Deterministic Distributed Algorithms and Lower Bounds in the Hybrid Model

Ioannis Anagnostides ✉
Department of Computer Engineering, National Technical University of Athens, Greece

Themis Gouleakis ✉
Max Planck Institute for Informatics, Saarbrücken, Germany

Abstract

The HYBRID model was recently introduced by Augustine et al. [6] in order to characterize from an algorithmic standpoint the capabilities of networks which combine multiple communication modes. Concretely, it is assumed that the standard LOCAL model of distributed computing is enhanced with the feature of all-to-all communication, but with very limited bandwidth, captured by the node-capacitated clique (NCC). In this work we provide several new insights on the power of hybrid networks for fundamental problems in distributed algorithms.

First, we present a deterministic algorithm which solves any problem on a sparse n-node graph in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds of HYBRID, where the notation $\widetilde{\mathcal{O}}(\cdot)$ suppresses polylogarithmic factors of n. We combine this primitive with several sparsification techniques to obtain efficient distributed algorithms for general graphs. Most notably, for the all-pairs shortest paths problem we give deterministic $(1+\epsilon)$- and $\log n/\log\log n$-approximate algorithms for unweighted and weighted graphs respectively with round complexity $\widetilde{\mathcal{O}}(\sqrt{n})$ in HYBRID, closely matching the performance of the state of the art randomized algorithm of Kuhn and Schneider [33]. Moreover, we[1] make a connection with the Ghaffari-Haeupler framework of low-congestion shortcuts [21], leading – among others – to a $(1+\epsilon)$-approximate algorithm for Min-Cut after $\mathcal{O}(\text{polylog}(n))$ rounds, with high probability, even if we restrict local edges to transfer $\mathcal{O}(\log n)$ bits per round. Finally, we prove via a reduction from the set disjointness problem that $\widetilde{\Omega}(n^{1/3})$ rounds are required to determine the radius of an unweighted graph, as well as a $(3/2-\epsilon)$-approximation for weighted graphs. As a byproduct, we show an $\widetilde{\Omega}(n)$ round-complexity lower bound for computing a $(4/3-\epsilon)$-approximation of the radius in the broadcast variant of the congested clique, even for unweighted graphs.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed Computing, Hybrid Model, Sparse Graphs, Deterministic Algorithms, All-Pairs Shortest Paths, Minimum Cut, Radius

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.5

Related Version *Full Version*: https://arxiv.org/abs/2108.01740

Acknowledgements We are indebted to Christoph Lenzen for carefully reviewing an earlier draft of our work, and proposing several improvements and interesting directions. Specifically, he suggested derandomizing Proposition 7 via the Garay-Kutten-Peleg algorithm, while he also pointed out the connection with low-congestion shortcuts, leading to the results of Section 4. We are also very grateful to the anonymous reviewers at DISC for carefully reviewing this paper, and for indicating many corrections and ways to improve the exposition. We are particularly thankful to a reviewer for providing very detailed arguments which strengthened our results in Section 3.4. All errors remain our own.

[1] See the acknowledgments.

1 Introduction

Hybrid networks have found numerous applications in real-life computer systems. Indeed, leveraging different communication modes has substantially reduced the complexity and has improved the efficiency of the system, measured in terms of the energy consumption, the latency, the number of switching links, etc. For instance, hybrid architectures have been extensively employed in data centers, augmenting the traditional electrical switching architecture with optical switches in order to establish direct connections [11, 17, 51]. Another notable example is the 5G standard, which enhances the traditional cellular infrastructure with device-to-device (D2D) connections in order to guarantee very low latency among communication users (see [50, 29, 37, 41], and references therein).

Despite the central role of hybrid architectures in communication systems, a rigorous investigation of their potential has only recently began to formulate in the realm of distributed algorithms. In particular, Augustine et al. [6] proposed HYBRID, a model which combines the extensively-studied *local* (LOCAL) [38, 44] model with the recently introduced *node-capacitated clique* (NCC) [5]. The former model captures the *locality* of a given problem – nodes are able to exchange messages of arbitrary size but only with adjacent nodes, while the latter model – which enables all-to-all communication but with severe capacity restrictions for every node – addresses the issue of *congestion*; these constitute the main challenges in distributed computing. From a practical standpoint, the local network captures the capabilities of *physical* networks, wherein dedicated edges (e.g. cables or optical fibers) offer large bandwidth and high efficiency, but lack flexibility as they cannot be dynamically adapted by the nodes. In contrast, the global mode relates to *logical* networks, which are formed as an overlay over a shared physical network [18]; here the feature of all-to-all communication comes at the cost of providing very limited throughput.

In this work we follow the recent line of research [6, 33, 18, 9, 25, 10] which endeavors to explore from a theoretical standpoint the power of hybrid models in distributed computing; specifically, the main issue that arises is whether combining two different communication modes offers a substantial improvement over each mode separately. This question is answered in the affirmative for a series of fundamental problems in distributed algorithms, while we also provide some hardness results mainly based on well-established communication-complexity lower bounds.

1.1 Contributions & Techniques

Sparsification

First, we consider the design of HYBRID algorithms in *sparse* graphs – i.e. the average degree is $\operatorname{polylog} n$, where n represents the number of communication entities in the graph. We prove the following general result:

▶ **Theorem 1.** *Consider a graph $G = (V, E, w)$ with $|E| = \widetilde{\mathcal{O}}(n)$. There exists a deterministic distributed algorithm such that every node learns the entire topology of the graph in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds of* HYBRID.

As a warm-up, we first provide a *randomized* HYBRID algorithm so that every node learns the topology in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds. More importantly, we also derandomize this communication pattern (Theorem 1), leading along the way to a derandomization of the token dissemination protocol of Augustine et al. [6], which is one of their main communication primitives. Specifically, we first employ the Garay-Kutten-Peleg algorithm [20] in order to construct a

"balanced" partition of the nodes, so that every cluster has "small" weak diameter. Then, we present several deterministic subroutines which allow to disseminate the composition of the clusters, perform load balancing, and finally broadcast the topology to the entire network within the desired round complexity.

Naturally, our guarantee for sparse graphs has an independent interest given that most communication networks of practical interest are very sparse [36]; the canonical example typically cited is the Internet [40]. Nonetheless, we leverage several *sparsification* techniques in order to design distributed algorithms for general graphs. In particular, we first employ a deterministic *multiplicative spanner* algorithm [23, 49] to obtain a $\log n/\log\log n$-approximation for the weighted *all-pairs shortest paths* (APSP) problem in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds. For unweighted graphs we leverage the recent deterministic *near-additive spanner* due to Elkin and Mater [16], leading to a $(1+\epsilon)$-approximate algorithm for APSP, for any constant $\epsilon > 0$. Although this does not quite reach the performance of the state of the art algorithm of Kuhn and Schneider [6], which yields an exact solution for weighted graphs with asymptotically the same round-complexity (modulo polylogarithmic factors), we stress that our algorithms are *deterministic*.

Moreover, we use *cut sparsifiers* in order to provide near-optimal algorithms for any cut-related problem in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds. Here it is important to point out that our algorithmic scheme "Sparsify & Conquer" is primarily meaningful when the output requirement is *global*. For example, for the Min-Cut problem, if we require that every node knows a cut at the end of the distributed algorithm, we show an $\widetilde{\Omega}(\sqrt{n})$ round-complexity lower bound for any non-trivial approximation based on a technical lemma in [6]. However, in many settings this approach may disseminate an overly amount of information. Indeed, under the usual requirement that each node has to know its "side" on the cut, we establish exponentially faster algorithms.

Simulating CONGEST-based Algorithms

This accelerated algorithm for Min-Cut is obtained through a connection with the concept of *low-congestion shortcuts*, due to Ghaffari and Haeupler [21]. Specifically, in this framework the performance-guarantee for a problem is parameterized in terms of the number of rounds required to solve the standard part-wise aggregation problem. A fascinating insight of Ghaffari and Haeupler [21] is that more "structured" topologies (e.g. planar graphs) enable faster algorithms for solving such problems, bypassing some notorious lower bounds under general graphs. Our observation is that a limited amount of global power, in the form of NCC, interacts particularly well with this line of work since NCC offers very fast primitives for the part-wise aggregation problem. As a result, this connection leads to the following result:

▶ **Theorem 2.** *There exists an $\mathcal{O}(\text{polylog}(n))$-round algorithm for $(1+\epsilon)$-approximate Min-Cut in* CONGEST + NCC.

Note that this guarantees applies even if local edges are restricted to transfer only $\mathcal{O}(\log n)$ bits per round, i.e. the local network is modeled with CONGEST instead of LOCAL. Another notable corollary of this connection is an approximate single-source shortest paths algorithm (Corollary 21) based on a result by Haeupler and Li [26], coming close to the algorithm of Augustine et al. [6] under the substantially more powerful HYBRID model. We also present another simulation argument, which in a sense eliminates the dependence of the performance of a CONGEST algorithm on the hop-diameter through an appropriate augmentation of the graph with global edges (see Proposition 22).

Distance Computation Tasks

Finally, we focus on distance computation tasks, and in particular, the complexity of determining the *radius* and the *diameter* of the underlying graph – the smallest and the largest of the eccentricities respectively. For the former, we show the following result:

▶ **Theorem 3.** *For any $\epsilon \in (0, 1/2]$, determining a $(3/2 - \epsilon)$-approximation for the radius of a weighted graph with probability $2/3$ requires $\widetilde{\Omega}(n^{1/3})$ rounds of* HYBRID*. For unweighted graphs, determining the radius requires $\widetilde{\Omega}(n^{1/3})$ rounds of* HYBRID*.*

This limitation applies for any randomized distributed algorithm even if we allow a substantial probability of failure (i.e. Monte Carlo algorithms), and/or public (common) randomness. We should point out that our lower bound for unweighted graphs matches the known upper bound for *approximate* radius, as the authors in [9] provide a $(1 + \epsilon)$-approximation for all the unweighted eccentricities in $\widetilde{\mathcal{O}}(n^{1/3})$ rounds of HYBRID, for any constant $\epsilon > 0$. Our theorem also supplements the hardness result of Kuhn and Schneider [33] who established analogous lower bounds for the diameter.

More precisely, we give a suitable *dense* gadget graph whose edges correspond to the input-strings of two players endeavoring to solve the set disjointness problem. Then, we show that there is a gap in the value of the radius depending on whether the input of the two players is *disjoint*. Our construction uses a *bit-gadget*, a component introduced in [1] (see also [2]) in order to show a linear lower bound for determining the radius in CONGEST, even for sparse graphs. Nonetheless, our reduction has several differences given that the source of the communication bottleneck is quite different in CONGEST (where it suffices to induce a bottleneck in the *communication cut* between the two players) compared to a model with all-to-all communication. As a result, we first prove an $\widetilde{\Omega}(n)$ round-complexity lower bound for determining a $(4/3 - \epsilon)$-approximation of the radius in the *broadcast* variant of the *congested clique* (BCC), for any $\epsilon \in (0, 1/3]$, even for unweighted graphs; we consider this result to be of independent interest. Next, with minor modifications in the construction we show Theorem 3. These results require simulation arguments, establishing that Alice and Bob can indeed employ (or simulate) the communication pattern of the distributed algorithm in order to solve the set disjointness problem. In this context, for the HYBRID model we make use of the simulation argument of Kuhn and Schneider [33].

Finally, for the weighted diameter the state of the art algorithm in HYBRID simply performs a Dijkstra search from an arbitrary source node and returns as the estimation the eccentricity (i.e. the largest distance) of the source node [9]; an application of the triangle inequality implies that this algorithm yields a 2-approximation of the actual diameter. We make a step towards improving this approximation ratio. Specifically, we show that for graphs with small degrees ($\Delta = \mathcal{O}(\text{polylog}\, n)$) we can obtain a 3/2-approximation of the diameter with asymptotically the same round-complexity, namely $\widetilde{\mathcal{O}}(n^{1/3})$ rounds. This result is based on the sequential algorithm of Roditty and Vassilevska W. [48]. Our contribution is to establish that their algorithm can be substantially parallelized in HYBRID; this is shown by employing some machinery developed in [9] for solving in parallel multiple single-source shortest paths problems.

Due to space constraints, most of the proofs, as well as additional results are presented in the full version of this paper [3].

1.2 Related Work

As we explained in our introduction, the HYBRID model was only recently introduced by Augustine, Hinnenthal, Kuhn, Scheideler, and Schneider [6]. Specifically, they developed several useful communication primitives in order to tackle distance computation tasks;

Table 1 An overview of our main results; it is assumed that $\epsilon > 0$ is an arbitrarily small constant.

Problem	Variant	Approximation	Model	Complexity	Technique
Deterministic APSP	Unweighted Weighted	$1+\epsilon$ $\log n/\log\log n$	HYBRID	$\widetilde{\mathcal{O}}(\sqrt{n})$	Sparsification: [49, 16]
MST Min-Cut SSSP	Weighted Weighted Weighted	Exact $1+\epsilon$ polylog(n)	CONGEST + NCC	$\mathcal{O}(\log^2 n)$ $\mathcal{O}(\text{polylog}(n))$ $\widetilde{\mathcal{O}}(n^\epsilon)$	Shortcuts: [21, 26]
Radius	Unweighted Unweighted Weighted	$4/3-\epsilon$ Exact $3/2-\epsilon$	BCC HYBRID	$\widetilde{\Omega}(n)$ $\widetilde{\Omega}(n^{1/3})$	Set Disjointness: [28, 1, 33]

most notably, for the SSSP problem they established a $(1+o(1))$-approximate solution in $\widetilde{\mathcal{O}}(n^{1/3})$ rounds, while they also presented an algorithm with round complexity $\widetilde{\mathcal{O}}(\sqrt{n})$ for approximately solving the weighted APSP problem with high probability.[2] Their lower bound for the APSP problem was matched in a subsequent work by Kuhn and Schneider [33], showing that $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds suffice in order to *exactly* solve APSP. They also presented an $\widetilde{\Omega}(n^{1/3})$ lower bound for determining the diameter based on a reduction from the two-party set disjointness problem.

Moreover, Censor-Hillel et al. [9] improved several aspects of the approach in [6], showing how to exactly solve multiple SSSP problems in $\widetilde{\mathcal{O}}(n^{1/3})$ rounds; they also presented near-optimal algorithms for approximating all the eccentricities in the graph. For the approximate SSSP problem an improvement over the result in [6] was recently achieved by Censor-Hillel et al. [10], obtaining a $(1+\epsilon)$-approximate algorithm in $\widetilde{O}(n^{5/17})$ rounds of HYBRID, for a sufficiently small constant $\epsilon > 0$. More restricted families of graphs (e.g. very sparse graphs or *cactus* graphs) were considered by Feldmann et al. [18], establishing an exponential speedup over some of the previous results even though they modeled the local network via CONGEST, which is of course substantially weaker than LOCAL. Finally, Götte et al. [25] provided several fast hybrid algorithms for problems such as connected components, spanning tree, and the maximal independent set.

The *node-capacitated clique* model (NCC) was recently introduced in [5]; it constitutes a much weaker – and subsequently much more realistic – model than the *congested clique* (CLIQUE) of Lotker et al. [39] in which every node can communicate with *any* other node (instead of only $\mathcal{O}(\log n)$ other nodes in NCC) with $\mathcal{O}(\log n)$-bit messages. Indeed, in CLIQUE a total of $\widetilde{\Theta}(n^2)$ bits can be transmitted in each round, whereas in NCC the cumulative broadcasting capacity is only $\widetilde{\Theta}(n)$ bits; as evidence for the power of CLIQUE we note that even slightly super-constant lower bounds would give new lower bounds in circuit complexity, as implied by a simulation argument in [15].

Reductions from communication complexity to distributed computing are by now fairly standard in the literature; see [45, 13, 19] and references therein. We also refer to [45, 13, 19] for reductions in the *broadcast* variant of CLIQUE where in each round every node can send the *same* $\mathcal{O}(\log n)$-bit message to all the nodes. Our construction for the radius is inspired by the gadget in [1], wherein the authors showed near-linear lower bounds for determining the radius in CONGEST, even for sparse networks. Finally, we refer to [22, 42, 24] for some of the state of the art technology for the Min-Cut problem.

[2] We will say that an event holds *with high probability* if it occurs with probability at least $1-1/n^c$ for some constant $c > 0$.

2 Preliminaries

We assume that the network consists of a set of n communication entities (e.g. processors) with $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ the set of IDs, and a local communication *topology* given by a graph $G = (V, E, w)$. We will tacitly posit that G is *undirected*, unless explicitly stated otherwise; we also assume that for all $e \in E, w(e) \in \{1, 2, \ldots, W\}$, for some $W = \text{poly}(n)$. At the beginning each node knows the identifiers of each node in its neighborhood, but has no further knowledge about the topology of the graph. Communication occurs in *synchronous rounds*; in every round nodes have unlimited computational power[3] to process the information they posses. The *local* communication mode will be modeled with LOCAL, for which in each round every node can exchange a message of *arbitrary* size with its neighbors in G via the *local* edges. The *global* communication mode uses NCC for which in each round every node can exchange $\mathcal{O}(\log n)$-bit messages with up to $\mathcal{O}(\log n)$ arbitrary nodes via *global* edges. More broadly, one can parameterize hybrid networks by the number of bits λ that can be exchanged via local edges, and the number of bits γ that can be exchanged via the global mode. Interestingly, all standard models can be seen as instances of this general parameterization; namely, LOCAL : $\lambda = \infty, \gamma = 0$, CONGEST : $\lambda = \mathcal{O}(\log n), \gamma = 0$, CLIQUE : $\lambda = 0, \gamma = \mathcal{O}(n \log n)$,[4] NCC : $\lambda = 0, \gamma = \mathcal{O}(\log^2 n)$.

If the capacity of some channel is exceeded the corresponding nodes will only receive an *arbitrary* (potentially adversarially selected) subset of the information according to the capacity of the network, while the rest of the messages are dropped. The performance of a distributed algorithm is measured in terms of its *round-complexity* – the number of rounds required so that every node knows its part of the output; for randomized protocols it will suffice to reach the desired state with high probability. Finally, all of the derived round-complexity upper bounds in HYBRID should be thought of as having a minimum with the (hop) diameter of the network.

2.1 Useful Communication Primitives

A *distributive aggregate function* f maps a multiset $S = \{x_1, \ldots, x_N\}$ of input values to some value $f(S)$, such that there exists an aggregate function g so that for any multiset S and any partition $S_1, \ldots S_\ell, f(S) = g(f(S_1), \ldots, f(S_\ell))$; typical examples that we will use include Max, Min, and Sum. Now consider that we are given a distributive aggregate function f and a set $A \subseteq V$, so that every member of A stores *exactly* one input value. The *aggregate-and-broadcast* problem consists of letting every node in the graph learn the value of f evaluated at the corresponding input.

▶ **Lemma 4** ([5], Theorem 2.2)**.** *There exists an algorithm in* NCC *which solves the aggregate-and-broadcast problem in* $\mathcal{O}(\log n)$ *rounds.*

In the (k, ℓ)-token dissemination problem (henceforth abbreviated as (k, ℓ)-TD) there are k (distinct) tokens (or messages), each of size $\mathcal{O}(\log n)$ bits, with every node initially having at most ℓ tokens. The goal is to guarantee that every node in the graph has collected all of the tokens.

▶ **Lemma 5** ([6], Theorem 2.1)**.** *There exists a randomized algorithm in* HYBRID *which solves the* (k, ℓ)*-TD problem on connected graphs in* $\widetilde{\mathcal{O}}(\sqrt{k} + \ell)$ *rounds with high probability.*

[3] Nonetheless, we remark that most of our algorithms use a reasonable amount of computation.
[4] This follows from Lenzen's routing [34].

2.2 Communication Complexity

Most of our lower bounds are established based on the communication complexity of *set disjointness*, arguably the most well-studied problem in communication complexity (e.g., see [27, 43, 47]). More precisely, consider two communication parties – namely Alice and Bob – with infinite computational power. Every player is given a binary string of k-bits, represented with $x, y \in \{0,1\}^k$ respectively, and their goal is to determine the value of a function $f(x, y)$ by interchanging messages between each other. The players are allowed to use randomization, and the complexity is measured by the expected number of communication in the worst case [52]. For probabilistic protocols the players are required to give the right answer with some probability bounded away from $1/2$, i.e. to outperform random guessing; for concreteness, we assume that the probability of being correct should be $2/3$. It is also interesting to point out that common (public) randomness is allowed, with Alice and Bob sharing an infinite string of independent coin tosses.

In the set disjointness problem (DISJ_k) the two parties have to determine whether there exists $i \in [k]$ such that $x_i = y_i = 1$; in other words, if the inputs x and y correspond to subsets of a universe Ω, the problem asks whether the two subsets are disjoint – with a slight abuse of notation this will be represented with $x \cap y = \emptyset$. We will use the following celebrated result due to Kalyanasundaram and Schnitger [28].

▶ **Theorem 6** ([28]). *The randomized communication complexity of* DISJ_k *is* $\Omega(k)$.

3 Sparsification in Hybrid Networks

As a warm-up, we commence this section by presenting an $\widetilde{\mathcal{O}}(\sqrt{n})$ *randomized* protocol for solving *any* problem on *sparse* graphs in the HYBRID model. More importantly, we also present a *deterministic* algorithm with asymptotically the same round complexity, up to polylogarithmic factors. Next, we present several applications of this result in general graphs via distributed *sparsification* techniques.

3.1 Randomized Protocol

▶ **Proposition 7** (Randomized Hybrid Algorithm for Sparse Networks). *Consider an n-node (connected) graph $G = (V, E, w)$. There exists a randomized algorithm so that every node in V can learn the entire topology of the graph in $\widetilde{\mathcal{O}}(\sqrt{|E|})$ rounds of* HYBRID *with high probability.*

Sketch of Proof. Suppose that the lowest degree node incident to each edge $e \in E$ is responsible for disseminating the information about that edge. Then, each node will be responsible for at most $\sqrt{2|E|}$ edges since otherwise it would have at least $\sqrt{2|E|}$ neighbors with degree at least $\sqrt{2|E|}$, which is a contradiction. Thus, the result follows from Lemma 5. ◀

3.2 Deterministic Protocol

Before we proceed with our deterministic algorithm let us first recall that the *strong diameter* of a subset $C \subset V$ is the diameter of the subgraph induced by C; in contrast, the *weak diameter* of C is measured in the original graph. We will analyze and explain every step of the algorithm separately. We stress that the round-complexity in some steps has not been optimized since it would not alter the asymptotic running time of the protocol. Also note that in the sequel we use the words component and cluster interchangeably.

Algorithm 1 Deterministic HYBRID Algorithm for Sparse Networks.

Input: An n-node graph $G = (V, E)$ such that $|E| = \widetilde{\mathcal{O}}(n)$.
Output Requirement: Every node knows the entire topology of G.

1. Determine a partition of the nodes V into $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ via the Garay-Kutten-Peleg algorithm such that for all i,
   (i) the strong diameter of $\mathcal{C}_i$ is $\mathcal{O}(\sqrt{n})$;
   (ii) $|\mathcal{C}_i| \geq \sqrt{n}$.
2. Let $C_1, C_2, \ldots, C_N = \textsc{Fragment}(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ such that for all i,
   (i) the weak diameter of C_i is $\mathcal{O}(\sqrt{n})$;
   (ii) $\sqrt{n} \leq |C_i| < 2\sqrt{n}$.
3. Broadcast the IDs of the components' leaders.
4. Distribute all the C_i's via the MatchingComponents subroutine.
5. Assign every edge on a component, and perform LoadBalancing.
6. Disseminate all the information.

Step 1. The first step of the algorithm partitions the set of nodes into a collection of connected components $\mathcal{C}_1, \ldots, \mathcal{C}_k$, so that the minimum size is at least $\sqrt{n}$ and the strong diameter in every component is $\mathcal{O}(\sqrt{n})$; for simplicity we will assume that $\sqrt{n}$ is an integer. This step will be implemented with the standard Garay-Kutten-Peleg (GKP) algorithm [20, 35]. Specifically, GKP is an MST algorithm which operates in two phases; we will only need the first phase. The main idea is to gradually perform merges but in a "balanced" manner. More precisely, GKP maintains a set of components. In each iteration i every component with diameter at most 2^i determines the minimum-weight outgoing edge, which is subsequently added to a set of "candidates" edges. Then, the algorithm determines a maximal matching on this set, updating the components accordingly. If a component with diameter smaller than 2^i did not participate in the maximal matching, the algorithm automatically incorporates the edge that was selected by it. This process is repeated for $i = 0, 1, \ldots, \lceil \log \sqrt{n} \rceil$, leading to a partition of V into $\mathcal{C}_1, \ldots, \mathcal{C}_k$.

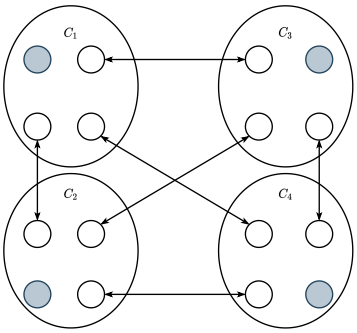
▶ **Lemma 8** ([35]). *At the end of the first phase of the* GKP *algorithm every component has strong diameter $\mathcal{O}(\sqrt{n})$, while every component has at least $\sqrt{n}$ nodes.*

This lemma verifies our initial claim for step 1. Moreover, note that the first phase of GKP can be implemented in $\mathcal{O}(\sqrt{n} \log^* n)$ rounds in CONGEST; naturally, in HYBRID we can substantially reduce the number of rounds, but this would not affect the overall asymptotic complexity as there is an inherent bottleneck in subsequent steps of the algorithm.

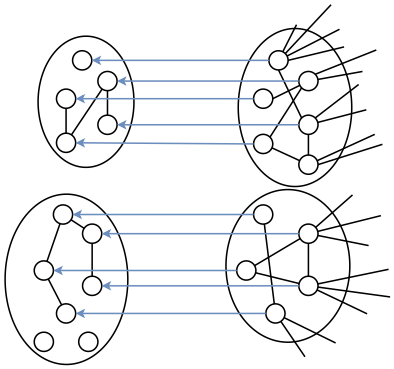
Step 2. The Fragment subroutine of the second step is particularly simple. If a component $\mathcal{C}_i$ is such that $|\mathcal{C}_i| < 2\sqrt{n}$ it remains intact. Otherwise, the component $\mathcal{C}_i$ is decomposed arbitrarily into disjoint fragments each of size between $\sqrt{n}$ and $2\sqrt{n}$. Let $C_1, \ldots, C_N$ be the induced partition of V. By virtue of Lemma 8 we know that the *weak* diameter of every C_i is $\mathcal{O}(\sqrt{n})$, although note that the induced graph on C_i is potentially disconnected. This step is made to ensure that the components have roughly the same size, while it can be trivially implemented in $\mathcal{O}(\sqrt{n})$ rounds of LOCAL.

Step 3. We assume that every component has elected a leader, e.g. the node with the smallest ID. There are overall $N \leq \sqrt{n}$ IDs to be broadcast to the entire graph. This can be implemented with N (deterministic) broadcasts in NCC, which requires $\mathcal{O}(\sqrt{n} \log n)$ rounds.

Step 4. The purpose of this step is to ensure that every node knows the composition – i.e. the set of IDs – of every other cluster. To this end, the leader of every component C_i selects arbitrarily $N - 1 \leq |C_i| - 1$ *representative* nodes from C_i, and devises a (bijective) mapping from these nodes to all the other components; the leader also informs via the local network the corresponding nodes. Then, the protocol proceeds in rounds: In every iteration a single component interacts with all the others, and specifically, every representative node sends its ID to the leader of its assigned component. This is repeated for all the components, and after $\mathcal{O}(\sqrt{n})$ rounds every representative node will be matched with some node on its corresponding component; see Figure 1a. Having established this matching every node can disseminate through the global network the IDs of all the nodes in its own component to its assigned node. This process requires at most $2\sqrt{n}$ rounds since every component has size less than $2\sqrt{n}$, and every node participates in at most one matching. Finally, the composition (the set of IDs) of every component is revealed to each node after $\mathcal{O}(\sqrt{n})$ additional rounds of the local network.



(a) An example of MatchingComponents.



(b) LoadBalancing: Transferring load from overloaded to underloaded components. We have highlighted with blue "global" edges.

Step 5. First of all, every edge with incident nodes residing on the same component is assigned to the component of its endpoints. Otherwise, the edge is assigned to one of the components according to some deterministic rule; e.g. the component with the smaller ID. In this context, the *load* of every component is the number of edges it has to disseminate. Notice that the nodes of each component can learn every component's load in $\mathcal{O}(\sqrt{n})$ rounds. Initially, the load of each component is distributed uniformly within the nodes of the component, which requires $\mathcal{O}(\sqrt{n})$ rounds. The LoadBalancing mechanism works as follows: It splits the components into a set of "overloaded" components with load more than $2|E|/N$, and a set of "underloaded" components with load less than $|E|/N$; every other component does not need any further processing. In every iteration we map arbitrarily (e.g. the overloaded component with the smallest ID is mapped to the underloaded component with the smallest ID, and so on) overloaded components to underloaded ones, so that the mapping is one-to-one and maximal. Then, every assigned overloaded component transmits as much load as is required to its corresponding component via the global network (see Figure 1b) until one of the two becomes balanced – according to the previous notion. This can be performed in $\widetilde{\mathcal{O}}(1)$ by virtue of Step 4 (recall that $|E| = \widetilde{O}(n)$). Then, we remove any components that have been balanced and we proceed recursively for the remaining ones.

It is easy to see that this process requires at most N iterations since in every iteration we eliminate at least one component from requiring further balancing, while at the end of this step every component will have $\widetilde{\mathcal{O}}(\sqrt{n})$ load.

Step 6. The final step is fairly straightforward. First, observe that a single component can transfer its entire load to another component in $\widetilde{\mathcal{O}}(1)$ rounds via the global network; this follows because (i) every component has $\widetilde{\mathcal{O}}(\sqrt{n})$ load due to the load balancing step, and (ii) every component has by construction roughly $\sqrt{n}$ nodes. Assume that the components $C_1, \ldots, C_N$ are sorted in ascending order with respect to their IDs. Then, at iteration i component C_j transfers its load to C_{r+1}, where $r = i + j \mod N$. This is repeated for $i = 0, 1, \ldots, N-2$. It is easy to see that this deterministic protocol guarantees that (i) no *collisions* occur, and (ii) every component eventually receives the load from all other components. As we previously argued every such iteration requires $\widetilde{\mathcal{O}}(1)$ rounds in HYBRID. Thus, overall this step requires $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds, leading to the following conclusion:

▶ **Theorem 1.** *Consider a graph $G = (V, E, w)$ with $|E| = \widetilde{\mathcal{O}}(n)$. There exists a deterministic distributed algorithm such that every node learns the entire topology of the graph in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds of* HYBRID.

More broadly, our deterministic protocol can be used for any m-edge (connected) graph by forming clusters of size $\Theta(\sqrt{m})$ nodes, so that every node learns the topology after $\widetilde{\mathcal{O}}(\sqrt{m})$ rounds. Consequently, this leads to a derandomization of the token dissemination algorithm of Augustine et al. [6] in the regime $k \geq n$:

▶ **Proposition 9.** *There exists a deterministic algorithm which solves the (k, ℓ)-TD problem in $\widetilde{\mathcal{O}}(\sqrt{k})$ rounds of* HYBRID*, assuming that $k \geq n$.*

3.3 Distributed Sparsification Techniques

Before we present applications of our protocol for general graphs, we first review some basic sparsification techniques. The goal is to efficiently sparsify the graph in a distributed fashion, while approximately preserving some *structure* in the graph. We will use two fundamental notions of sparsifiers.

Spanners

The first structure one may wish to retain with sparsification is every pairwise distance in the graph. To this end, we will employ the notion of a graph *spanner*, a fundamental object in graph theory with numerous applications in distributed computing [46]. To be more precise, for a graph $G = (V, E)$, a subgraph H is an *α-stretch spanner* if every pairwise distance in H is at most an $\alpha \geq 1$ factor larger than the distance in G, without ever underestimating; i.e., for all $u, v \in G, d_G(u, v) \leq d_H(u, v) \leq \alpha \cdot d_G(u, v)$. Naturally, we desire spanners with small stretch and a limited number of edges. It is well-known that any n-node graph admits a $(2k-1)$-stretch spanner with $\mathcal{O}(n^{1+1/k})$ number of edges, while this trade-off is optimal conditioned on Erdős girth conjecture. In the distributed context, we will use the following result of Rozhon and Ghaffari:

▶ **Theorem 10** ([49])**.** *Consider an n-node weighted graph $G = (V, E)$. There exists a deterministic distributed algorithm in* CONGEST *which computes a $(2k-1)$-stretch spanner of size $\widetilde{\mathcal{O}}(kn^{1+1/k})$ in* polylog(n) *rounds.*

Besides *multiplicative* spanners, we will also use *near-additive* spanners. More precisely, a subgraph H of G is an (α, β)-stretch spanner if for all $u, v \in G, d_G(u,v) \leq d_H(u,v) \leq \alpha \cdot d_G(u,v) + \beta$; for $\beta = 0$ this recovers the previous notion of a multiplicative spanner. Moreover, for $\alpha = 1 + \epsilon$, for an arbitrarily small $\epsilon > 0$, the spanner is called near-additive. In this context, we will leverage the following recent result due to Elkin and Matar [16]:

▶ **Theorem 11** ([16]). *Consider an n-node unweighted graph $G = (V, E)$. For any constants $\epsilon \in (0,1)$ and $\rho \in (0, 1/2)$, there is an algorithm in* CONGEST *which computes a $(1+\epsilon, \beta)$-stretch spanner of G with $\widetilde{\mathcal{O}}(n)$ number of edges after $\mathcal{O}(\beta n^{\rho})$ rounds, where $\beta = \mathcal{O}((\log\log n/\rho + 1/\rho^2)^{\log\log n + 1/\rho})$.*

Cut Sparsifiers

Another fundamental class of sparsifiers endeavors to approximately preserve the weight of every *cut* in the graph. Recall that for a subset of vertices $S \subset V$ we define

$$\mathrm{cut}_G(S) = \sum_{u \in S, v \in V \setminus S} w(u,v). \tag{1}$$

To this end, we will employ the sparsification algorithm developed by Koutis [32]. We should remark that the algorithm of Koutis actually returns a *spectral* sparsifier, which is a strictly stronger notion than a cut sparsifier [7], but we will not use this property here.

▶ **Theorem 12** ([32], Theorem 5). *Consider a graph $G = (V, E, w)$. There exists a distributed algorithm in* CONGEST *such that for any $\epsilon > 0$ outputs a graph $H = (V, \widehat{E}, \widehat{w})$ after $\widetilde{\mathcal{O}}(1/\epsilon^2)$ rounds such that (i) $(1-\epsilon)\,\mathrm{cut}_H(S) \leq \mathrm{cut}_G(S) \leq (1+\epsilon)\,\mathrm{cut}_H(S)$ for any $S \subset V$, and (ii) the expected number of edges in H is $\widetilde{\mathcal{O}}(n/\epsilon^2)$.*

3.4 Applications

Deterministic APSP

In the α-approximate *all-pairs shortest paths* problem every node $u \in V$ has to learn a value $d'(u,v)$ such that $d(u,v) \leq d'(u,v) \leq \alpha \cdot d(u,v)$, for all $v \in V$. In this context, we establish the following result:

▶ **Proposition 13** (Weighted APSP). *Consider an n-node weighted graph $G = (V, E)$. There exists a deterministic $\log n / \log\log n$-approximation algorithm for the APSP problem which runs in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds of* HYBRID.

For unweighted graphs we use near-additive spanners (Theorem 11) to improve upon the approximation ratio established for weighted graphs.

▶ **Proposition 14** (Unweighted APSP). *Consider an n-node unweighted graph $G = (V, E)$. For any constant $\epsilon \in (0,1)$, there exists a deterministic $(1+\epsilon)$-approximation algorithm for the APSP problem which runs in $\widetilde{\mathcal{O}}(\sqrt{n})$ rounds of* HYBRID.

Cut Problems

Moreover, we will leverage the distributed algorithm of Koutis in order to obtain efficient algorithms for cut-related problems in the HYBRID model. We wish to convey the robustness of our approach by presenting a guarantee for a series of cut problems. First, we recall the following: The *minimum cut* problem consists of identifying a partition of the vertices V into

S and $V \setminus S$ in order to minimize the weight of $\text{cut}_G(S)$; it admits an efficient centralized solution, for example, via Karger's celebrated algorithm [30]. Note that for an unweighted graph the minimum cut coincides with the *edge connectivity*. The $s-t$ *minimum cut* problem is similar to the minimum cut problem, but the nodes s and t are restricted to reside on different sets of the partition; see [8, 12]. Finally, in the *sparsest cut* problem we are searching for a partition $S, V \setminus S$ that minimizes the quantity $\text{cut}_G(S)/(|S| \cdot |V \setminus S|)$; it is known that the sparsest cut problem is $\mathcal{NP}$-hard [4, 31].

▶ **Proposition 15** (HYBRID Algorithms for Cut Problems)**.** *For any n-node graph $G = (V, E, w)$ and for any $\epsilon \in (0, 2)$, we can compute with high probability a $(1+\epsilon)$-approximation in expected $\widetilde{\mathcal{O}}(\sqrt{n}/\epsilon + 1/\epsilon^2)$ rounds of* HYBRID *for the following problems: (i) the minimum $s-t$ cut, (ii) the minimum cut, and (iii) the sparsest cut.*

This approach is meaningful for cut-related problems once we impose a stronger output requirement. Namely, we guarantee that every node will know at the end of the distributed algorithm the entire composition of an approximate cut. In fact, for such an output requirement we can establish an almost-matching lower bound based on a technique developed in [6]:

▶ **Proposition 16.** *Determining a W/n-approximation for the minimum cut problem requires $\widetilde{\Omega}(\sqrt{n})$ rounds of* HYBRID*, where $W \geq n$ is the maximum edge-weight, assuming that every node has to know a cut at the end of the distributed algorithm.*

4 Simulating CONGEST Algorithms

The approach developed in the previous section is primarily meaningful for problems with a very demanding output requirement; for example, APSP, or cut-related problems for which nodes have to learn the exact composition of the cut. In contrast, in this section we will show that, for the minimum cut problem, we can obtain substantially faster algorithms when the nodes have to simply learn their "side" on the cut, which constitutes the usual output requirement in distributed algorithms. This result (Corollary 20) will be established through a connection with the concept of *low-congestion shortcuts*, which also implies other important results as well; e.g. for the SSSP problem (Corollary 21). We also present another simulation argument, leading to an accelerated algorithm for approximating the diameter. It should be stress that for this section we model the local network via the weaker CONGEST model.

4.1 Low-Congestion Shortcuts

Consider a graph $G = (V, E)$ under the CONGEST model, and a partition of V into k parts $P_1, \ldots, P_k$ such that the induced graph $G[P_i]$ is connected. A recurring scenario in distributed algorithms consists of having to perform simultaneous aggregations in each part; this will be referred to as the *part-wise aggregation* problem. For example, an instance of this problem corresponds to determining the minimum-weight outgoing edge in the context of Boruvka's celebrated algorithm. A very insightful observation by Ghaffari and Haeupler [21] was to parameterize the performance of algorithms based on the complexity of the part-wise aggregation problem. For instance, if it admits a solution in Q rounds, under any collection of parts, we can compute an MST in $\mathcal{O}(Q \log n)$ rounds via Boruvka's algorithm. Now although in general graphs $Q = \mathcal{O}(\sqrt{n} + D)$ rounds, with the bound being existential tight for certain topologies, a key insight of Ghaffari and Haeupler [21] is that special classes of graphs allow for accelerated algorithms via *shortcuts*; most notably, for planar graphs they showed that the part-wise aggregation problem can be solved in $\widetilde{\mathcal{O}}(D)$ rounds of CONGEST, bypassing the notorious $\Omega(\sqrt{n})$ rounds for "global" problems under general graphs.

In the hybrid model this connection is particularly useful since the NCC model enables very fast algorithms for solving the part-wise aggregation problem:

▶ **Lemma 17** ([5]). *The part-wise aggregation problem admits a solution with high probability in $\mathcal{O}(\log n)$ rounds in* NCC.

As a result, we can directly derive a near-optimal algorithm for the minimum spanning tree problem through an implementation based on Boruvka's algorithm:

▶ **Corollary 18.** *There exists a distributed algorithm which computes with high probability an MST in $\mathcal{O}(\log^2 n)$ rounds of* CONGEST + NCC.

It should be noted that a deterministic $\mathcal{O}(\log^2 n)$ algorithm in CONGEST + NCC for the MST problem was developed in [18] with very different techniques. More importantly, Ghaffari and Haeupler [21] managed to establish the following:

▶ **Theorem 19** ([21]). *If we can solve the part-wise aggregation problem in Q rounds of* CONGEST*, there exists an $\widetilde{\mathcal{O}}(Q \operatorname{poly}(1/\epsilon))$ distributed algorithm for computing with high probability a $(1+\epsilon)$-approximation of the minimum cut, for any sufficiently small $\epsilon > 0$.*

We should remark that in [21] the authors establish this result only for planar graphs, but their argument can be directly extended in the form of this theorem. As a result, if we use Lemma 17 we arrive at the following conclusion:

▶ **Corollary 20.** *Consider any n-node weighted graph. There exists an $\mathcal{O}(\operatorname{polylog}(n))$-round algorithm in* CONGEST + NCC *for computing with high probability a $(1+\epsilon)$-approximation of Min-Cut, for any sufficiently small constant $\epsilon > 0$.*

In terms of *exact* Min-Cut, one can obtain an $\widetilde{\mathcal{O}}(\sqrt{n})$-round algorithm in CONGEST + NCC by simulating the recent algorithm due to Dory et al. [14], which requires $\widetilde{\mathcal{O}}(D + \sqrt{n})$ rounds of CONGEST; an analogous simulation argument is employed in the next subsection, so we omit the proof here. However, this leaves a substantial gap between exact and approximate Min-Cut. Moreover, analogous results can be established for computing approximate shortest paths by virtue of a result by Haeupler and Li [26]:

▶ **Corollary 21.** *Consider any n-node weighted graph. There exists with high probability a $\operatorname{polylog}(n)$-approximate algorithm for the single-source shortest paths problem which runs in $\widetilde{O}(n^{\epsilon})$ rounds in* CONGEST + NCC*, for any constant $\epsilon > 0$.*

We remark that Haeupler and Li [26] actually provide a more general result, but we state this special case for the sake of simplicity. Of course, there are other applications as well, as we have certainly not exhausted the literature. Overall, this connection illustrates another very concrete motivation of low-congestion shortcuts.

4.2 Diameter

We also provide another notable simulation argument. In particular, the main idea is to augment the local topology with a limited number of "global" edges so that the resulting graph has a small diameter, and at the same time the solution to the underlying problem remains invariant.

▶ **Proposition 22.** *There exists a distributed algorithm in* CONGEST + NCC *which determines a $3/2$-approximation of the diameter in $\mathcal{O}(\sqrt{n \log n})$ rounds with high probability.*

5 Distance Computations

5.1 Lower Bound for the Radius

In this subsection we show a lower bound of $\widetilde{\Omega}(n^{1/3})$ rounds for computing the radius – the smallest eccentricity of the graph – in the HYBRID model, even for unweighted graphs. We commence by constructing a suitable "gadget" in the BCC model, and then we will massage it appropriately to establish a guarantee for HYBRID as well. Our construction is inspired by that in [1] which established a sharp lower bound for sparse networks in CONGEST. First, consider a set of nodes $U \cup V \cup V' \cup U'$, and we let $U = \{u_0, u_1, \ldots, u_{k-1}\}, V = \{v_0, v_1, \ldots, v_{k-1}\}, V' = \{v'_0, v'_1, \ldots, v'_{k-1}\}$, and $U' = \{u'_0, u'_1, \ldots, u'_{k-1}\}$. We also incorporate edges of the form $\{v_i, v'_i\}$ for all $i \in \{0, 1, \ldots, k-1\} = [k]^*$. An important additional ingredient is the *bit-gadget* [1], which works as follows: every node in U and U' will inherit some edge-connections based on the binary representation of their index; the role of this component will become clear as we proceed with the construction. Formally, consider a (different) set of nodes F, T, F', T', and let $F = \{f_0, f_1, \ldots, f_{l-1}\}, T = \{t_0, t_1, \ldots, t_{l-1}\}, F' = \{f'_0, f'_1, \ldots, f'_{l-1}\}$, and $T' = \{t'_0, t'_1, \ldots, t'_{l-1}\}$, where $l = \lceil \log k \rceil$. Now consider a node $u_i \in U$, and let $i = \overline{b_{l-1} \ldots b_1 b_0}$ be the binary representation of its index; for every $j \in [l]^*$ we add the edge $\{u_i, f_j\}$ if $b_j = 0$; otherwise, we add the edge $\{u_i, t_j\}$. This process is also repeated for the nodes in U' (with respect to the sets F' and T'). Next, we add the edges $\{f_j, t_j\}$ and $\{f'_j, t'_j\}$ for all $j \in [l]^*$, while a critical element of the construction is the set of edges $\{\{f_j, t'_j\} : j \in [l]^*\} \cup \{\{t_j, f'_j\} : j \in [l]^*\}$. We also incorporate in the graph two nodes w, w' such that w is connected to all the nodes in U and V, and w' is connected to all the nodes in U' and V'. Finally, we add three nodes z_0, z_1, z_2, as well as the set of edges $\{\{z_0, z_1\}\} \cup \{\{z_1, z_2\}\} \cup \{\{z_0, u_i\} : i \in [k]^*\}$.

Having constructed this base graph the next step is to encode the input of Alice and Bob as edges on the induced graph. Specifically, let $x \in \{0,1\}^{k^2}$ and $y \in \{0,1\}^{k^2}$ represent the input strings of Alice and Bob respectively. We let $x_{i,j} = 1 \iff \{u_i, v_j\} \in E$, and $y_{i,j} = 1 \iff \{v'_j, u'_i\} \in E$. We denote the induced graph with $G_k^{x,y}$; an example of our construction is illustrated in Figure 2.

▷ Claim 23. For every node u in $G_k^{x,y}$ besides the nodes in U it follows that $\mathrm{ecc}(u) \geq 4$.

▷ Claim 24. The radius R of $G_k^{x,y}$ is 3 if $x \cap y \neq \emptyset$; otherwise, $R = 4$.

▶ **Theorem 25.** *For any $\epsilon \in (0, 1/3]$, determining a $(4/3 - \epsilon)$-approximation for the radius of an unweighted graph with probability $2/3$ requires $\widetilde{\Omega}(n)$ rounds of* BCC.

Next, we will show how to adapt this construction for the HYBRID model. Specifically, if $\ell \in \mathbb{N}$ is some parameter, we introduce the following modifications: instead of connecting the corresponding nodes in V with V', F with T', and F' with T directly via edges, we will connect them via paths of length ℓ edges; moreover, we create the path $z_0 \to z_1 \to \cdots \to z_{\ell+1}$ (in place of $z_0 \to z_1 \to z_2$). As before, the players' inputs $x, y \in \{0,1\}^{k^2}$ shall be encoded as edges between U with V, and V' with U' for Alice and Bob respectively. Let $G_{k,\ell}^{x,y}$ be the induced graph; the following claim admits an analogous proof to Claim 24:

▷ Claim 26. The radius R of $G_{k,\ell}^{x,y}$ is $\ell + 2$ if $x \cap y \neq \emptyset$; otherwise, $R \geq \ell + 3$.

Consequently, we are ready to state the implied lower bound in the HYBRID model for unweighted graphs.

▶ **Theorem 27.** *Determining the radius of an unweighted graph with probability $2/3$ requires $\widetilde{\Omega}(n^{1/3})$ rounds of* HYBRID.

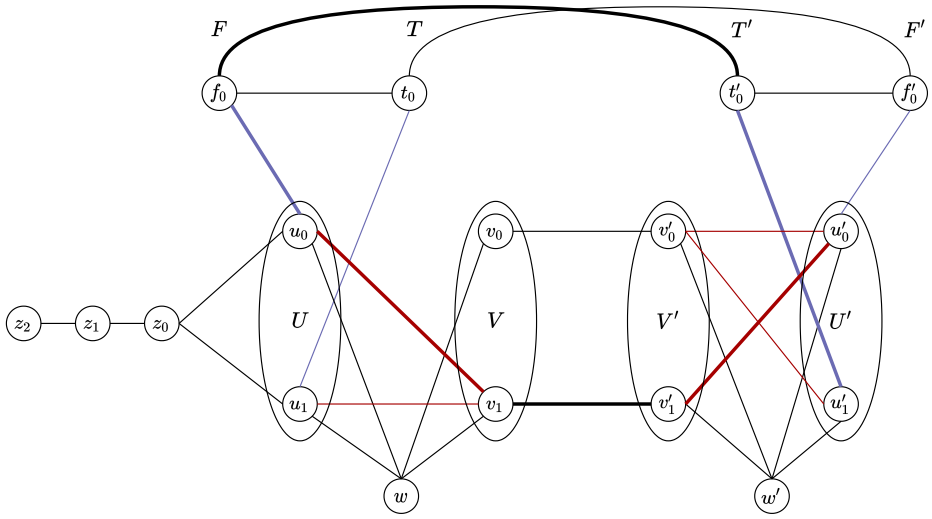


Figure 2 An example of our construction for the radius. The red edges correspond to the players' inputs, while the blue edges map nodes of U and U' to their *bit-gadget*. Observe that $d(u_0, u'_0) = 3$ (we have highlighted the corresponding path in the figure) as $\{u_0, v_1\} \in E$ and $\{v'_1, u'_0\} \in E$, implying that $x \cap y \neq \emptyset$. We have also highlighted the path of length 3 from u_0 to u'_1 through the bit-gadget.

Weighted Graphs

Next, we further modify our construction in order to obtain stronger lower bounds for weighted graphs. Specifically, if W represents the maximum-weight edge, we endow every edge of the graph $G^{x,y}_{k,\ell}$ with weight W, with the following exceptions: (i) all the edges belonging in paths connecting V to V'; (ii) all the edges belonging in paths connecting F to T' and T to F'; and (iii) all the edges belonging in the path $z_1 \to z_2 \to \cdots \to z_{\ell+1}$. We will represent the induced weighted graph as $G^{x,y}_{k,\ell,W}$.

▷ Claim 28. For every node u in $G^{x,y}_{k,\ell,W}$ besides the nodes in U it follows that $\text{ecc}(u) \geq \ell + 3W$.

▷ Claim 29. The radius R of $G^{x,y}_{k,\ell,W}$ is $\ell + 2W$ if $x \cap y \neq \emptyset$; otherwise, $R = \ell + 3W$.

Observe that for sufficiently large W this claim implies an asymptotically 3/2-gap depending on whether $x \cap y = \emptyset$. As a result, we are ready to establish the following theorem:

▶ **Theorem 30.** *For any $\epsilon \in (0, 1/2]$, determining a $(3/2 - \epsilon)$-approximation for the radius of a weighted graph with probability 2/3 requires $\widetilde{\Omega}(n^{1/3})$ rounds of* HYBRID*, assuming that $W = \omega(n^{1/3})$*

5.2 Diameter

Finally, we employ the machinery developed in [9] for solving in parallel multiple single-source shortest paths problems in HYBRID in order to derive an efficient implementation of the sequential algorithm of Roditty and Vassilevska W. [48] in HYBRID. This leads to an improved approximation algorithm for the diameter under graphs with small degrees:

▶ **Proposition 31.** *For any weighted graph G with max-degree $\Delta = \mathcal{O}(\text{polylog}\, n)$ we can determine a 3/2-approximation of the diameter in $\widetilde{\mathcal{O}}(n^{1/3})$ rounds of* HYBRID*.*

References

1 Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-linear lower bounds for distributed distance computations, even in sparse networks. In Cyril Gavoille and David Ilcinkas, editors, *Distributed Computing - 30th International Symposium, DISC 2016*, volume 9888 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2016. doi:10.1007/978-3-662-53426-7_3.

2 Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, pages 1681–1697. SIAM, 2015. doi:10.1137/1.9781611973730.112.

3 Ioannis Anagnostides and Themis Gouleakis. Deterministic distributed algorithms and lower bounds in the hybrid model, 2021. arXiv:2108.01740.

4 Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009. doi:10.1145/1502793.1502794.

5 John Augustine, Mohsen Ghaffari, Robert Gmyr, Kristian Hinnenthal, Christian Scheideler, Fabian Kuhn, and Jason Li. Distributed computation in node-capacitated networks. In Christian Scheideler and Petra Berenbrink, editors, *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019*, pages 69–79. ACM, 2019. doi:10.1145/3323165.3323195.

6 John Augustine, Kristian Hinnenthal, Fabian Kuhn, Christian Scheideler, and Philipp Schneider. Shortest paths in a hybrid network model. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1280–1299. SIAM, 2020. doi:10.1137/1.9781611975994.78.

7 Joshua D. Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: theory and algorithms. *Commun. ACM*, 56(8):87–94, 2013. doi:10.1145/2492007.2492029.

8 András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 1996*, pages 47–55. ACM, 1996. doi:10.1145/237814.237827.

9 Keren Censor-Hillel, Dean Leitersdorf, and Volodymyr Polosukhin. Distance computations in the hybrid network model via oracle simulations, 2020. arXiv:2010.13831.

10 Keren Censor-Hillel, Dean Leitersdorf, and Volodymyr Polosukhin. On sparsity awareness in distributed computations. In Kunal Agrawal and Yossi Azar, editors, *SPAA '21: 33rd ACM Symposium on Parallelism in Algorithms and Architectures, 2021*, pages 151–161. ACM, 2021. doi:10.1145/3409964.3461798.

11 Tao Chen, Xiaofeng Gao, and Guihai Chen. The features, hardware, and architectures of data center networks: A survey. *Journal of Parallel and Distributed Computing*, 96:45–74, 2016. doi:10.1016/j.jpdc.2016.05.009.

12 Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 273–282, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993674.

13 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. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 363–372. Association for Computing Machinery, 2011. doi:10.1145/1993636.1993686.

14 Michal Dory, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Distributed weighted min-cut in nearly-optimal time. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021*, pages 1144–1153. ACM, 2021. doi:10.1145/3406325.3451020.

15 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14*, pages 367–376. ACM, 2014. doi:10.1145/2611462.2611493.

16 Michael Elkin and Shaked Matar. Ultra-sparse near-additive emulators. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, 2021*, pages 235–246. ACM, 2021. doi:10.1145/3465084.3467926.

17 Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: A hybrid electrical/optical switch architecture for modular data centers. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, page 339–350, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1851182.1851223.

18 Michael Feldmann, Kristian Hinnenthal, and Christian Scheideler. Fast hybrid network algorithms for shortest paths in sparse graphs, 2020. arXiv:2007.01191.

19 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 1150–1162. Society for Industrial and Applied Mathematics, 2012. doi:10.1137/1.9781611973099.91.

20 Juan A. Garay, Shay Kutten, and David Peleg. A sublinear time distributed algorithm for minimum-weight spanning trees. *SIAM J. Comput.*, 27(1):302–316, 1998. doi:10.1137/S0097539794261118.

21 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks II: low-congestion shortcuts, mst, and min-cut. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 202–219. SIAM, 2016. doi:10.1137/1.9781611974331.ch16.

22 Mohsen Ghaffari and Fabian Kuhn. Distributed minimum cut approximation. In Yehuda Afek, editor, *Distributed Computing*, pages 1–15, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-41527-2_1.

23 Mohsen Ghaffari and Fabian Kuhn. Derandomizing distributed algorithms with small messages: Spanners and dominating set. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018*, volume 121 of *LIPIcs*, pages 29:1–29:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.DISC.2018.29.

24 Mohsen Ghaffari and Krzysztof Nowicki. Congested clique algorithms for the minimum cut problem. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, page 357–366, New York, NY, USA, 2018. Association for Computing Machinery. URL: https://dl.acm.org/citation.cfm?id=3212750.

25 Thorsten Götte, Kristian Hinnenthal, Christian Scheideler, and Julian Werthmann. Time-optimal construction of overlay networks, 2020. arXiv:2009.03987.

26 Bernhard Haeupler and Jason Li. Faster distributed shortest path approximations via shortcuts. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018*, volume 121 of *LIPIcs*, pages 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.DISC.2018.33.

27 Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory Comput.*, 3(1):211–219, 2007. doi:10.4086/toc.2007.v003a011.

28 Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, 1992. doi:10.1137/0405044.

29 Udit Narayana Kar and Debarshi Kumar Sanyal. An overview of device-to-device communication in cellular networks. *ICT Express*, 4(4):203–208, 2018. doi:10.1016/j.icte.2017.08.002.

30 David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, page 21–30, USA, 1993. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=313559.313605.

31 P. Klein, C. Stein, and É. Tardos. Leighton-rao might be practical: Faster approximation algorithms for concurrent flow with uniform capacities. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 310–321. Association for Computing Machinery, 1990. doi:10.1145/100216.100257.

32 Ioannis Koutis. Simple parallel and distributed algorithms for spectral graph sparsification, 2014. arXiv:1402.3851.

33 Fabian Kuhn and Philipp Schneider. Computing shortest paths and diameter in the hybrid network model. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC '20, page 109–118. Association for Computing Machinery, 2020. doi:10.1145/3382734.3405719.

34 Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In Panagiota Fatourou and Gadi Taubenfeld, editors, *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*, pages 42–50. ACM, 2013. doi:10.1145/2484239.2501983.

35 Christoph Lenzen. *Lectures notes on Theory of Distributed Systems.* , 2016. URL: https://www.mpi-inf.mpg.de/fileadmin/inf/d1/teaching/winter15/tods/ToDS.pdf.

36 Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

37 P. Li, S. Guo, and I. Stojmenovic. A truthful double auction for device-to-device communications in cellular networks. *IEEE Journal on Selected Areas in Communications*, 34(1):71–81, 2016. doi:10.1109/JSAC.2015.2452587.

38 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992. doi:10.1137/0221015.

39 Zvi Lotker, Elan Pavlov, Boaz Patt-Shamir, and David Peleg. MST construction in O(log log n) communication rounds. In Arnold L. Rosenberg and Friedhelm Meyer auf der Heide, editors, *SPAA 2003: Proceedings of the Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 94–100. ACM, 2003. doi:10.1145/777412.777428.

40 Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. The graph structure in the web – analyzed on different aggregation levels. *The Journal of Web Science*, 1(1):33–47, 2015. doi:10.1561/106.00000003.

41 A. Murkaz, R. Hussain, S. F. Hasan, M. Y. Chung, B. . Seet, P. H. J. Chong, S. T. Shah, and S. A. Malik. Architecture and protocols for inter-cell device-to-device communication in 5G networks. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 489–492, 2016. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2016.95.

42 Danupon Nanongkai and Hsin-Hao Su. Almost-tight distributed minimum cut algorithms. In Fabian Kuhn, editor, *Distributed Computing*, pages 439–453, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-45174-8_30.

43 Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006. doi:10.1016/j.jet.2004.10.007.

44 David Peleg. *Distributed Computing: A Locality-Sensitive Approach.* Society for Industrial and Applied Mathematics, USA, 2000.

45 David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed mst construction. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, page 253, USA, 1999. IEEE Computer Society. doi:10.1109/SFFCS.1999.814597.

46 David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989. doi:10.1137/0218050.

47 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684.

48 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13*, pages 515–524. ACM, 2013. doi:10.1145/2488608.2488673.

49 Václav Rozhon and Mohsen Ghaffari. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 350–363. ACM, 2020. doi:10.1145/3357713.3384298.

50 C. Wang, F. Haider, X. Gao, X. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir. Cellular architecture and key technologies for 5G wireless communication networks. *IEEE Communications Magazine*, 52(2):122–130, 2014. doi:10.1109/MCOM.2014.6736752.

51 Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T.S. Eugene Ng, Michael Kozuch, and Michael Ryan. C-through: Part-time optics in data centers. *SIGCOMM Comput. Commun. Rev.*, 40(4):327–338, August 2010. doi:10.1145/1851182.1851222.

52 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, page 209–213, New York, NY, USA, 1979. Association for Computing Machinery. doi:10.1145/800135.804414.