Almost Universally Optimal Distributed Laplacian Solvers via Low-Congestion Shortcuts*

Ioannis Anagnostides ✉
Carnegie Mellon University, Pittsburgh, PA, USA

Christoph Lenzen ✉
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Bernhard Haeupler ✉
ETH Zürich, Switzerland

Goran Zuzic ✉
ETH Zürich, Switzerland

Themis Gouleakis ✉
National University of Singapore, Singapore

Abstract

In this paper, we refine the (almost) *existentially optimal* distributed Laplacian solver recently developed by Forster, Goranci, Liu, Peng, Sun, and Ye (FOCS '21) into an (almost) *universally optimal* distributed Laplacian solver.

Specifically, when the topology is known (i.e., the Supported-CONGEST model), we show that any Laplacian system on an n-node graph with *shortcut quality* $\mathrm{SQ}(G)$ can be solved after $n^{o(1)}\mathrm{SQ}(G)\log(1/\varepsilon)$ rounds, where ε is the required accuracy. This almost matches our lower bound that guarantees that any correct algorithm on G requires $\widetilde{\Omega}(\mathrm{SQ}(G))$ rounds, even for a crude solution with $\varepsilon \leq 1/2$. Several important implications hold in the unknown-topology (i.e., standard CONGEST) case: for excluded-minor graphs we get an almost universally optimal algorithm that terminates in $D \cdot n^{o(1)}\log(1/\varepsilon)$ rounds, where D is the hop-diameter of the network; as well as $n^{o(1)}\log(1/\varepsilon)$-round algorithms for the case of $\mathrm{SQ}(G) \leq n^{o(1)}$, which holds for most networks of interest. Conditioned on improvements in state-of-the-art constructions of low-congestion shortcuts, the CONGEST results will match the Supported-CONGEST ones.

Moreover, following a recent line of work in distributed algorithms, we consider a hybrid communication model which enhances CONGEST with limited global power in the form of the node-capacitated clique (NCC) model. In this model, we show the existence of a Laplacian solver with round complexity $n^{o(1)}\log(1/\varepsilon)$.

The unifying thread of these results, and our main technical contribution, is the study of a novel ρ-*congested* generalization of the standard *part-wise aggregation* problem. We develop near-optimal algorithms for this primitive in the Supported-CONGEST model, almost-optimal algorithms in (standard) CONGEST (with the additional overhead due to standard barriers), as well as a simple algorithm for bounded-treewidth graphs with a quadratic dependence on the congestion ρ. This primitive can be readily used to accelerate the Laplacian solver of Forster, Goranci, Liu, Peng, Sun, and Ye, and we believe it will find further independent applications in the future.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed algorithms, Laplacian solvers, low-congestion shortcuts

Digital Object Identifier 10.4230/LIPIcs.DISC.2022.6

Related Version *Full Version*: `https://arxiv.org/abs/2109.05151`

* The author ordering was randomized using `https://www.aeaweb.org/journals/policies/random-author-order/generator`. It is requested that citations of this work list the authors separated by `\textcircled{r}` instead of commas: Anagnostides ⓡ Lenzen ⓡ Haeupler ⓡ Zuzic ⓡ Gouleakis.

36th International Symposium on Distributed Computing (DISC 2022).
Editor: Christian Scheideler; Article No. 6; pp. 6:1–6:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Bernhard Haeupler*: Supported in part by NSF grants CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808, a Sloan Research Fellowship, funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (ERC grant agreement 949272), and the Swiss National Foundation (project grant 200021-184735).
Goran Zuzic: Supported in part by the Swiss National Foundation (project grant 200021-184735).
Themis Gouleakis: Supported in part by an NRF Fellowship for AI (R-252-000-A33-133). Part of the work was done while visiting the Simons Institute for Theory of Computing).

Acknowledgements We are grateful to the anonymous reviewers for their valuable feedback.

1 Introduction

The *Laplacian paradigm* has emerged as one of the cornerstones of modern algorithmic graph theory. Integrating techniques from combinatorial optimization with powerful machinery from numerical linear algebra, it was originally pioneered in [47] who established the first nearly-linear time solvers for a (linear) Laplacian system. Thereafter, there has been a considerable amount of interest in providing simpler and more efficient solvers [34, 33, 37]. Indeed, this framework has led to some state of the art algorithms for a wide range of fundamental graph-theoretic problems; e.g., see [5, 40, 10, 48, 32, 43, 4], and references therein. In the distributed setting, a major breakthrough was very recently made in [18]. In particular, the authors developed a distributed algorithm that solves any Laplacian system on an n-node graph after $n^{o(1)}(\sqrt{n} + D)\log(1/\varepsilon)$ rounds of the standard CONGEST model, where D represents the hop-diameter of the underlying network and $\varepsilon > 0$ is the error of the solver. Moreover, they showed that their algorithm is *existentially optimal*, up to the $n^{o(1)}$ factor, establishing a lower bound of $\widetilde{\Omega}(\sqrt{n} + D)$ rounds via a reduction from the $s - t$ connectivity problem [13].

This *existential* lower bound in the CONGEST model of distributed computing should hardly come as any surprise. Indeed, it is well-known by now that a remarkably wide range of *global* optimization problems, including minimum spanning tree (MST), minimum cut (Min-Cut), maximum flow, and single-source shortest paths (SSSP), require $\widetilde{\Omega}(\sqrt{n} + D)$ rounds[1] [41, 15, 13]. The same limitation generally applies to any non-trivial approximation and even under randomization. Nonetheless, these lower bounds are constructed on some pathological graph instances that arguably do not occur in practice. This begs the question: *Can we obtain more refined performance guarantees based on the underlying topology of the communication network?* The framework of *low-congestion shortcuts*, introduced by [20], demonstrated that bypassing the notorious $\Omega(\sqrt{n})$ lower bound is possible: MST and Min-Cut on *planar graphs* can be solved in $\widetilde{O}(D)$ rounds. This is crucial, given that in many graphs of practical significance the diameter is remarkably small; e.g., $D = \text{polylog}(n)$ (as is folklore, this holds for most social networks), implying *exponential improvements* over generic algorithms used for general graphs. In the context of the distributed Laplacian paradigm, we raise the following question:

Is there a faster distributed Laplacian solver under "non-worst-case" families of graphs in the CONGEST model?

The only known technique in distributed computing for designing algorithms that go below the $\sqrt{n}$-bound is the low-congestion shortcut framework of Ghaffari and Haeupler [20], and the large ecosystem of tools built around it [25, 26, 29, 21, 50, 23, 27]. However, the "ρ-congested minor" primitive introduced and extensively used in the novel distributed Laplacian

[1] As usual, we use the notation $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to suppress polylogarithmic factors on n.

solver [18] is out of reach from the current set of tools available in the low-congestion shortcut framework. We address this issue by introducing an analogous primitive called *ρ-congested part-wise aggregation*, which greatly simplifies the interface used by [18]. We then extend the low-congestion shortcut framework with new techniques that enables it to near-optimally solve this primitive: we provide both an algorithm that utilizes the very recent hop-constrained expander decompositions for shortcut construction [27] to solve the primitive in general graphs with a linear dependence on ρ, as well as a very simple algorithm with a quadratic ρ-dependence for bounded-treewidth graphs. Finally, we settle our original question in the positive by establishing that our new primitive can be readily used to accelerate the distributed Laplacian solver for non-worst-case topologies.

Specifically, we show our new techniques are sufficient to lift the existentially optimal algorithm [18] to a *universally optimal* algorithm – modulo $n^{o(1)}$ factor inherent in the prior approach – for distributedly solving a Laplacian system, meaning that, *for any topology*, our algorithm is essentially as fast as possible. In other words, for any graph, our algorithm almost matches the best possible (correct) algorithm for that graph. This result is unconditional in essentially all settings of interest (see Theorem 2 for details), but relies on conjectured improvements of current state-of-the-art constructions of low-congestion shortcuts to achieve unqualified universal optimality – like all other results in the area.

Furthermore, another concrete way of bypassing the $\widetilde{\Omega}(\sqrt{n} + D)$ lower bound, besides investigating non-worst-case families of graphs, is by enhancing the local communication network with a limited amount of *global power*. Indeed, research concerning *hybrid* networks was recently initiated in the realm of distributed algorithms [3], although networks combining different communication modes have already found numerous applications in real-life computing systems; as such, hybrid networks have been intensely studied in other areas of distributed computing (see [9, 49, 31], and references therein). In this paper, we will enhance the standard CONGEST model with the recently introduced *node-capacitated clique* (henceforth NCC) [2]. The latter model enables all-to-all communication, but with severe capacity restrictions for every node. The integration of these models will be referred to as the HYBRID model for the rest of this work. This leads to the following central question:

Is there a faster distributed Laplacian solver in the HYBRID model?

Our paper essentially settles this question by showing the same ρ-congested part-wise aggregation primitive can be efficiently solved in $\widetilde{O}(\rho)$ rounds of NCC, implying an almost optimal $n^{o(1)}$-round distributed algorithm for solving Laplacian systems in the HYBRID model. A conceptual contribution of our approach is that we treat both CONGEST, Supported-CONGEST, and HYBRID in a *unified way* through the lens of the low-congestion shortcut framework, by designing our algorithm using high-level primitives and leaving the model-specific translations to the framework itself. A similar unified view of PRAM (i.e., parallel) and CONGEST (i.e., distributed) graph algorithms through the same lens has led to very recent breakthroughs on long-standing open problems for both of these settings [45].

1.1 Overview of our Contributions and Techniques

The unifying thread and the main technical ingredient of our (almost) universally optimal distributed Laplacian solvers is a new fundamental communication primitive referred to as the *congested part-wise aggregation problem.* Specifically, we develop near-optimal algorithms for solving this problem in the (Supported-)CONGEST and the NCC model (Section 3), and then we utilize this primitive to develop almost universally optimal Laplacian solvers.

1.1.1 The Congested Part-Wise Aggregation Problem

To introduce the congested part-wise aggregation problem, let us first give some basic background. The aforementioned Ghaffari-Haeupler framework of low-congestion shortcuts revolves around the so-called *part-wise aggregation problem* posed as follows: "The graph is partitioned into *disjoint* and individually-connected parts, and we need to compute some simple aggregate function for each part, e.g., the minimum of the values held by the nodes in a given part" [20] (see Definition 4 for a formal definition). Importantly, it has been shown that this primitive can be solved efficiently in *structured* topologies and that many problems (including the MST, shortest path, min-cut, etc.) reduce to a small number of calls to a part-wise aggregation oracle, leading to universally optimal algorithms. Unfortunately, it is not clear how to reduce solving a Laplacian system to (a small number of) part-wise aggregation calls; in this paper, we primarily address this issue.

Our first technical contribution is to extend the framework of low-congestion shortcuts by studying a more general primitive: one that incorporates *congestion* (of the input parts) into the underlying *part-wise aggregation* instance. More precisely, unlike the standard part-wise aggregation problem, we allow each node to participate in up to $\rho \in \mathbb{Z}_{\geq 1}$ aggregation parts (see Definition 13). We later show that efficient solutions to this primitive leads to efficient distributed Laplacian solvers.

We first remark that a natural strategy for solving congested part-wise aggregation instances does not work: congested instances *cannot*, in general, be directly reduced to a "small" collection of 1-congested instances, thereby necessitating a more refined approach. To this end, our approach is based on "lifting" the underlying communication network $\overline{G}$ into its *ρ-layered version* $\widehat{G}_{O(\rho)}$: every edge is replaced with a matching and every node with a ρ-clique. The importance of this transformation is that, as we show in Lemma 15, the ρ-congested part-wise aggregation problem can be reduced to a 1-congested instance on the ρ-layered graph (Section 3.1.1). This is first established under the assumption that individual parts correspond to simple paths, and then we extend our results to general parts by following [29]. In light of this reduction, we next focus on solving the 1-congested part-wise aggregation instance on the layered graph.

As a warm-up, we treat graphs with bounded *treewidth* $\mathrm{tw}(G)$ (Definition 11). It is known from [26] that on a graph G with treewidth $\mathrm{tw}(G)$, a 1-congested part-wise aggregation instance can be solved in $\widetilde{O}(\mathrm{tw}(G)D)$ rounds of CONGEST. Keeping this in mind, we first show that the treewidth of the ρ-layered graph $\widehat{G}_\rho$ can only increase by a factor of ρ compared to the original graph (Lemma 19). Hence, we can solve 1-congested instances in $\widehat{G}_{O(\rho)}$ in $\widetilde{O}(\rho\, \mathrm{tw}(\overline{G})D)$ rounds (when the underlying network is $\widehat{G}_{O(\rho)}$), which in turn allows us to solve ρ-congested instances on $\overline{G}$ in $\widetilde{O}(\rho^2\, \mathrm{tw}(G)D)$ time in G (another ρ factor is necessary to simulate $\widehat{G}_{O(\rho)}$ in $\overline{G}$). This positive result poses a natural question: can we achieve similar results on graphs with bounded *minor density* $\delta(G)$ (Definition 9)? However, the answer to this question is negative: *minor density* can blow up even for a 2-layered planar graph (see Observation 21), making such a result impossible.

Then, we look at arbitrary graphs G: it is known that 1-congested part-wise aggregation instances can be solved in a number of rounds that is controlled by $\mathrm{SQ}(G)$, where $\mathrm{SQ}(G)$ is the *shortcut quality* of G (a certain graph parameter we formalize in Definition 7). Specifically, it can be solved in $\widetilde{O}(\mathrm{SQ}(G))$ rounds when the topology is known in advance[2] [29] and $\mathrm{poly}(\mathrm{SQ}(G)) \cdot n^{o(1)}$ in general CONGEST [27]. The shortcut quality parameter is significant

[2] This model is also known as the *supported* CONGEST. That is, CONGEST under the assumption that the topology is known; see Section 2 for a formal description of the model. Our techniques also apply in the full generality of CONGEST, as we explain in the sequel.

since many distributed problems (including the MST, shortest path, min-cut, and – Laplacian solving, as we show later) require $\widetilde{\Omega}(\mathrm{SQ}(G))$ rounds in CONGEST to be solved on G [29]. Therefore, algorithms that have an upper bound close to $\mathrm{SQ}(G)$ are *universally optimal*.

With the end goal of solving the 1-congested part-wise aggregations on layered graphs $\widehat{G}_\rho$ in time controlled by $\mathrm{SQ}(G)$, our main result established that *the shortcut quality of the ρ-layered graph does not increase* (modulo polylogarithmic factors) as compared to the original graph (Theorem 22). This has a plethora of important consequences: (1) when $\mathrm{SQ}(G) \leq n^{o(1)}$, we can unconditionally solve ρ-congested part-wise aggregation instances in $\rho \cdot n^{o(1)}$ CONGEST rounds using the state-of-the-art shortcut construction [27], and (2) when the topology of G is known (i.e., Supported-CONGEST), there exists a distributed algorithm that solves any ρ-congested part-wise aggregation problem in $\rho \cdot \widetilde{O}(\mathrm{SQ}(G))$ rounds via [29]. As a consequence of our general result, the shortcut quality of any 2-layered planar graph is $\widetilde{O}(D)$ since the shortcut quality of a planar graph is $\widetilde{O}(D)$ [20]. This is perhaps the most natural example of a graph whose minor density is very far from the shortcut quality; the only other example documented in the literature so far is that of *expander graphs*.

Our proof proceeds by employing alternative characterizations of the shortcut quality in terms of certain communication tasks. Specifically, shortcut quality can be shown to be equal (modulo polylogarithmic factors) to the following two-player max-min game: the first (max) player chooses k sources and k sinks in the graph such that we can find k node-disjoint paths matching the sources with the sinks; then the second (min) player finds the smallest so-called *quality* Q such that there exist k paths matching the sources with the sinks with the path lengths being at most Q and each edge of the underlying graph supporting at most Q of second player's paths. This characterization allows us to compare the shortcut quality of $\widehat{G}_\rho$ with $\overline{G}$ as follows: take the worst-case (first player's) set of sources and sinks in $\widehat{G}_\rho$. Project them to $\overline{G}$ and note they have node congestion ρ (due to the construction of $\widehat{G}_\rho$). Then, we show we can decompose (i.e., partition) these set of sources and sinks into $\widetilde{O}(\rho)$ pairs of sub-sources and sub-sinks that are node-disjointly connectable in G. However, each such set enjoys paths of quality $\mathrm{SQ}(G)$, hence embedding each such pair in a separate layer of $\widehat{G}_\rho$ shows that the shortcut quality of $\mathrm{SQ}(\widehat{G}_\rho)$ is at most $\widetilde{O}(\mathrm{SQ}(\overline{G}))$. Although this general approach improves over our result for treewidth-bounded graphs we previously described, our approach for the latter class of graphs is substantially simpler and more suited in practice.

1.1.2 Almost Universally Optimal Laplacian Solvers

First, we note that any distributed Laplacian solver that always correctly outputs an answer on a fixed graph G must take at least $\tilde{\Omega}(\mathrm{SQ}(G))$ rounds, giving us a lower bound to compare ourselves with. Our refined lower bound uses the hardness result recently shown by [29] for the spanning connected subgraph problem, applicable for *any* (i.e., non-worst-case) graph G. Specifically, we show that a Laplacian solver can be leveraged to solve the spanning connected subgraph problem, thereby substantially strengthening the lower bound in [18].

▶ **Theorem 1.** *Consider a graph $\overline{G}$ with shortcut quality $\mathrm{SQ}(\overline{G})$. Then, solving a Laplacian system on $\overline{G}$ with $\varepsilon \leq \frac{1}{2}$ requires $\tilde{\Omega}(\mathrm{SQ}(\overline{G}))$ rounds in both CONGEST and Supported-CONGEST models.*

On the upper-bound side, we utilize the congested part-wise aggregation primitive to improve and refine the Laplacian solver of [18], leading to a substantial improvement in the round complexity under *structured* network topologies.

▶ **Theorem 2.** *Consider any n-node graph G with shortcut quality $\mathrm{SQ}(G)$ and hop-diameter D. There exists a distributed Laplacian solver with error $\varepsilon > 0$ with the following guarantees:*

- *In the Supported-CONGEST model, it requires $n^{o(1)}\,\mathrm{SQ}(G)\log(1/\varepsilon)$ rounds.*
- *In the CONGEST model, it requires $n^{o(1)}\,\mathrm{poly}(\mathrm{SQ}(G))\log(1/\varepsilon)$ rounds.*
- *In the CONGEST model on graphs with minor density δ, it requires $n^{o(1)}\delta D\log(1/\varepsilon)$ rounds.*

We note that the above algorithm is almost (up to inherent $n^{o(1)}$ factors) universally optimality for most settings of interest. Since it is (almost) matching the $\mathrm{SQ}(G)$-lower-bound, it is unconditionally universally optimal when the topology is known in advance (i.e., Supported-CONGEST). Furthermore, in standard CONGEST, we give almost universally optimal $Dn^{o(1)}\log(1/\varepsilon)$-round algorithms for topologies that include planar graphs, $n^{o(1)}$-*genus* graphs, $n^{o(1)}$-treewidth graphs, excluded-minor graphs, since all of them are graphs with minor density $\delta(G) = n^{o(1)}$. Furthermore, for the realistic case of $D \leq n^{o(1)}$, it holds for most networks of interest that $\mathrm{SQ}(G) \leq n^{o(1)}$ (e.g., expanders, hop-constrained expanders, as well as all classes mentioned earlier), for which we get $n^{o(1)}\log(1/\varepsilon)$-round solvers. Finally, the conjectured improvements of the state-of-the-art of almost-optimal low-congestion shortcut constructions would immediately lift our results to be unconditionally universally optimal in CONGEST; this issue is orthogonal and not within the scope of this paper.

Furthermore, in HYBRID we obtain an almost optimal complexity in *general graphs*:

▶ **Theorem 3.** *Consider any n-node graph. There exists a distributed Laplacian solver in the HYBRID model with round complexity $n^{o(1)}\log(1/\varepsilon)$, where $\varepsilon > 0$ is the error of the solver.*

This implies a remarkably fast subroutine for solving a Laplacian system in HYBRID under arbitrary topologies. As a result, we corroborate the observation that a very limited amount of global power can lead to substantially faster algorithms for certain optimization problems, supplementing a recent line of work [8, 3, 35, 16, 7, 24, 36, 11]. Furthermore, our framework based on the congested part-wise aggregation problem allows for a unifying treatment of both (Supported-)CONGEST and HYBRID, and we consider this to be an important conceptual contribution of our work. Indeed, as we previously explained, both of our accelerated Laplacian solvers rely on faster algorithms for solving the congested part-wise aggregation problem. In particular, for (Supported-)CONGEST we have already described our approach in detail, while in the HYBRID model we employ certain communication primitives developed in [2] for dealing with congestion in part-wise aggregations. A byproduct of our results is that the framework of low-congestion shortcuts interacts particularly well with the HYBRID model, as was also observed in [1].

1.2 Further Related Work

Our main reference point is the recent Laplacian solver of [18] with *existentially* almost-optimal complexity of $n^{o(1)}(\sqrt{n} + D)\log(1/\varepsilon)$ rounds, where $\varepsilon > 0$ represents the error of the solver. Specifically, they devised several new ideas and techniques to circumvent certain issues which mostly relate to the bandwidth restrictions of the CONGEST model; these building blocks, as well as the resulting Laplacian solver are revisited in our work to refine the performance of the solver. We are not aware of any previous research addressing this problem in the distributed context. On the other hand, the Laplacian paradigm has attracted a considerable amount of interest in the community of parallel algorithms [44, 6].

Research concerning hybrid communication networks in distributed algorithms was recently initiated by [3]. Specifically, they investigated the power of a model which integrates the standard LOCAL model [38] with the recently introduced node-capacitated clique (NCC) [2], focusing mostly on distance computation tasks. Several of their results were subsequently improved and strengthened in subsequent works [35, 7] under the same model of computation. In our work we consider a substantially weaker model, imposing a severe limitation on the communication over the "local edges". This particular variant has been already studied in some recent works for a variety of fundamental problems [16, 24].

The NCC model, which captures the global network in all hybrid models studied thus far, was introduced in [2] partly to address the unrealistic power of the *congested clique* (CLIQUE) [39]. In the latter model each node can communicate *concurrently and independently* with *all* other nodes by $O(\log n)$-bit messages. In contrast, the NCC model allows communication with $O(\log n)$ (arbitrary) nodes per round. As a result, in the HYBRID model and under a sparse local network, only $\widetilde{\Theta}(n)$ bits can be exchanged overall per round, whereas CLIQUE allows for the exchange of up to $\widetilde{\Theta}(n^2)$ (distinct) 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 [14]. Finally, we remark a subsequent work that leverages tools from the Laplacian paradigm in the *broadcast* variant of the congested clique [17].

2 Preliminaries

General notation. We denote with $[k] := \{1, 2, \dots, k\}$. Graphs throughout this paper are undirected. The nodes and the edges of a given graph G are denoted as $V(G)$ and $E(G)$, respectively. We also use $n := |V(G)|$ for brevity. The graphs are often weighted, in which case we assume (as is standard) that for all $e \in E(G), \boldsymbol{w}(e) \in \{1, 2, \dots, \text{poly}(n)\}$. We will denote the hop-diameter of a graph G with $D(G)$ (the hop-diameter ignores weights). Moreover, we use $A \uplus B$ to denote the multiset union, i.e., each element is repeated according to its multiplicity; this operation corresponds to disjoint unions when $A \cap B = \emptyset$.

Communication models. The communication network consists of a set of $\overline{n}$ entities with $[\overline{n}] := \{1, 2, \dots, \overline{n}\}$ being the set of their IDs, and a local communication *topology* given by a graph $\overline{G}$.[3] We define $D := D(\overline{G})$ to be the (hop-)diameter of the underlying network. At the beginning, each node knows its own unique $O(\log \overline{n})$-bit identifier as well as the weights of the incident edges. Communication occurs in *synchronous rounds*, and in every round nodes have unlimited computational power to process the information they possess. We will consider models with both *local* and *global* communication modes.

The *local* communication mode will be modeled with the *CONGEST model* [42] and *Supported-CONGEST model* [46], for which in each round every node can exchange an $O(\log \overline{n})$-bit message with each of its neighbors in $\overline{G}$ via the *local* edges. In the (standard) CONGEST model, each node $v \in V(\overline{G})$ initially only knows the identifiers of each node in v's own neighborhood, but has no further knowledge about the topology of the graph. On the other hand, in the Supported-CONGEST model, all nodes know the entire topology of $\overline{G}$ upfront, but not the input.

The *global* communication mode will be modeled using *NCC* [2], for which in each round every node can exchange $O(\log \overline{n})$-bit messages with $O(\log \overline{n})$ arbitrary nodes via *global* edges. If the capacity of some channel is exceeded, i.e., too many messages are sent to the

[3] To avoid any possible confusion we point out that, for consistency with the nomenclature of [18], we henceforth reserve $\overline{G}$ to denote the underlying *communication network*, while G is used in statements regarding arbitrary graphs.

same node, it will only receive an *arbitrary* (potentially adversarially selected) subset of the information based on the capacity of the network; the rest of the messages are dropped. In this context, we will let HYBRID be the integration of CONGEST and NCC (i.e., nodes have both a *local* and a *global* communication mode at their disposal).

The performance of a distributed algorithm will be measured in terms of its *round complexity* – the number of rounds required so that every node knows its part of the output. For randomized algorithms it will suffice to reach the desired state with high probability.[4] We will assume throughout this work that nodes have access to a common source of randomness; this comes without any essential loss of generality in our setting [19]. When talking about a distributed algorithm for a specific problem (e.g., Laplacian solving, part-wise aggregation, etc.) we assume the input is appropriately *distributedly stored* (i.e., each node will know its own part) and, upon termination, it will be required that the output is appropriately distributedly stored. The appropriate way to distributedly store the input and output will be explained in the problem definition.

Low-Congestion Shortcuts. A recurring scenario in distributed algorithms for global problems (e.g. MST) boils down to solving the following part-wise aggregation problem:

▶ **Definition 4** (Part-Wise Aggregation Problem). *Consider an n-node graph G whose node set $V(G)$ is* partitioned *into k (disjoint) parts $P_1 \uplus \cdots \uplus P_k \subseteq V(G)$ such that each induced subgraph $G[P_i]$ is* connected*. In the* part-wise aggregation *problem, each node $v \in V$ is given its part-ID (if any) and an $O(\log n)$-bit value $\boldsymbol{x}(v)$ as input. The goal is that, for every part P_i, all nodes in P_i learn the part-wise aggregate $\bigoplus_{w \in P_i} \boldsymbol{x}(w)$, where $\bigoplus$ is an arbitrary pre-defined* aggregation function*.*

Throughout this paper, we will assume that the aggregation function $\bigoplus$ is commutative and associative (e.g. min, sum, logical-AND), although this is not strictly needed (e.g., see [23]). To solve such problems, [20] introduced a natural combinatorial graph structure that they refer to as *low-congestion shortcuts*.

▶ **Definition 5** (Low-Congestion Shortcuts). *Consider a graph G whose node set $V(G)$ is* partitioned *into k (disjoint) parts $P_1 \uplus \cdots \uplus P_k \subseteq V(G)$ such that each induced subgraph $G[P_i]$ is* connected*. A collection of subgraphs $H_1, \ldots, H_k$ is a* shortcut *of G with* congestion *c and* dilation *d if the following properties hold: (i) the (hop) diameter of each subgraph $G[P_i] \cup H_i$ is at most d, and (ii) every edge is included in at most c many of the subgraphs H_i. The quantity $Q = c + d$ will be referred to as the* quality *of the shortcut.*

Importantly, a shortcut of quality Q allows us to solve the part-wise aggregation problem in $\widetilde{O}(Q)$ rounds of CONGEST, as formalized below.

▶ **Proposition 6.** *Suppose that $P_1, \ldots, P_k$ is any part-wise aggregation instance in a communication network $\overline{G}$. Given a shortcut of quality Q, we can solve with high probability the part-wise aggregation problem in $\widetilde{O}(Q)$ CONGEST rounds.*

Shortcut Quality and Construction of Shortcuts. Shortcut quality, introduced below, is a fundamental graph parameter that has been proven to characterize the complexity of many important problems in distributed computing.

[4] We say that an event holds with high probability if it occurs with probability at least $1 - 1/n^c$ for a (freely choosable) constant $c > 0$.

▶ **Definition 7.** *Given a graph $G = (V, E)$, we define the* shortcut quality $\mathrm{SQ}(G)$ *of G as the optimal (smallest) shortcut quality of the worst-case partition of V into disjoint and connected parts $P_1 \uplus P_2 \uplus \ldots \uplus P_k \subseteq V$.*

For fundamental problems such as MST, SSSP, and Min-Cut any correct algorithm requires $\widetilde{\Omega}(\mathrm{SQ}(\overline{G}))$ rounds on any network $\overline{G}$, even if we allow randomized solutions and (non-trivial) approximation factors. In fact, this limitation holds even when the network topology $\overline{G}$ is known to all nodes in advance [29]. We remark that $\widetilde{\Omega}(D(\overline{G})) \le \mathrm{SQ}(\overline{G}) \le O(D(\overline{G}) + \sqrt{\overline{n}})$, and the upper bound is known to be tight in certain (pathological) worst-case graph instances.

Moreover, *assuming* fast distributed algorithms for constructing shortcuts of quality competitive with $\mathrm{SQ}(\overline{G})$, all of the aforementioned problems can be solved in $\widetilde{O}(\mathrm{SQ}(\overline{G}))$ rounds [20, 50, 23]. However, the key issue here is the algorithmic construction of the shortcuts upon which the above papers rely. While there has been a lot of recent progress in this regard, current algorithms are quite complicated and have sub-optimal guarantees. We recall below these state-of-the-art $\mathrm{SQ}(\overline{G})$-competitive construction results.

▶ **Theorem 8.** *There exists a distributed algorithm that, given any part-wise aggregation instance on any $\overline{n}$-node graph $\overline{G}$, computes with high probability a shortcut with the following guarantees:*

- *In CONGEST, the shortcut has quality* $\mathrm{poly}\big(\mathrm{SQ}(\overline{G})\big) \cdot \overline{n}^{o(1)}$ *and the algorithm terminates in* $\mathrm{poly}\big(\mathrm{SQ}(\overline{G})\big) \cdot \overline{n}^{o(1)}$ *rounds [27].*
- *In Supported-CONGEST, the shortcut has quality $\widetilde{O}(\mathrm{SQ}(\overline{G}))$ and the algorithm terminates in $\widetilde{O}(\mathrm{SQ}(\overline{G}))$ rounds [29].*

Universal Optimality. A distributed algorithm is said to be α-*universally optimal* if, on every network graph $\overline{G}$, it is α-competitive with the fastest correct algorithm on $\overline{G}$ [29]. Even the existence of such algorithms is not at all clear as it would seem possible that vastly different algorithms are required to leverage the structure of different networks. Nevertheless, a remarkable consequence of Theorem 8 is that in Supported-CONGEST we can design $\widetilde{O}(1)$-universally optimal algorithms for many fundamental optimization problems. Moreover, efficient shortcut construction is the only obstacle towards achieving these results in the full generality of CONGEST, which is an orthogonal issue and out of scope for this paper. Still, the aforementioned results are sufficient to design $\overline{n}^{o(1)}$-universally optimal algorithms on graphs that have shortcut quality $\mathrm{SQ}(\overline{G}) = \overline{n}^{o(1)}$.

Graphs Excluding Dense Minors. It turns out that the crucial issue of efficient shortcut construction can be resolved with a near-optimal, simple, and even deterministic algorithm for the rich class of graphs with *bounded minor density*. Formally, let us first recall the following definition.

▶ **Definition 9** (Minor Density). *The* minor density $\delta(G)$ *of a graph G is defined as*

$$\delta(G) = \max\left\{\frac{|E'|}{|V'|} : H = (V', E') \text{ is a minor of } G\right\}.$$

Any family of graphs closed under taking minors (such as planar graphs) has a constant minor density. For such graphs, [21] established efficient shortcut construction:

▶ **Theorem 10** ([21]). *Any graph G with hop-diameter D and minor density $\delta(G)$ admits shortcuts of quality $\widetilde{O}(\delta D)$, which can be constructed with high probability in $\widetilde{O}(\delta D)$ rounds of CONGEST.*

Some of our results apply for communication networks with *bounded treewidth*, so let us recall the following definition.

▶ **Definition 11** (Tree Decomposition and Treewidth). *A* tree decomposition *of a graph G is a tree T with tree-nodes $X_1, \ldots, X_k$, where each X_i is a subset of $V(G)$ satisfying the following properties:*

1. *$V = \bigcup_{i=1}^{k} X_i$;*
2. *For any node $u \in V(G)$, the tree-nodes containing u form a connected subtree of T;*
3. *For every edge $\{u, v\} \in E(G)$, there exists a tree-node X_i which contains both u and v.*

The width *w of the tree decomposition is defined as $w := \max_{i \in [k]} |X_i| - 1$. Moreover, the* treewidth $\mathrm{tw}(G)$ *of G is defined as the minimum of the width among all possible tree decompositions of G.*

Bounded-treewidth graphs inherit all of the nice properties guaranteed by Theorem 10, as implied by the following well-known fact.

▶ **Lemma 12.** *For any graph G, $\delta(G) \leq \mathrm{tw}(G)$.*

3 The Congested Part-Wise Aggregation Problem

This section is concerned with a *congested* generalization of the standard part-wise aggregation problem (Definition 4), formally introduced below.

▶ **Definition 13** (Congested Part-Wise Aggregation Problem). *Consider an n-node graph G with a collection of k subsets of nodes $P_1, \ldots, P_k \subseteq V(G)$ called* parts *such that each induced subgraph $G[P_i]$ is* connected *and each node $v \in V(G)$ is contained in at most $\rho \in \mathbb{Z}_{\geq 1}$ many parts, i.e., $\forall v \in V(G) \ \ |\{i : P_i \ni v\}| \leq \rho$. In the ρ-*congested part-wise aggregation *problem, each node v is given the following as input: for each part $P_i \ni v$ node v knows the part-ID i and an $O(\log n)$-bit part-specific value $\boldsymbol{x}_i(v)$. The goal is that, for each part P_i, all nodes in P_i learn the part-wise aggregate $\bigoplus_{w \in P_i} \boldsymbol{x}_i(w)$, where $\bigoplus$ is a pre-defined* aggregation function.

This congested generalization of the standard part-wise aggregation problem that we study in this section turns out to be a central ingredient in our refined Laplacian solver; this is further explained in Section 4. The remainder of this section is organized as follows. In Section 3.1 we establish near-optimal algorithms for solving congested part-wise aggregations in CONGEST, which is also the main focus of this section. We conclude by pointing out the construction for NCC in Section 3.2. Due to space limitations, all the omitted proofs are deferred to the full version of this paper.

3.1 Solving Congested Instances in the CONGEST Model

The first natural strategy for solving the ρ-congested part-wise aggregation problem of Definition 13 is through a reduction to $\mathrm{poly}(\rho)$ 1-congested instances. However, this approach immediately fails even if we allow $\rho = 2$. Indeed, there exist congested part-wise aggregation instances for which every two (distinct) parts share a common node, even when $\rho = 2$, leading to the following observation.

▶ **Observation 14.** *For an infinite family of values $\overline{n}$, there exists an $\overline{n}$-node planar graph $\overline{G}$ and a 2-congested part-wise aggregation instance $\mathcal{I}$ with $k = \Theta(\sqrt{\overline{n}})$ parts such that reducing $\mathcal{I}$ to the union of k' 1-congested part-wise aggregation instances on $\overline{G}$ requires $k' = \Omega(\sqrt{\overline{n}})$.*

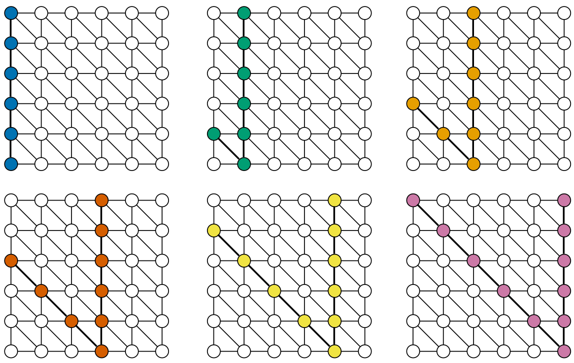

Figure 1 A 2-congested part-wise aggregation problem on a 6×6 grid (the instance immediately extends to an $\sqrt{n} \times \sqrt{n}$ topology). Different colors highlight different parts of the instance.

Such a pattern is illustrated in Figure 1. Indeed, in that 2-congested part-wise aggregation instance every two distinct parts share a common node. As a result, directly employing a 1-congested part-wise aggregation oracle is of little use since it would introduce an overhead depending on the number of parts. In light of this, we develop a more refined approach that leverages what we refer to as the *layered graph*.

3.1.1 The Layered Graph

Here we introduce the *layered graph* $\widehat{G}_\rho$, associated with the underlying graph $\overline{G}$. Then, we reduce any ρ-congested part-wise aggregation on $\overline{G}$ to a 1-congested instance on $\widehat{G}_{O(\rho)}$.

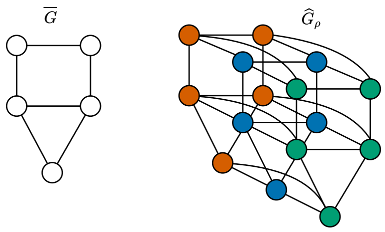

Figure 2 An example of a transformation from $\overline{G}$ to the layered graph $\widehat{G}_\rho$ with $\rho = 3$. We have highlighted with different colors different layers of the graph.

The Layered Graph. Consider an underlying network $\overline{G}$ and some $\rho \in \mathbb{Z}_{\geq 1}$, corresponding to the congestion parameter in Definition 13. The *layered graph* $\widehat{G}_\rho$ is constructed in the following way. First, we let $\widehat{G}_\rho$ be a disjoint union of ρ copies of $\overline{G}$ (called *layers*), namely $\overline{G}_1, \overline{G}_2, \dots, \overline{G}_\rho$. Each node $v \in V(\overline{G})$ is associated with its copies $v_1, v_2, \dots, v_\rho \in V(\widehat{G}_\rho)$. We also add an edge between each two copies that originate from the same node (i.e., we add a clique to $\widehat{G}_\rho$ on the set of copies associated with the same node $v \in V(\overline{G})$); this construction is illustrated in Figure 2. The layered graph induces a natural *projection* operation $\pi : V(\widehat{G}_\rho) \to V(\overline{G})$ which maps a copy v_i to its original node $v = \pi(v_i)$. Furthermore, we often talk about simulating $\widehat{G}_\rho$ in $\overline{G}$, by which we mean that each node v simulates – learns all the inputs and can generate all outputs – for its copies $v_1, \dots, v_\rho$. Throughout this paper, we will assume that $\rho = \text{poly}(\overline{n})$ so that any $O(\log n)$-bit message on $\widehat{G}_\rho$ can be sent within $O(1)$ rounds in $\overline{G}$; this also keeps the $\widetilde{O}$-notation well-defined.

The main goal of this section is to establish that the ρ-congested part-wise aggregation problem on $\overline{G}$ can be reduced to a 1-congested instance on $\widehat{G}_{O(\rho)}$, as formalized below.

▶ **Lemma 15** (Unrestricted Congested Part-Wise Aggregation). *Let $\overline{G}$ be an $\overline{n}$-node graph and let $\mathbb{Z}_{\geq 1} \ni \rho \leq \text{poly}(\overline{n})$. Suppose that any (1-congested) part-wise aggregation on $\widehat{G}_{O(\rho)}$ can be solved with a τ-round CONGEST algorithm on $\widehat{G}_{O(\rho)}$. Then, there exists an $\widetilde{O}(\rho \cdot \tau)$-round CONGEST algorithm on $\overline{G}$ that solves any ρ-congested part-wise aggregation instance on $\overline{G}$.*

Towards establishing this reduction, we first point out that any CONGEST algorithm on $\widehat{G}_\rho$ can be simulated with only a ρ multiplicative overhead in the round complexity.

▶ **Lemma 16** (Simulating $\widehat{G}_\rho$ in $\overline{G}$). *For any $\overline{G}$ and any $\mathbb{Z}_{\geq 1} \ni \rho \leq \text{poly}(\overline{n})$, we can simulate any τ-round CONGEST algorithm on $\widehat{G}_\rho$ with a $(\rho \cdot \tau)$-round CONGEST algorithm on $\overline{G}$.*

Furthermore, we will use a folklore result showing how to color a (multi)graph of maximum degree Δ in $O(\Delta)$ colors in $O(\log n)$ rounds of CONGEST. By multigraph here we simply mean that there can be multiple parallel edges between the same pair of nodes, and every such edge can carry an independent message per round.

▶ **Lemma 17** (Folklore, [30]). *Given a (multi)graph G with n nodes and maximum degree $\Delta \leq \text{poly}(n)$, there exists a randomized CONGEST algorithm that colors the edges of G with $O(\Delta)$ colors and completes in $O(\log n)$ rounds, with high probability. The coloring is proper, i.e., two edges that share an endpoint are assigned a different color.*

Using this lemma, we first prove a version of our main reduction (Lemma 15), but with the slight twist that we restrict each part of the ρ-congested part-wise aggregation problem to be a simple path.

▶ **Lemma 18** (Path-Restricted Congested Part-Wise Aggregation). *Let $\overline{G}$ be an $\overline{n}$-node graph and let $\mathbb{Z}_{\geq 1} \ni \rho \leq \text{poly}(\overline{n})$. Suppose that there exists a τ-round CONGEST algorithm solving the (1-congested) part-wise aggregation on $\widehat{G}_{O(\rho)}$. Then, there exists an $\widetilde{O}(\rho \cdot \tau)$-round CONGEST algorithm on $\overline{G}$ that solves any ρ-congested part-wise aggregation instance on $\overline{G}$ when each part is* restricted to be a simple path[5] *(nodes are not repeated in simple paths).*

Finally, our reduction in Lemma 15 follows by reformulating [29, Lemma 7.2].

3.1.2 Treewidth-Bounded Graphs

Here we leverage the reduction we established in Lemma 15 to obtain a simple algorithm for solving the congested part-wise aggregation problem in treewidth-bounded graphs. The crucial observation is that the treewidth of the layered graph can only grow by a factor of ρ compared to the treewidth of the underlying graph, as we show below.

▶ **Lemma 19.** *If the treewidth of $\overline{G}$ is $\text{tw}(\overline{G})$, then $\text{tw}(\widehat{G}_\rho) \leq \rho\, \text{tw}(\overline{G}) + \rho - 1$.*

Combining this guarantee with Lemmas 12 and 15 and Theorem 10, we obtain the following immediate consequence.

▶ **Corollary 20.** *Let $\overline{G}$ be an $\overline{n}$-node communication network of diameter at most D and treewidth $\text{tw}(\overline{G})$. Then, we can solve with high probability any ρ-congested part-wise aggregation problem in $\overline{G}$ within $\widetilde{O}(\rho^2 \cdot \text{tw}(\overline{G}) \cdot D)$ rounds of CONGEST.*

---

[5] I.e., there exists a simple path traversing all the nodes of the part, and each node knows the corresponding incident edges of that path.

Minor Density in the Layered Graph. In light of Lemma 19, a natural question is whether an analogous bound holds with respect to the minor density of the underlying graph; i.e., whether $\delta(\widehat{G}_\rho) = \text{poly}(\rho)\delta(G)$. Unfortunately, this is not possible, as illustrated in Figure 3.

▶ **Observation 21.** *There exists an n-node graph G with minor density $\delta(G) = \widetilde{O}(1)$, but its 2-layered version $\widehat{G}_2$ has minor density $\delta(\widehat{G}_2) = \Omega(\sqrt{n})$.*

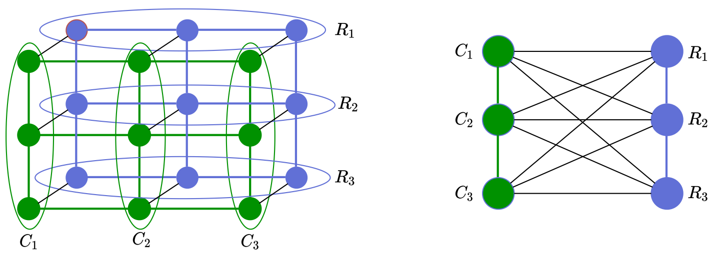

Figure 3 The layered graph $\widehat{G}_\rho$ of a 3×3 grid with every node having congestion $\rho = 2$ (left), and a minor of $\widehat{G}_\rho$ induced by the connected components $\{C_1, C_2, C_3, R_1, R_2, R_3\}$ (right).

3.1.3 General Graphs

We conclude with our main result of Section 3.1: a near-optimal distributed algorithm for solving the ρ-congested part-wise aggregation problem in general graphs. In light of our reduction in Lemma 15, the technical crux is to control the degradation in the shortcut quality incurred by the transformation into the layered graph. Surprisingly, we show that the shortcut quality of $\widehat{G}_\rho$ does not increase by more than a polylogarithmic factor even when the number of layers is polynomial:

▶ **Theorem 22.** *For any $\overline{n}$-node graph $\overline{G}$ and any $\mathbb{Z}_{\geq 1} \ni \rho \leq \text{poly}(\overline{n})$, we have that $\text{SQ}(\widehat{G}_\rho) = \widetilde{O}(\text{SQ}(\overline{G}))$.*

This theorem improves over our previous result for treewidth-bounded graphs (Lemma 19) since the latter guarantee inevitably induces a linear factor of ρ in the shortcut quality of $\widehat{G}_\rho$. While this will not affect the asymptotic performance of the Laplacian solver, this improvement might prove to be important for future applications. Assuming that we have shown Theorem 22, we can then utilize the efficient shortcut constructions given in Theorem 8 to solve ρ-congested part-wise aggregations on any graph.

▶ **Corollary 23.** *There exists a randomized distributed algorithm that, for any $\overline{n}$-node graph $\overline{G}$ and $\rho \in \mathbb{Z}_{\geq 1} \leq \text{poly}(\overline{n})$, solves with high probability any ρ-congested part-wise aggregation instance on $\overline{G}$ with the following guarantees:*

- *In the CONGEST, the algorithm terminates in at most $\rho \cdot \text{poly}\big(\text{SQ}(\overline{G})\big) \cdot \overline{n}^{o(1)}$ rounds.*
- *In the CONGEST model on graphs with minor density δ, it requires $\widetilde{O}(\rho \cdot \delta \cdot D)$ rounds.*
- *In the Supported-CONGEST, the algorithm terminates in $\widetilde{O}(\rho \cdot \text{SQ}(\overline{G}))$ rounds.*

The rest of this subsection is dedicated to the proof of Theorem 22. First, to argue about the shortcut quality of the layered graph, we need to develop several generalized notions of node connectivity.

Pair Node Connectivity. Given a (multi)set of source-sink pairs $\mathcal{P} = \{(s_i, t_i)\}_{i=1}^k$ in G, we say that $\mathcal{P}$ has pair node connectivity ρ if there exist paths $P_1, \ldots, P_k$, with s_i and t_i being the endpoints of each P_i, such that every node $v \in V(G)$ is contained in at most ρ many paths, i.e., for all v we have $|\{i : V(P_i) \ni v\}| \leq \rho$. If $\mathcal{P}$ has pair node connectivity 1 we say that they are pair *node-disjointly connectable.*

Any-to-Any Node Connectivity. Suppose that we are given multisets of k *sources* $S = \{s_1, \ldots, s_k\}$ and k *sinks* $T = \{t_1 \ldots, t_k\}$. We say that (S, T) have any-to-any node connectivity ρ if there is a permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ such that the pairs $\{(s_i, t_{\pi(i)})\}_{i=1}^k$ have pair node connectivity ρ. If (S, T) have any-to-any node connectivity 1 we say they are any-to-any *node-disjointly connectable.*

The following decomposition lemma states that two sets with any-to-any node connectivity ρ can be decomposed into $\widetilde{O}(\rho)$ many pairs of subsets that are any-to-any node-disjointly connectable.

▶ **Lemma 24.** *Given a graph G, suppose we are given any two multisets of nodes $S \subseteq V(G)$ and $T \subseteq V(G)$ of size $k := |S| = |T|$ that have any-to-any node connectivity ρ. Then, we can partition $S = S_1 \uplus S_2 \uplus \ldots \uplus S_{O(\rho \log k)}$ and $T = T_1 \uplus T_2 \uplus \ldots T_{O(\rho \log k)}$ such that $|S_i| = |T_i|$ and (S_i, T_i) are any-to-any node-disjointly connectable.*

Next, we introduce two communication tasks that will be useful for characterizing the shortcut quality.

Multiple-Unicast Problem. Suppose that we are given k source-sink pairs $\mathcal{P} = \{(s_i, t_i)\}_{i=1}^k$. The goal is to find the smallest possible *completion time* τ such that there are k paths $P_1, \ldots, P_k$ for which (1) the endpoints of each P_i are exactly s_i and t_i; (2) the dilation is τ, i.e., each path P_i has at most τ hops; and (3) the congestion is τ, i.e., each edge $e \in E(G)$ is contained in at most τ many paths.

Any-to-Any-Cast Problem. Suppose we are given k *sources* $S = \{s_1, \ldots, s_k\}$ and k *sinks* $T = \{t_1 \ldots, t_k\}$. The goal is to find the smallest *completion time* τ such that there exists a permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ for which the multiple-unicast problem on $\{(s_i, t_{\pi(i)})\}_{i=1}^k$ has a completion time of at most τ.

Finally, we now recall (a reinterpretation of) a result characterizing shortcut quality from [28, 29]. Shortcut quality was originally defined as the smallest completion-time of the worst-case generalized (with respect to parts) multiple-unicast (i.e., multi-commodity) problem over a *pair* node-disjointly connectable instance (Definition 7). Using recent network coding gap results, we can equivalently express shortcut quality as the smallest completion-time of the worst-case any-to-any-cast (i.e., single-commodity) problem over sources and sinks that are *any-to-any* node-disjointly connectable. The formal statement follows.

▶ **Theorem 25** ([28, 29])**.** *Consider any graph G and let τ be the worst-case completion time of any-to-any-cast problems taken over all any-to-any node-disjointly connectable sets $(S \subseteq V(G), T \subseteq V(G))$. Then, $\tau = \widetilde{\Theta}(\mathrm{SQ}(G))$.*

Finally, combining all of the previous ingredients, we are ready to show Theorem 22.

Proof of Theorem 22. Let $S \subseteq V(\widehat{G}_\rho)$ and $T \subseteq V(\widehat{G}_\rho)$ be any-to-any node-disjointly connectable sets such that the completion time of any-to-any-cast between S and T is $\widetilde{\Theta}(\mathrm{SQ}(\widehat{G}_\rho))$ (Theorem 25). Let $k := |S| = |T|$, and suppose that $S' := \biguplus_{s \in S}\{\pi(s)\} \subseteq V(\overline{G})$

and $T' := \biguplus_{t \in T}\{\pi(t)\} \subseteq V(\overline{G})$ are the multisets induced by projecting S and T to $\overline{G}$, respectively. By construction of $\widehat{G}_\rho$, S' and T' have any-to-any node connectivity ρ; to see this, consider the witness paths disjointly connecting them in $\widehat{G}_\rho$ and project them to $\overline{G}$. Therefore, we can partition $S' = S'_1 \uplus \ldots \uplus S'_{O(\rho \log k)}$ and $T' = T'_1 \uplus \ldots \uplus T'_{O(\log k)}$ such that $|S'_i| = |T'_i|$ and (S'_i, T'_i) are any-to-any node-disjointly connectable in $\overline{G}$ (Lemma 24).

By definition of shortcut quality, for each $i \in \{1, \ldots, O(\rho \log k)\}$ there exists a set of paths $(P^i_j)_{j=1}^{|S'_i|}$ in $\overline{G}$ between S'_i and T'_i of quality (i.e., both congestion and dilation) at most $\mathrm{SQ}(\overline{G})$. Then, we inject the first $O(\log k)$ collections of paths $(P^1_j)_j, (P^2_j)_j, \ldots, (P^{O(\log k)}_j)_j$ to the first layer $\overline{G}_1$ of $\widehat{G}_\rho$; the second $O(\log k)$ collections to the second layer $\overline{G}^2$, and so on, until we finally inject the last $O(\log k)$ collections to the last layer $\overline{G}_\rho$. Note that only the paths on the same layer interact, so both the congestion and dilation after injecting all paths into $\widehat{G}_\rho$ is $O(\mathrm{SQ}(\overline{G}) \log k)$. Hence, the same applies for the shortcut quality. Finally, to solve the any-to-any-cast problem on S and T one might need to add an between-layer edge at the beginning and at the end since each injected path is restricted to some adversarially chosen layer. However, this only increases the congestion and dilation by $O(1)$. Hence, the completion time of any-to-any-cast between S and T is $\widetilde{O}(\mathrm{SQ}(\overline{G}))$, implying that $\mathrm{SQ}(\widehat{G}_\rho) = \widetilde{O}(\mathrm{SQ}(\overline{G}))$. ◀

3.2 The NCC Model

We next turn our attention to the NCC model. We observe that the ρ-congested part-wise aggregation problem admits a solution in $\mathrm{poly}(\rho, \log \overline{n})$ rounds of NCC. This is established after appropriately translating the communication primitives established for NCC in [2].

▶ **Lemma 26.** *Let $\overline{G}$ be an $\overline{n}$-node graph. Then, we can solve with high probability any ρ-congested part-wise aggregation problem on $\overline{G}$ after $O(\rho + \log \overline{n})$ rounds of NCC.*

4 Almost Universally Optimal Laplacian Solvers

In this section, we relate the congested part-wise aggregation problem we studied in the previous section with the Laplacian solver in [18]. To present a unifying analysis for both CONGEST and HYBRID, as well as for future applications and extensions, we analyze the distributed Laplacian solver under the following hypothesis.

▶ **Assumption 27.** *Consider a model of computation which incorporates CONGEST. We assume that we can solve with high probability any ρ-congested part-wise aggregation problem in $Q(\rho) = O(\rho^c Q(1))$ rounds, for some universal constant $c \geq 1$.*

One of our crucial observations is that the performance of the Laplacian solver of [18] can be parameterized in terms of the complexity of the congested part-wise aggregation problem. Indeed, we revisit and refine the main building blocks of their solver, leading to the following.

▶ **Theorem 28.** *Consider a weighted $\overline{n}$-node graph $\overline{G}$ for which Assumption 27 holds for some $Q(\rho) = O(\rho^c \mathcal{Q})$, where c is a universal constant and $\mathcal{Q} = \mathcal{Q}(\overline{G})$ is some parameter. Then, we can solve any Laplacian system after $\overline{n}^{o(1)} \mathcal{Q} \log(1/\varepsilon)$ rounds.*

Combining this theorem with Corollary 23 and Lemma 26 yields the following immediate consequences.

▶ **Theorem 2.** *Consider any n-node graph G with shortcut quality $\mathrm{SQ}(G)$ and hop-diameter D. There exists a distributed Laplacian solver with error $\varepsilon > 0$ with the following guarantees:*

- *In the Supported-CONGEST model, it requires $n^{o(1)}\,\mathrm{SQ}(G)\log(1/\varepsilon)$ rounds.*
- *In the CONGEST model, it requires $n^{o(1)}\,\mathrm{poly}(\mathrm{SQ}(G))\log(1/\varepsilon)$ rounds.*
- *In the CONGEST model on graphs with minor density δ, it requires $n^{o(1)}\delta D\log(1/\varepsilon)$ rounds.*

▶ **Theorem 3.** *Consider any n-node graph. There exists a distributed Laplacian solver in the HYBRID model with round complexity $n^{o(1)}\log(1/\varepsilon)$, where $\varepsilon > 0$ is the error of the solver.*

Lower Bound in Supported-CONGEST. Finally, we complement our positive results with an almost-matching lower bound on any graph $\overline{G}$, applicable even under the Supported-CONGEST model, thereby establishing universal optimality up to an $\overline{n}^{o(1)}$ factor. Our reduction leverages the refined hardness result established in [29] for the *spanning connected subgraph* problem [13]. In this problem a subgraph $\overline{H}$ of $\overline{G}$ is specified with nodes knowing all of the incident edges belonging to $\overline{H}$. The goal is to let every node learn whether $\overline{H}$ is connected and spans the entire network.

▶ **Theorem 29** ([29])**.** *Let $\mathcal{A}$ be any algorithm which is always correct with probability*[6] *at least $\frac{2}{3}$ for the spanning connected subgraph problem, and $T(\overline{G}) = \max_{\mathcal{I}} T_{\mathcal{A}}(\mathcal{I};\overline{G})$ be the worst-case round-complexity of $\mathcal{A}$ under $\overline{G}$. Then, $T(\overline{G}) = \widetilde{\Omega}(\mathrm{SQ}(\overline{G}))$.*

In this context, we show that a Laplacian solver can be leveraged to solve the spanning connected subgraph problem, leading to the following lower bound.

▶ **Theorem 1.** *Consider a graph $\overline{G}$ with shortcut quality $\mathrm{SQ}(\overline{G})$. Then, solving a Laplacian system on $\overline{G}$ with $\varepsilon \le \frac{1}{2}$ requires $\widetilde{\Omega}(\mathrm{SQ}(\overline{G}))$ rounds in both CONGEST and Supported-CONGEST models.*

5 Conclusions

In this paper, we have established almost universally optimal Laplacian solvers for both the (Supported-)CONGEST and the HYBRID model. One of our main technical contributions was to introduce and study a congested generalization of the standard part-wise aggregation problem, which we believe may find further applications beyond the Laplacian paradigm in the future. For example, one candidate problem would be to refine the distributed algorithm for max-flow due to [22].

We also hope that our accelerated Laplacian solvers will be used as a basic primitive for obtaining improved distributed algorithms for other fundamental optimization problems as well. For example, our results directly imply an exact $O(m^{1/2+o(1)}\,\mathrm{SQ}(G))$ algorithm for the max-flow problem using a standard reduction [12]. On the other hand, there are substantial obstacles in obtaining further improvements and strengthening the max-flow algorithm of [18], which in turn relies on the more recent techniques of [40, 10], as that would require solving exactly the single-source shortest paths problem in almost $\mathrm{SQ}(G)$ rounds.

[6] Note that [29] only proved this for always-correct algorithms with probability 1, but the extension we claim here follows readily from their argument.

References

1 Ioannis Anagnostides and Themis Gouleakis. Deterministic distributed algorithms and lower bounds in the hybrid model. In *35th International Symposium on Distributed Computing, DISC 2021*, volume 209 of *LIPIcs*, pages 5:1–5:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.DISC.2021.5.

2 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.

3 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*, pages 1280–1299. SIAM, 2020. doi:10.1137/1.9781611975994.78.

4 Kyriakos Axiotis, Aleksander Madry, and Adrian Vladu. Circulation control for faster minimum cost flow in unit-capacity graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 93–104. IEEE, 2020. doi:10.1109/FOCS46700.2020.00018.

5 Kyriakos Axiotis, Aleksander Madry, and Adrian Vladu. Faster sparse minimum cost flow by electrical flow localization. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 528–539. IEEE, 2021. doi:10.1109/FOCS52979.2021.00059.

6 Guy E. Blelloch, Anupam Gupta, Ioannis Koutis, Gary L. Miller, Richard Peng, and Kanat Tangwongsan. Nearly-linear work parallel SDD solvers, low-diameter decomposition, and low-stretch subgraphs. *Theory Comput. Syst.*, 55(3):521–554, 2014. doi:10.1007/s00224-013-9444-5.

7 Keren Censor-Hillel, Dean Leitersdorf, and Volodymyr Polosukhin. Distance computations in the hybrid network model via oracle simulations. In *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021*, volume 187 of *LIPIcs*, pages 21:1–21:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.21.

8 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*, pages 151–161. ACM, 2021. doi:10.1145/3409964.3461798.

9 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.

10 Michael B. Cohen, Aleksander Madry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in õ $(m^{10/7} \log W)$ time (extended abstract). In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 752–771. SIAM, 2017. doi:10.1137/1.9781611974782.48.

11 Sam Coy, Artur Czumaj, Michael Feldmann, Kristian Hinnenthal, Fabian Kuhn, Christian Scheideler, Philipp Schneider, and Martijn Struijs. Near-shortest path routing in hybrid communication networks, 2022. arXiv:2202.08008.

12 Samuel I. Daitch and Daniel A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008*, pages 451–460. ACM, 2008. doi:10.1145/1374376.1374441.

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, pages 363–372, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993686.

14 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.

15 Michael Elkin. Unconditional lower bounds on the time-approximation tradeoffs for the distributed minimum spanning tree problem. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 331–340, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1007352.1007407.

16 Michael Feldmann, Kristian Hinnenthal, and Christian Scheideler. Fast hybrid network algorithms for shortest paths in sparse graphs. In *24th International Conference on Principles of Distributed Systems, OPODIS 2020*, volume 184 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.OPODIS.2020.31.

17 Sebastian Forster and Tijn de Vos. The laplacian paradigm in the broadcast congested clique. In *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, 2022*, pages 335–344. ACM, 2022. doi:10.1145/3519270.3538436.

18 Sebastian Forster, Gramoz Goranci, Yang P. Liu, Richard Peng, Xiaorui Sun, and Mingquan Ye. Minor sparsifiers and the distributed laplacian paradigm. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 989–999. IEEE, 2021. doi:10.1109/FOCS52979.2021.00099.

19 Mohsen Ghaffari. *Near-Optimal Scheduling of Distributed Algorithms*, pages 3–12. Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2767386.2767417.

20 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.

21 Mohsen Ghaffari and Bernhard Haeupler. Low-congestion shortcuts for graphs excluding dense minors. In *PODC '21: ACM Symposium on Principles of Distributed Computing, 2021*, pages 213–221. ACM, 2021. doi:10.1145/3465084.3467935.

22 Mohsen Ghaffari, Andreas Karrenbauer, Fabian Kuhn, Christoph Lenzen, and Boaz Patt-Shamir. Near-optimal distributed maximum flow: Extended abstract. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015*, pages 81–90. ACM, 2015. doi:10.1145/2767386.2767440.

23 Mohsen Ghaffari and Goran Zuzic. Universally-optimal distributed exact min-cut. In *PODC '22: ACM Symposium on Principles of Distributed Computing, 2022*, pages 281–291. ACM, 2022. doi:10.1145/3519270.3538429.

24 Thorsten Götte, Kristian Hinnenthal, Christian Scheideler, and Julian Werthmann. Time-optimal construction of overlay networks. In *PODC '21: ACM Symposium on Principles of Distributed Computing*, pages 457–468. ACM, 2021. doi:10.1145/3465084.3467932.

25 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Low-congestion shortcuts without embedding. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016*, pages 451–460. ACM, 2016. doi:10.1145/2933057.2933112.

26 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Near-optimal low-congestion shortcuts on bounded parameter graphs. In *Distributed Computing – 30th International Symposium, DISC 2016*, volume 9888 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 2016. doi:10.1007/978-3-662-53426-7_12.

27 Bernhard Haeupler, Harald Räcke, and Mohsen Ghaffari. Hop-constrained expander decompositions, oblivious routing, and distributed universal optimality. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022*, pages 1325–1338. ACM, 2022. doi:10.1145/3519935.3520026.

28 Bernhard Haeupler, David Wajc, and Goran Zuzic. Network coding gaps for completion times of multiple unicasts. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 494–505. IEEE, 2020. doi:10.1109/FOCS46700.2020.00053.

29 Bernhard Haeupler, David Wajc, and Goran Zuzic. Universally-optimal distributed algorithms for known topologies. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1166–1179. ACM, 2021. doi:10.1145/3406325.3451081.

30 Öjvind Johansson. Simple distributed $\delta + 1$-coloring of graphs. *Information Processing Letters*, 70(5):229–232, 1999.

31 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.

32 Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 217–226. SIAM, 2014. doi:10.1137/1.9781611973402.16.

33 Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, 2013*, pages 911–920. ACM, 2013. doi:10.1145/2488608.2488724.

34 Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD linear systems. *SIAM J. Comput.*, 43(1):337–354, 2014. doi:10.1137/110845914.

35 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, pages 109–118. Association for Computing Machinery, 2020.

36 Fabian Kuhn and Philipp Schneider. Routing schemes and distance oracles in the hybrid model, 2022. arXiv:2202.06624.

37 Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians – fast, sparse, and simple. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 573–582. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.68.

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

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, 2003*, pages 94–100. ACM, 2003. doi:10.1145/777412.777428.

40 Aleksander Madry. Computing maximum flow with augmenting electrical flows. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 593–602. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.70.

41 D. Peleg and V. Rubinovich. A near-tight lower bound on the time complexity of distributed mst construction. In *40th Annual Symposium on Foundations of Computer Science*, pages 253–261, 1999. doi:10.1109/SFFCS.1999.814597.

42 David Peleg. *Distributed computing: a locality-sensitive approach.* SIAM, 2000.

43 Richard Peng. Approximate undirected maximum flows in $O(m\text{polylog}(n))$ time. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*, pages 1862–1867. SIAM, 2016. doi:10.1137/1.9781611974331.ch130.

44 Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014*, pages 333–342. ACM, 2014.

45 Václav Rozhon, Christoph Grunau, Bernhard Haeupler, Goran Zuzic, and Jason Li. Undirected $(1+\epsilon)$-shortest paths via minor-aggregates: near-optimal deterministic parallel and distributed algorithms. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, 2022*, pages 478–487. ACM, 2022. doi:10.1145/3519935.3520074.

46 Stefan Schmid and Jukka Suomela. Exploiting locality in distributed sdn control. *HotSDN 2013 – Proceedings of the 2013 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, pages 121–126, 2013. `doi:10.1145/2491185.2491198`.

47 Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, 2014. `doi:10.1137/090771430`.

48 Jan van den Brand, Yin Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 919–930. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00090`.

49 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. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 327–338. Association for Computing Machinery, 2010. `doi:10.1145/1851182.1851222`.

50 Goran Zuzic, Gramoz Goranci, Mingquan Ye, Bernhard Haeupler, and Xiaorui Sun. Universally-optimal distributed shortest paths and transshipment via graph-based ℓ_1-oblivious routing. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2549–2579. SIAM, 2022. `doi:10.1137/1.9781611977073.100`.