On Packing Low-Diameter Spanning Trees

Julia Chuzhoy

Toyota Technological Institute at Chicago, USA
https://ttic.uchicago.edu/~cjulia/
cjulia@ttic.edu

Merav Parter

The Weizmann Institute of Science, Israel.
http://www.weizmann.ac.il/math/parter/home
merav.parter@weizmann.ac.il

Zihan Tan

Computer Science Department, University of Chicago, USA.
https://sites.google.com/view/zihantan
zihantan@uchicago.edu

Abstract

Edge connectivity of a graph is one of the most fundamental graph-theoretic concepts. The celebrated *tree packing* theorem of Tutte and Nash-Williams from 1961 states that every k-edge connected graph G contains a collection $\mathcal{T}$ of $\lfloor k/2 \rfloor$ edge-disjoint *spanning* trees, that we refer to as a *tree packing*; the *diameter* of the tree packing $\mathcal{T}$ is the largest diameter of any tree in $\mathcal{T}$. A desirable property of a tree packing for leveraging the high connectivity of a graph in distributed communication networks, is that its diameter is low. Yet, despite extensive research in this area, it is still unclear how to compute a tree packing of a low-diameter graph G, whose diameter is sublinear in $|V(G)|$, or, alternatively, how to show that such a packing does not exist.

In this paper, we provide first non-trivial upper and lower bounds on the diameter of tree packing. We start by showing that, for every k-edge connected n-vertex graph G of diameter D, there is a tree packing $\mathcal{T}$ containing $\Omega(k)$ trees, of diameter $O((101k\log n)^D)$, with edge-congestion at most 2.

Karger's edge sampling technique demonstrates that, if G is a k-edge connected graph, and $G[p]$ is a subgraph of G obtained by sampling each edge of G independently with probability $p = \Theta(\log n/k)$, then with high probability $G[p]$ is connected. We extend this result to show that the diameter of $G[p]$ is bounded by $O(k^{D(D+1)/2})$ with high probability. This immediately gives a tree packing of $\Omega(k/\log n)$ edge-disjoint trees of diameter at most $O(k^{D(D+1)/2})$. We also show that these two results are nearly tight for graphs with a small diameter: we show that there are k-edge connected graphs of diameter $2D$, such that any packing of k/α trees with edge-congestion η contains at least one tree of diameter $\Omega\left((k/(2\alpha\eta D))^D\right)$, for any k, α and η. Additionally, we show that if, for every pair u, v of vertices of a given graph G, there is a collection of k edge-disjoint paths connecting u to v, of length at most D each, then we can efficiently compute a tree packing of size k, diameter $O(D\log n)$, and edge-congestion $O(\log n)$. Finally, we provide several applications of low-diameter tree packing in the distributed settings of network optimization and secure computation.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms;

Keywords and phrases Spanning tree, packing, edge-connectivity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.33

Funding *Julia Chuzhoy*: Part of the work was done while the author was a Weston visiting professor in the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science.

Supported in part by NSF grant CCF-1616584.
Merav Parter: Supported in part by an ISF grant no. 2084/18.
Zihan Tan: Part of the work was done while the author was visiting the Department of Computer Science and Applied Mathematics, Weizmann Institute of Science. Supported in part by NSF grant CCF-1616584.

1 Introduction

Edge connectivity of a graph is one of the most basic graph theoretic parameters, with various applications to network reliability and information dissemination. A key tool for leveraging high edge connectivity of a given graph is *tree packing*: a large collection of spanning trees that are (nearly) edge-disjoint. A celebrated result of Tutte [24] and Nash-Williams [19] shows that for every k-edge connected graph, there is a tree packing $\mathcal{T}$ containing $\lfloor k/2 \rfloor$ edge-disjoint trees. This beautiful theorem has numerous algorithmic applications, but unfortunately it provides no guarantee on the diameter of the individual trees in $\mathcal{T}$. In the worst case, trees in $\mathcal{T}$ may have diameter that is as large as $\Omega(|V(T)|)$, even if the diameter of the original graph is very small. Given a graph G and a collection $\mathcal{T}$ of trees in G, we say that the trees in $\mathcal{T}$ are *edge-disjoint* iff every edge of G lies in at most one tree of $\mathcal{T}$, and we say that they cause *edge-congestion* η iff every edge of G lies in at most η trees of $\mathcal{T}$. The *diameter* of a tree-packing $\mathcal{T}$ is the maximum diameter of any tree in $\mathcal{T}$.

The diameter of a graph is a central graph measure that determines the round complexity of distributed algorithms for various central graph problems, including minimum spanning tree, global minimum cut, shortest s-t path, and so on. All these problems admit a trivial lower bound of $\Omega(D)$ for the round complexity (where D is the diameter of the graph), and in fact a stronger lower bound of $\Omega(D + \sqrt{n})$, which is almost tight for general n-vertex graphs, that was shown by Das-Sarma et al. [23]. Despite attracting a significant amount of attention over the last decade (see e.g., [22, 10, 18, 2, 3, 16, 6, 1, 5, 4]), algorithms that exploit large edge connectivity of the input graph in the distributed setting are quite rare. The only examples that we are aware of are recent algorithms for minimum cut by Daga et al. [4] and by Ghaffari et al. [11].

Censor-Hillel et al. [2] presented several distributed algorithms, that, given a k-edge connected n-vertex graph of diameter D, computes a fractional tree packing of $\Omega(k/\log n)$ trees that are fractionally edge-disjoint[1] in $\widetilde{O}(D + \sqrt{n})$ rounds. These trees have been used to parallelize the flow of information, obtaining nearly optimal *throughput* for store-and-forward algorithms[2]. However, as these trees might have diameter as large as $\Omega(n)$ in the worst case, it is not clear how to use them in order to improve the *round complexity* of the problem at hand, as opposed to improving the throughput. In particular, in terms of optimizing the number of communication rounds, it may still be preferable to send the entire information over a single BFS tree rather than spreading it over *many* trees of potentially *large* diameter.

The problem of computing a low-diameter tree packing was studied later by Ghaffari [6] from the perspective of optimization. Specifically, he studied the multi-message broadcast problem,

[1] In the fractional setting, each tree T in the packing has a weight $w(T)$ and for each edge e, the sum of weights of all trees that contain e is at most 1.

[2] In this class of algorithms, the nodes can only forward the messages they receive (e.g., network coding is not allowed).

where a designated source vertex is required to send k messages to all other nodes in the network. Denoting by $\mathrm{OPT}(G)$ the minimum number of rounds required for the broadcast on an input graph G, he constructed a tree packing of size k, where both the diameter and the congestion are bounded by $\widetilde{O}(\mathrm{OPT}(G))$. While this approach provides a nearly optimal broadcast scheme, it does not provide absolute upper bounds on the diameter of the tree packing, and moreover, the congestion caused by the tree packing can be large.

A recent work of Ghaffari and Kuhn [10] provides the following negative result for packing low-diameter trees into a graph: they show that for any large enough n and any $k \geq 1$, there is a k-edge-connected n-vertex graph of diameter $\Theta(\log n)$, such that, in any partitioning of the graph into spanning subgraphs, all but $O(\log n)$ of the subgraphs have diameter $\Omega(n/k)$. In light of this result, it is natural to consider the following key question:

> *(1) Is it possible to compute a tree packing whose diameter is strongly* sublinear *in $|V(G)|$, provided that the diameter of the input graph G is* sublogarithmic *in $|V(G)|$?*

Our second key question aims at crystallizing the main challenge to computing low-diameter tree packing. So far, we have compared the diameter of the tree packing to the diameter of the original graph. However, as observed above, the results of [10] indicate that there may be a large gap between these two measures, even for graphs whose diameter is logarithmic in n. A more natural reference point is the following. We say that a graph G is (k, D)*-connected*, iff for every pair $u, v \in V(G)$ of distinct vertices, there are k edge-disjoint paths connecting u to v in G, such that the length of each path is bounded by D. Clearly, if there is a tree packing of edge-disjoint trees of diameter at most D into G, then G must be (k, D)-connected. The question is whether the reverse is also true, if we allow a small congestion and a small slack in the diameter of the trees. The celebrated result of Tutte and Nash-Williams shows that, if every pair of vertices in G has k edge-disjoint paths connecting them, then there are $\lfloor k/2 \rfloor$ edge-disjoint spanning trees in G. However, this result is not length-preserving, in the sense that the tree paths may be much longer than the original paths connecting pairs of vertices. Our goal is then to provide such a length-preserving transformation from collections of short edge-disjoint paths connecting pairs of nodes in G to a low-diameter tree packing.

> *(2) Given a (k, D)-connected graph G, can one obtain a tree packing of $\widetilde{\Omega}(k)$ trees of diameter $\widetilde{O}(D)$ into G, with small edge-congestion?*

In this paper, we address both questions. For the first question, we show two efficient algorithms, that, given a k-edge connected n-vertex graph G of diameter at most D, construct a low-diameter tree packing. We complement this result by an almost matching lower bound. We address the second question by providing an efficient algorithm, that, given a (k, D)-connected graph G, computes a collection of k spanning trees of diameter at most $O(D \log n)$ each, that cause edge-congestion of $O(\log n)$.

Our Results

Our graph-theoretic results consider two main settings: in the first setting, the input graph is k-edge connected, and has diameter at most D; in the second setting, the input graph is (k, D)-connected. We only consider unweighted graphs, that is, all edge lengths are unit. Graphs are allowed to have parallel edges, unless we explicitly state that the graph is simple. Throughout the paper, we use the term *efficient algorithm* to refer to a sequential algorithm whose running time is polynomial in its input size.

Packing Trees into Low-Diameter Graphs.

We prove the following two theorems that allow us to pack low-diameter trees into low-diameter graphs.

▶ **Theorem 1.** *There is an efficient randomized algorithm, that, given any positive integers D, n, k, and an n-vertex k-edge-connected graph G of diameter at most D, computes a collection $\mathcal{T}' = \{T'_1, \ldots, T'_{\lfloor k/2 \rfloor}\}$ of $\lfloor k/2 \rfloor$ spanning trees of G, such that each edge of G appears in at most two of the trees in $\mathcal{T}'$, and, with high probability, each tree $T'_i \in \mathcal{T}'$ has diameter $O((101k \ln n)^D)$.*

As we show later, the diameter bound of Theorem 1 is close to the best possible. Unfortunately, the trees in the packing provided by Theorem 1 may share edges. Next, we generalize the classical result of Karger [14] to obtain a packing of completely edge-disjoint trees of small diameter, in the following theorem.

▶ **Theorem 2.** *There is an efficient randomized algorithm that, given an n-vertex k-edge-connected graph G of diameter at most D, such that $k > 1000 \ln n$, computes a collection $\{T_1, \ldots, T_r\}$ of $r = \Omega(k/\ln n)$ edge-disjoint spanning trees of G, such that with probability $1 - 1/\mathsf{poly}(n)$, each resulting tree T_i has diameter $O(k^{D(D+1)/2})$.*

We note that while the diameter bound in Theorem 2 is slightly weaker than that obtained in Theorem 1, and the number of the spanning trees is somewhat lower, its advantage is that the resulting trees are guaranteed to be edge-disjoint. Moreover, the algorithm in Theorem 2 is very simple: we construct r graphs $G_1, \ldots, G_r$ with $V(G_i) = V(G)$ for all i, by sampling every edge of G into one of these graphs independently. We then compute a spanning tree T_i in each such graph G_i, and show that its diameter is suitably bounded. As such, this algorithm is easy to use in the distributed setting.

Lastly, we show that our upper bounds are close to the best possible if $k \gg D$, by proving the following lower bound.

▶ **Theorem 3.** *For all positive integers n, k, D, η, α such that $k/(4D\alpha\eta)$ is an integer and $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$, there exists a k-edge connected simple graph G on n vertices of diameter at most $2D + 2$, such that, for any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η, some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.*

Note that, in particular, any collection $\mathcal{T}$ of $\Omega(k)$ trees that are either edge-disjoint, or cause a constant edge-congestion, must contain a tree of diameter $\Omega\left(\left(\frac{k}{cD}\right)^D\right)$ for some constant c. Even if we are willing to allow a polylogarithmic edge-congestion, and to settle for $\Theta(k/\mathsf{poly}\log n)$ trees, at least one of the trees must have diameter $\Omega\left(\left(\frac{k}{D\mathsf{poly}\log n}\right)^D\right)$. Moreover, we show that the lower bound from Theorem 3 continues to hold even for the weaker notion of *edge-independent* trees[3], introduced in [12].

Packing Trees into (k, D)-connected Graphs.

We next consider (k, D)-connected graphs and show an algorithm that computes a tree packing, that is near-optimal in both the number of trees and in the diameter.

[3] A collection $\mathcal{T}$ of spanning trees is edge-independent, iff all trees in $\mathcal{T}$ are rooted at the same vertex v^*, and for every vertex $v \in V(G)$, if we denote by $\mathcal{P}(v)$ the collection of paths that contains, for each tree $T \in \mathcal{T}$, the unique path connecting v to v^* in T, then all paths in $\mathcal{P}(v)$ are edge-disjoint.

▶ **Theorem 4.** *There is an efficient randomized algorithm, that, given any positive integers D, k, n with $k \leq n$, and a (k, D)-connected n-vertex graph G, computes a collection $\mathcal{T} = \{T_1, \ldots, T_k\}$ of k spanning trees of G, such that, for each $1 \leq \ell \leq k$, tree T_ℓ has diameter at most $O(D \log n)$, and with probability at least $1 - 1/\mathsf{poly}(n)$, each edge of G appears in $O(\log n)$ trees of $\mathcal{T}$.*

Improved Distributed Algorithms for Highly Connected Graphs. We present several applications of low-diameter tree packing in the standard CONGEST model of distributed computation [21]. By the proof of Theorem 2 and the $O(\log n)$-approximation algorithm for edge connectivity by [10], we obtain the following result.

▶ **Theorem 5.** *There is a randomized distributed algorithm, that, given an n-vertex graph G of constant diameter $D = O(1)$ and an integer λ, with high probability solves the problem of $O(\log n)$-approximate verification of λ-edge connectivity in G in $\mathsf{poly}(\lambda \cdot \log n)$ rounds.*

This improves upon the state of the art bound of $O(\sqrt{n})$ for graphs with constant diameter $D \geq 3$, and $\lambda \leq n^c$ for some positive constant $c < 1/(2D^2)$. From now on, we restrict our attention to k-edge connected graphs with a constant diameter $D = O(1)$. We employ the modular approach for distributed optimization introduced by Ghaffari and Haeupler in [8] which is based on the notion of *low-congestion shortcuts*. Roughly speaking, these shortcuts augment vertex-disjoint connected subgraphs by adding nearly-edge disjoint subsets of "shortcut" edges (that is, edges that reduce the diameter of each subgraph). Using our tree packing construction, we provide improved shortcuts for highly connected graphs of small diameter. This immediately leads to $o(\sqrt{n})$-round algorithms for several classical graph problems. For example, we prove the following:

▶ **Theorem 6.** *There is a randomized distributed algorithm, that, given a k-edge connected weighted n-vertex graph G of diameter D, such that the nodes know an $O(\log n)$ approximation of k, computes an MST of G in $\widetilde{O}(\min\{\sqrt{n/k}+n^{D/(2D+1)}, n/k\})$ rounds with high probability.*

If the nodes do not know an $O(\log n)$-approximation of the value of k, then such an approximation can be computed in $\mathsf{poly}(k \log n)$ rounds for $D = O(1)$ using Theorem 5, w.h.p. For general graphs (of an arbitrary connectivity) with diameter $D = 3, 4$, Kitamura et al. [15] showed nearly optimal constructions of MST's (based on shortcuts) with round complexities of $\widetilde{O}(n^{1/4})$ and $\widetilde{O}(n^{1/3})$ respectively. Turning to lower bounds, we slightly modify the construction of Lotker et al. [17] to obtain a lower bound of $\Omega((n/k)^{1/3})$ rounds for computing an MST in k-edge connected graphs of diameter 4, assuming that $k = O(n^{1/4})$.

Finally, we consider the basic task of *information dissemination*, where a given source vertex s is required to send N bits of information to the designated target vertex t in a k-edge connected n-vertex graph. This problem was first addressed in [10], who showed a lower bound of $\Omega(\min\{N/\log^2 n, n/k\})$ rounds, provided that the diameter of the graph is $\Theta(\log n)$. Using our low-diameter tree packing we obtain the first improved upper bounds for sublogarithmic diameter. We also show a new lower bound for simple store-and-forward algorithms, for the regime where $D = o(\log n)$.

▶ **Theorem 7.** *There is a randomized distributed algorithm, that, given any k-edge connected n-vertex graph G of diameter D with a source vertex s and a destination vertex t, sends an input sequence of N bits from s to t. The number of rounds is bounded by $\widetilde{O}(N^{1-1/(D+1)}+N/k)$ with high probability.*

In addition, for all integers n, N, D and $k \leq n$, there exists a k-edge connected n-vertex graph $G = (V, E)$ of diameter $2D$, and a pair s, t of its vertices, such that sending N bits from s to t in a store-and-forward manner requires at least $\Omega(\min\{(N/(D \log n))^{1-1/(D+1)}, n/k\} + N/k + D)$ rounds.

Applications to Secure Distributed Computation. Recently, Parter and Yogev [20] presented a general simulation result that converts any non-secure distributed algorithm to an equivalent secure algorithm, while paying a small overhead in the number of rounds. This transformation is based on the combinatorial graph structure of low-congestion cycle cover, namely, a collection of nearly edge-disjoint short cycles that cover all edges in the graph. The security provided by [20] was limited to adversaries who can manipulate at most one edge of the graph in a given round; in fact if the graph is only 2-edge connected, no stronger security guarantees, in terms of the number of edges that an adversary is allowed to corrupt is possible. In this paper we provide technical tools for handling stronger adversaries, who collude with $f(k)$ edges in a k-edge connected graph in each given round. In order to do so, we define a stronger variant of cycle cover that is adapted to the highly connected setting. This generalization is formalized by the notion of k-connected cycle cover, in which each edge in the graph is covered by k almost-disjoint cycles. Our key contribution is an algorithm that transforms any tree packing with k trees of diameter D into a $(k-1)$-connected cycle cover with cycle length $O(D \log n)$ and congestion $\widetilde{O}(k \log n)$. This yields a simple secure simulation of distributed algorithms in the presence of an adversary who colludes with $O(k/\log n)$ edges of the graph in each round[4]. Finally, we also use low-diameter tree packing to provide a simple store-and-forward algorithm for the problem of secure broadcast.

Organization. We provide the proof of Theorem 1 in Section 2, the proof of Theorem 2 in Section 3, the proof of Theorem 3 in Section 4, and the proof of Theorem 4 in Section 5. We discuss applications of our graph theoretic results to distributed computation in Section 6. Lastly, we discuss open problems in Section 7. Due to lack of space, some of the proofs are only sketched; the full formal proofs are deferred to the full version of the paper.

2 Low-Diameter Tree Packing with Small Edge-Congestion: Proof of Theorem 1

We start by showing that, if we are given a graph G, and a collection $\{T_1, \ldots, T_k\}$ of edge-disjoint spanning trees of G, such that the diameter of the tree T_k is at most $2D$ (but other trees may have arbitrary diameters), then we can efficiently compute another collection $\{T'_1, \ldots, T'_{k-1}\}$ of edge-disjoint spanning trees of G, such that the diameter of each resulting tree T'_i is bounded by $O((101k \ln n)^D)$ with high probability.

▶ **Theorem 8.** *There is an efficient randomized algorithm, that, given any positive integers D, k, n, an n-vertex graph G, and a collection $\{T_1, \ldots, T_k\}$ of k spanning trees of G, such that the trees $T_1, \ldots, T_{k-1}$ are edge-disjoint, and the diameter of T_k is at most $2D$, computes a collection $\{T'_1, \ldots, T'_{k-1}\}$ of edge-disjoint spanning trees of G, such that, with probability at least $1 - 1/\mathsf{poly}(n)$, for each $1 \leq i \leq k-1$, the diameter of tree T'_i is bounded by $O((101k \ln n)^D)$.*

[4] We note that an adversary may choose a different set of $O(k/\log n)$ edges to listen to or to corrupt in each round.

Theorem 1 easily follows by combining Theorem 8 with the results of Kaiser [13], who gave a short elementary proof of the tree-packing theorem of Tutte [24] and Nash-Williams [19]. His proof directly translates into an efficient algorithm, that, given a k-edge connected graph G, computes a collection of $\lfloor k/2 \rfloor$ edge-disjoint spanning trees of G. In order to complete the proof of Theorem 1, we use the algorithm of Kaiser [13] to compute an arbitrary collection $\mathcal{T} = \{T_1, \ldots, T_{\lfloor k/2 \rfloor}\}$ of edge-disjoint spanning trees of G, and compute another arbitrary BFS tree T^* of G. Since the diameter of G is at most D, the diameter of T^* is at most $2D$. We then apply Theorem 8 to the collection $\{T_1, \ldots, T_{\lfloor k/2 \rfloor}, T^*\}$ of spanning trees, to obtain another collection $\mathcal{T}' = \{T'_1, \ldots, T'_{\lfloor k/2 \rfloor}\}$ of spanning trees, such that each edge of G belongs to at most 2 trees of $\mathcal{T}'$, and with high probability, the diameter of each tree in $\mathcal{T}'$ is at most $O((101k \ln n)^D)$. We note that, since we allow parallel edges, the trees in the set $\{T_1, \ldots, T_{\lfloor k/2 \rfloor}, T^*\}$ are edge-disjoint in graph $G \cup E(T^*)$.

The main technical tool that we use in order to prove of Theorem 8 is the following theorem, that allows one to "fix" a diameter of a connected graph using a low-diameter tree.

▶ **Theorem 9.** *Let H be a connected graph with $|V(H)| \le n$, and let T be a rooted tree of depth D, such that $V(T) = V(H)$. For a real number $0 < p < 1$, let R be a random subset of the edges of T, where each edge $e \in E(T)$ is added to R independently with probability p. Then with probability at least $1 - \frac{D}{n^{48}}$, the diameter of the graph $H \cup R$ is at most $(\frac{101 \ln n}{p})^D$.*

Theorem 8 easily follows from Theorem 9: For each $1 \le i < k$, we construct a graph G_i as follows. Start with $G_i = T_i$ for all $1 \le i \le k$. Compute a random partition $E_1, \ldots, E_{k-1}$ of the edges of $E(T_k)$, by adding each edge $e \in E(T_k)$ to a set E_i chosen uniformly at random from $\{E_1, \ldots, E_{k-1}\}$ independently from other edges. Using Theorem 9 with $p = 1/(k-1)$, it is immediate to see that with high probability, the diameter of each resulting graph G_i is bounded by $O((101k \ln n)^D)$. We then let T'_i be a BFS tree of graph G_i, rooted at an arbitrary vertex. In order to complete the proof of Theorem 1, it is now enough to prove Theorem 9.

Proof of Theorem 9. Recall that we are given a connected graph H with $|V(H)| \le n$, and a rooted tree T of depth D, such that $V(T) = V(H)$, together with a parameter $0 < p < 1$. We let R be a random subset of $E(T)$, where each edge $e \in E(T)$ is added to R independently with probability p. Our goal is to show that the diameter of the graph $H \cup R$ is at most $\left(\frac{101 \ln n}{p}\right)^D$ with probability at least $1 - \frac{D}{n^{48}}$. Denote $V = V(H) = V(T)$. For each $0 \le i \le D$, let V_i be the set of nodes lying at level i of the tree T (that is, at distance i from the tree root), and denote $V_{\le i} = \bigcup_{t=0}^{i} V_t$. Let $H' = H \cup R$.

We say that a node $x \in V$ is *good* if either (i) $x \in V_{\le D-1}$; or (ii) $x \in V_D$, and there is an edge in R connecting x to a node in V_{D-1}. We assume that $V = \{v_1, \ldots, v_{n'}\}$, where the vertices are indexed in an arbitrary order. Given an ordered pair (x, x') of vertices in H, and a path P connecting x to x', let $\sigma(P)$ be a sequence of vertices that lists all the vertices appearing on P in their natural order, starting from vertex x (so in a sense, we think of P as a directed path). For an ordered pair $(x, x') \in V$ of vertices, let $P_{x,x'}$ be shortest path connecting x to x' in H, and among all such paths P, choose the one whose sequence $\sigma(P)$ is smallest lexicographically. Observe that $P_{x,x'}$ is unique, and, moreover, if some pair u, u' of vertices lie on $P_{x,x'}$, with u lying closer to x than u' on $P_{x,x'}$, then the sub-path of $P_{x,x'}$ from u to u' is precisely $P_{u,u'}$.

Let $M = \frac{50 \ln n}{p}$. For a pair x, x' of vertices of V, we let $B(x, x')$ be the bad event that length of $P_{x,x'}$ is greater than M and there is no good internal node on $P_{x,x'}$. Notice that event $B(x,x')$ may only happen if every inner vertex on $P_{x,x'}$ lies in V_D, and for each such vertex, the unique edge of T that is incident to it was not added to R. Therefore, the probability that event $B(x,x')$ happens for a fixed pair x,x' of vertices is at most $(1-p)^M = (1-p)^{(50\ln n)/p} \le n^{-50}$. Let B be the bad event that $B(x,x')$ happens for some pair $x,x' \in V$ of nodes. From the union bound over all pairs of nodes in V, the probability of B is bounded by n^{-48}.

Recall that H is a subgraph of H' and $\mathrm{dist}_H(\cdot,\cdot)$ is the shortest-path distance metric on H. We use the following immediate observation.

▷ Observation 10. If the event B does not happen, then for every node $x \in V$, there is a good node $x' \in V$ such that $\mathrm{dist}_H(x,x') \le M$.

We prove Theorem 9 by induction on D. The base of the induction is when $D=1$. In this case, T is a star graph. Let c denote the vertex that serves as the center of the star. For any pair $x_1,x_2 \in V$ of vertices, we denote by x'_1 the good node that is closest to x_1 in H, and we define x'_2 similarly for x_2. Notice that, from the definition of good vertices, either $x'_1 = c$, or it is connected to c by an edge of R, and the same holds for x'_2. Therefore, $\mathrm{dist}_{H'}(x'_1,x'_2) \le 2$ must hold. If the event B does not happen, then, since H is a subgraph of H', $\mathrm{dist}_{H'}(x_1,x_2) \le \mathrm{dist}_{H'}(x_1,x'_1) + \mathrm{dist}_{H'}(x'_1,x'_2) + \mathrm{dist}_{H'}(x_2,x'_2) \le \mathrm{dist}_H(x_1,x'_1) + \mathrm{dist}_{H'}(x'_1,x'_2) + \mathrm{dist}_H(x_2,x'_2) \le 2M+2 \le \frac{101\ln n}{p}$. Therefore, with probability at least $1-n^{-48}$, $\mathrm{dist}_{H'}(x_1,x_2) \le \frac{101\ln n}{p}$.

Assume now that Theorem 9 holds for every connected graph H and every tree T of depth at most $D-1$, with $V(T)=V(H)$. Consider now some connected graph H, and a rooted tree T of depth D, with $V(T)=V(H)$. We partition the edges of $E(T)$ into two subsets: set E_1 contains all edges incident to the vertices of V_D, and set E_2 contains all remaining edges. Let $E'_1 = E_1 \cap R$, and let $E'_2 = E_2 \cap R$. Notice that the definition of good vertices only depends on the edges of E'_1, and so the event B only depends on the random choices made in selecting the edges of E'_1, and is independent from the random choices made in selecting the edges of E'_2.

Let L be a subgraph of H', obtained by starting with $L=H$, and then adding all edges of E'_1 to the graph. Finally, we define a new graph $\hat{H}$, whose vertex set is $V_{\le D-1}$, and there is an edge between a pair of nodes w,w' in $\hat{H}$ iff the distance between w and w' in L is at most $M+2$. We also let $\hat{T}$ be the tree obtained from T, by discarding from it all vertices of V_D and all edges incident to vertices of V_D. Observe that $V(\hat{H}) = V(\hat{T}) = V_{\le D-1}$. The idea is to use the induction hypothesis on the graph $\hat{H}$, together with the tree $\hat{T}$. In order to do so, we need to prove that $\hat{H}$ is a connected graph, which we do next.

▷ Observation 11. If the event B does not happen, then graph $\hat{H}$ is connected.

Proof. Assume that the event B does not happen, and assume for contradiction that graph $\hat{H}$ is not connected. Let $\mathcal{C} = \{C_1,\ldots,C_r\}$ be the set of all connected components of graph $\hat{H}$. For every pair C_i,C_j of distinct components of $\mathcal{C}$, consider the set $\mathcal{P}_{i,j} = \{P_{x,x'} \mid x \in V(C_i), x' \in V(C_j)\}$ of paths (recall that $P_{x,x'}$ is the shortest path connecting x to x' in H with $\sigma(P_{x,x'})$ lexicographically smallest among all such paths). We let $P_{i,j}$ be a shortest path in $\mathcal{P}_{i,j}$. Choose two distinct components $C_i,C_j \in \mathcal{C}$, whose path $P_{i,j}$ has the shortest length, breaking ties arbitrarily. Assume that $P_{i,j}$ connects a vertex $v \in C_i$ to a vertex $u \in C_j$, so $P_{i,j} = P_{v,u}$. Recall that $H \subseteq L$, and so the path $P_{i,j}$ is contained in graph L. Since we did not add edge (u,v) to $\hat{H}$, the length of $P_{i,j}$ is greater than $M+2$. Since we have assumed that event B does not happen, there is at least one good inner vertex on path $P_{i,j}$. Let X be the set of all good vertices that serve as inner vertices of $P_{i,j}$.

We first show that for each $x \in X$, $x \notin V(\hat{H})$ must hold. Indeed, assume for contradiction that $x \in V(\hat{H})$, so x belongs to some connected component of $V(\hat{H})$. Assume first that $x \in V(C_i)$. Recall that the sub-path of $P_{i,j}$ from x to u is precisely $P_{x,u}$, so this path lies in $\mathcal{P}_{i,j}$. But its length is less than the length of $P_{i,j}$, contradicting the choice of $P_{i,j}$. Otherwise, x belongs to some connected component C_ℓ of $\mathcal{C}$ with $\ell \neq i$. The sub-path of $P_{i,j}$ from v to x is precisely $P_{v,x}$, so this path must lie in $\mathcal{P}_{i,\ell}$. Since its length is less than the length of $P_{i,j}$, this contradicts the choice of the components C_i, C_j. We conclude that $x \notin V(\hat{H})$.

Since $V(\hat{H})$ contains all vertices of $V_{\leq D-1}$, and every vertex in X is a good vertex, it must be the case that $X \subseteq V_D$. Consider again some vertex $x \in X$. Since x is a good vertex and $x \in V_D$, there must be an edge $e_x = (x, x') \in E'_1$, connecting x to some vertex $x' \in V_{\leq D-1}$. In particular, x' must belong to some connected component of $\mathcal{C}$, and the edge e_x lies in graph L. Assume that $X = \{x_1, x_2, \ldots, x_q\}$, where the vertices are indexed in the order of their appearance on $P_{i,j}$, from v to u. Consider the sequence $\tilde{\sigma} = (v, x'_1, x'_2, \ldots, x'_q, u)$ of vertices. All these vertices belong to $V(\hat{H})$, and $v \in C_i$, while $u \in C_j$. For convenience, denote $v = x'_0 = x_0$ and $u = x'_{q+1} = x_{q+1}$. Then there must be an index $1 \leq a \leq q$, such that x'_a and x'_{a+1} belong to distinct connected components of $\mathcal{C}$. Note that the sub-path of $P_{i,j}$ between x_a and x_{a+1} is precisely $P_{x_a, x_{a+1}}$ – the shortest path connecting x_a to x_{a+1} in H. Since no good vertices lie between x_a and x_{a+1} on this path, and since we have assumed that event B does not happen, the length of this path is at most M. Therefore, there is a path in graph L, connecting x'_a to x'_{a+1}, whose length is at most $M + 2$. This path connects a pair of vertices that belong to different connected components of $\hat{H}$, contradicting the construction of $\hat{H}$. ◀

Consider now the tree $\hat{T}$ and the graph $\hat{H}$. Recall that $\hat{T}$ is a rooted tree of depth $D - 1$, $V(\hat{T}) = V(\hat{H})$, $|V(\hat{H})| \leq |V(H)| \leq n$, and, assuming the event B did not happen, $\hat{H}$ is a connected graph. Moreover, set E'_2 of edges is a subset of $E(\hat{T}) = E_2$, obtained by adding every edge of $E(\hat{T})$ to E'_2 with probability p, independently from other edges. Therefore, assuming that event B did not happen, we can use the induction hypothesis on the graph $\hat{H}$, the tree $\hat{T}$, and the set E'_2 of edges as R. Let B' be the bad event that the diameter of $\hat{H} \cup E'_2$ is greater than $(\frac{101 \ln n}{p})^{D-1}$. Note that the event B' only depends on the random choices made in selecting the edges of E'_2. From the induction hypothesis, the probability that B' happens is at most $\frac{D-1}{n^{48}}$.

Lastly, we show that, if neither of the events B, B' happens, then $\mathrm{diam}(H') \leq (\frac{101 \ln n}{p})^D$.

▷ Observation 12. If neither of the events B, B' happens, then $\mathrm{diam}(H') \leq (\frac{101 \ln n}{p})^D$.

Proof. Consider any pair $x_1, x_2 \in V$ of vertices. It is sufficient to show that, if events B, B' do not happen, then $\mathrm{dist}_{H'}(x_1, x_2) \leq (\frac{101 \ln n}{p})^D$.

Let x'_1 be a good node in $V(H)$ that is closest to x_1, and define x'_2 similarly for x_2. From Observation 10, $\mathrm{dist}_H(x_1, x'_1) \leq M$. If $x'_1 \in V_{\leq D-1}$, then we define $x''_1 = x'_1$, otherwise we let x''_1 be the node of V_{D-1} that is connected to x'_1 by an edge of E'_1, and we define x''_2 similarly for x_2. Therefore, $x''_1, x''_2 \in V_{\leq D-1} = V(\hat{H})$, and, assuming event B does not happen, $\mathrm{dist}_{H'}(x_1, x''_1) \leq M + 1$, and $\mathrm{dist}_{H'}(x_2, x''_2) \leq M + 1$. Since we have assumed that the bad event B' does not happen, $\mathrm{dist}_{\hat{H} \cup E'_2}(x''_1, x''_2) \leq (\frac{101 \ln n}{p})^{D-1}$. Recall that for every edge $e = (u, v) \in \hat{H} \cup E'_2$, if $e \in E'_2$ then $e \in E(H')$; otherwise, $e \in E(\hat{H})$, and there is a path in graph $H \cup E_1'$ of length at most $M+2$ connecting u to v in H. Therefore, $\mathrm{dist}_{H'}(x_1'', x_2'') \le (M+2) \cdot \mathrm{dist}_{\hat{H}}(x_1'', x_2'') \le (\frac{101 \ln n}{p})^{D-1} \cdot (M+2)$.

Altogether, since $M = (50 \ln n)/p$,

$$\begin{aligned}\mathrm{dist}_{H'}(x_1, x_2) &\le \mathrm{dist}_{H'}(x_1, x_1'') + \mathrm{dist}_{H'}(x_1'', x_2'') + \mathrm{dist}_{H'}(x_2, x_2'') \\ &\le \left(\frac{101 \ln n}{p}\right)^{D-1} \cdot (M+2) + (2M+2) \\ &\le \left(\frac{101 \ln n}{p}\right)^{D}.\end{aligned}$$

◀

The probability that either B or B' happen is bounded by $\frac{D}{n^{48}}$. Therefore, with probability at least $1 - \frac{D}{n^{48}}$, neither of the events happens, and $\mathrm{diam}(H') \le (\frac{101 \ln n}{p})^D$. ◀

3 Low-Diameter Packing of Edge-Disjoint Trees: Proof of Theorem 2

The main tool in the proof of Theorem 2 is the following theorem.

▶ **Theorem 13.** *Let k, D, n be any positive integers with $k > 1000 \ln n$, let $\frac{707 \ln n}{k} \le p \le 1$ be a real number, and let G be an n-vertex k-edge-connected graph of diameter D. Let G' be a sub-graph of G with $V(G') = V(G)$, where every edge $e \in E(G)$ is added to G' with probability p independently from other edges. Then, with probability at least $1 - 1/\mathsf{poly}(n)$, G' is a connected graph, and its diameter is bounded by $k^{D(D+1)/2}$.*

Karger [14] has shown that, if G is a k-connected graph, and G' is obtained by sub-sampling the edges of G with probability $\Omega(\log n/k)$, then G' is a connected graph with high probability. Theorem 13 further shows that the diameter of G' is with high probability bounded by $k^{D(D+1)/2}$, where D is the diameter of G.

Theorem 2 easily follows from Theorem 13: Let $r = \lfloor k/(707 \ln n) \rfloor$. We partition $E(G)$ into subsets $E_1, \ldots, E_r$ by choosing, for each edge $e \in E(G)$, an index i independently and uniformly at random from $\{1, 2, \ldots, r\}$ and then adding e to E_i. For each $1 \le i \le r$, we define a graph G_i by setting $V(G_i) = V(G)$ and $E(G_i) = E_i$. Finally, for each graph G_i, we compute an arbitrary BFS tree T_i, and return the resulting collection $\mathcal{T} = \{T_1, \ldots, T_r\}$ of trees. It is immediate to verify that the graphs $G_1, \ldots, G_r$ are edge-disjoint, and so are the trees of $\mathcal{T}$. Moreover, applying Theorem 13 to each graph G_i with $p = 1/r$, we get that with probability $1 - 1/\mathsf{poly}(n)$, $\mathrm{diam}(T_i) \le 2\,\mathrm{diam}(G_i) \le O(k^{D(D+1)/2})$. Using the union bound over all $1 \le i \le r$ completes the proof of Theorem 2. It now remains to prove Theorem 13. We provide a proof sketch here; a formal proof appears in the full version of the paper.

Proof Sketch of Theorem 13: We use the well known result of Karger [14], that shows that the probability that the graph G' is not connected is at most $O(1/\mathsf{poly}(n))$. It remains to bound the diameter of G'. Throughout the proof, for a graph H, we denote by $\mathcal{D}(H, p)$ be the distribution of graphs, where the vertex set of the resulting graph is $V(H)$, and each edge of H is included in the graph with probability p independently from other edges.

Denote $G = (V, E)$, and let T be a BFS tree of G, rooted at an arbitrary node r of G. Since G has diameter at most D, the depth of T is at most D. Recall that $G' \sim \mathcal{D}(G, p)$. We define a different (but equivalent) sampling algorithm for generating a random graph G' from $\mathcal{D}(G,p)$ as follows. The algorithm consists of $D+1$ phases. In the 0th phase, we sample all edges in $E \setminus E(T)$ independently with probability p each. For each $1 \le i \le D$, in the ith phase, we sample all edges that connect a vertex at distance $(D-i+1)$ from r to a vertex at distance $(D-i)$ from r in T. Let E' be the set of all sampled edges at the end of this algorithm. We denote by $G' = (V, E')$ the final graph that we obtain. Clearly, G' is generated from the distribution $\mathcal{D}(G,p)$. We denote by T' the subgraph of T with $V(T') = V(T)$ and $E(T') = E(T) \cap E'$. Clearly, $T' \sim \mathcal{D}(T,p)$.

Consider a pair $u, u' \in V$ of distinct vertices. We say that they are *joined at phase i* for $0 \le i \le D$, if u and u' belong to the same connected component of the graph induced by all edges sampled in the first i phases, but they lie in different connected components of the graph induced by all edges sampled in the first $(i-1)$ phases. Note that, if G' is connected, then every pair (u,u') of distinct vertices of V are joined at phase i for some $0 \le i \le D$. The following lemma allows us to bound the diameter of G'.

▶ **Lemma 14.** *For each $0 \le i \le D$, with probability $1 - O(1/\mathsf{poly}(n))$, for every pair x, y of vertices that are joined at phase i, x and y are at distance at most $7^i(101 \ln n/p)^{D+(D-1)+\cdots+(D-i)}$ in G'.*

Observe that, by applying the union bound over all $0 \le i \le D$, Lemma 14 implies Theorem 13, since $k \ge 707 \ln n/p$. We defer the proof of Lemma 14 to the full version of the paper, and only provide its proof sketch here. Assume for simplicity that the edges in $E \setminus E(T)$ only connect vertices that are at distance D from r in T (this also turns out to be the hardest case). The proof is by induction on i. In the base case where $i = 0$, let C be a connected component of the graph induced by all edges sampled in phase 0. Intuitively, we can view the algorithm as using a random subgraph of T to "fix" the diameter of C, like in Theorem 9. Therefore, with high probability, for every pair x, y of vertices of C, the distance from x to y in $C \cup T'$ is at most $(101 \ln n/p)^D$. Similarly, let C' be a connected component of the the subgraph of G induced by all edges sampled in phases $0, 1, \ldots, i$; we call C' a *phase-i cluster*. We view C' as consisting of a number of phase-$(i-1)$ clusters $C''_1, \ldots, C''_k$, connected to each other by edges that were sampled in the ith phase. Therefore, if $\hat{C}'$ is a graph obtained from C' by contracting each cluster $C''_1, \ldots, C''_k$ into a single vertex, then $\hat{C}'$ is a connected graph. Denote by T_i the subtree of T induced by all nodes that are at distance at most $(D-i)$ from r in T, and denote $T'_i = T' \cap T_i$. Clearly $T'_i \sim \mathcal{D}(T_i, p)$. We can again view our algorithm as using a random subgraph T'_i of T_i to "fix" the diameter of $\hat{C}'$, like in Theorem 9. Therefore, with high probability, for every pair x, y of vertices of $\hat{C}'$, the distance from x to y in $\hat{C}' \cup T'_i$ is at most $(101 \ln n/p)^{D-i}$. Note however that every vertex of $\hat{C}'$ is in fact a contracted level-$(i-1)$ cluster. Moreover, from the induction hypothesis, if C'' is a level-$(i-1)$ cluster, and x', y' is a pair of vertices in C'', then with high probability, the distance from x' to y' in $C'' \cup T'$ is at most $7^{i-1} \cdot (101 \ln n/p)^{D+(D-1)+\cdots+(D-i+1)}$. Therefore, with high probability, the distance between a pair $u, v \in V(C')$ of vertices in $C' \cup T'$ is at most $7^i \cdot (101 \ln n/p)^{D+(D-1)+\cdots+(D-i)}$.

4 Lower Bound: Proof of Theorem 3

In this section we provide the proof of Theorem 3. We start by proving the following slightly weaker theorem; we then extend it to obtain the proof of Theorem 3.

▶ **Theorem 15.** *For all positive integers k, D, η, α such that $k/(4D\alpha\eta)$ is an integer, there* *exists a k-edge connected graph G with $|V(G)| = O\left(\left(\frac{k}{2D\alpha\eta}\right)^D\right)$ and diameter at most $2D$, such that, for any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η, some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.*

Notice that the main difference from Theorem 3 is that the graph G is no longer required to be simple; the number of vertices of $V(G)$ is no longer fixed to be a prescribed value; and the diameter of G is $2D$ instead of $2D+2$.

Proof. For a pair of integers $w > 1, D \geq 1$, we let $T_{w,D}$ be a tree of depth D, such that every vertex lying at levels $0, \ldots, D-1$ of $T_{w,D}$ has exactly w children. In other words, $T_{w,D}$ is the full w-ary tree of depth D. We denote $N_{w,D} = |V(T_{w,D})| = 1 + w + w^2 + \cdots + w^D \leq w^{D+1}/(w-1)$. We assume that for every inner vertex $v \in V(T_{w,D})$, we have fixed an arbitrary ordering of the children of v, denoted by $a_1(v), \ldots, a_w(v)$.

A *traversal* of a tree T is an ordering of the vertices of T. A *post-order traversal* on a tree T, $\pi(T)$, is defined as follows. If the tree consists of a single node v, then $\pi(T) = (v)$. Otherwise, let r be the root of the tree and consider the sequence $(a_1(r), \ldots, a_w(r))$ of its children. For each $1 \leq i \leq w$, let T_i be the sub-tree of T rooted at the vertex $a_i(r)$. We then let $\pi(T)$ be the concatenation of $\pi(T_1), \pi(T_2), \ldots, \pi(T_w)$, with the vertex r appearing at the end of the sequence; see Figure 1 for an illustration. For simplicity, we assume that $V(T_{w,D}) = \{v_1, v_2, \ldots, v_{N_{w,D}}\}$, where the vertices are indexed in the order of their appearance in $\pi(T_{w,D})$, so the traversal visits these vertices in this order.

Next, we define a graph $G_{w,D}$, as follows. The vertex set of $G_{w,D}$ is the same as the vertex set of $T_{w,D}$, namely $V(G_{w,D}) = V(T_{w,D})$. The edge set of $G_{w,D}$ consists of two subsets: $E_1 = E(T_{w,D})$, and another set E_2 of edges that contains, for each $1 \leq i < N_{w,D}$, k parallel copies of the edge (v_i, v_{i+1}). We then set $E(G_{w,D}) = E_1 \cup E_2$. For convenience, we call the edges of E_1 *blue edges*, and the edges of E_2 *red edges*; see Figures 1 and 2.

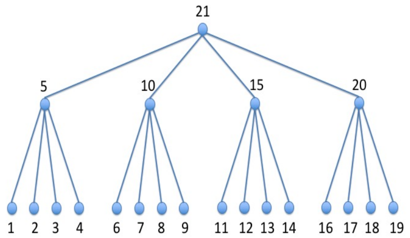

Figure 1 Tree $T_{4,2}$ with vertices indexed according to post-order traversal.

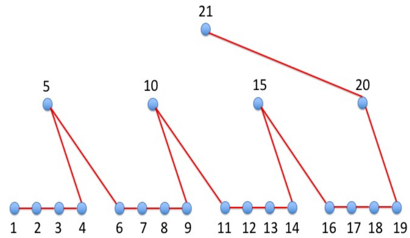

Figure 2 The edge set E_2 in $G_{4,2}$ (only a single copy of each edge is shown).

It is easy to verify that graph $G_{w,D}$ must be k-edge connected, since for any partition of $V(G_{w,D})$, there is some index $1 \leq i < N_{w,D}$ with v_i, v_{i+1} separated by the partition, and so k parallel edges connecting v_i to v_{i+1} must cross the partition.

We now fix an integer $w = k/(2D\alpha\eta)$ (note that $w \geq 2$), and we let $T = T_{w,D}$ be the corresponding tree and $G = G_{w,D}$ the corresponding graph. For convenience, we denote $N_{w,D}$ by N. Recall that $N \leq w^{D+1}/(w-1) = O\left(\left(\frac{k}{2D\alpha\eta}\right)^D\right)$. As observed before, G is

k-edge connected. Since the depth of T is D, and $T \subseteq G$, it is easy to see that the diameter of G is at most $2D$.

We now consider any collection $\mathcal{T} = \{T_1, \ldots, T_{k/\alpha}\}$ of k/α spanning trees of G that causes edge-congestion at most η. Our goal is to show that some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.

For convenience, we denote $V(G) = V(T) = V$. We say that a vertex $x \in V$ is an *ancestor* of a vertex $y \in V$ if x is an ancestor of y in the tree T, that is, $x \neq y$, and x lies on the unique path connecting y to the root of T.

Let $L \subseteq V$ be the set of vertices that serve as leaves of the tree T. We denote by $u = v_1$ a vertex of L that has the lowest index, and by u' the vertex of L with the largest index. It is easy to see that $u' = v_{N-D}$, as every vertex whose index is greater than that of u' is an ancestor of u'. For each $1 \leq j \leq k/\alpha$, we denote by P_j the unique path that connects u to u' in tree T_j. Let $\mathcal{P} = \{P_j \mid 1 \leq j \leq k/\alpha\}$. It is enough to show that at least one of the paths P_j has length at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. In order to do so, we show that $\sum_{j=1}^{k/\alpha} |E(P_j)|$ is sufficiently large. At a high level, we consider the red edges (v_i, v_{i+1}) (the edges of E_2), and show that many of the paths in $\mathcal{P}$ must contain copies of each such edge. This in turn will imply that $\sum_{P_j \in \mathcal{P}} |E(P_j)|$ is large, and that some path in $\mathcal{P}$ is long enough.

For each vertex $v_i \in L$ such that $v_i \neq u'$, we let $S_i = \{v_1, \ldots, v_i\}$, and we let $\overline{S}_i = \{v_{i+1}, \ldots, v_N\}$. Notice that, since $u \in S_i$ and $u' \in \overline{S}_i$, every path in $\mathcal{P}$ must contain an edge of $E_G(S_i, \overline{S}_i)$. Note that the only red edges in $E_G(S_i, \overline{S}_i)$ are the k parallel copies of the edge (v_i, v_{i+1}). In the next observation, we show that the number of blue edges in $E_G(S_i, \overline{S}_i)$ is bounded by Dw.

▷ Observation 16. For each vertex $v_i \in L$ such that $v_i \neq u'$, for every blue edge $e \in E_G(S_i, \overline{S}_i)$, at least one endpoint of e must be an ancestor of v_i.

Proof. We consider a natural layout of the tree T, where for every inner vertex x of the tree, its children $a_1(x), \ldots, a_w(x)$ are drawn in this left-to-right order (see Figure 3). Consider the path Q connecting the root of T to v_i, so every vertex on Q (except for v_i) is an ancestor of v_i. All vertices lying to the left of Q in the layout are visited before v_i by $\pi(T)$. All vertices lying to the right of Q, and on Q itself (excluding v_i) are visited after v_i. It is easy to see that the vertices of Q separate the two sets in T, and so the only blue edges connecting S_i to $\overline{S}_i$ are edges incident to the vertices of $V(Q) \setminus \{v_i\}$. ◀

Since every vertex of the tree T has at most w children, and since the depth of the tree is D, we obtain the following corollary of Observation 16.

▶ **Corollary 17.** *For each vertex $v_i \in L$ such that $v_i \neq u'$, at most Dw blue edges lie in $E_G(S_i, \overline{S}_i)$.*

Since the trees in $\mathcal{T}$ cause edge-congestion η, at most $Dw\eta$ trees of $\mathcal{T}$ may contain blue edges in $E_G(S_i, \overline{S}_i)$. Each of the remaining $\frac{k}{\alpha} - Dw\eta \geq \frac{k}{2\alpha}$ trees contains a copy of the red edge (e_i, e_{i+1}) (recall that $w = k/(2D\alpha\eta)$.) Therefore, $\sum_{P_j \in \mathcal{P}} |E(P_j)| \geq |L| \cdot \frac{k}{2\alpha} \geq \frac{Nk}{4\alpha}$, since $|L| \geq |N|/2$. We conclude that at least one path $P_j \in \mathcal{P}$ must have length at least $\frac{Nk}{4\alpha} / \frac{k}{\alpha} \geq \frac{N}{4}$, and so the diameter of T_j is at least $\frac{N}{4}$. Since $N \geq w^D \geq \left(\frac{k}{2D\alpha\eta}\right)^D$, the diameter of T_j is at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. ◀

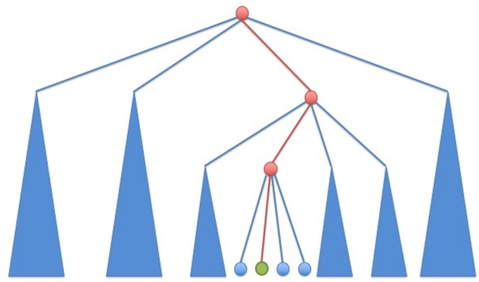

Figure 3 A layout of the tree T. Vertex v_i is shown in green and path Q in red. All vertices lying to the left of Q in this layout appear before v_i in $\pi(T)$, and all vertices lying to the right of Q or on Q (except for v_i) appear after v_i in $\pi(T)$.

We are now ready to complete the proof of Theorem 3. First, we show that we can turn the graph G into a simple graph, and ensure that $|V(G)| = n$, if $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$. Let $G'_{w,D}$ be the graph obtained from $G_{w,D}$ as follows. For each $1 \leq i \leq N$, we replace the vertex v_i with a set $X_i = \{x_i^1, x_i^2 \dots, x_i^k\}$ of k vertices that form a clique. For each $1 \leq i < N$, the k red edges connecting v_i to v_{i+1} are replaced by the perfect matching $\{(x_i^t, x_{i+1}^t)\}_{1 \leq t \leq k}$ between vertices of X_i and vertices of X_{i+1}. Each blue edge (v_i, v_j) is replaced by a new edge (x_i^1, x_j^1). Since $n \geq 3k \cdot \left(\frac{k}{2D\alpha\eta}\right)^D > k|V(G)| + k$, we add $n - k|V(G)| > k$ new vertices that form a clique, and for each newly-added vertex, we add an edge connecting it to x_N^1 (recall that the vertex v_N is the root of T). We denote $G' = G'_{w,D}$ for simplicity. It is not hard to see that G' has n vertices and it is k-edge connected. Moreover, G' has diameter at most $2D + 2$, since its subgraph induced by vertices of $\{x_i^1\}_{1 \leq i \leq N}$ has diameter $2D$, and every other vertex of G' is a neighbor of one of the vertices in $\{x_i^1\}_{1 \leq i \leq N}$. The tree T' is defined exactly as before, except that every original vertex v_j is now replaced with its copy x_j^1. Let L denote the set of all leaf vertices in T'.

Assume that we are given a collection $\mathcal{T} = \{T_1, \dots, T_{k/\alpha}\}$ of k/α spanning trees of G' that causes edge-congestion at most η. For each $1 \leq i \leq k/\alpha$, we denote by Q_i the unique path that connects x_1^1 to x_{N-D}^1 in T_i and denote $\mathcal{Q} = \{Q_i \mid 1 \leq i \leq k/\alpha\}$. For each every leaf vertex $x_j^1 \in L$, we define a cut $(W_j, \overline{W}_j)$ as follows: $W_j = \bigcup_{1 \leq s \leq j} X_s$ and $\overline{W_j} = V(G') \setminus W_j$. Using reasoning similar to that in Corollary 17, it is easy to see that for every leaf vertex $x_j^1 \in L$, the set $E_{G'}(W_j, \overline{W_j})$ of edges contains at most Dw blue edges – the edges of the tree T'. Since the trees in $\mathcal{T}$ cause edge-congestion at most η, at most $Dw\eta$ trees of $\mathcal{T}$ may contain blue edges in $E_{G'}(W_j, \overline{W_j})$. Therefore, for each of the remaining $\frac{k}{\alpha} - Dw\eta \geq \frac{k}{2\alpha}$ trees T_i, path Q_i must contain a red edge from $\{(x_j^t, x_{j+1}^t)\}_{1 \leq t \leq k}$. Therefore, the sum of lengths of all paths of $\mathcal{Q}$ is at least $\frac{Nk}{4\alpha}$, and so at least one path $Q_i \in \mathcal{Q}$ must have length at least $\frac{N}{4}$. We conclude that some tree $T_i \in \mathcal{T}$ has diameter at least $\frac{1}{4} \cdot \left(\frac{k}{2D\alpha\eta}\right)^D$.

Lastly, we extend our results to edge-independent trees. We use the same simple graph G' and the same tree T' as before, setting the congestion parameter $\eta = 2$. Assume that we are given a collection $\mathcal{T}' = \{T'_1, \dots, T'_{k/\alpha}\}$ of k/α edge-independent spanning trees of G' and let $x \in V(G')$ be their common root vertex. For each $1 \leq i \leq k/\alpha$, we denote by Q'_i the unique path that connects vertex x_1^1 to vertex x_{N-D}^1 in tree T'_i, and we denote $\mathcal{Q}' = \{Q'_i \mid 1 \leq i \leq k/\alpha\}$. Note that, for each $1 \leq i \leq k/\alpha$, the path Q'_i is a sub-path of the path obtained by concatenating the path Q''_i, connecting x_1^1 to x in T'_i, with the path

Q_i''', connecting x^1_{N-D} to x in T_i'. Since the trees in $\mathcal{T}'$ are edge-independent, the paths in $\{Q_i''\}_{1\le i\le k/\alpha}$ are edge-disjoint and so are the paths in $\{Q_i'''\}_{1\le i\le k/\alpha}$. Therefore, the paths of $\mathcal{Q}'$ cause edge-congestion at most 2. The remainder of the proof is the same as before and is omitted here.

5 Tree Packing for (k, D)-Connected Graphs: Proof of Theorem 4

In this section we provide a proof sketch of Theorem 4. The full proof is deferred to the full version of the paper. The main tool that we use is the following theorem, whose proof appears in the full version of the paper.

▶ **Theorem 18.** *There is an efficient algorithm, that, given a (k, D)-connected graph G and a subset $S \subseteq V(G)$ of its vertices, computes a bi-partition (S', S'') of S, and a flow f from vertices of S'' to vertices of S', such that the following hold:*

1. *every vertex of S'' sends at least $k/2$ flow units;*
2. *every flow-path has length at most $2D$;*
3. *the total amount of flow through any edge is at most 3; and*
4. $|S'| \le \frac{|S|}{2} + 1$.

Our algorithm consists of two phases. In the first phase, we define a partition of the vertices of G into layers $L_1, \ldots, L_h$, where $h = O(\log n)$. Additionally, for each $1 \le i \le h$, we define a flow f_i in graph G between vertices of L_i and vertices of $L_1 \cup \cdots \cup L_{i-1}$. In the second phase, we use the layers and the flows in order to construct the desired set of spanning trees.

Phase 1: Partitioning into layers. We use a parameter $h = \Theta(\log n)$, whose exact value will be set later. We now define the layers $L_h, \ldots, L_1$ in this order, and the corresponding flows $f_h, \ldots, f_1$. In order to define the layer L_h, we let $S = V(G)$, and we apply Theorem 18 to the graph G and the set S of its vertices, to obtain a partition (S', S'') of S, with $|S'| \le |S|/2 + 1$, and the flow f between the vertices of S'' and the vertices of S', where every vertex of S'' sends at least $k/2$ units of flow, each flow-path has length at most $2D$, and the edge-congestion caused by f is at most 3. We then set $L_h = S''$ and $f_h = f$, and continue to the next iteration.

Assume now that we have constructed layers $L_h, \ldots, L_i$. We now show how to construct layer L_{i-1}. Let $S = V(G) \setminus (L_h \cup \cdots \cup L_i)$. We apply Theorem 18 to the graph G and the set S of its vertices, to obtain a partition (S', S'') of S, with $|S'| \le |S|/2 + 1$, and the corresponding flow f. We then set $L_{i-1} = S''$, $f_{i-1} = f$, and continue to the next iteration. If we reach an iteration where $|S| \le 2$, we arbitrarily designate one of the two vertices as s and the other as s', and compute a flow of value at least k between the two vertices, such that the edge-congestion of the flow is at most 2, and every flow-path has length at most $2D$. We add vertex s' to the current layer, and we add vertex s to the final layer L_1. If we reach an iteration where $|S| = 1$, then we add the vertex of S to the final layer L_1 and terminate the algorithm. The number h of layers is chosen to be exactly the number of iterations in this algorithm. Notice that $h \le 2 \log n$ must hold. Also observe that, for all $1 < i \le h$, flow f_i originates at vertices of L_i, terminates at vertices of $L_1 \cup \cdots \cup L_{i-1}$, uses flow-paths of length at most $2D$, and causes edge-congestion at most 3.

Phase 2: Constructing the trees. In order to construct the spanning trees $T_1, \ldots, T_k$, we start by letting each tree contain all vertices of G and no edges. We then process every vertex $v \in V(G)$ one-by-one. Assume that $v \in L_i$, for some $1 \le i \le h$. Consider the following experiment. Let $\mathcal{Q}(v)$ be the set of all flow-paths that carry non-zero flow in f_i, and connect v to vertices of $L_1 \cup \cdots \cup L_{i-1}$. Let $F(v)$ be the total amount of flow that f_i sends on all paths $P \in \mathcal{Q}(v)$; recall that $F(v) \ge k/2$ must hold. We choose a path $P \in \mathcal{Q}(v)$ at random, where the probability to choose a path P is precisely $f_i(P)/F(v)$. We repeat this experiment k times, obtaining paths $P_1(v), \ldots, P_k(v)$. For each $1 \le j \le k$, we add all edges of $P_j(v)$ to T_j. Consider the graphs $T_1, \ldots, T_k$ at the end of this process. Notice that each such graph T_j may not be a tree. We fist show that the diameter of each such tree is $O(D \log n)$.

▷ Claim 19. For all $1 \le j \le k$, $\mathrm{diam}(T_j) \le O(D \log n)$.

Proof. Fix an index $1 \le j \le k$. Let r be the unique vertex lying in L_1. We prove that for all $1 \le i \le h$, for every vertex $v \in L_i$, there is a path connecting v to r in T_j, of length at most $2D(i-1)$, by induction on i. The base of the induction is when $i = 1$ and the claim is trivially true. Assume now that the claim holds for layers $L_1, \ldots, L_{i-1}$. Let v be any vertex at layer L_i. Consider the path $P_j(v)$ that we have selected. Recall that this path has length at most $2D$, and it connect v to some vertex $u \in L_1 \cup \cdots \cup L_{i-1}$. By the induction hypothesis, there is a path P in T_j of length at most $2D(i-2)$, that connects u to r. Since all edges of $P_j(v)$ are added to T_j, the path $P_j(v)$ is contained in T_j. By concatenating path $P_j(v)$ with path P, we obtain a path connecting v to r, of length at most $2D(i-1)$. ◀

Lastly, using standard analysis of the Randomized Rounding technique, we show that, with probability at least $(1-1/\mathsf{poly}(n))$, every edge of G lies in at most $O(\log n)$ graphs $T_1, \ldots, T_k$. For each $1 \le j \le k$, we can now let T'_j be a BFS tree of the graph T_j, rooted at the vertex r. We conclude that each tree T'_j has diameter at most $O(D \log n)$, and the resulting set $\{T'_1, \ldots, T'_k\}$ of trees cause edge-congestion $O(\log n)$ with high probability.

6 Overview of the Applications to Distributed Computation

Our improved distributed algorithms in highly-connected graphs are based on the following basic tool, which follows by combining Karger's edge sampling and the diameter-fixing Theorem 9.

▷ Claim 20 (Basic Distributed Tool). There is a randomized algorithm that, given a k-edge connected n-vertex graph G and a congestion bound $\eta \in [1, k]$, computes, in $\widetilde{O}((101k \ln n/\eta)^D)$ rounds, a collection of k spanning trees that cause total edge-congestion at most $O(\eta \cdot \log n)$, and have diameter at most $O((101k \ln n/\eta)^D)$ each. Moreover, the algorithm can compute k spanning subgraphs with similar congestion and diameter bounds in $O(D + \eta \log n)$ rounds. The round complexity, the diameter, and the congestion bounds hold with high probability.

Approximation of Minimum-Cut. Ghaffari and Kuhn [10, 7] gave a very simple approach for finding an $O(\log n)$-approximation for the minimum cut problem that is based on Karger's edge sampling technique. The round complexity of their algorithm is $O(\sqrt{n})$ for constant diameter graphs. Combining Theorem 13 with Ghaffari and Kuhn's algorithm immediately leads to an $\widetilde{O}(\lambda)$ algorithm for graphs with constant diameter, where λ is the size of the minimum-cut.

To provide a more general approach for improved algorithms in highly-connected graphs, we next describe the notion of low-congestion shortcuts.

Low-Congestion Shortcuts. This notion, introduced by Ghaffari and Haeupler [9], provides a modular framework for solving global graph problems in the distributed setting.

▶ **Definition 21** (Low-Congestion Shortcuts, [9])**.** *Given a graph $G = (V, E)$, and a partition $S_1, \ldots, S_N$ of V into disjoint subsets, such that for all $1 \le i \le N$, graph $G[S_i]$ is connected, an (α, β)-shortcut is a collection $\{H_1, \ldots, H_N\}$ of subgraphs of G, that satisfy the following:*

- *(1) for each edge $e \in E$, there are at most α subgraphs $G[S_i] \cup H_i$ containing e; and*
- *(2) the diameter of each subgraph $G[S_i] \cup H_i$ is at most β.*

Ghaffari and Haeupler [9] showed that the quality of algorithms for several basic problems depend on the sum of α (i.e., congestion) and β (i.e., the dilation). The quantity of $\alpha + \beta$ is usually referred to as the *quality* of the shortcuts. As observed by [9] for every n-vertex graph G and any collection of vertex-disjoint subsets $S_1, \ldots, S_N$, there exist (α, β) shortcuts for with $\alpha + \beta = O(D + \sqrt{n})$. Our key result is in providing a nearly optimal construction for low-congestion shortcuts in highly connected graphs of constant diameter.

▶ **Theorem 22.** *[Improved Shortcuts in Highly Connected Graphs] There is a randomized algorithm that, for a sufficiently large n, given any k-connected n-vertex graph G of diameter $D = O(\log n / \log \log n)$, together with a partition $\{S_1, \ldots, S_N\}$ of $V(G)$, such that for all $1 \le i \le N$, $G[V_i]$ is a connected graph, w.h.p. computes (α, β) shortcuts, with*

$$\alpha + \beta = \widetilde{O}(\min\{\sqrt{n/k} + n^{D/(2D+1)}\}, n/k),$$

in $\widetilde{O}(\alpha + \beta)$ rounds.

The construction of the shortcuts from Theorem 22 serves the basis for the proof of Theorem 6. In the full version of the paper we describe further algorithmic applications of our results for additional graph problems. The proof of Theorem 7 is based on a careful implementation of Claim 20.

7 Open Problems

For brevity, let us say that a collection $\mathcal{T}$ of spanning trees of a (k, D)-connected graph G is an (α, D')-packing iff $|\mathcal{T}| \ge k/\alpha$ and the diameter of every tree in $\mathcal{T}$ is at most D'. A major remaining open question is: for which values of α and D' can we guarantee the existence of an (α, D')-packing $\mathcal{T}$ of edge-disjoint spanning tree in every (k, D)-connected graph. In particular, is the following statement true: every (k, D)-connected graph G contains a collection of $\Omega(k/\mathsf{poly}\log n)$ edge-disjoint trees of diameter $O(D \cdot \mathsf{poly}\log n)$ each. The only upper bounds that we have are the ones guaranteed by Theorem 2, and we do not have any lower bounds. We also do not have any upper bounds, except for those guaranteed by Theorem 1, if we allow a constant, or more generally any sub-logarithmic congestion. Additionally, obtaining an analogue of the algorithm from Theorem 4 in the distributed setting remains a very interesting open question.

Finally, most of our results are mainly meaningful for the setting where $k = \Omega(\log n)$. It will be very interesting to consider the case of small connectivity $k = O(1)$. One can show that any k-edge connected graph with $k = O(1)$ of diameter D is a $(k, \mathsf{poly}(D))$-connected graph. Is it possible to show that any k-edge-connected graph of diameter D, for some constant $k \ge 3$, has at least two edge-disjoint trees of depth at most $\mathsf{poly}(D)$?

References

1 Keren Censor-Hillel, Mohsen Ghaffari, George Giakkoupis, Bernhard Haeupler, and Fabian Kuhn. Tight bounds on vertex connectivity under vertex sampling. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 2006–2018. Society for Industrial and Applied Mathematics, 2015.

2 Keren Censor-Hillel, Mohsen Ghaffari, and Fabian Kuhn. Distributed connectivity decomposition. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 156–165, 2014.

3 Keren Censor-Hillel, Mohsen Ghaffari, and Fabian Kuhn. A new perspective on vertex connectivity. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 546–561. Society for Industrial and Applied Mathematics, 2014.

4 Mohit Daga, Monika Henzinger, Danupon Nanongkai, and Thatchaphol Saranurak. Distributed edge connectivity in sublinear time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 343–354. ACM, 2019.

5 Michal Dory. Distributed approximation of minimum k-edge-connected spanning subgraphs. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 149–158. ACM, 2018.

6 Mohsen Ghaffari. Distributed broadcast revisited: Towards universal optimality. In *International Colloquium on Automata, Languages, and Programming*, pages 638–649. Springer, 2015.

7 Mohsen Ghaffari. *Improved Distributed Algorithms for Fundamental Graph Problems*. PhD thesis, MIT, USA, 2017. URL: `https://groups.csail.mit.edu/tds/papers/Ghaffari/PhDThesis-Ghaffari.pdf`.

8 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks ii: Low-congestion shortcuts, mst, and min-cut. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 202–219. SIAM, 2016.

9 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks II: low-congestion shortcuts, mst, and min-cut. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 202–219, 2016.

10 Mohsen Ghaffari and Fabian Kuhn. Distributed minimum cut approximation. In *International Symposium on Distributed Computing*, pages 1–15. Springer, 2013.

11 Mohsen Ghaffari, Krzysztof Nowicki, and Mikkel Thorup. Faster algorithms for edge connectivity via random 2-out contractions. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1260–1279. SIAM, 2020.

12 Alon Itai and Michael Rodeh. The multi-tree approach to reliability in distributed networks. *Information and Computation*, 79(1):43–59, 1988.

13 Tomáš Kaiser. A short proof of the tree-packing theorem. *Discrete Mathematics*, 312(10):1689–1691, 2012.

14 David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

15 Naoki Kitamura, Hirotaka Kitagawa, Yota Otachi, and Taisuke Izumi. Low-congestion shortcut and graph parameters. In *33rd International Symposium on Distributed Computing (DISC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

16 Fabian Kuhn. A distributed perspective on graph connectivity and cuts. In *26th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '14, Prague, Czech Republic - June 23 - 25, 2014*, page 1, 2014.

17 Zvi Lotker, Boaz Patt-Shamir, and David Peleg. Distributed MST for constant diameter graphs. *Distributed Computing*, 18(6):453–460, 2006.

18 Danupon Nanongkai and Hsin-Hao Su. Almost-tight distributed minimum cut algorithms. In *International Symposium on Distributed Computing*, pages 439–453. Springer, 2014.

19 CSJA Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

20 Merav Parter and Eylon Yogev. Low congestion cycle covers and their applications. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1673–1692. SIAM, 2019.

21 David Peleg. *Distributed Computing: A Locality-sensitive Approach.* SIAM, 2000.

22 David Pritchard and Ramakrishna Thurimella. Fast computation of small cuts via cycle space sampling. *ACM Transactions on Algorithms (TALG)*, 7(4):46, 2011.

23 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM Journal on Computing*, 41(5):1235–1265, 2012.

24 William Thomas Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 1(1):221–230, 1961.