Distance Computations in the Hybrid Network Model via Oracle Simulations

Keren Censor-Hillel
Technion – Israel Institute of Technology, Haifa, Israel

Dean Leitersdorf
Technion – Israel Institute of Technology, Haifa, Israel

Volodymyr Polosukhin
Technion – Israel Institute of Technology, Haifa, Israel

Abstract

The Hybrid network model was introduced in [Augustine et al., SODA '20] for laying down a theoretical foundation for networks which combine two possible modes of communication: One mode allows high-bandwidth communication with neighboring nodes, and the other allows low-bandwidth communication over few long-range connections at a time. This fundamentally abstracts networks such as hybrid data centers, and class-based software-defined networks.

Our technical contribution is a *density-aware* approach that allows us to simulate a set of *oracles* for an overlay skeleton graph over a Hybrid network.

As applications of our oracle simulations, with additional machinery that we provide, we derive fast algorithms for fundamental distance-related tasks. One of our core contributions is an algorithm in the Hybrid model for computing *exact* weighted shortest paths from $\tilde{O}(n^{1/3})$ sources which completes in $\tilde{O}(n^{1/3})$ rounds w.h.p. This improves, in both the runtime and the number of sources, upon the algorithm of [Kuhn and Schneider, PODC '20], which computes shortest paths from a single source in $\tilde{O}(n^{2/5})$ rounds w.h.p.

We additionally show a 2-approximation for weighted diameter and a $(1+\epsilon)$-approximation for unweighted diameter, both in $\tilde{O}(n^{1/3})$ rounds w.h.p., which is comparable to the $\tilde{\Omega}(n^{1/3})$ lower bound of [Kuhn and Schneider, PODC '20] for a $(2-\epsilon)$-approximation for weighted diameter and an exact unweighted diameter. We also provide fast distance *approximations* from multiple sources and fast approximations for eccentricities.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed graph algorithms, Hybrid network model, Distance computations

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.21

Related Version *Full Version*: `https://arxiv.org/abs/2010.13831`

Funding This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 755839-ERC-BANDWIDTH.

Acknowledgements The authors would like to thank Michal Dory and Yuval Efron for various helpful conversations. We also thank Fabian Kuhn for sharing a preprint of [30] with us.

1 Introduction

The Hybrid model of computation was recently introduced by Augustine et al. [9], for abstracting networks which can utilize both high-bandwidth local communication links, as well as very few low-bandwidth global communication channels. This model abstracts fundamental systems, such as a combination of device-to-device communication with cellular networks (e.g. 5G) [7], wired data centers with wireless links (hybrid DCNs) [17, 26, 29, 41], and Class-Based Hybrid software-defined networks (SDNs) [40].

The pioneering works of [9, 20, 30] provide fast algorithms for various distance-related tasks in the Hybrid model. At the heart of many of these algorithms lies a framework for using skeleton overlay graphs for computation and approximation of distances, as well as for fast communication.

In this paper, we define and show how to efficiently simulate *oracles* over skeleton graphs in the Hybrid model. Using additional machinery that we provide, the implications of our simulations are faster algorithms for distance computations in the Hybrid model. Our oracle models could also be of independent interest, presenting a generic approach which can potentially be applied elsewhere.

1.1 Our Contributions

The Hybrid model, which we consider in this paper, abstracts a synchronous network of nodes, where in each round, every node can send and receive arbitrarily many messages of $O(\log n)$ bits to/from each of its neighbors (over *local edges*) and an additional $O(\log n)$ messages, in total, to/from any other nodes in the network (over *global edges*). The high bandwidth permissible over the local edges is aligned with previous research in the Hybrid model as well as with the extensively studied LOCAL distributed model.

The main idea which our results hinge upon is exploiting an inherent *asymmetry* in the Hybrid model which we observe. This asymmetry allows nodes with *dense* neighborhoods to effectively receive significantly more information. To see this, note that every node can use the global edges of the Hybrid model to send and receive a limited number of messages every round. However, since in the next round a node can communicate with its neighborhood in the graph using local edges and share with them the information which it received, this implies that a node is able to learn much more information from the entire graph if it is in a more dense neighborhood. Thus, *density-aware* algorithms are inherently useful in the Hybrid model.

We use this asymmetry to simulate Oracle and Tiered Oracles models. To capture what an oracle can do, we introduce the Oracle and Tiered Oracles models. Roughly speaking, in the Oracle model there is a node ℓ, the *oracle*, which can receive $\deg(v)$ messages from each node v, within *a single round*. In particular, this implies that the oracle can learn the entire communication graph.

We cannot afford to directly simulate the Oracle model as it requires too much communication in the Hybrid model. Instead, we simulate the Oracle model over a *skeleton graph*. Roughly speaking, given an input graph G, a skeleton graph is a subset of the nodes of G, connected by virtual edges which represent paths in G. Skeleton graphs are a common tool for distance computations in various models [9, 30, 33, 37, 39], and it has been shown in [9, 30] that some distances on the skeleton graph can be efficiently extended to distances on the entire graph in the Hybrid model.

As a warm-up, we show that a single round of the Oracle model over certain skeleton graphs can be simulated in $\tilde{O}(n^{1/3})$ rounds, w.h.p.[1], in the Hybrid model. Combining this with a simple, constant-round algorithm for exact weighted *single source shortest paths (SSSP)* which we show in the Oracle model, gives the following theorem.

▶ **Theorem 1** (Exact SSSP). *Given a weighted graph $G = (V, E)$, there is an algorithm in the* Hybrid *model that computes an exact weighted SSSP in $\tilde{O}(n^{1/3})$ rounds w.h.p.*

[1] As common, w.h.p. indicates a probability that is at least $1 - n^{-c}$, for some constant $c > 1$.

This result should be compared with the previous state-of-the-art algorithms for exact weighted SSSP in $\tilde{O}(n^{2/5})$ rounds [30], and in $\tilde{O}(\sqrt{SPD})$ rounds [9], where SPD is the length of the shortest path diameter. Further, it improves upon the $\tilde{O}(n^{1/3}/\epsilon^6)$ round algorithm for a $(1+\epsilon)$-approximation of weighted SSSP [9], in both the runtime and in being exact. We stress that this is a warm up, and later on we extend this result to shortest path distances from $O(n^{1/3})$ sources, instead of a single source, in the same round complexity of $\tilde{O}(n^{1/3})$.

It is well known that one can approximate the diameter using a solution to SSSP, and so as a byproduct we get the following result.

▶ **Corollary 2** (2-Approx. Weighted Diameter). *There is an algorithm in the* Hybrid *model that computes a 2-approximation of weighted diameter in $\tilde{O}(n^{1/3})$ rounds w.h.p.*

Notably, $\tilde{\Omega}(n^{1/3})$ rounds are necessary for a $(2-\epsilon)$-approximation for weighted diameter [30]. Our algorithm in Corollary 2 thus raises the interesting open question of whether one can go below this complexity for a 2-approximation.

While efficiently simulating an oracle is powerful, it still does not exploit the full capacity of the Hybrid model. This observation brings us to enhance the Oracle model and introduce the Tiered Oracles model, which consists of multiple oracles with varying abilities. In a nutshell, in the Tiered Oracles model, in each round every node v can send (the same) $\deg(v)$ messages to all nodes u with $\deg(u) \geq \deg(v)/2$. This basically means that nodes are bucketed according to degrees and each node is an oracle for all nodes in buckets below it. One can notice that the node with the highest degree in the graph is equivalent to the oracle in the Oracle model, but here, the other nodes in the graph also have some *partial* oracle capabilities.

We show how to simulate the Tiered Oracles model over skeleton graphs in the Hybrid model within $\tilde{O}(n^{1/3})$ rounds. Subsequently, we present an algorithm which solves all pairs shortest paths (APSP) using one round of the Tiered Oracles model and $O(\log n)$ rounds of the Congested Clique model[2]. We then utilize our Tiered Oracles model simulation, along with a previously known simulation of the Congested Clique model from [30], to simulate the APSP algorithm over skeleton graphs in the Hybrid model. Our efficient computation of APSP over a skeleton graph in the Hybrid model then leads to computing multi-source shortest paths from *random* sources in the Hybrid model.

Shortest paths from random sources is a crucial stepping stone for our later results. We show that computing shortest path distances from random sources to the entire graph, allows us to subsequently obtain fast algorithms for other distance problems. We call the problem of computing distances from sources sampled with probability n^{x-1} i.i.d n^x-RSSP.

▶ **Theorem 3** (n^x-RSSP). *Given a graph $G=(V,E)$, $0<x<1$, and a set of nodes M sampled independently with probability n^{x-1}, there is an algorithm in the* Hybrid *model that ensures that every $v \in V$ knows the exact, weighted distance from itself to every node in M within $\tilde{O}(n^{1/3}+n^{2x-1})$ rounds w.h.p.*

We complement Theorem 3 with a lower bound, following the lines of [9, 30], for approximating distances from many random sources, to any reasonable approximation factor, which tightly matches the upper bound when $x = 2/3$.

▶ **Theorem 4** (Lower Bound Exact Shortest Paths, Sources Sampled i.i.d.). *Let $p = \Omega(\log n/n)$ and $\alpha < \sqrt{n/p} \cdot \log(n)/2$. Any α-approximate unweighted algorithm from random sources sampled independently with probability p in the* Hybrid *network model takes $\Omega\left(\sqrt{p \cdot n}/\log n\right)$ rounds w.h.p.*

[2] The Congested Clique is a synchronous distributed model where every two nodes in the graph can exchange messages of $O(\log n)$ bits in every round.

We leverage our *near-optimal* (tight up to polylogarithmic factors) algorithm for shortest paths from a set M of $\tilde{O}(n^{2/3})$ random sources in order to obtain exact weighted shortest paths from any *given* set U of $O(n^{1/3})$ sources. We achieve this by adapting the behavior of the given fixed source nodes to the density of their neighborhoods, as follows. A source node $s \in U$ in a sparse neighborhood broadcasts the distances to all the random source nodes from M it sees in its neighborhood. A source node $s \in U$ in a dense neighborhood takes control of one of the random sources in M in its neighborhood and uses it as a proxy in order to communicate enough information to all the other nodes in the graph so that they could determine their distances from s. We remark that this proxy approach is a *key insight* which we later encapsulate as a general tool in the Hybrid model and may potentially be of independent interest. Our approach gives the following theorem.

▶ **Theorem 5** (Exact $n^{1/3}$ Sources Shortest Paths). *Given a weighted graph $G = (V, E)$, and a set of sources U, such that $|U| = O(n^{1/3})$, there exists an algorithm, at the end of which each $v \in V$ knows its distance from every $s \in U$, which runs in $\tilde{O}(n^{1/3})$ rounds w.h.p.*

Theorem 5 raises an interesting open question of whether the complexity of SSSP in the Hybrid model is below that of computing shortest paths from $\tilde{O}(n^{1/3})$ sources.

We also exploit our aforementioned solution for computing APSP on the skeleton graph to obtain approximate distances from a larger set of given sources (n^x-SSP), as follows.

▶ **Theorem 6** (Approximate Multiple Source Shortest Paths). *Given a graph $G = (V, E)$, a set of sources U, where $|U| = \tilde{\Theta}(n^y)$ for some constant $0 < y < 1$, and a value $0 < \epsilon$, there is an algorithm in the* Hybrid *model which ensures that every node $v \in V$ knows an approximation to its distance from every $s \in U$, where the approximation factor is $(1+\epsilon)$ if G is unweighted and 3 if G is weighted. The complexity of the algorithm is $\tilde{O}(n^{1/3}/\epsilon + n^{y/2})$ rounds, w.h.p.*

This result improves both in round complexity and approximation factors upon the previous results in [30]. The reason for this is that we compute APSP over skeleton graphs using the efficient, exact algorithm from the Tiered Oracles oracle model, while [30] simulate the slower, approximate algorithms of [12, 14] in the Congested Clique model. Particularly, this result is tight up to polylogarithmic factors for $y \geq 2/3$ due to a lower bound of [30].

We can also approximate unweighted eccentricities by a combination of computing shortest path distances from $n^{2/3}$ random sources and performing local explorations using the local edges of the model. For approximating weighted eccentricities, this is insufficient, and here our approach is to additionally broadcast required information from each random source node regarding its $\tilde{O}(n^{1/3})$-hop neighborhood in the graph. We obtain the following result.

▶ **Theorem 7** (Approx. Eccentricities). *Given a graph $G = (V, E)$, there is an algorithm in the* Hybrid *model that computes a $(1+\epsilon)$-approximation of unweighted and 3-approximation of weighted eccentricities in $\tilde{O}(n^{1/3}/\epsilon)$ rounds, w.h.p.*

Finally, the unweighted eccentricities approximation directly implies a $(1+\epsilon)$ approximation for unweighted diameter. This should be compared with the lower bound of $\tilde{\Omega}(n^{1/3})$ rounds for exact unweighted diameter due to [30].

▶ **Corollary 8** ($(1+\epsilon)$-Approx. Unweighted Diameter). *Let $G = (V, E)$ be an unweighted graph, and let $\epsilon > 0$. There exists an algorithm in the* Hybrid *model which computes a $(1+\epsilon)$-approximation of the diameter in $\tilde{O}(n^{1/3}/\epsilon)$ rounds, w.h.p.*

We refer the reader to Table 1, for a visual summary of our end results with comparison to related work.

Roadmap. The remainder of the current section is dedicated towards surveying related work. In Section 2, we provide all formal definitions. Next, in Section 3 we formally define the Oracle and Tiered Oracles models, and show how to simulate them over skeleton graphs in the Hybrid model. Finally, Section 4, gives our algorithms for distance problems in the oracle models and, using our simulations, also in the Hybrid model. Some additional results are deferred to the the full version of the paper [15]. There we show the approximation for shortest path from n^x given sources, eccentricities, diameter and lower bound for shortest paths from sources sampled i.i.d.

Table 1 Comparison of our results. SPD is the length of the shortest path diameter. The results for n^x-RSSP, and weighed diameter approximation upper bounds from previous works are implicit in [9, 30]. Our upper bound for $n^{2/3}$-RSSP is tight up to poly-logarithmic factors due to our lower bound. Our approximations for n^x-SSP are also tight up to poly-logarithmic factors for $x \geq 2/3$, due to [30].

Problem	Variant	Approximation	This work	Previous works
SSSP	weighted	exact	$\tilde{O}(n^{1/3})$	$\tilde{O}(n^{2/5})$ [30], $\tilde{O}(\sqrt{SPD})$ [9]
	weighted	$1+\epsilon$		$\tilde{O}(n^{1/3} \cdot \epsilon^{-6})$ [9]
	weighted	$(1/\epsilon)^{O(1/\epsilon)}$		n^{ϵ} [9]
n^x-RSSP	unweighted	$\tilde{O}(n^{1-x/2})$	$\tilde{\Omega}(n^{x/2})$	
	weighted	exact	$\tilde{O}(n^{1/3} + n^{2x-1})$	
	weighted	$2+\epsilon$		$\tilde{O}(n^{1/3} + n^{2x-1})$ [30]
$n^{1/3}$-SSP	unweighted	$1+\epsilon$		$\tilde{O}(n^{1/3}/\epsilon)$ [30]
	weighted	exact	$\tilde{O}(n^{1/3})$	
	weighted	$3+\epsilon$		$\tilde{O}(n^{1/3}/\epsilon)$ [30]
n^x-SSP	unweighted	$\tilde{O}(n^{1-x/2})$		$\tilde{\Omega}(n^{x/2})$ [30]
	unweighted	$1+\epsilon$	$\tilde{O}(n^{1/3}/\epsilon + n^{x/2})$	
	unweighted	$2+\epsilon$		$\tilde{O}(n^{1/3}/\epsilon + n^{x/2})$ [30]
	weighted	3	$\tilde{O}(n^{1/3} + n^{x/2})$	
	weighted	$3+\epsilon$		$\tilde{O}(n^{0.397} + n^{x/2})$ [30]
	weighted	$7+\epsilon$		$\tilde{O}(n^{1/3}/\epsilon + n^{x/2})$ [30]
eccentricities	unweighted	$1+\epsilon$	$\tilde{O}(n^{1/3}/\epsilon)$	
	weighted	3	$\tilde{O}(n^{1/3})$	
diameter	unweighted	exact		$\tilde{\Omega}(n^{1/3})$ [30]
	unweighted	$1+\epsilon$	$\tilde{O}(n^{1/3}/\epsilon)$	$\tilde{O}(n^{0.397}/\epsilon)$ [30]
	unweighted	$3/2+\epsilon$		$\tilde{O}(n^{1/3}/\epsilon)$ [30]
	weighted	$2-\epsilon$		$\tilde{\Omega}(n^{1/3})$ [30]
	weighted	2	$\tilde{O}(n^{1/3})$	$\tilde{O}(n^{2/5})$ [30]
	weighted	$2+\epsilon$		$\tilde{O}(n^{1/3} \cdot \epsilon^{-6})$ [9]
	weighted	$2 \cdot (1/\epsilon)^{O(1/\epsilon)}$		n^{ϵ} [9]

1.2 Related Work

Hybrid Models. The Hybrid network model was studied in [9, 20, 30]. In [20], distance results are obtained in one of the harsher variants of the model, where the local edges are restricted to have $\log n$ bandwidth. However, these apply only to extremely sparse graphs of at most $n + O(n^{1/3})$ edges and cactus graphs. In [2, 25], slightly different models of hybrid nature are studied.

Augustine et al. [8] proposed the Node-Capacitated Clique model, which is similar to the Congested Clique model, but each node has $\log n$ bandwidth. This model is also a special case of the generalised Hybrid model [9] without local edges. This allows one to use the results from the Node-Capacitated Clique model in the Hybrid model without modifications.

Distributed Distance Computations. Distance related problems have been extensively studied in many distributed models. For example, in the CONGEST model, there is a long line of research on APSP [3–5, 11, 19, 32, 34, 36] which culminated in tight, up to polylogarithmic factors, $\tilde{O}(n)$ round exact weighted APSP randomized algorithm of Bernstein and Nanongkai [11] and a $\tilde{O}(n^{4/3})$ round deterministic algorithm of Agarwal and Ramachandran [4]. [34, 36] develop an $\tilde{O}(n)$ round algorithm, optimal up to polylogarithmic factors, for unweighted APSP. The study of approximate SSSP algorithms was the focus of many recent paper [10, 27, 38] and lately Becker et al. [10] showed the solution which is close to the lower bound of Das Sarma et al. [38]. In case of exact SSSP, after recent works [19, 21, 24], there still is a gap between upper and lower bounds. The diameter and eccentricities problems are studied in the CONGEST model in [1, 6, 22, 36].

In the Congested Clique model, k-SSP, APSP and diameter are extensively studied in [14, 16, 23, 35] and approximate versions of the k-SSP and APSP problem are solved in polylogarithmic [13] and even polyloglogaritmic [18] time. In the more restricted Broadcast Congested Clique model, in which each message a node sends in a round is the same for all recipients, distance computations are researched by [10, 28].

2 Preliminaries

We provide here some definitions and claims that are critical for reading the main part of the paper. Full version of the paper [15] contains additional definitions and basic claims. We use the following variant of the Hybrid model, introduced in [9].

▶ **Definition 9** (Hybrid Model). *In the* Hybrid *model, a synchronous network of n nodes with identifiers in $[n]$, is given by a graph $G = (V, E)$. In each round, every node can send and receive λ messages of $O(\log n)$ bits to/from each of its neighbors (over* local edges*) and an additional γ messages in total to/from any other nodes in the network (over* global edges*). If in some round more than γ messages are sent via global edges to/from a node, only γ messages selected adversarially are delivered.*

We follow the previous work of [9, 30] and consider $\lambda = \infty, \gamma = O(\log n)$. Notice that the Hybrid model can also capture the classic LOCAL[3] model, with $\lambda = \infty, \gamma = 0$, the classic CONGEST model, with $\lambda = O(1), \gamma = 0$, the Congested Clique model, with $\lambda = O(1), \gamma = 0$ and G being a clique, the Congested Clique + Lenzen's Routing with $\lambda = 0, \gamma = n$ and the Node-Capacitated Clique model [8], with $\lambda = 0, \gamma = O(\log(n))$.

Many of our results hold for weighted graphs $G = (V, E, w)$. We assume an edge weight is given by a function $w \colon E \mapsto \{1, 2, \ldots, W\}$ for a W which is polynomial in n. When we *send* an edge as part of a message in any algorithm, we assume it is sent along with its weight.

[3] The LOCAL and CONGEST models are synchronous distributed models where every two neighbors in the graph can exchange messages of unlimited size or of $O(\log n)$ bits, respectively, in each round.

2.1 Notations and Problem Definitions

We use the following definitions related to graphs. Given a graph $G = (V, E)$ and a pair of nodes $u, v \in V$, we denote by $hop(u, v)$ the hop distance between u and v, by $N_G^h(v)$ the h-hop neighborhood of v, by $d_G^h(u, v)$ the weight of the lightest path between u and v of at most h-hops, and if there is no path of at most h-hops then $d_G^h(u, v) = \infty$. In the special case of $h = 1$, we denote by $N_G(v)$ the neighbors of v and in the special case of $h = \infty$, we denote by $d_G(u, v)$ the weight of the lightest path between u and v. We also denote by $\deg_G(v)$ the degree of v in G. Whenever it is clear from the context we drop the subscript of G and just write N, N^h, d, d^h or $\deg(v)$.

We define the following problems in the Hybrid model.

▶ **Definition 10** (k-Source Shortest Paths (k-SSP)). *Given a graph $G = (V, E)$, and a set $S \subseteq V$ of k sources. Every $u \in V$ is required to learn the distance $d_G(u, s)$ for each $s \in S$. The case where $k = 1$, is called the* single source shortest paths *problem (SSSP).*

▶ **Definition 11** (n^x-Random Sources Shortest Path (n^x-RSSP)). *Given a graph $G = (V, E)$, and a set $M \subseteq V$ of sources, such that each $v \in V$ is sampled independently with probability n^{x-1} to be in M. Every $u \in V$ is required to learn the distance $d_G(u, s)$ for each $s \in M$.*

In the approximate versions of these problems, each $u \in V$ is required to learn an (α, β)-approximate distance $\widetilde{d}(u, v)$ which satisfies $d(u, v) \leq \widetilde{d}(u, v) \leq \alpha \cdot d(u, v) + \beta$, and in case $\beta = 0$, $\widetilde{d}(u, v)$ is called an α-approximate distance.

▶ **Definition 12** (Eccentricity and diameter). *Given a graph $G = (V, E)$ and node $v \in V$, the* eccentricity *of v is the farthest shortest path distance from v, i.e., $ecc(v) = \max_{u \in V} d(v, u)$ and the diameter $D = \max_{v \in V} \{ ecc(v) \}$ is the maximum eccentricity. An α-approximation of all eccentricities is a function $\widetilde{ecc}(v)$ which satisfies $ecc(v)/\alpha \leq \widetilde{ecc}(v) \leq ecc(v)$ for all nodes v. An α-approximation of the diameter is a value $\widetilde{D}$ which satisfies $D/\alpha \leq \widetilde{D} \leq D$.*

2.2 Skeleton Graphs

In a nutshell, given a graph $G = (V, E)$, a skeleton graph $S_x = (M, E_S)$, for some constant $0 < x < 1$, is generated by letting every node in V independently join M with probability n^{x-1}. Two nodes in M have an edge in E_S if there exists a path between them in G of at most $h = \tilde{O}(n^{1-x})$ hops. This graph w.h.p. satisfies many useful properties in terms of distance computation, which for simplicity of presentation we add to its definition, provided below. A crucial property is that for any two nodes, if the shortest path between them in G has more than h hops, then there exists a shortest path between them in G on which every roughly h nodes there is a node from M (all such skeleton properties hold w.h.p.).

▶ **Definition 13** (Skeleton Graph, Combined Definition of [9, 30]). *Given a graph $G = (V, E)$ and a value $0 < x < 1$, a graph $S_x = (M, E_S)$ is called a skeleton graph in G, if all of the following hold.*

1. *Each $v \in V$ is included to M independently with probability n^{x-1}.*
2. *$\{v, u\} \in E_S$ if and only if there is a path of at most $h = \tilde{\Theta}(n^{1-x})$ edges between v, u in G.*
3. *Every node $v \in M$ knows all its incident edges in E_S.*
4. *S_x is connected.*
5. *For any two nodes $v, v' \in M$, $d_S(v, v') = d_G(v, v')$.*
6. *For any two nodes $u, v \in V$ with $hop(u, v) \geq h$, there is at least one shortest path P from u to v in G, such that any sub-path Q of P with at least h nodes contains a node $w \in M$.*
7. *$|M| = \tilde{O}(n^x)$.*

The following claim summarizes what is proven in [9] regarding the construction of a skeleton graph from a set of random marked nodes, w.h.p.

▷ Claim 14 (Skeleton from Random Nodes). Given a graph $G = (V, E)$, a value $0 < x < 1$, and a set of nodes M marked independently with probability n^{x-1}, there is an algorithm in the Hybrid model which constructs a skeleton graph $S_x = (M, E_S)$ in $\tilde{O}(n^{1-x})$ rounds w.h.p. If also given a single node $s \in V$, it is possible to construct $S_x = (M \cup \{ s \}, E_S)$, without damaging the properties of S_x.

We extract the following basic claim, used in the proof of [9, Theorem 2.7] for a $(1+\epsilon)$-approximation for SSSP, and slightly extend it to use for multiple source problem and arbitrary approximation factors. It states that given a skeleton graph and a set of sources, if every skeleton node knows any approximation to its distance from every source, then it is possible to efficiently reach a state where every node in the graph knows the approximation for its own distance from any of the sources. The idea is that each node locally explores its $\tilde{O}(n^{1-x})$ neighborhood and identifies for each source the best skeleton node in its neighborhood to go through.

▷ Claim 15 (Extend Distances). [9, Theorem 2.7] Let $G = (V, E)$, let $S_x = (M, E_S)$ be a skeleton graph, and let $V' \subseteq V$ be the set of source nodes. If for each source node $s \in V'$, each skeleton node $v \in M$ knows the (α, β)-approximate distance $\tilde{d}(v, s)$ such that $d(v, s) \leq \tilde{d}(v, s) \leq \alpha d(v, s) + \beta$, then each node $u \in V$ can compute for all source nodes $s \in V'$, a value $\tilde{d}(u, s)$ such that $d(u, s) \leq \tilde{d}(u, s) \leq \alpha d(u, s) + \beta$ in $\tilde{O}(n^{1-x})$ rounds.

3 Oracles in the Hybrid model

This section is split into three parts. Initially, as preliminaries, we show simulations of the LOCAL and Congested Clique models in the Hybrid model, citing [30] for the Congested Clique simulation. Then, we devote a section to each of the two new oracle models in order to introduce them and present their simulations in the Hybrid model.

3.1 Model Simulation Preliminaries

We will use simulations of the LOCAL and Congested Clique models as follows.

▶ **Lemma 16** (LOCAL Simulation). *Given a graph $G = (V, E)$, and a skeleton graph $S_x = (M, E_S)$, it is possible to simulate one round of the* LOCAL *model over S_x within $\tilde{O}(n^{1-x})$ rounds in G in the* Hybrid *model. That is, within $\tilde{O}(n^{1-x})$ rounds in G in the* Hybrid *model, any two adjacent nodes in S_x can communicate any amount of data between each other.*

The proof follows trivially due to the definition of the Hybrid model and Property 2 in the definition of a skeleton graph S_x, since in S_x two skeleton nodes are connected if they are within $\tilde{\Theta}(n^{1-x})$ hops in the original graph G. Thus, one round of the LOCAL model over S_x is obtained in the Hybrid network in $\tilde{O}(n^{1-x})$ rounds, by having neighboring skeleton nodes communicate through the local edges.

▶ **Lemma 17** (Congested Clique Simulation). *[30, Corollary 4.1.] Given a graph $G = (V, E)$, and a skeleton graph $S_x = (M, E_S)$, for some constant $0 < x < 1$, it is possible to simulate one round of the* Congested Clique *model over S_x in $\tilde{O}(n^{2x-1} + n^{\frac{x}{2}})$ rounds of the* Hybrid *model on G, w.h.p. That is, within $\tilde{O}(n^{2x-1} + n^{\frac{x}{2}})$ rounds of the* Hybrid *model on G, w.h.p., every node $v \in M$ can, for each node $u \in M$, each send a unique $O(\log n)$ bit message to u.*

3.2 Simulating the Oracle Model

Here, we define the Oracle model and then show how to efficiently simulate it over a skeleton graph in the Hybrid model.

▶ **Definition 18** (Oracle Model). *In the* Oracle *model over a network G, there exists one* oracle *node ℓ, which in every round can send to and receive from every node v a number of $O(\log n)$-bit messages that is equal to the degree of v in G.*

▶ **Theorem 19** (Oracle Simulation). *Given a graph $G = (V, E)$, for every constant $0 < x < 1$, there is an algorithm which simulates one round of the* Oracle *model, on a skeleton graph $S_x = (M, E_S)$, in $\tilde{O}(n^{1-x} + n^{2x-1})$ rounds of the* Hybrid *model on G, w.h.p.*

Proof. We prove the claim by showing how to simulate a round of the Oracle model in $O(1)$ rounds of the Congested Clique model and 1 round of the LOCAL model. Then, invoking the simulations of Lemmas 16 and 17, gives the desired round complexity in the Hybrid model.

We show how to send messages to the oracle, and the receiving part is symmetric. The pseudocode is given by Algorithm 1. First, each $v \in M$ broadcasts its degree in S_x using one round of the Congested Clique model (Line 1) and selects as an oracle ℓ the node with largest degree in S_x, breaking ties by identifier (Line 2). Then, the identifiers of the neighbors of ℓ are broadcast using one round of the Congested Clique model (Line 3). The actual messages are sent to these neighbors instead of to ℓ itself (Line 4) and ℓ learns all these messages in 1 round of the LOCAL model in Line 5.

Clearly, all the nodes select the same oracle ℓ (Line 2). Due to the definition of the Oracle model, each node $v \in M$ has $\deg_{S_x}(v)$ messages to send, and since $\deg_{S_x}(\ell) \geq \deg_{S_x}(v)$, there are enough neighbors of ℓ to receive one message from v per neighbor, which is why Line 4 can work. ◀

Algorithm 1 Simulating the Oracle model in the Congested Clique with LOCAL.

1 Congested Clique model: $v \in M$ broadcasts $\deg_{S_x}(v)$
2 Select an oracle $\ell \leftarrow \arg\max_{v \in M} \{ (\deg_{S_x}(v), v) \}$
3 Congested Clique model: $v \in M$ broadcasts if it is a neighbor of ℓ
4 Congested Clique model: $v \in M$ sends i-th message to i-th neighbor of ℓ for each i
5 LOCAL model: ℓ collects the messages from its neighbors

3.3 Simulating the Tiered Oracles Model

We further enhance our Oracle model and define the Tiered Oracles model, where, roughly speaking, all nodes in parallel can learn all the edges adjacent to nodes with degrees in lower degree buckets. To simulate the stronger Tiered Oracles model over a skeleton graph in the Hybrid model, we need additional insights. Here, we use the fact that when we scatter messages independently at random, denser neighborhoods are more likely to receive a given message than sparse neighborhoods. In other words, while for simulating the Oracle model we used the LOCAL round only to concentrate information in a single node ℓ, here we exploit the information that *each* node can gather from its neighborhood.

▶ **Definition 20** (Tiered Oracles Model). *In the* Tiered Oracles *model over a network G, in every round, suppose each node v has a set of $O(\log n)$-bit messages M_v of size $|M_v| = \deg(v)$, then each node u can receive all messages in M_v for every v such that $\deg(u) \geq \deg(v)/2$.*

To simulate the Tiered Oracles model, we first prove the following model-independent tool.

▶ **Lemma 21** (Sampled neighbors). *Given is a graph $G = (V, E)$. For a value $c \leq n$, there is a value $x = \tilde{O}(n/c)$ such that the following holds w.h.p.: Let $V' \subseteq V$ be a subset of $|V'| = x$ nodes sampled uniformly at random from M. Then each node $u \in V$ with $\deg(u) \geq c$ has a neighbor in V'.*

Proof. For some node $u \in V$, the probability of not having a neighbor sampled to the set V' is $(1 - \deg(u)/n)^x \leq e^{-x \cdot \det(u)/n} \leq e^{x \cdot c/n}$. Thus, there exists $x = \tilde{O}(n/c)$ such that node u has a neighbor in the set V', w.h.p. ◀

Finally, we show how to simulate the Tiered Oracles model over the skeleton graph in the Hybrid model.

▶ **Theorem 22** (Tiered Oracles Simulation). *Given a graph $G = (V, E)$, for every constant $0 < x < 1$, there is an algorithm which simulates one round of the* Tiered Oracles *model, on a skeleton graph $S_x = (M, E_S)$, in $\tilde{O}(n^{1-x} + n^{2x-1})$ rounds of the* Hybrid *model on G, w.h.p.*

Proof. We prove the claim by reducing one round of the Tiered Oracles model to $\tilde{O}(1)$ rounds of the Congested Clique model followed by a round of the LOCAL model on the skeleton graph S_x. By Lemmas 16 and 17, we obtain that the resulting round complexity is $\tilde{O}(n^{1-x} + n^{2x-1})$.

For each $v \in M$, let M_v be the set of messages, of size $|M_v| = \deg_{S_x}(v)$, which v desires to broadcast. For each message in M_v node $v \in M$ samples uniformly at random $x = \tilde{O}(2 \cdot n / \deg_{S_x}(v))$ nodes of M and sends the message to those nodes. As each node sends and receives $\tilde{O}(|M|)$ messages this can be done using with the well known routing theorem of Lenzen [31, Theorem 3.7] by simulating $\tilde{O}(1)$ rounds of the Congested Clique model. Alternatively, this can be done in the same round complexity by applying the algorithm for *token routing* [30, Theorem 2.2]. Afterwards, we simulate one round of the LOCAL model over S_x for each node to learn tokens received by its neighbors in S. Due to Lemma 21 (*Sampled neighbors*), each node $u \in V$ learns messages from each v such that $\deg_{S_x}(u) \geq \deg_{S_x}(v)/2$ w.h.p. ◀

4 Shortest Paths Algorithms

4.1 Warm-Up: Exact SSSP

As a warm-up, we show how to compute exact SSSP in the Oracle model, and then we simulate this on a skeleton graph in the Hybrid model in order to get exact SSSP in the Hybrid model within $\tilde{O}(n^{1/3})$ rounds. We note that later, in Section 4.3, we obtain this complexity for exact distances from a much larger set, of $O(n^{1/3})$ sources.

▶ **Lemma 23** (Exact SSSP in the Oracle Model). *There is a* deterministic *algorithm in the* Oracle *model that given a weighted graph $G = (V, E)$ and source $s \in V$ solves exact SSSP in $O(1)$ rounds.*

Proof. Let $s \in V$ be the source node. We solve the problem in two communication rounds. In the first round, oracle ℓ learns all of E by receiving from each node v its adjacent edges. Afterwards, oracle ℓ, given all the edges in the graph G, locally computes the distance from s to every other node. In the second round, oracle ℓ sends for each $v \in V$ the value $d(s, v)$. It is clear that the algorithm computes SSSP from $s \in V$, and that it takes two rounds in the Oracle model. ◀

▶ **Theorem 1** (Exact SSSP). *Given a weighted graph $G = (V, E)$, there is an algorithm in the* Hybrid *model that computes an exact weighted SSSP in $\tilde{O}(n^{1/3})$ rounds w.h.p.*

Proof. Let s be the source node, and let $x = 2/3$. We start by constructing a skeleton graph $S_x = (M, E_S)$, by sampling nodes with probability $n^{-1/3}$ and using Claim 14 (*Skeleton from Random Nodes*). Then, we simulate the algorithm given in Lemma 23 in the Oracle model, which computes the distance $d_S(s, v)$ from s to each node $v \in M$. By Property 5 of the skeleton graph, for every $v \in M$, it holds that $d_S(s, v) = d_G(s, v)$. To extend this and compute the distance from s for each node $v \in V$, we apply Claim 15 (*Extend Distances*).

Constructing the skeleton graph takes $O(h) = \tilde{O}(n^{1/3})$ rounds w.h.p., by Claim 14 (*Skeleton from Random Nodes*). Simulating the algorithm from Lemma 23 completes in $\tilde{O}(n^{1/3})$ rounds w.h.p. by Theorem 19 (Oracle *Simulation*). Applying Claim 15 (*Extend Distances*) takes $\tilde{O}(n^{1/3})$ rounds. Therefore, overall, the execution of the algorithm completes in $\tilde{O}(n^{1/3})$ rounds w.h.p. ◀

4.2 Exact n^x-RSSP

Recall that in Definition 11 (*n^x-Random Sources Shortest Path (n^x-RSSP)*), we are given set of roughly n^x sources sampled independently with probability n^{x-1}, and we need for each node to compute its distance to each source. We do so by constructing a skeleton graph S_x from the random sources. We show that using one round of the Tiered Oracles model, and $O(\log n)$ rounds of the Congested Clique model, one can solve APSP over S_x. To do so, we split the nodes of the graph into $\lceil \log n \rceil$ tiers by degree and compute APSP by proceeding tier after tier and computing distances from current tier to all the tiers below.

▶ **Lemma 24** (APSP in Congested Clique with Tiered Oracles). *There is a* deterministic *algorithm which, given a weighted graph $G = (V, E)$, solves exact APSP on G using $O(\log |V|)$ rounds of the* Congested Clique *model and one round of the* Tiered Oracles *model.*

Proof. The pseudocode for the algorithm appears in Algorithm 2. We partition the nodes V by their degrees into $\lceil \log |V| \rceil$ tiers, $T_j = \left\{ v \in V : 2^j \le \deg(v) < 2^{j+1} \right\}$ for $0 \le j < \lceil \log |V| \rceil$. Denote by $T_{>i} = \bigcup_{k>i} T_k$ the nodes in all tiers $k > i$ and by $T_{\le i} = \bigcup_{k \le i} T_k$ the nodes in all tiers $k \le i$. Similarly, define $T_{\ge i}$ and $T_{<i}$. Denote by $d_{\le i}(u, v)$ the weight of the shortest path between u and v that uses only edges adjacent to at least one node in $T_{\le i}$.

Algorithm 2 **Exact-APSP**: Computes exact APSP using the Congested Clique and Tiered Oracles models.

```
1 Tiered Oracles model: each v ∈ T_i broadcasts to u ∈ T_{≥i} its incident edges
2 for i = ⌈log |V|⌉ − 1 downto 0 do
3     For each node u ∈ T_{≤i}, each node v ∈ T_i computes
        d̃(v, u) ← min { d_{≤i}(v, u), min_{w∈T_{>i}} { d̃(v, w) + d_{≤i}(w, u) } }
4     Congested Clique model: v ∈ T_i sends to u ∈ T_{≤i}, the value d̃(v, u)
```

The outline of our algorithm is as follows. We start by having each node $v \in T_i$ broadcast its incident edges to all the nodes in tiers greater than or equal to its own, that is, to all $u \in T_{\ge i}$, using one round of the Tiered Oracles model (Line 1). Afterwards, in the loop in Line 2, we compute the solution tier by tier, starting from the topmost tier, which contains nodes knowing all the edges in the graph. While processing the i-th tier, every node $v \in T_i$ already knows its distance to every node in $T_{>i}$, and so computes its distances to every node

$u \in T_{\leq i}$. A shortest path between such v and u can either pass through edges which are all known to v, or be broken into a subpath from v to some node $w \in T_{>i}$ and then a path from w to u which is known to v. Thus, we compute the distance from $v \in T_i$ to the nodes $T_{\leq i}$ (Line 3). On Line 4, node $v \in T_i$, which knows for each node $u \in T_{\leq i}$ the distance to u, sends it to u.

For each $u, v \in V$, Algorithm 2 outputs a value $\tilde{d}(u,v)$. We show that it is the correct distance in G, that is $\tilde{d}(u,v) = d(u,v)$.

One round of the Tiered Oracles model suffices for ensuring that for each tier, T_i, every node $v \in T_i$ knows all the edges incident to all the nodes $u \in T_{\leq i}$. Let $v \in T_i$, and $u \in T_j$ such that $i \geq j$, observe that it holds that $\deg(v) \geq 2^i \geq \frac{1}{2}2^j = \frac{1}{2}\deg(u)$, and therefore after Line 1 node v knows the edges incident to u. Thus, each node $v \in T_i$ knows enough information to compute the function $d_{\leq i}$, which is the distance function in G limited to edges incident to nodes in $T_{\leq i}$.

By induction on tier index i, we show that after iteration i of the loop in Line 2 all the nodes in V know the exact distances to all nodes in tiers $T_{\geq i}$.

Base case: In iteration $i = \lceil \log |V| \rceil - 1$, node $v \in T_{\lceil \log |V| \rceil - 1}$ (if exists) in the topmost tier knows about all the edges in E since it knows about all edges incident to nodes $T_{\leq \lceil \log |V| \rceil - 1} = V$. Thus, v can compute the solution to the entire APSP on G, since $d_{\leq \lceil \log |V| \rceil - 1} = d$. Since the set

$$\{\, d(v,w) + d_{\leq \lceil \log |V| \rceil - 1}(w,u) \,\}_{w \in T_{> \lceil \log |V| \rceil - 1}}$$

is empty, we get that $\tilde{d}(v,u) = d_{\leq \lceil \log |V| \rceil - 1}(v,u) = d(v,u)$. That is, node $v \in T_{\lceil \log |V| \rceil - 1}$ computes for each other node $u \in V$ its weighted distance $d(v,u)$ and sends it to u on Line 4.

Induction Step: In iteration $i < \lceil \log |V| \rceil - 1$, consider $v \in T_i$ and $u \in T_{\leq i}$, and let P be a shortest path between them. Recall that node v can locally compute $d_{\leq i}$, and thus knows the value $d_{\leq i}(v,u)$ and for each $w \in T_{>i}$, it knows the value $d_{\leq i}(w,u)$. Further, for each $w \in T_{>i}$, the value $\tilde{d}(v,w) = d(v,w)$ is known to v from one of the previous iterations of the loop in Line 2, by the induction assumption. All values in the set $\{\, \tilde{d}(v,w) + d_{\leq i}(w,u) \,\}_{w \in T_{>i}} \cup \{\, d_{\leq i}(v,u) \,\}$ are either infinite or correspond to some (not necessary simple) path from v to u, thus $\tilde{d}(v,u) \geq d(v,u)$. To show that $\tilde{d}(v,u) \leq d(v,u)$, we consider two cases. If P does not contain nodes from $T_{>i}$, then $d_{\leq i}(v,u) = d(v,u)$ is the length of P. Otherwise, let $w' \in T_{>i}$ be the last node on P (closest to u) which belongs to $T_{>i}$. By the induction hypothesis, v knows $\tilde{d}(v,w') = d(v,w')$. Moreover, the subpath from w' to u only contains edges with at least one endpoint incident to node in $T_{\leq i}$, thus $d_{\leq i}(w',u) = d(w',u)$. For this node w' the value $\{\, \tilde{d}(v,w') + d_{\leq i}(w',u) \,\}$ belongs to $\{\, \tilde{d}(v,w) + d_{\leq i}(w,u) \,\}_{w \in T_{>i}}$. Thus, in both cases the computed $\tilde{d}(v,u)$ is at most the weighted length of P. Hence, $\tilde{d}(v,u) = d(v,u)$. On Line 4, node v informs u about the correct $d(v,u)$, which completes the induction proof.

Lines 1 and 4 each take a single round of the Tiered Oracles model and the Congested Clique model, respectively, and thus the execution of the entire algorithm takes $O(\log |V|)$ rounds of the Congested Clique model and one round of the Tiered Oracles model. ◀

By simulating the algorithm given in Lemma 24 (*APSP in* Congested Clique *with* Tiered Oracles) using Theorem 22 (Tiered Oracles *Simulation*) and Lemma 17 (Congested Clique *Simulation*), we get exact APSP over the skeleton graph, as follows.

▶ **Corollary 25** (Exact APSP on Skeleton Graph)**.** *For any constant $0 < x < 1$, there is an algorithm in the* Hybrid *model that computes an exact weighted APSP on a skeleton graph $S_x = (M, E_S)$, in $\tilde{O}(n^{1-x} + n^{2x-1})$ rounds w.h.p.*

Finally, we extend the result to n^x-RSSP on G, by having each node in the graph learn the information stored in the skeletons in its $\tilde{O}(n^{1-x})$ neighborhood.

▶ **Theorem 3** (n^x-RSSP)**.** *Given a graph $G = (V, E)$, $0 < x < 1$, and a set of nodes M sampled independently with probability n^{x-1}, there is an algorithm in the* Hybrid *model that ensures that every $v \in V$ knows the exact, weighted distance from itself to every node in M within $\tilde{O}(n^{1/3} + n^{2x-1})$ rounds w.h.p.*

Proof. Primarily, assume that $x \geq \frac{2}{3}$. Otherwise, we add each node outside of M with probability $(n^{-1/3} - n^{x-1})/(1 - n^{x-1})$ into the set M. Thus, each node has probability exactly $(n^{x-1} \cdot 1) + (1 - n^{x-1}) \cdot (n^{-1/3} - n^{x-1})/(1 - n^{x-1}) = n^{-1/3}$ to be sampled into M, ensuring $x = 2/3$. We use Claim 14 (*Skeleton from Random Nodes*) to build a skeleton graph $S_x = (M, E_S)$ in $\tilde{O}(n^{1/3})$ rounds w.h.p. Then, we compute exact APSP on the skeleton graph using Corollary 25 (*Exact APSP on Skeleton Graph*) in $\tilde{O}(n^{1/3} + n^{2x-1})$ rounds w.h.p. By Property 5 of the skeleton graph, for each $v, u \in M$ it holds that $d_S(v, u) = d(v, u)$, where $d_S(v, u)$ is the distance in the skeleton graph. So, we apply Claim 15 (*Extend Distances*) with $\alpha = 1, \beta = 0$ and set of sources $V' = M$, to compute an exact weighted shortest paths distances, from M to all of V, in additional $\tilde{O}(n^{1/3})$ rounds w.h.p. ◀

Instantiating Theorem 3 with $x = 2/3$ gives $n^{2/3}$-RSSP in $\tilde{\Theta}(n^{1/3})$ rounds w.h.p., which is tight due to our lower bound given in Theorem 4 (*Lower Bound Exact Shortest Paths, Sources Sampled i.i.d*). We extensively use our $n^{2/3}$-RSSP algorithm for our following results.

4.3 Exact $n^{1/3}$-SSP

We now present an improvement over the warm-up exact SSSP algorithm which we showed previously, by providing an algorithm for exact shortest paths from a *given* set of $n^{1/3}$ nodes ($n^{1/3}$-SSP) in $\tilde{O}(n^{1/3})$ rounds. To do so, we create a skeleton graph and use our algorithm for $n^{2/3}$-RSSP algorithm to compute exact distances from the skeleton nodes to the entire graph. Then, we adapt the behavior of the source nodes depending on the number of skeleton nodes in their neighborhood (which is proportional to the density of the neighborhoods). That is, nodes in sparse neighborhoods can broadcast the distances from themselves to all the skeleton nodes which they see surrounding them, while a node in dense neighborhoods can take over a skeleton node surrounding it and use it as a proxy to communicate efficiently with the other skeleton nodes in the graph. We formalize this in this section, as well as refer to Lemma 26 (*Reassign Skeletons*) which is a generic tool which performs this action of *taking over* skeleton nodes as proxies.

We show the following fundamental algorithm, which allows *assigning* skeletons to help other skeletons. That is, given a set of nodes A where each node in A sees many skeleton nodes in its neighborhood, it is possible to assign skeleton nodes to service the nodes of A. We use this to increase sending and receiving *capacity* of the nodes of A. This is a key tool which we use in the proof of Theorem 5 (*Exact $n^{1/3}$ Sources Shortest Paths*) and we believe it may be useful for additional tasks.

▶ **Lemma 26** (Reassign Skeletons)**.** *Given graph $G = (V, E)$, a skeleton graph $S_x = (M, E_S)$, a value k which is known to all the nodes, and nodes $A \subseteq V$ such that each $u \in A$ has at least $\tilde{\Theta}(k \cdot |A|)$ nodes $M_u \subseteq M$ in its $\tilde{\Theta}(n^{1-x})$ neighborhood, there is an algorithm that assigns $K_u \subseteq M_u$ nodes to u, where $|K_u| = \tilde{\Omega}(k)$, such that each node in M is assigned to at most $\tilde{O}(1)$ nodes in A. With respect to the set A, it is only required that every node in G must know whether or not it itself is in A – that is, the entire contents of A do not have to be globally known. The algorithm runs in $\tilde{O}(n^{1-x})$ rounds in the* Hybrid *model, w.h.p.*

Proof. The pseudocode is provided by Algorithm 3.

Algorithm 3 Reassign-Skeletons(A, k).

1 Compute $|A|$ by running Aggregate-And-Broadcast
2 Skeleton node $v \in M$ learns its $\tilde{O}(n^{1-x})$-hop neighborhood
3 Skeleton node $v \in M$ samples each $u \in A \cap N_G^{\tilde{\Theta}(n^{1-x})}(v)$ with probability $\frac{1}{|A|}$
4 Skeleton node $v \in M$ informs each sampled node u about $v \in K_u$

First, each node w learns the size of the set A by invoking using aggregate and broadcast routine [8, Theorem 2.2] with value 1 if $w \in A$ and 0 otherwise, and the summation function (Line 1). Then, each skeleton node $v \in M$, learns its $\tilde{\Theta}(n^{1-x})$-hop neighborhood (Line 2), and in particular it learns the nodes $A_v = A \cap N_G^{\tilde{\Theta}(n^{1-x})}(v)$. Then, v samples each $u \in A_v$ independently with probability $\frac{1}{|A|}$ (Line 3). Afterwards, v informs each node u it sampled on the previous stage that $v \in K_u$ (Line 4).

For every $v \in M$, since $|A_v| \leq |A|$, and since v samples nodes from there A_v independently with probability $\frac{1}{|A|}$, by Chernoff Bounds each v assigns itself to at most $\tilde{O}(1)$ nodes $a \in A_v$ w.h.p. Hence, by a union bound over all skeleton nodes, each skeleton node is assigned to $\tilde{O}(1)$ nodes w.h.p.

For every $u \in A$, since it is sampled by at least $\tilde{\Omega}(k \cdot |A|)$ skeleton nodes independently with probability $\frac{1}{|A|}$, by Chernoff Bounds it is sampled by $|K_u| = \tilde{\Omega}(1)$ skeleton nodes w.h.p. Thus, by union bound over all skeleton nodes, each $u \in A$ has $|K_u| = \tilde{\Omega}(1)$ assigned nodes w.h.p.

By [8, Theorem 2.2], Line 1 takes $\tilde{O}(1)$ rounds w.h.p., and Lines 2 and 4 take $\tilde{O}(n^{1-x})$ rounds, and thus the entire execution completes in $\tilde{O}(n^{1-x})$ rounds w.h.p. ◀

Now we apply Token Dissemination [9, Theorem 2.1] or Lemma 26 (*Reassign Skeletons*) depending on density of each source's neighborhood and show how to compute exact $n^{1/3}$-SSP in $\tilde{O}(n^{1/3})$ rounds. For sources in "sparse" neighborhoods, in which there is a small number of skeleton nodes, we use Token Dissemination to inform all nodes about their distances to those skeletons. For source v with "dense" neighborhood, in which there are many skeleton nodes, we use Lemma 26 (*Reassign Skeletons*) to get at least one skeleton node u which participates in the round of the Congested Clique model on behalf of that source and sends each other skeleton node v' the distance $d(v, v')$.

▶ **Theorem 5** (Exact $n^{1/3}$ Sources Shortest Paths)**.** *Given a weighted graph $G = (V, E)$, and a set of sources U, such that $|U| = O(n^{1/3})$, there exists an algorithm, at the end of which each $v \in V$ knows its distance from every $s \in U$, which runs in $\tilde{O}(n^{1/3})$ rounds w.h.p.*

Proof. The pseudocode for the algorithm appears in Algorithm 4.

Without loss of generality the set of nodes U is globally known (it can be disseminated in $\tilde{O}(n^{1/6})$ rounds w.h.p. using Token Dissemination from [9, Theorem 2.1]). We build $M \subseteq V$ by marking nodes independently with probability $n^{-1/3}$ (Line 1). Then we run the algorithm from Theorem 3 (n^x*-RSSP*) with $x = 2/3$ to obtain w.h.p. $n^{2/3}$-RSSP from the set of nodes M (Line 2), such that w.h.p. every $u \in V$ knows its distance to every node in M. Afterwards, we apply Claim 14 (*Skeleton from Random Nodes*) to construct a skeleton graph $S_{2/3} = (M, E_S)$ w.h.p. Then, each source learns the information in its h-hop neighborhood (Line 4), for $h \in \tilde{\Theta}(n^{1/3})$. In particular, it counts the skeleton nodes in its h-hop neighborhood.

Algorithm 4 **Exact-$n^{1/3}$-SSP**: Computes an exact weighted $n^{1/3}$-SSP. Routine for node $u \in V$.

1 Join M independently with probability $n^{-1/3}$
2 Compute $n^{2/3}$-RSSP from M
3 Construct skeleton graph $S_{2/3} = (M, E_S)$
4 Learn $h = \tilde{\Theta}(n^{1/3})$-hop neighborhood
5 **if** $u \in U$ **then**
6   **if** $\left|N_G^h(u) \cap M\right| = \tilde{O}(n^{1/3})$ **then**
7   Participate in `Token-Dissemination` with a token $\langle u, v', d(v', u)\rangle$ for each $v' \in M \cap N_G^h(s)$
8   **else**
9         $K_u \leftarrow$`Reassign-Skeletons` $\left(\left\{ u \colon \left|N_G^h(u) \cap M\right| = \tilde{\Omega}(n^{1/3}) \right\}, \tilde{O}(1)\right)$
10   Send each $v \in K_u$ the values $d(u, v')$ for each $v' \in M$
11 **if** $u \in M$ **then**
12   In the Congested Clique model: for each $v' \in M$ and each $v \in U$ such that $u \in K_v$ send $d(v, v')$ to v'
13   For each $s \in U$, compute $\tilde{d}(u, s)$ by Equation (1) and output it
14 Apply Claim 15, which given distances from skeleton to sources $\tilde{d} \colon M \times U \mapsto \mathbb{N}$ extends it to distances from each nodes to sources $\tilde{d} \colon V \times U \mapsto \mathbb{N}$

If a source finds that the number of skeleton nodes in its h-hop neighborhood is $\tilde{O}(n^{1/3})$, then it participates in a token dissemination protocol ([9, Theorem 2.1]) and w.h.p. informs all the graph about its distance to these skeleton nodes.

Otherwise, each source $u \in U$ which finds that there are at least $\tilde{\Omega}(n^{1/3})$ skeleton nodes in its h-hop neighborhood, applies Lemma 26 (*Reassign Skeletons*) with $k = \tilde{O}(1)$ and $A = \left\{ u \colon \left|N_G^h(u) \cap M\right| = \tilde{\Omega}(n^{1/3}) \right\}$ and receives $K_u \subseteq N_G^h(u) \cap M$, a set of $\tilde{\Omega}(1)$ skeletons. Such a source $u \in U$ sends by local edges to $v \in K_u$ the distance $d(u, v')$ to each $v' \in M$ (Line 10). Each skeleton node $u \in M$ sends the distances $d(s, v')$ to each $v' \in M$, for each source node $s \in U$ that it is assigned to, by simulating the Congested Clique model. If skeleton node $u \in M$ for source $s \in U$ did not receive the distance from s, it computes it using Equation (1) (Line 13) based on the information it received in Line 7.

$$\tilde{d}(u, s) = \min \left\{ d^h(u, s), \min_{v' \in M} \left\{ d(u, v') + d^h(v', s) \right\} \right\}^4 \tag{1}$$

After each skeleton knows the distance to each source, we apply Claim 15 to compute distances from sources to all the nodes.

▶ **Lemma 27.** *After Line 13, each $u \in M$ knows $d(v, s)$ for each $s \in U$ w.h.p.*

By Lemma 27, whose proof appears in the full version of the paper [15], each node in M knows the distance to each node in U, thus by Claim 15 (*Extend Distances*) with $\alpha = 1, \beta = 0$ there is an algorithm to compute shortest paths distance from U.

By Theorem 3 (*n^x-RSSP*) with $x = \frac{2}{3}$, Line 2 completes in $\tilde{O}(n^{1/3})$ rounds w.h.p. For Line 3, by Claim 14 (*Skeleton from Random Nodes*), the round complexity, w.h.p., is $\tilde{O}(n^{1/3})$ as well. Line 4 completes in $\tilde{O}(n^{1/3})$ rounds. Since there are at most $\ell = \tilde{O}(n^{1/3})$ tokens per source and $k = \Omega\left(n^{2/3}\right)$ tokens overall, Line 7 takes $\tilde{O}(n^{1/3})$ rounds w.h.p. by [9, Theorem 2.1]. By Lemma 26 (*Reassign Skeletons*), Line 9 takes $\tilde{O}(n^{1/3})$ rounds w.h.p. All skeleton nodes are assigned to some helpers in their $\tilde{O}(n^{1/3})$-hop neighborhood by Line 9, so Line 4 takes $\tilde{O}(n^{1/3})$ rounds. Since each skeleton selects $\tilde{O}(1)$ sources w.h.p. in Line 9 by Lemma 26 (*Reassign Skeletons*), Line 12 simulates $\tilde{O}(1)$ rounds of the Congested Clique model and takes $\tilde{O}(n^{1/3})$ rounds by Lemma 17 (Congested Clique *Simulation*) w.h.p. Finally by Claim 15 (*Extend Distances*), Line 14 for $x = \frac{2}{3}$ takes $\tilde{O}(n^{1/3})$ rounds as well. Thus, the overall execution of the algorithm takes $\tilde{O}(n^{1/3})$ rounds. ◀

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, Paris, France, September 27-29, 2016. Proceedings*, volume 9888 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2016. doi:10.1007/978-3-662-53426-7_3.

2 Yehuda Afek, Gad M. Landau, Baruch Schieber, and Moti Yung. The power of multimedia: Combining point-to-point and multiaccess networks. *Inf. Comput.*, 84(1):97–118, 1990. doi:10.1016/0890-5401(90)90035-G.

3 Udit Agarwal and Vijaya Ramachandran. Distributed weighted all pairs shortest paths through pipelining. In *2019 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2019, Rio de Janeiro, Brazil, May 20-24, 2019*, pages 23–32. IEEE, 2019. doi:10.1109/IPDPS.2019.00014.

4 Udit Agarwal and Vijaya Ramachandran. Faster deterministic all pairs shortest paths in congest model. In Christian Scheideler and Michael Spear, editors, *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*, pages 11–21. ACM, 2020. doi:10.1145/3350755.3400256.

5 Udit Agarwal, Vijaya Ramachandran, Valerie King, and Matteo Pontecorvi. A deterministic distributed algorithm for exact weighted all-pairs shortest paths in õ(n $^{3/2}$) rounds. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 199–205. ACM, 2018. doi:10.1145/3212734.3212773.

6 Bertie Ancona, Keren Censor-Hillel, Mina Dalirrooyfard, Yuval Efron, and Virginia Vassilevska Williams. Distributed distance approximation. *CoRR*, abs/2011.05066, 2020. arXiv:2011.05066.

7 Arash Asadi, Vincenzo Mancuso, and Rohit Gupta. An sdr-based experimental study of outband D2D communications. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, pages 1–9. IEEE, 2016. doi:10.1109/INFOCOM.2016.7524372.

8 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, Phoenix, AZ, USA, June 22-24, 2019*, pages 69–79. ACM, 2019. doi:10.1145/3323165.3323195.

9 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.

10 Ruben Becker, Andreas Karrenbauer, Sebastian Krinninger, and Christoph Lenzen. Near-optimal approximate shortest paths and transshipment in distributed and streaming models. In Andréa W. Richa, editor, *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, volume 91 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.DISC.2017.7.

11 Aaron Bernstein and Danupon Nanongkai. Distributed exact weighted all-pairs shortest paths in near-linear time. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 334–342. ACM, 2019. doi:10.1145/3313276.3316326.

12 Keren Censor-Hillel, Michal Dory, Janne H. Korhonen, and Dean Leitersdorf. Fast approximate shortest paths in the congested clique. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 74–83. ACM, 2019.

13 Keren Censor-Hillel, Michal Dory, Janne H. Korhonen, and Dean Leitersdorf. Fast approximate shortest paths in the congested clique. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 74–83. ACM, 2019.

14 Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. *Distributed Comput.*, 32(6):461–478, 2019. doi:10.1007/s00446-016-0270-2.

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

16 Keren Censor-Hillel, Dean Leitersdorf, and Elia Turner. Sparse matrix multiplication and triangle listing in the congested clique model. *Theor. Comput. Sci.*, 809:45–60, 2020. doi:10.1016/j.tcs.2019.11.006.

17 Yong Cui, Hongyi Wang, and Xiuzhen Cheng. Channel allocation in wireless data center networks. In *INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10-15 April 2011, Shanghai, China*, pages 1395–1403. IEEE, 2011. doi:10.1109/INFCOM.2011.5934925.

18 Michal Dory and Merav Parter. Exponentially faster shortest paths in the congested clique. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 59–68. ACM, 2020. doi:10.1145/3382734.3405711.

19 Michael Elkin. Distributed exact shortest paths in sublinear time. *J. ACM*, 67(3):15:1–15:36, 2020. doi:10.1145/3387161.

20 Michael Feldmann, Kristian Hinnenthal, and Christian Scheideler. Fast hybrid network algorithms for shortest paths in sparse graphs. *CoRR*, abs/2007.01191, 2020. arXiv:2007.01191.

21 Sebastian Forster and Danupon Nanongkai. A faster distributed single-source shortest paths algorithm. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 686–697. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00071.

22 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012. doi:10.1137/1.9781611973099.91.

23 François Le Gall. Further algebraic algorithms in the congested clique model and applications to graph-theoretic problems. In Cyril Gavoille and David Ilcinkas, editors, *Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, volume 9888 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2016. doi:10.1007/978-3-662-53426-7_5.

24 Mohsen Ghaffari and Jason Li. Improved distributed algorithms for exact shortest paths. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 431–444. ACM, 2018. doi:10.1145/3188745.3188948.

25 Robert Gmyr, Kristian Hinnenthal, Christian Scheideler, and Christian Sohler. Distributed monitoring of network properties: The power of hybrid networks. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 137:1–137:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.137.

26 Kai Han, Zhiming Hu, Jun Luo, and Liu Xiang. RUSH: routing and scheduling for hybrid data center networks. In *2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, pages 415–423. IEEE, 2015. doi:10.1109/INFOCOM.2015.7218407.

27 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. A deterministic almost-tight distributed algorithm for approximating single-source shortest paths. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 489–498. ACM, 2016. doi:10.1145/2897518.2897638.

28 Stephan Holzer and Nathan Pinsker. Approximation of distances and shortest paths in the broadcast congest clique. In Emmanuelle Anceaume, Christian Cachin, and Maria Gradinariu Potop-Butucaru, editors, *19th International Conference on Principles of Distributed Systems, OPODIS 2015, December 14-17, 2015, Rennes, France*, volume 46 of *LIPIcs*, pages 6:1–6:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.OPODIS.2015.6.

29 He Huang, Xiangke Liao, Shanshan Li, Shaoliang Peng, Xiaodong Liu, and Bin Lin. The architecture and traffic management of wireless collaborated hybrid data center network. In Dah Ming Chiu, Jia Wang, Paul Barford, and Srinivasan Seshan, editors, *ACM SIGCOMM 2013 Conference, SIGCOMM'13, Hong Kong, China, August 12-16, 2013*, pages 511–512. ACM, 2013. doi:10.1145/2486001.2491724.

30 Fabian Kuhn and Philipp Schneider. Computing shortest paths and diameter in the hybrid network model. In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 109–118. ACM, 2020. doi:10.1145/3382734.3405719.

31 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.

32 Christoph Lenzen and Boaz Patt-Shamir. Fast partial distance estimation and applications. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 153–162. ACM, 2015. doi:10.1145/2767386.2767398.

33 Christoph Lenzen, Boaz Patt-Shamir, and David Peleg. Distributed distance computation and routing with small messages. *Distributed Comput.*, 32(2):133–157, 2019. doi:10.1007/s00446-018-0326-6.

34 Christoph Lenzen and David Peleg. Efficient distributed source detection with limited bandwidth. In Panagiota Fatourou and Gadi Taubenfeld, editors, *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*, pages 375–382. ACM, 2013. doi:10.1145/2484239.2484262.

35 Danupon Nanongkai. Distributed approximation algorithms for weighted shortest paths. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 565–573. ACM, 2014. doi:10.1145/2591796.2591850.

36 David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 660–672. Springer, 2012. doi:10.1007/978-3-642-31585-5_58.

37 Liam Roditty and Uri Zwick. On dynamic shortest paths problems. *Algorithmica*, 61(2):389–401, 2011. doi:10.1007/s00453-010-9401-5.

38 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.*, 41(5):1235–1265, 2012. doi:10.1137/11085178X.

39 Jeffrey D. Ullman and Mihalis Yannakakis. High-probability parallel transitive-closure algorithms. *SIAM J. Comput.*, 20(1):100–125, 1991. doi:10.1137/0220006.

40 Stefano Vissicchio, Laurent Vanbever, and Olivier Bonaventure. Opportunities and research challenges of hybrid software defined networks. *Comput. Commun. Rev.*, 44(2):70–75, 2014. doi:10.1145/2602204.2602216.

41 Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T. S. Eugene Ng, Michael Kozuch, and Michael P. Ryan. c-through: part-time optics in data centers. In Shivkumar Kalyanaraman, Venkata N. Padmanabhan, K. K. Ramakrishnan, Rajeev Shorey, and Geoffrey M. Voelker, editors, *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New Delhi, India, August 30 -September 3, 2010*, pages 327–338. ACM, 2010. doi:10.1145/1851182.1851222.