathsf{H}) < 2$. This lemma proves the theorem for $k = O(\log n)$. To prove the theorem for the case of larger vertex-connectivity compared to n, in Lemma A.2, we look at randomly chosen sub-graphs of H and apply the probabilistic method [4].

Lemma A.1. *For any k, there exists an n-node graph H with vertex connectivity k and $n \in [2^k, 4^k]$ such that $K'_{CDS}(\mathsf{H}) < 2$.*

Proof. We obtain graph H by simple modifications to the graph presented by Sanders et al. [37] for proving an $\Omega(\log n)$ network coding gap in the model where network is directed and each node can send *distinct* unit-size messages to its different outgoing neighbors.

The graph H has two layers. The first layer is a clique of $2k$ nodes. The second layer has $\binom{2k}{k}$ nodes, one for each subset of size k of the nodes of the first layer. Each second layer node is connected to the k first-layer nodes of the corresponding subset. Note that the total number of nodes is $\binom{2k}{k} + 2k \in [2^k, 4^k]$. Let $\mathcal{A}$ and $\mathcal{B}$ denote the set of nodes in the first and second layer, respectively.

First, we show that H has vertex connectivity k. Since the degree of each second layer node is exactly k, it is clear that the vertex connectivity of H is at most k. To prove that the vertex connectivity of H is at least k, let u and v be two arbitrary nodes of H. We show that there are at least k internally vertex disjoint paths between u and v. If u and v are both in $\mathcal{A}$, then there is one direct edge between v and u and there are

$2k-2$ paths of length 2 between them. If exactly one of v and u is in $\mathcal{A}$, e.g., suppose $u \in \mathcal{A}$ and $v \in \mathcal{B}$, then u is directly connected to k neighbors of v. Otherwise, if both u and v are in $\mathcal{B}$, then let p be the size of the intersection of the neighbors of v and u. Note that these neighbors are all in $\mathcal{A}$. It is clear that u and v have exactly p paths of length 2 between themselves and $k-p$ paths of lengths 3, and that these paths are internally vertex disjoint.

To see that $K'_{CDS}(H) < 2$, first note that each CDS τ must include at least $k+1$ nodes of $\mathcal{A}$. This is because, otherwise, there are at least k nodes of $\mathcal{A}$ that are not included in τ and thus, there is a node in $\mathcal{B}$—corresponding to a subset of size k of these uncovered nodes of $\mathcal{A}$—which is not dominated by τ. Thus we have,

$$\sum_{v \in \mathcal{A}} \sum_{\substack{\tau \in CDS(\mathsf{H}) \\ s.t.\ v \in \tau}} x_\tau \geq (k+1) \cdot \sum_{\tau \in CDS(\mathsf{H})} x_\tau.$$

On the other hand we have,

$$\sum_{v \in \mathcal{A}} \sum_{\substack{\tau \in CDS(\mathsf{H}) \\ s.t.\ v \in \tau}} x_\tau \leq \sum_{v \in \mathcal{A}} 1 = |\mathcal{A}| = 2k,$$

and thus we can conclude that $\sum_{\tau \in CDS(\mathsf{H})} x_\tau \leq \frac{2k}{k+1} < 2$. Since this holds for any CDS-Packing of H, we get $K'_{CDS}(H) < 2$. □

Note that in the above construction, we have $K_{CDS}(\mathsf{H}) = 1$ as H is connected and $K_{CDS}(\mathsf{H})$ has to be an integer.

Lemma A.2. *For each large enough k and $\eta \in [4k, 2^k]$, there exists a sub-graph $H' \subseteq \mathsf{H}$ that has η nodes and vertex connectivity k but $K'_{CDS}(H') = O(\frac{k}{\log \eta})$.*

Proof. Pick an arbitrary $k \geq 64$, fix an $\eta \in [4k, 2^k]$ and let $\beta = \frac{\log \eta}{8}$. Consider a random subset $V_z \subseteq V$, where V_z includes all nodes of $\mathcal{A}$ and for each node $u \in \mathcal{B}$, u is independently included in V_z with probability p, where

$$p = \frac{65\beta^2}{\binom{2k-\beta}{k}}.$$

We now look at the sub-graph H_z of H induced on V_z. With the same argument as for H, we get that for any such V_z, the graph H_z has vertex connectivity exactly k. We show that (a) with probability at least $\frac{1}{2}$, V_z is such that $K'_{CDS}(H_z) < \frac{2k}{\beta} = O(\frac{k}{\log \eta})$, and (b) with probability at least $\frac{3}{4}$, we have $|V_z| \leq \eta$. A union bound then completes the proof.

Property (a) We first show that with probability at least $\frac{1}{2}$, V_z is such that there does not exist a subset of size β of the nodes of $\mathcal{A}$ that dominates V_z. For each subset $W \subset \mathcal{A}$ such that $|W| = \beta$, there are $\binom{2k-\beta}{k}$ nodes in $\mathcal{B}$ which are not dominated by W. Thus, for W to dominate V_z, none of these second layer nodes should be included in V_z. The probability for this to happen is

$$(1-p)^{\binom{2k-\beta}{k}} \leq e^{-65\beta^2}$$

There are $\binom{2k}{\beta}$ possibilities for set W. Hence, using a union bound, the probability that there exists such a set W that dominates V_z is at most

$$\begin{aligned} e^{-65\beta^2}\binom{2k}{\beta} \leq e^{-65\beta^2} \cdot (\frac{2ek}{\beta})^\beta &\overset{(\dagger)}{<} e^{-65\beta^2} \cdot (\eta^2)^\beta \\ &= e^{-65\beta^2+64\beta^2} \leq \frac{1}{2}, \end{aligned}$$

where Inequality (†) follows since $64 \leq k \leq \frac{\eta}{4}$, which gives $2ek < k^2 < \eta^2$.

Thus, with probability at least $\frac{1}{2}$, V_z is such that each CDS of H_z includes at least $\beta+1$ nodes of $\mathcal{A}$. From this, similar to the last part of the proof of Lemma A.1, we have that, $\sum_{\tau \in CDS(\mathsf{H})} x_\tau \leq \frac{2k}{\beta+1} < \frac{2k}{\beta}$. Since this holds for any packing of H_z, we get that with probability at least $\frac{1}{2}$, V_z is such that $K'_{CDS}(H_z) < \frac{2k}{\beta}$.

Property (b) Note that $\mathbb{E}[|V_z|] = 2k + p \cdot \binom{2k}{k}$. Substituting $p = \frac{65\beta^2}{\binom{2k-\beta}{k}}$ and noting that $\beta \leq \frac{k}{2}$, we get

$$\begin{aligned}
\mathbb{E}[|V_z|] - 2k &= p \cdot \binom{2k}{k} = 65\beta^2 \cdot \frac{\binom{2k}{k}}{\binom{2k-\beta}{k}} \\
&= 65\beta^2 \cdot \frac{2k}{2k-\beta} \cdot \frac{2k-1}{2k-\beta-1} \cdots \frac{k+1}{k-\beta+1} \\
&\leq 65\beta^2 \cdot (1 + \frac{2\beta}{k})^k \\
&\leq \frac{65 \log^2 \eta}{64} \cdot \eta^{\frac{1}{4}} \leq \frac{\eta}{4}.
\end{aligned}$$

As the second-layer nodes are picked independently, for η sufficiently large, we can apply a Chernoff bound to get $\Pr[|V_z| - 2k > \frac{\eta}{2}] \leq \frac{1}{4}$. Since $2k \leq \frac{\eta}{2}$, we then obtain $\Pr[|V_z| > \eta] \leq \frac{1}{4}$. If desired, we can adjust the number of nodes to exactly η by adding enough nodes in the second layer which are each connected to all nodes of the first layer. □

B Missing Proof of Theorem 1.5: CDS Packing vs. Throughput

In this section, we prove Theorem 1.5. For simplicity, we restate the theorem.

Theorem 1.5. *(restated) A CDS packing with size K provides a store-and-forward backbone with broadcast throughput $\Omega(K)$ messages per round. Inversely, a store-and-forward broadcast algorithm with throughput K messages per round induces a CDS packing of size at least K.*

Proof of Theorem 1.5. First consider a CDS τ and suppose that the graph induced by τ has diameter D_τ. Using τ, we can perform p broadcasts (or multicast or unicasts) in time $O(p + D_\tau)$. This can be seen as follows: Since τ is a dominating set, we can deliver each message to a node of τ in at most p rounds. Because τ is connected, $O(p + D_\tau)$ rounds are enough to broadcast the messages to all nodes in τ. Finally, because τ is a dominating set, at most p more rounds are enough to deliver the messages to all the desired destination nodes. Hence, a CDS structure allows for performing broadcasts with an (amortized) rate of $\Omega(1)$ messages per round. In other words, a CDS can be viewed as a communication backbone with throughout $\Omega(1)$ messages per round.

Consequently, K vertex-disjoint CDS sets form a communication backbone with throughput of $\Omega(K)$ messages per round. Intuitively, we can use those K vertex-disjoint sets in parallel with each other and get throughput of $\Omega(1)$ message per round from each of them. For a more formal description, consider p broadcasts such that no more than q broadcasts have the same source node. We first deliver each messages to a randomly and uniformly chosen CDS set. This can be done in time at most q. With high probability, the number of messages in each CDS is $O(\frac{p}{K} + \log n)$ and thus, we can simultaneously broadcast messages in time $O(\frac{p}{K} + \log n + D_{\max})$ where $D_{\max}$ is the maximum diameter of the CDSs. Thus, the total time for completing all the broadcasts is $O(q + \frac{p}{K} + \log n + D_{\max})$. That is, we can perform the broadcasts with a rate (throughput) of $\Omega(\min\{K, p/q\})$. Note that since each source can only send one packet per round, if $q \leq K$, then the maximum achievable throughout with any algorithm including network coding approaches is at most q packets per round. In other words, in that case, the bottleneck is not the communication protocol but rather the sources of the messages. As long as no node is the source of more than $\Theta(p/K)$ messages, K vertex-disjoint CDS sets form a communication backbone with throughput $\Omega(K)$.

Similarly one can see that a CDS packing with size K provides a backbone with a throughput of $\Omega(K)$ messages per round. The only change with respect to above description is that now each node v spends a x_τ-fraction of its time for sending the messages assigned to CDS τ for every τ such that $v \in \tau$. Further, messages are assigned to each CDS τ with probability proportional to x_τ. We remark here that even though this scheme provides a backbone with throughput $\Omega(K)$, if the weights x_τ are too small, the outlined time sharing might impose a considerably large additive term on the overall time for completing the broadcasts. In fact since the number of potential CDS sets can be exponential, the time sharing might lead to exponentially large additive terms. Note that in the CDS packing we present in Theorem 1.1, each CDS has weight at least $\Omega(1/\log n)$ and thus using the partition also leads to an asymptotically optimal throughput for a relatively small number of broadcast messages.

Let us now argue that a broadcast protocol with throughput K also leads to a CDS packing of size K. Suppose that there exists a (possibly large enough) number p and a store-and-forward algorithm which broadcasts p messages (originating from potentially different sources) in $T \leq \frac{p}{K}$ rounds. For each message σ that is being broadcast, define set $S(\sigma)$ to be the set of nodes that send σ in some round of the algorithm. Clearly $S(\sigma)$ induces a connected sub-graph and because every node needs to receive the message $S(\sigma)$ also is a dominating set. For each node v and message σ, let $N_\sigma(v)$ be the number of rounds in which node v sends message σ and let $y_\sigma(v) = \frac{N_\sigma(v)}{T}$. Moreover, for each CDS τ such that $v \in \tau$, let

$$z_\tau(v) = \sum_{\substack{\sigma \\ S(\sigma)=\tau \,\wedge\, v\in\tau}} y_\tau(v).$$

Finally, let $x_\tau = \min_{v\in\tau}\{z_\tau(v)\}$. Given these parameters, first notice that for each node v, we have

$$\sum_{\substack{\tau \\ v\in\tau}} x_\tau \leq \sum_{\substack{\tau \\ v\in\tau}} z_\tau(v) = \sum_{\substack{\tau \\ v\in\tau}} \sum_{\substack{\sigma \\ S(\sigma)=\tau}} y_\sigma(v) = \sum_{\sigma} \frac{N_\sigma(v)}{T} \overset{(\dagger)}{\leq} 1.$$

Here, Inequality (†) is because in each round, node v can send at most one message and thus, $\sum_\sigma N_\sigma(v) \leq T$. On the other hand, we show that $\sum_\tau x_\tau \geq \frac{p}{T} = \Omega(K)$. For this purpose, consider a CDS τ and let u^* be a node such that $z_\tau(u^*) = x_\tau$. Since each message σ such that $S(\sigma) = \tau$ is sent at least once by u^*, we have

$$\begin{aligned}\sum_{\substack{\sigma \\ S(\sigma)=\tau}} 1 \leq \sum_{\substack{\sigma \\ S(\sigma)=\tau}} N_\sigma(u^*) = & \sum\nolimits_{\substack{\sigma \\ S(\sigma)=\tau}} y_\sigma(u^*) \cdot T \\ = & \; z_\tau(u^*) \cdot T = x_\tau \cdot T\end{aligned}$$

Moreover, we have that

$$p = \sum_\sigma 1 = \sum_\tau \sum_{\substack{\sigma \\ S(\sigma)=\tau}} 1 \leq \sum_\tau x_\tau \cdot T$$

Thus, $\sum_\tau x_\tau \geq \frac{p}{T}$. Since $T \leq \frac{p}{K}$, we get that $\sum_\tau x_\tau \geq K$. □

Remark We remark that in the general formulation of CDS packings, each node might participate in arbitrarily many (in fact up to exponentially many) CDSs. This would make CDS packing inefficient from a practical point of view if the number of messages is small compared to the number of CDSs used. Fortunately, in our construction (cf., Theorem 1.1), each node only participates in $O(\log n)$ CDSs, which makes the CDS packing efficient even for a small number of messages.

C Observation C.1

Now we prove Observation C.1, which shows that the bound of $\Theta(\frac{\log n}{k})$ which we proved in Section 1.2 for vertex-sampling probability so that the sampled vertices induce a connected graph is optimal up to an $O(\sqrt{\log n})$ factor.

Observation C.1. *For every k and n, there exists an n-vertex graph G with vertex connectivity k such that if we independently sample vertices with probability $p \leq \sqrt{\log(n/2k)}/\sqrt{2k}$, then the subgraph induced by the sampled vertices is disconnected with probability at least $1/2$.*

Proof of Observation C.1. Consider a graph G composed of a chain of $\frac{n}{k}$ cliques, each of size k, where each two consecutive cliques on the chain are connected by a matching of size k. For simplicity, assume that $\frac{n}{k}$ is an even integer. Clearly, this graph has vertex connectivity k. Suppose that the sampling probability p is less than $\sqrt{\log(\frac{n}{2k})/(2k)}$. We show that, with probability at least $1/2$, the graph induced on the sampled vertices is disconnected.

Let us number the cliques from 1 to $\frac{n}{k}$. First note that some nodes of each clique are sampled w.h.p. Let $J = \{j | j \in [1, \frac{n}{k}] \wedge j \equiv 0 \pmod 2\}$, i.e., the set of even numbers in the range $[1, \frac{n}{k}]$. If the sampled graph is connected, then for each $j \in J$, at least on of the edges of the matching connecting clique $j-1$ with clique j has to be in the sampled graph. For this, there have to be two adjacent nodes u from clique $j-1$ and v from clique j that are both sampled. We call such a pair of nodes *connecting*. For each $j \in J$, there are k possible choices for a connecting pair and each choice has probability $\frac{1}{s^2}$ to be sampled (to be connecting). Moreover, these pairs are vertex disjoint. Thus, for each $j \in J$, the probability that the sampled graph has a connecting pair between cliques $j-1$ and j is

$$1 - \left(1 - \frac{1}{s^2}\right)^k \leq 1 - 4^{-k/s^2} < 1 - 4^{-\log(\frac{n}{2k})/2} = 1 - \frac{2k}{n}.$$

Since J only contains even values, the pairs of consequent cliques that we consider are disjoint. Thus, the events whether they have a connecting pair or not are independent. Therefore, the probability that we have at least one connecting pair for each value of $j \in J$ is at most $\left(1 - \frac{2k}{n}\right)^{|J|} = \left(1 - \frac{2k}{n}\right)^{\frac{n}{2k}} < \frac{1}{e} < 1/2$. $\square$

D Missing Proofs of Section 3

Proof of Lemma 3.2 . Since G has vertex connectivity k, for each node real node v, v has at least k real neighbors and in these k real neighbors, in expectation at least $\frac{kq}{2t} = \Omega(\log n)$ virtual nodes are sampled, have layer number 1 to $\frac{L}{2}$, and join class i, i.e., are in $\mathcal{S}^i_{\frac{L}{2}}$. Thus, the claim follows from a standard Chernoff argument combined with a union bound over all choices of v and over all classes. $\square$

Proof of Lemma 3.4. Fix a layer $\ell \in [L/2, L-1]$. Let $\mathcal{D}$ be the set of dominating sets of $\mathcal{G}$ consisting only of nodes from layers $1, \ldots, L/2$. Further, for all $\ell \geq L/2$, let $\mathcal{D}_\ell$ contain all sets $\mathcal{T} \subseteq \mathcal{V}_\ell$ such that there exists a $D \in \mathcal{D}$ for which $D \subseteq \mathcal{T}$. That is, $\mathcal{D}_\ell$ is the collection of all sets of virtual nodes in layers $1, \ldots, \ell$ that contain a dominating set $D \in \mathcal{D}$. Fix an arbitrary set $\mathcal{T} \in \mathcal{D}_\ell$ and fix $\mathcal{S}^i_\ell = \mathcal{T}$.

Consider the projection $\Psi(\mathcal{S}^i_\ell)$ onto G and recall Menger's theorem: Between any pair (u, v) of non-adjacent nodes of a k-vertex connected graph, there are k internally vertex-disjoint paths connecting u and v. Applying Menger's theorem to a node in $\Psi(\mathcal{C})$ and a node in $\Psi(\mathcal{S}^i_\ell \setminus \mathcal{C})$, we obtain at least k internally vertex-disjoint paths between $\Psi(\mathcal{C})$ and $\Psi(\mathcal{S}^i_\ell \setminus \mathcal{C})$ in G. We first show that these paths can be shortened so that each of them has at most 2 internal nodes i.e., to get property (B) of potential connector paths. Pick an arbitrary one of these k paths and denote it $P = v_1, v_2, \ldots, v_r$, where $v_1 \in \Psi(\mathcal{C})$ and $v_r \in \Psi(\mathcal{S}^i_\ell \setminus \mathcal{C})$. By the assumption that $\mathcal{S}^i_{L/2}$ dominates $\mathcal{G}$, we get that $\Psi(\mathcal{S}^i_\ell)$ dominates G. Hence, since $v_1 \in \Psi(\mathcal{C})$ and

$v_r \in \Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$, either there is a node v_i along P that is connected to both $\Psi(\mathcal{C})$ and $\Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$, or there must exist two consecutive nodes v_i, v_{i+1} along P, such that one of them is connected to $\Psi(\mathcal{C})$ and the other is connected to $\Psi(\mathcal{S}_\ell^i \setminus \mathcal{C})$. In either case, we can derive a new path P' which satisfies (B) and is internally vertex-disjoint from the other $k-1$ paths since its internal nodes are a subset of the internal nodes of P and are not in $\Psi(\mathcal{S}_\ell^i)$. After shortening all the k internally vertex-disjoint paths, we get k internally vertex-disjoint paths in graph G that satisfy conditions (A) and (B), as stated Section 3.2.

Now using rules (D) and (E) in Section 3.2, we get k internally vertex-disjoint potential connector paths on the virtual nodes of layer $l+1$. It is clear that during the transition from the real nodes to the virtual nodes, the potential connector paths remain internally vertex-disjoint. Now, for each fixed potential connector path on the virtual nodes, the probability that the internal nodes of this path are sampled is at least q^2. Hence, in expectation, $\mathcal{C}$ has kq^2 internally vertex-disjoint connector paths (on virtual nodes). A simple application of Markov's inequality shows that with probability at least $1/2$, $\mathcal{C}$ has at least $t = \Omega(kq^2)$ internally vertex-disjoint connector paths. □

Proof of Lemma 3.5. For part (a) of lemma, note that since for each class i, set $\mathcal{S}_{L/2}^i$ is a dominating set of $\mathcal{G}$ and for each layer $\ell \geq L/2$, $\mathcal{S}_{L/2}^i \subseteq S_\ell^i$, we get that S_ℓ^i dominates set $\mathcal{V}$. Thus, each virtual sampled node in layer $\ell+1$ has a neighbor in S_ℓ^i which means that each connected component of $\mathcal{G}[\mathcal{S}_{\ell+1}^i]$ contains at least one connected component of $\mathcal{G}[\mathcal{S}_\ell^i]$. Hence, $N_{\ell+1}^i \leq N_\ell^i$, which also means that $M_{\ell+1} \leq M_\ell$.

For part (b), consider a class i such that $N_\ell^i \geq 2$ and let $\mathcal{C}_1$ be a connected component of $\mathcal{G}[\mathcal{S}_\ell^i]$. We say component $\mathcal{C}_1$ is *good* if for at least one connector path p of $\mathcal{C}_1$, all internal nodes of p — one or two nodes depending on whether p is short or long — join class i. Note that if $\mathcal{C}_1$ is good, then it gets connected to another component of $\mathcal{G}[\mathcal{S}_\ell^i]$. In order to prove the lemma, we first show that with probability at least $1/2$, at least $\frac{M_\ell}{3}$ connected components (summed up over all classes) are good. This is achieved using a simple accounting method by considering the number of remaining connector paths as the budget.

We know that with probability at least $1/4$, initially we have a budget of $M_\ell \cdot t$. We show that the greedy algorithm spends this budget in a manner that at the end, we get $M_{\ell+1} \leq \frac{5}{6}M_\ell$.

In each step of each of stages I or II, if respectively a type-1 node or a type-3 nodes and some associated type-2 nodes join a class, then at least Δ components become good where Δ is defined as explained in the algorithm description. We show that in that case, we remove at most $3\Delta t$ connector paths in the related bookkeeping part. Thus, in the accounting argument, we get that at most $3\Delta \cdot t$ amount of budget is spent and Δ components become good. Hence, in total over all steps, at least $\frac{M_\ell}{3}$ components become good.

Let us first check the case of short connector paths, which is performed in stage I. Let v be the new type-1 node under consideration in this step and suppose that the related $\Delta \geq 2$, and node v joins class i^*. For class i^*, we remove all paths of all connected components of i^* that have v on their short connector paths. This includes Δ such connected components, and t connector paths for each such component. Thus, in total we remove at most $\Delta \cdot t$ connector paths of components of class i^*. For each class $i \neq i^*$, we remove at most Δ connector paths. This is because v can be on short connector paths of at most Δ components, at most once for each such component. These are respectively because of definition of Δ and due to internally vertex-disjointedness of connector paths of each component. There are less than t classes other than i^*, so in total over all classes other than i^*, we remove at most $\Delta \cdot t$ connector paths. Therefore, we can conclude that the total amount of decrease in budget is at most $2\Delta \cdot t$.

Now we check the case of long connector paths, performed in stage II. Suppose that in this step, we are working on a type-3 new node u, it has $\Delta \geq 1$, and we assign node u and associated type-2 new nodes v_1, …, v_Δ to class i^*. It follows from Proposition 3.3 that nodes v_1, …, v_Δ are not on long connector paths of components of class i^* other than the Δ components which have long paths through u. Thus, any connector path of class i^* that goes through any of v_1 to v_Δ also goes through u. For each component of class i^* that has a long connector path through u, we remove all the connector paths. By definition of Δ, there are Δ

such components and from each such component, we remove at most t paths. Hence, the number of such connector paths that are removed is at most $\Delta \cdot t$. On the other hand, for each class $i \neq i^*$, we remove at most 2Δ connector paths. This is because by definition of Δ, removing just node u removes at most Δ long paths from each class. Moreover, because of Proposition 3.3 and internally vertex-disjointedness of connector paths of each component, removing each type-2 node v_j (where $j \in \{1, 2, \ldots, \Delta\}$) removes at most one long connector path of one connected component of class $i \neq i^*$. Over all classes $i \neq i^*$, in total we remove at most $2\Delta \cdot t$ connector paths. Hence, when summed up with removed connector paths related to class i^*, we get that the total amount of decrease in the budget is at most $3\Delta \cdot t$.

Now we know that with probability at least $1/4$, at least $\frac{M_\ell}{3}$ connected components (summed up over all classes) are good. Recall that each good component gets merged with at least one other component of its class. Thus, $\Pr\left[M_{\ell+1} \leq \frac{5}{6} \cdot M_\ell \,|\, \mathcal{S}_\ell = \mathcal{T}\right] \geq 1/4$. □

E Missing Proofs of Section 4

Proof of Lemma 4.1. Let W^*_ℓ be the set of all nodes with a layer number in $\{\ell + 1, \ldots, L\}$. Since the probability of each node to be in W^*_ℓ is at least $1/2$ (because $\ell \leq L/2$), Theorem 1.4 shows that, w.h.p, the vertex-connectivity of $G[W^*_\ell]$ is $\Omega(\frac{k}{\log^3 n})$. It is easy to see that therefore, the vertex-connectivity of $G[W^*_\ell \cup V^i_\ell]$ is also $\Omega(\frac{k}{\log^3 n})$. Thus, for each component $\mathcal{C}$ of $G[V^i_\ell]$, we can follow the first part of the proof of Lemma 3.4, this time using Menger's theorem on $G[W^*_\ell \cup V^i_\ell]$, and find $\Omega(\frac{k}{\log^3 n})$ internally vertex-disjoint potential connector paths in graph $G[W^*_\ell \cup V^i_\ell]$. It is clear that the internal nodes of these potential connector paths are not in V^i_ℓ, which means they are in W^*_ℓ. For each potential connector path, for each of its internal nodes, given that this node is in W^*_ℓ, the probability that the node is in layer $\ell + 1$ and has the type which satisfies rules (A) and (B) of Section 3.2 is at least $\Theta(1/L) = \Theta(1/\log n)$. Hence, the probability of each of these potential connector paths being a connector path is at least $\Theta(1/\log^2 n)$. From internally vertex-disjointedness of the potential connector paths, and since there are $\Omega(\frac{k}{\log^3 n})$ of them, it follows that with probability at least $1/2$, component $\mathcal{C}$ has at least $\Omega(\frac{k}{\log^5 n})$ internally vertex-disjoint connector paths. □

Proof of Lemma 4.2. Fix a class i. We start by studying the connected components of $G[V^i_1]$. Each node has probability $\frac{1}{tL} = \frac{\log n}{\delta \lambda k}$ to be in V^i_1. Therefore, $\mathbb{E}[|V^i_1|] = \frac{n \log n}{\delta \lambda k}$. Using a Chernoff bound we get that for δ sufficiently small, w.h.p., $|V^i_1| \leq \frac{2n \log n}{\delta \lambda k}$. Moreover, since each node $v \in V$ has at least k neighbors in G, the expected number of neighbors of v in V^i_1 is at least $\Omega(\log n)$. Using another Chernoff bound (and δ sufficiently small) and then a union bound over all choices of v, we get that w.h.p., each node $v \in V$ has $\Omega(\log n)$ neighbors in V^i_1. Therefore, in particular, each node $v \in V^i_1$ has $\Omega(\log n)$ neighbors in V^i_1. In other words, w.h.p., the degree of each node in $G[V^i_1]$ is $\Omega(\log n)$. Thus, w.h.p., $G[V^i_1]$ has at most $\frac{2n \log n}{\delta \lambda k}$ nodes, each of degree $\Omega(\log n)$. Therefore, for an appropriate choice of the constant δ, w.h.p., the number of connected components of $G[V^i_1]$ is at most $\varepsilon \cdot \frac{n}{k}$ for a given constant $\varepsilon > 0$. Let Σ^i be the set of all connected components of $G[V^i_1]$.

We call each nonempty set A which is a strict subset of Σ^i, i.e., $A \subset \Sigma^i$, a *component cut* of $G[V^i_1]$. Since, w.h.p., we have $|\Sigma^i| \leq \varepsilon \frac{n}{k}$, the number of component cuts of $G[V^i_1]$ is at most $2^{\varepsilon n/k}$, w.h.p.

For each layer $\ell \in [2, \frac{L}{2}]$ and each component cut A of $G[V^i_1]$, we say A is *ℓ-rich* if there are at least $\frac{k}{8}$ internally vertex-disjoint paths p which satisfy the following conditions: (1) p has one end point $s \in A$ and the other endpoint $t \in S \setminus A$, (2) p has at most two internal nodes, (3) if p has two internal nodes and has the form s, u, w, t, then u does not have a neighbor in $S - A$ and w does not have a neighbor in S, (4) all internal nodes of p are in layers $[\ell + 1, L]$. We first show that with high probability, for each layer $\ell \in [2, \frac{L}{2}]$ and each component cut A of $G[V^i_1]$, A is *ℓ-rich*.

Consider an arbitrary component cut A of $G[V^i_1]$. Since graph G is k-vertex connected and because V^i_1 is a dominating set (cf. Lemma 3.2), there are at least k internally vertex-disjoint paths which satisfy conditions (1) and (2). The details of this argument are similar to the first part of the proof of Lemma 3.4. Each of these k paths can be *trimmed* to also satisfy condition (3). To see this, consider a path p as described in condition (3). If v has a neighbor in $S \setminus A$ then there is a trimmed path p' from some node in A to v to some node in $S \setminus A$, which satisfies (3). Similarly, if w has a neighbor in A, then there is a trimmed path p' from A to w to some node in $S \setminus A$, which satisfies (3). In either case, the trimmed path p' remains internally vertex-disjoint from the other $k-1$ paths.

Thus far, we have found k internally vertex-disjoint paths of component cut A satisfying conditions (1) to (3). Now, each internal node u on each of these k paths has probability at least $\frac{1}{2}$ to be in one of layers $[\ell+1, L]$. Formally, this is because, the layer number of u is chosen randomly and so far, the only information exposed about u is that it is not in V_1^i. Let $layer(u)$ be the layer number of u. We get

$$\begin{aligned}\Pr[layer(u) \in [\ell+1, L] | u \notin V_1^i] &= \frac{\Pr[layer(u) \in [\ell+1, L] \wedge u \notin V_1^i]}{\Pr[u \notin V_1^i]} \\ &= \frac{\Pr[layer(u) \in [\ell+1, L]]}{\Pr[u \notin V_1^i]} = \frac{\frac{L-\ell}{L}}{1-\frac{1}{Lt}} \geq \frac{L-\ell}{L} \geq \frac{1}{2},\end{aligned}$$

where the last inequality holds because $\ell \leq \frac{L}{2}$. Hence, each internal node u on each path p out of the k internally vertex-disjoint paths for A has probability at least $1/2$ to be in layer $[\ell+1, L]$. The expected number of paths that also satisfy condition (4) therefore is at least $\frac{k}{4}$. Since the paths are internally vertex-disjoint and layer numbers are chosen independently, we can use a Chernoff bound and conclude that with probability at least $1-e^{-k/32}$, at least $\frac{k}{8}$ of the paths satisfy condition (4). Consequently, with probability at least $1-e^{-k/32}$, the component cut A of $G[V_1^i]$ is ℓ-rich.

Now there are at most $2^{\varepsilon n/k}$ component cuts for $G[V_1^i]$. Thus, using a union bound, the probability that there exists one of them that is not ℓ-rich is at most $2^{\varepsilon n/k} \cdot e^{-k/48}$. Since $k = \Omega(\sqrt{n})$, for ε small enough, this probability is less than $2^{-\Omega(\sqrt{n})} = 2^{-\omega(\log n)}$. Hence, w.h.p., each component cut of $G[V_1^i]$ is ℓ-rich. Using a union bound over all choices of $\ell \in [2, \frac{L}{2}]$, we can also conclude that for each such ℓ, each component cut of $G[V_1^i]$ is ℓ-rich.

We are now ready to show that each connected component $\mathcal{C}$ of $G[V_\ell^i]$ has at least $\Omega(k/L^2)$ internally vertex-disjoint connector paths. Note that each component $\mathcal{C}$ is composed of a subset of the connected components of $G[V_1^i]$ and some nodes of layers $2, \ldots, \ell$. Hence, there is a component cut A' of $G[V_1^i]$ such that $\mathcal{C} \cap V_1^i = A'$. Since the component cut A' is ℓ-rich w.h.p., there are at least $\frac{k}{8}$ paths which satisfy conditions (1) to (4) above. Each such path p satisfies conditions (A), (B). However, path p might not directly satisfy condition (C). This is because, it is possible that p is defined as s, v, w, t which satisfies condition (3) but in graph $G[V_\ell^i]$, node v has a neighbor in $W_\ell^i - \mathcal{C}$ or node w has a neighbor in $\mathcal{C}$. Though, in either case, we can get a path p' of length 2 which satisfies conditions (A), (B), and (C) with just one internal node, either v or w. Note that p' still remains internally-vertex-disjoint from the other paths. Moreover, the internal node of p' is in layers $[\ell+1, L]$.

Now we have $\frac{k}{8}$ potential connectors for $\mathcal{C}$ which have their internal nodes in layers in $[\ell+1, L]$. For each internal node v of any of these paths, it holds that v is in layer $\ell+1$ with probability greater than $\frac{1}{L}$. Formally this is true because, given that node v is in a layer in $[\ell+1, L]$, the exact layer number of v can be chosen after making all the decisions for the first ℓ layers. Therefore, each of the paths that we have found so far has probability at least $\frac{1}{L^2}$ to satisfy the corresponding rule conditions (D) or (E) such that it would be a connector path for $\mathcal{C}$. Hence, the expected number of connector paths of $\mathcal{C}$ is at least $\frac{k}{8L^2}$. Since the potential connectors we found (which have internal nodes in layers $\ell+1$ to L) are internally vertex-disjoint, the events of them being connector paths for $\mathcal{C}$ are independent. Thus, using a Chernoff bound we get that w.h.p., $\mathcal{C}$ has at least $\Omega(k/L^2) = \Omega(k/\log^2 n) = \omega(\log n)$ connector paths. A union bound over all connected components of class i, over all layers $\ell \in [2, \frac{L}{2}]$, and over all choices of the class i completes the proof. □